

Počítačové systémy, periférie, operační systémy

texty pro distanční studium

Doc. Ing. Cyril Klimeš, CSc.
Ing. Eva Burianová

Ostravská univerzita v Ostravě, Pedagogická fakulta
Katedra Informatiky a počítačů

OBSAH

1	ARCHITEKTURA POČÍTAČŮ	9
1.1	ZÁKLADNÍ POJMY	9
1.2	VON NEUMANNOVO SCHÉMA	11
1.3	HISTORIE POČÍTAČŮ	18
1.4	KATEGORIE POČÍTAČŮ.....	20
2	ČÍSELNÉ SOUSTAVY, PŘEVODY ČÍSEL, ALGEBRAICKÉ VÝPOČTY	21
2.1	PROBLEMATIKA ČÍSELNÝCH SYSTÉMŮ	21
2.2	DESÍTKOVÁ SOUSTAVA	23
2.3	ČÍSELNÝ SYSTÉM, ČÍSLO, ZÁPIS ČÍSLA.....	24
2.4	DALŠÍ ČÍSELNÉ SOUSTAVY:	25
2.4.1	<i>Algebraická pravidla</i>	30
2.5	KÓD BCD	35
3	BOOLEOVA ALGEBRA A ZÁKLADNÍ LOGICKÉ ČLENY A OBVODY	38
3.1	LOGICKÉ PROMĚNNÉ A LOGICKÉ FUNKCE	38
3.1.1	<i>Základní logické operátory</i>	38
3.1.2	<i>Zákony a pravidla Booleovy algebry</i>	40
3.2	DEFINICE LOGICKÉ FUNKCE.....	41
3.2.1	<i>Pravdivostní tabulka</i>	41
3.2.2	<i>Zápis logické funkce</i>	42
3.3	LOGICKÁ FUNKCE N-PROMĚNNÝCH	45
3.3.1	<i>Funkce jedné proměnné</i>	45
3.3.2	<i>Funkce dvou proměnných</i>	45
3.3.3	<i>Funkce více proměnných</i>	47
3.4	ZJEDNODUŠOVÁNÍ ZÁPISU LOGICKÉ FUNKCE	47
3.4.1	<i>Algebraická minimalizace</i>	47
3.4.2	<i>Grafická – Karnaughova metoda</i>	48
3.4.3	<i>Zjednodušení úplně zadané funkce</i>	48
3.4.4	<i>Zjednodušení neúplně zadané funkce</i>	50
3.5	OBVODOVÉ ZNÁZORNĚNÍ BOOLEOVY ALGEBRY	51
3.6	SHEFFEROVA ALGEBRA	52
3.7	PEIRCEOVA ALGEBRA	52
3.8	FYZIKÁLNÍ PODSTATA SIGNÁLŮ.....	53
3.9	TECHNOLOGIE TTL (TRANZISTOR-TRANZISTOR LOGIC)	54
3.10	ZÁKLADNÍ LOGICKÉ ČLENY	55
3.10.1	<i>Invertor</i>	55
3.10.2	<i>AND</i>	56
3.10.3	<i>OR</i>	56
3.10.4	<i>NAND</i>	56
3.10.5	<i>NOR</i>	57
3.11	OSTATNÍ LOGICKÉ ČLENY	57
3.11.1	<i>Nonekvivalence – XOR</i>	57
3.11.2	<i>Ekvivalence - NOXOR</i>	58
3.12	LOGICKÉ OBVODY	58
3.12.1	<i>Multiplexor</i>	58

Počítačové systémy, periférie, operační systémy

3.12.2	4-vstupý multiplexor	58
3.12.3	Dekodér	60
3.13	SČÍTAČKY	61
3.13.1	Sčítačka MODULO 2.....	61
3.13.2	Polosčítačka.....	61
3.13.3	Úplná sčítačka.....	62
4	ÚVOD DO ARCHITEKTURY WINDOWS.....	63
4.1	OPERAČNÍ SYSTÉM.....	64
4.2	ČÁSTI OPERAČNÍHO SYSTÉMU.....	65
4.3	APLIKACE	66
4.4	APLIKAČNÍ PROGRAMOVÉ ROZHRAŇÍ API	66
4.5	JÁDRO.....	67
4.6	OVLADAČE ZAŘÍZENÍ.....	67
4.7	SOUČÁSTI WINDOWS 98	68
4.8	MODULY JÁDRA.....	68
	<i>Kernel16 a Kernel32</i>	<i>68</i>
	<i>GDI – Rozhraní grafických zařízení</i>	<i>69</i>
	<i>User16 a User 32.....</i>	<i>69</i>
4.9	SPRÁVCE VIRTUÁLNÍCH ZAŘÍZENÍ (VMM32) A OVLADAČE VxD	69
	<i>Správce virtuálních zařízení</i>	<i>69</i>
	<i>Správa virtuální paměti</i>	<i>70</i>
	<i>Virtuální ovladače zařízení (VxD).....</i>	<i>71</i>
	<i>Otázky a úkoly.....</i>	<i>72</i>
4.10	REGISTR	72
4.11	INSTALOVANÉ SOUBOROVÉ SYSTÉMY – IFS	73
	<i>Správce konfigurace</i>	<i>73</i>
	<i>Plug and Play.....</i>	<i>73</i>
4.12	REGISTR	75
	<i>Co je to Registr ?</i>	<i>75</i>
	<i>Proč je vlastně Registr uložen ve dvou souborech (USER.DAT a SYSTEM.DAT)?</i>	<i>75</i>
	<i>K čemu je Registr?</i>	<i>76</i>
	<i>Co obsahuje Registr?</i>	<i>76</i>
	<i>Jak to bylo dříve?</i>	<i>76</i>
	<i>Otázky a úkoly.....</i>	<i>77</i>
4.13	POPIS REGISTRAČNÍ DATABÁZE.....	77
	<i>Funkce podvětví Registru</i>	<i>78</i>
4.14	ÚPRAVY V LOKÁLNÍM REGISTRU	78
	<i>Zálohování Registru</i>	<i>79</i>
	<i>Obnovení Registru</i>	<i>79</i>
4.15	SLOVNÍČEK POJMŮ.....	80
4.16	ŘEŠENÍ A ODPOVĚDI.....	81
5	INTERNET A VYHLEDÁVACÍ SLUŽBY	83
5.1	CO JE TO INTERNET	83
	<i>. Nároky na uživatele</i>	<i>83</i>
	<i>Kdy a jak vznikl Internet.....</i>	<i>84</i>
	<i>Co je to Internet a jak funguje</i>	<i>84</i>
	<i>Internet jako informační zdroj</i>	<i>84</i>

<i>Shrnutí</i>	85
5.2 SLUŽBY INTERNETU	86
<i>World Wide Web</i>	86
<i>Hypertext</i>	86
<i>Domovské stránky</i>	87
<i>Záložky</i>	88
<i>Shrnutí</i>	88
5.3 SÍŤOVÁ ARCHITEKTURA INTERNETU	89
<i>Sít'ová vrstva TCP/IP</i>	89
<i>Sít'ový protokol TCP/IP</i>	89
<i>Protokol IP (Internet Protocol)</i>	89
<i>Kritika bezpečnosti Internetu (TCP/IP)</i>	90
<i>Kritika nedostatku adres Internetu (TCP/IP)</i>	90
<i>Kritika charakteru přenosu Internetu (TCP/IP)</i>	90
5.4 NÁSTROJE A SLUŽBY INTERNETU	90
<i>FTP (File Transfer Protocol)</i>	91
<i>Telnet</i>	93
<i>Elektronická pošta (e-mail)</i>	94
<i>Usenet</i>	95
<i>Web (World Wide Web)</i>	95
5.5 TAXONOMIE ADRES A DOMÉN SÍŤE INTERNET	96
<i>HTTP (Hypertext Transfer Protocol)</i>	96
<i>URL (Uniform Resource Locator)</i>	96
<i>IP adresa a její třídy</i>	97
<i>Domény</i>	97
5.6 KATALOGY	99
<i>Co jsou katalogy</i>	99
<i>Vyhledávání nových zdrojů</i>	99
<i>Příklady katalogových služeb</i>	100
<i>Shrnutí</i>	100
5.7 VYHLEDÁVACÍ SLUŽBY DATABÁZOVÉHO TYPU	101
<i>Vyhledávací stroje</i>	101
<i>Jak probíhá hledání</i>	101
<i>Zadávání vyhledávacího řetězce</i>	102
<i>Vývoj databázových služeb</i>	104
5.8 PŘÍKLADY NĚKTERÝCH DATABÁZOVÝCH SLUŽEB	104
<i>Shrnutí</i>	105
5.9 DALŠÍ FORMY VYHLEDÁVÁNÍ	105
<i>Seznamy zdrojů</i>	105
<i>Registrace zdrojů</i>	106
<i>Vyhledávací meta-slужby</i>	106
<i>Další trendy vývoje</i>	107
<i>Příklady vyhledávacích služeb</i>	107
<i>Atlas</i>	107
5.10 CVIČENÍ	108
5.11 ZÁVĚR	109
6 KOMPRIMACE DAT	111
6.1 KOMPRIMAČNÍ PROGRAMY	112
6.2 PARAMETRY KOMPRIMAČNÍCH PROGRAMŮ A JEJICH TESTY	113

Počítačové systémy, periférie, operační systémy

6.3	ZÁKLADNÍ DRUHY A PŘÍSTUPY V KODOVÁNÍ	115
6.4	BEZZTRÁTOVÁ KOMPRIMACE	118
	<i>RLE (Run – length Encoding)</i>	118
	<i>Lempel-Ziv-Welch algoritmus</i>	121
	<i>Řetězec</i>	122
	<i>Přečtený znak</i>	122
	<i>Výstup</i>	122
	<i>Nová položka ve slovníku</i>	122
	<i>Huffmanovo kódování a jemu podobné algoritmy</i>	123
	<i>Shannon-Fanovo kódování</i>	125
	<i>Aritmetické kódování</i>	126
	<i>Řešení a odpovědi na otázky:</i>	129
6.5	ZTRÁTOVÁ KOMPRIMACE	129
	<i>Metody ztrátové komprese obrazu</i>	130
	<i>Vlastnosti:</i>	130
	<i>Druhy obrázků JPEG</i>	131
	<i>Kompresní algoritmus JPEG</i>	131
	<i>Shrnutí</i>	133
	<i>Skalární a vektorová kvantifikace</i>	135
	<i>Metody ztrátové komprese videa a audiosignálu</i>	136
	<i>Shrnutí MPEG:</i>	137
7	POČÍTAČOVÉ VIRY	139
7.1	ÚVOD	139
7.2	TROCHU HISTORIE	140
7.3	INFILTRACE A JEJICH TYPY	140
7.4	HACKERSTVÍ A PRŮNIKY	141
7.5	TROJSKÉ KONĚ = TROJAN HORSES	141
7.6	ČERVI - WORMS	142
7.7	SPAMMING	142
	<i>Čím spamming škodí?</i>	143
7.8	POČÍTAČOVÉ VIRY	143
7.9	OBECNÉ VLASTNOSTI VIRU:	144
	<i>Nejobvyklejší ničivé akce virů :</i>	144
7.10	DRUHY VIRŮ	145
	<i>Klasifikace podle způsobu umístění do paměti</i>	145
	<i>Klasifikace podle napadených oblastí</i>	145
	<i>Klasifikace podle chování a možnosti detekce</i>	145
	<i>Mutační generátory virů</i>	145
7.11	KLASIFIKACE PODLE ZPŮSOBU UMÍSTĚNÍ DO PAMĚTI	145
	<i>Rezidentní viry</i>	145
	<i>Nerezidentní viry - viry přímé akce</i>	146
7.12	KLASIFIKACE PODLE NAPADENÝCH OBLASTÍ	146
	<i>Bootové viry – boot-sektorové viry</i>	146
	<i>Souborové viry</i>	147
	<i>Retroviry</i>	151
	<i>Makro viry - Macro viruses</i>	151
7.13	KLASIFIKACE PODLE CHOVÁNÍ A MOŽNOSTI DETEKCE	152
	<i>Tunelující viry</i>	152
	<i>Mutační generátory virů</i>	153

7.14	SLOVNÍK POJMŮ	154
8	KONFIGURACE POČÍTAČE PC	158
8.1	ZÁKLADNÍ DESKA (MAINBOARD, MOTHERBOARD)	158
8.2	PROCESOR (MIKROPROCESOR).....	160
8.3	PAMĚTI.....	162
8.3.1	<i>Paměti Flash</i>	165
8.3.2	<i>Paměti RAM</i>	165
8.3.3	<i>Organizace paměti v PC</i>	165
8.3.4	<i>Cache paměti</i>	167
8.3.5	<i>CMOS paměť</i>	168
8.4	ČIPOVÉ SADY	169
8.5	SBĚRNICE (BUS)	169
	<i>Sběrnice pro PC</i>	170
8.5.4	<i>Sběrnice ISA (AT bus)</i>	170
8.5.5	<i>Sběrnice MCA (MicroChannel)</i>	171
8.5.6	<i>Sběrnice EISA</i>	171
8.5.7	<i>Sběrnice VL bus</i>	171
8.5.8	<i>Sběrnice PCI</i>	172
8.6	PRUŽNÉ DISKY (FLOPPY DISKY, DISKETY)	173
8.6.4	<i>Mechaniky pružných disků</i>	174
8.7	PEVNÉ DISKY (WINCHESTER DISKY, HARD DISKY)	175
8.7.4	<i>Geometrie pevných disků</i>	177
8.7.5	<i>Rozhraní pevných disků</i>	180
8.8	VIDEOKARTY.....	185
	<i>Typy videokaret</i>	187
8.8.4	<i>Grafická karta VGA</i>	187
8.8.5	<i>Grafická karta SVGA</i>	187
8.9	IO KARTA.....	189
8.10	ZVUKOVÁ KARTA	190
8.11	SÍŤOVÁ KARTA	195
8.12	SKŘÍŇ POČÍTAČE.....	196
8.13	MONITOR	201
8.13.4	<i>Parametry monitorů</i>	204
8.14	KLÁVESNICE A MYŠ.....	206
8.15	MYŠ	208
8.16	PCMCIA.....	209
8.17	EXTERNÍ PAMĚŤOVÁ MÉDIA	210
8.18	CD MECHANIKY	211
8.18.4	<i>CD-ROM</i>	211
8.18.5	<i>Mechaniky CD-R</i>	212
8.18.6	<i>Mechaniky CD-RW</i>	213
8.19	TYPY EXTERNÍCH PAMĚŤOVÝCH MÉDIÍ.....	213
8.19.4	<i>Páskové paměti</i>	213
8.19.5	<i>SyQuest disk</i>	215
8.19.6	<i>Bernoulliho disk</i>	215
8.19.7	<i>Floptical disk</i>	216
8.19.8	<i>Flash disky</i>	216
8.19.9	<i>Magnetooptické disky</i>	216
8.19.10	<i>ZIP disky</i>	217

Počítačové systémy, periférie, operační systémy

8.19.11	<i>Disky LS120</i>	217
8.19.12	<i>JAZZ disky</i>	217
8.19.13	<i>SyJet disky</i>	218
8.20	TISKÁRNA.....	218
8.20.1	<i>Jehličková tiskárna</i>	220
8.20.2	<i>Tepelná tiskárna</i>	220
8.20.3	<i>Inkoustová tiskárna</i>	221
8.20.4	<i>Laserová tiskárna</i>	221
8.21	SCANNER	223
8.22	ZDROJE VYUŽÍVANÉ ZAŘÍZENÍMI	224
8.23	TECHNOLOGIE PLUG & PLAY (PNP)	225
8.24	DALŠÍ ZAŘÍZENÍ.....	226

1 Architektura počítačů

1.1 Základní pojmy

Počítač:

Stroj na zpracování informací

Informace:

1. data, která se strojově zpracovávají
2. vše co nám nebo něčemu podává (popř. předává) zprávu o věcech nebo událostech, které se staly nebo které nastanou

Data:

údaje, hodnoty, čísla, znaky, symboly, grafy, ...

Program:

Algoritmus zapsaný v programovacím jazyce, který řeší nějaký konkrétní úkol. Jedná se o posloupnost instrukcí.

Instrukce:

Předpis k provedení nějaké (většinou jednoduché) činnosti realizovatelný přímo technickým vybavením počítače (např. přičtení jedničky, uložení hodnoty do paměti apod.)

Hardware:

Technické vybavení počítače - souhrnný název pro veškerá fyzická zařízení, kterými je počítač vybaven.

Software:

Programové vybavení počítače - souhrnný název pro veškeré programy, které mohou na počítači pracovat. Software je možné rozdělit do dvou skupin:

systémový software: operační systémy, pomocné programy pro správu systému (utility), překladače programovacích jazyků

aplikační software: programy umožňující řešení specifických problémů uživatele: textové editory
grafické editory
tabulkové procesory
databázové systémy
CAD programy (Computer Aided Design)
DTP programy (Desktop Publishing)
počítačové hry

Firmware:

Programové vybavení, které tvoří součást technického vybavení. Toto programové vybavení až na naprosté výjimky nemůže být uživatelem modifikováno.

bit:

1 bit (binary digit - dvojková číslice) je základní jednotka informace. Poskytuje množství informace potřebné k rozhodnutí mezi dvěma možnostmi. Jednotka bit se označuje **b** a může nabývat pouze dvou hodnot - 0, 1.

Byte:

Jednotka informace, která se označuje **B** a platí $1 B = 8 b$.

Word:



Počítačové systémy, periférie, operační systémy

Jednotka informace. Platí $1 W = 2 B = 16 b$. Kromě této jednotky se také někdy užívá ještě 1 doubleword (DW), pro který platí $1 DW = 2 W = 4 B = 32 b$.

Paměť:

Zařízení, které slouží pro uchování informací (konkrétně binárně kódovaných dat). Množství informací, které je možné do paměti uložit, se nazývá **kapacita paměti** a udává se v bytech. Protože byte je poměrně malá jednotka, používá se často následujících předpon:



Předpona	Značka	Zápis	Mocnina (B)	Převod (B)
kilo	k, K	1 kB	$2^{10} B$	1024 B
mega	M	1 MB	$2^{20} B$	1048576 B
giga	G	1 GB	$2^{30} B$	1073741824 B
tera	T	1 TB	$2^{40} B$	1099511627776 B

Paměť bývá rozdělena na buňky určité velikosti, z nichž každá je jednoznačně identifikována svým číslem. Toto číslo se nazývá **adresa paměti** a velikost takovéto buňky, která má svou vlastní adresu, se označuje jako **nejmenší adresovatelná jednotka**. Paměti je možné rozdělit do následujících základních skupin:

- **Vnitřní (operační):** paměť sloužící pro uchování momentálně zpracovávaných dat a programů. Realizovaná většinou pomocí polovodičových součástek.
- **Vnější (periferní):** paměť sloužící k dlouhodobějšímu uchování dat. Realizovaná většinou na principu magnetického (popř. optického) záznamu dat. Ve srovnání s operační pamětí bývá přístup k jejím datům pomalejší.
- **RAM:** paměť určená ke čtení i zápisu dat
- **ROM:** paměť určená pouze ke čtení dat
- **Paměť s přímým přístupem:** paměť, která dovoluje přistoupit okamžitě k místu s libovolnou adresou
- **Paměť se sekvenčním přístupem:** paměť, u které je nutné při přístupu k místu s adresou n nejdříve postupně přečíst všechna předcházející místa (0 až n-1)

Registr:

Velmi rychlé paměťové místo malé kapacity (jednotky bytů) umístěné většinou uvnitř procesoru počítače.

Řadič (Controller):

Zařízení převádějící příkazy v symbolické formě (instrukce) na posloupnost signálů ovládajících připojené zařízení. Jedná se tedy o zařízení, které řídí činnost jiného zařízení.

Diskrétní režim práce počítače:

Způsob práce počítače, kdy je do paměti počítače zaveden program, data a pak probíhá výpočet. V průběhu výpočtu již není možné s počítačem dále interaktivně komunikovat. Tento způsob práce byl charakteristický pro počítače první generace.

Počítačové systémy, periférie, operační systémy

Integrovaný obvod

Elektronická součástka realizující určité množství obvodových prvků neoddělitelně spojených na povrchu nebo uvnitř určitého spojitého tělesa, aby se dosáhlo ucelené funkce elektronického obvodu

Multitasking:

Současný provoz více úloh na jednom počítači, kdy jedna úloha probíhá na popředí a ostatní probíhají na pozadí. Dovoluje lepší využití CPU. V případě, že uživatel pracuje interaktivně s nějakým programem, který většinu času čeká na zadání jeho požadavků, je možné, aby procesor prováděl např. nějaký náročný matematický výpočet. Je-li na počítači s jedním procesorem provozováno více programů, je procesor přidělován postupně vždy na určitou dobu, tzv. **časové kvantum** (asi 0.1 s), všem provozovaným programům. Podle způsobu práce rozlišujeme dva druhy multitaskingu:

- **kooperativní multitasking:** procesor je operačním systémem přidělen jednomu programu, který jej má v držení tak dlouho, dokud jej sám nevrátí zpět operačnímu systému. Ten jej pak přidělí jinému programu. Nevýhodou je, že program nemusí procesor navrátit v dostatečně krátkém časovém úseku, což způsobí dojem, že ostatní programy nepracují. Ještě horší případ nastane ve chvíli, kdy program procesor nevrátí vůbec (např. zhavaruje). Tato situace vede ve většině případů k havárii celého systému.
- **preemptivní multitasking:** procesor je přidělen programu pouze na určitou dobu a po jejím uplynutí jej sám operační systém programu odebere a přidělí jinému programu. Z toho vyplývá, že nemohou nastat stavy uvedené u kooperativního multitaskingu. Nevýhodou tohoto řešení je vyšší náročnost na hardwarové vybavení počítače.

Vstupní / výstupní zařízení (I/O devices - Input / Output):

Zařízení určená pro vstup i výstup dat. Např.: disky (pevné, pružné), páskové mechaniky

BIOS (ROM BIOS) (Basic Input Output System):

Programové vybavení uložené v paměti ROM (EPROM, EEPROM, Flash) zajišťující nezákladnější funkce (např. zavedení OS).

1.2 Von Neumannovo schéma

Klíčová slova:

Operační paměť

ALU - Arithmetic-Logic Unit (aritmetickologická jednotka)

Řadič

Vstupní jednotka

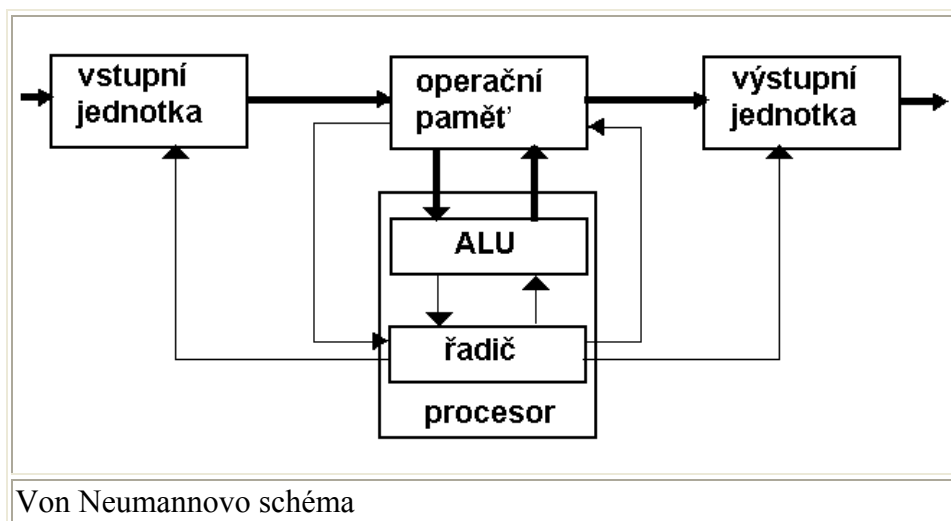
Výstupní jednotka





Již od dávnověku si lidé snaží zjednodušit a hlavně zrychlit počítání. Vznikaly různá počítadla jako je řecký *abakus*, japonský *saroban*, čínský *suan-pchan* atd. V sedmnáctém století po objevu logaritmu byly sestaveny první logaritmická pravítka. Patent na něj získal Angličan **E. Wingate**. Ale již v šestnáctém století vznikaly první počítačové stroje. Jeden z prvních takovýchto strojů navrhl **Wilhelm Schickard** (1592 - 1635). Nestačil však svůj nápad realizovat, neboť během třicetileté války zahynul během epidemie moru. Před několika lety se však podle zachovaného návodu podařilo vytvořit funkční stroj. Autorem jednoho z nejznámějších počítačích mechanických strojů zvaného "paskalina" byl významný francouzský fyzik **Blais Pascal** (1623 - 1662), který stroj postavil pro svého otce, který pracoval jako daňový úředník. V roce 1671 vytvořil německý matematik a fyzik **Gottfried Wilhelm Leibnitz** (1646 - 1716) počítač, který zvládal čtyři základní početní úkony. Leibnitz pochopil, že desítková soustava pro tyto stroje není nejvhodnější a sestavil *dvojkovou* neboli *binární soustavu*. Mezi další známé počítačové stroje patří "počítač" **M. Hahna** z roku 1770, nedokončený avšak na svou dobu moderně navržený diferenční počítačový stroj (Difference engine) Charlese Babbageho (1791 – 1871), kalkulátory švédského inženýra V. T. Odhnera z přelomu našeho století atd. 8. ledna 1889 dostal **Hermann Hollerith** pracující v registračním úřadu Spojených států patent na soupravu děroštitkových strojů, které značně zjednodušily zaznamenávání údajů o obyvatelstvu. O využití děroštitkových strojů projevil značný zájem banky, pojišťovny a velké firmy, a proto Hollerith založil v roce 1896 společnost Tabulating Machine Company. V roce 1924 tato firma a několik dalších podobných se spojila v mamutí koncern International Business Machine (IBM).

Von Neumannovo schéma bylo navrženo roku 1945 americkým matematikem (naroděným v Maďarsku) Johnem von Neumannem jako model samočinného počítače. Tento model s jistými výjimkami zůstal zachován dodnes.



Podle tohoto schématu se počítač skládá z pěti hlavních modulů:

Počítačové systémy, periférie, operační systémy

- **Operační paměť** : slouží k uchování zpracovávaného programu, zpracovávaných dat a výsledků výpočtu
- **ALU - Arithmetic-Logic Unit (aritmetickologická jednotka)**: jednotka provádějící veškeré aritmetické výpočty a logické operace. Obsahuje sčítačky, násobičky (pro aritmetické výpočty) a komparátory (pro porovnávání)
- **Řadič**: řídicí jednotka, která řídí činnost všech částí počítače. Toto řízení je prováděno pomocí **řídících signálů**, které jsou zasílány jednotlivým modulům. Reakce na řídicí signály, stavy jednotlivých modulů jsou naopak zasílány zpět řadiči pomocí **stavových hlášení**
- **Vstupní jednotka**: zařízení určená pro vstup programu a dat.
- **Výstupní jednotka**: zařízení určená pro výstup výsledků, které program zpracoval

Ve von Neumannově schématu je možné ještě vyznačit dva další moduly vzniklé spojením předcházejících modulů:

- **Processor**: Řadič + ALU
- **CPU - Central Processor Unit (centrální procesorová jednotka)**: Procesor



Princip činnosti počítače podle von Neumannova schématu

1. Do operační paměti se pomocí vstupních zařízení přes ALU umístí program, který bude provádět výpočet.
2. Stejným způsobem se do operační paměti umístí data, která bude program zpracovávat
3. Proběhne vlastní výpočet, jehož jednotlivé kroky provádí ALU. Tato jednotka je v průběhu výpočtu spolu s ostatními moduly řízena řadičem počítače. Mezivýsledky výpočtu jsou ukládány do operační paměti.
4. Po skončení výpočtu jsou výsledky poslány přes ALU na výstupní zařízení.



Z hlediska systémového se výpočetní systém skládá z:

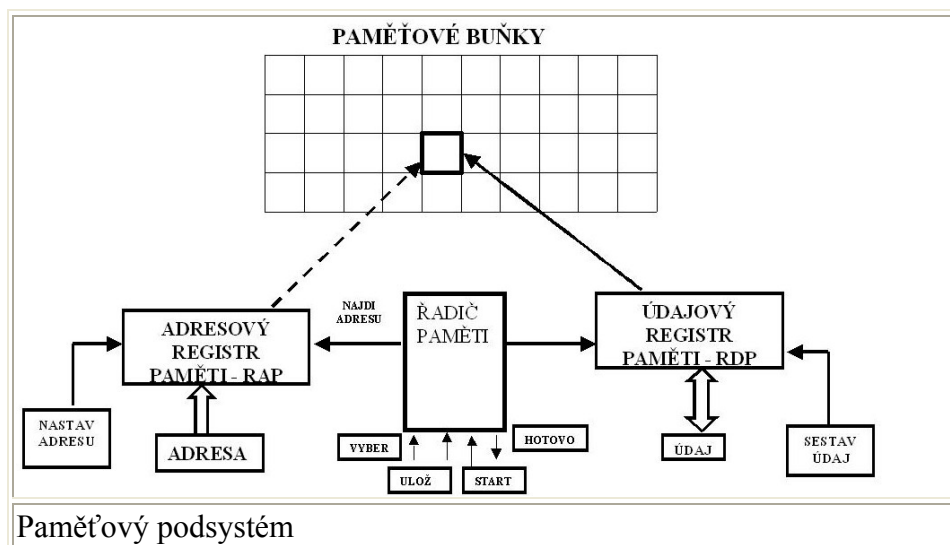
- paměťového podsystemu
- operačního podsystemu
- řídicího podsystemu
- vstupního a výstupního podsystemu.



Centrálním podsystemem je paměť. Přejímá informace od vstupního podsystemu a předává je výstupnímu podsystemu. Řídicí podsystem (řadič) získává z paměti instrukce programu a přiděluje ostatním podsystemům postupně úkoly. Operační podsystem získává podle pokynů řídicího podsystemu z paměťového podsystemu data, zpracovává je a vrací zpět do paměti.

Paměťový podsystem je schopen si zapamatovat tj. uchovat beze změny určité množství informace. Podsystem tvoří:

- blok adresovatelných paměťových buněk
- adresový registr paměti (registr adresy paměti – RAP)
- údajový registr paměti (registr dat paměti – RDP)
- řadič paměti



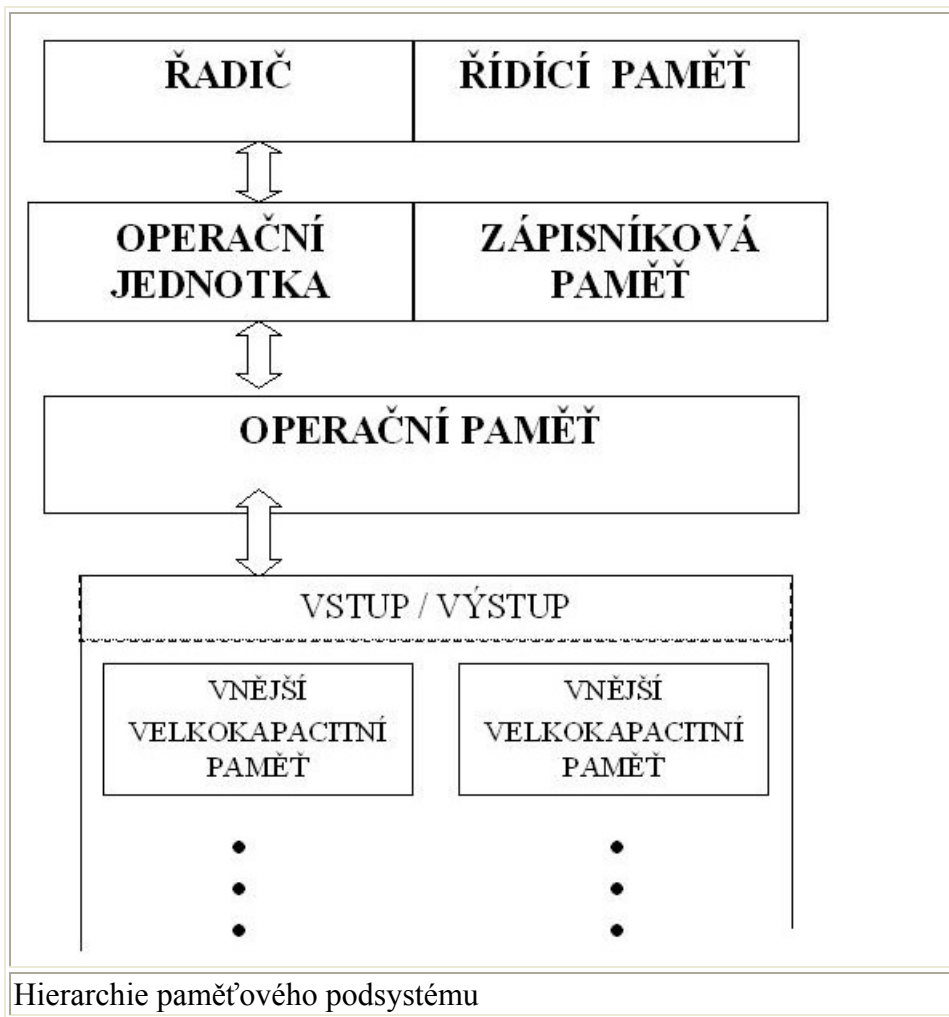
Funkce každé paměťové buňky je dána následujícími předpoklady:

- Obsahuje jedno slovo – slabiku – byte
- Při získání údaje čtením se obsah neporuší
- Lze do ní zapsat novou informaci s podmínkou že se zruší existující informace
- Má jednoznačně přidělenou adresu.

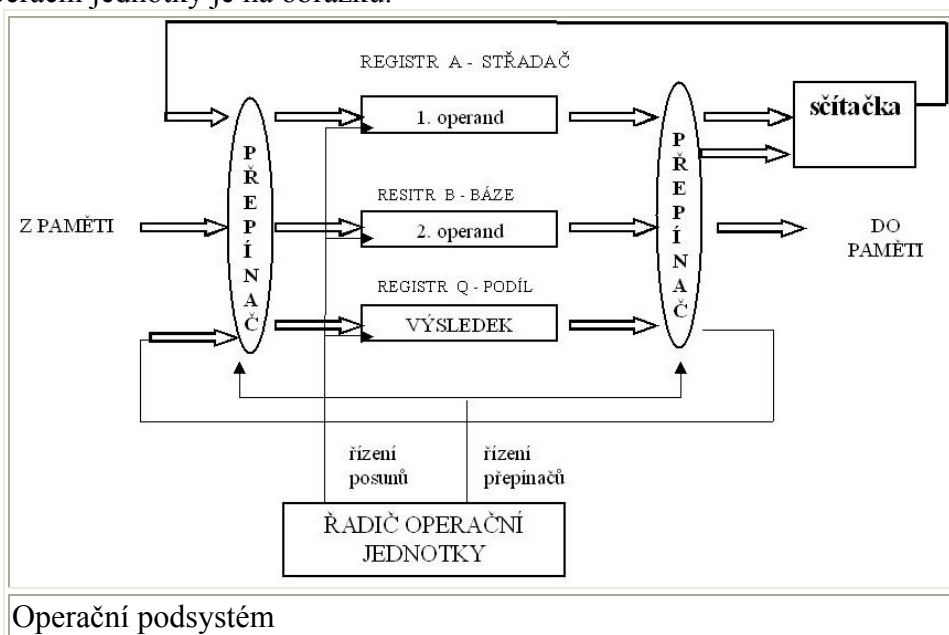
Adresový registr paměti (RAP) uchovává adresu buňky, se kterou paměť v daném okamžiku pracuje. Tyto adresy získává z některých jiných podsystémů. Na základě signálu „nastav adresu“ od zdrojového podsystému se paralelně (najednou celá adresa) zapíše do RAP. Údajový registr paměti (RDP) uchovává data. Při výběru informace z paměti zde řadič paměti dočasně uchovává údaj z buňky, na kterou odkazuje adresový registr paměti. Údaj zde uchovaný je dostupný pro žádající podsystém v okamžiku, kdy řadič vyše signál "hotovo". Při ukládání informace do paměti se údaj z vysílacího podsystému posílá do registru dat paměti na základě signálu "Nastav údaj". Jakmile se nalezne místo, na které odkazuje adresa v adresovaném registru paměti, obsah registru dat se řadičem paměti odešle do této buňky.

Radič paměti řídí cyklus paměti. Jednotka požadující spolupráci s pamětí, zda jde o výběr nebo vkládání informace, při čemž je do RAP uložena adresa korespondující buňky. Při ukládání do paměti se naplní RDP požadovanou informací. Radič paměti uzavře přístup k oběma registrům RAP i RDP, najde buňku a sleduje tok informace mezi registrem dat paměti a zvolenou buňkou. Jakmile je úkol paměti hotov, řadič paměti generuje signál o ukončení "hotovo".

Požadavky, které se v počítačovém systému kladou na paměti se zatím nedají ekonomicky splnit jedinou pamětí. Čím větší je kapacita paměti, tím větší je totiž i její poměrná cena. Podobně roste cena i se zkracováním vybavovací doby. Proto rychlé paměti (s krátkou vybavovací dobou) mají obvykle malou kapacitou, pomalé paměti (s delší vybavovací dobou) mají velkou kapacitu. Začlenění jednotlivých úrovní paměti je naznačeno na obrázku.



Operační podsystem provádí všechny aritmetické a logické transformace. Přitom spolupracuje s pamětí, z níž vybírá operandy a ukládá do ní výsledky. Operaci, která se má provádět, určuje řídicí jednotka. Typické složení sériové operační jednotky je na obrázku.



Počítačové systémy, periférie, operační systémy

Tři registry, každý na jedno slovo, tvoří dočasnou paměť. Registry uchovávají operandy, mezivýsledky i konečně výsledky. Označení registrů vychází z jejich funkce:

RA - stádač (acumulator)

RB - báze (base)

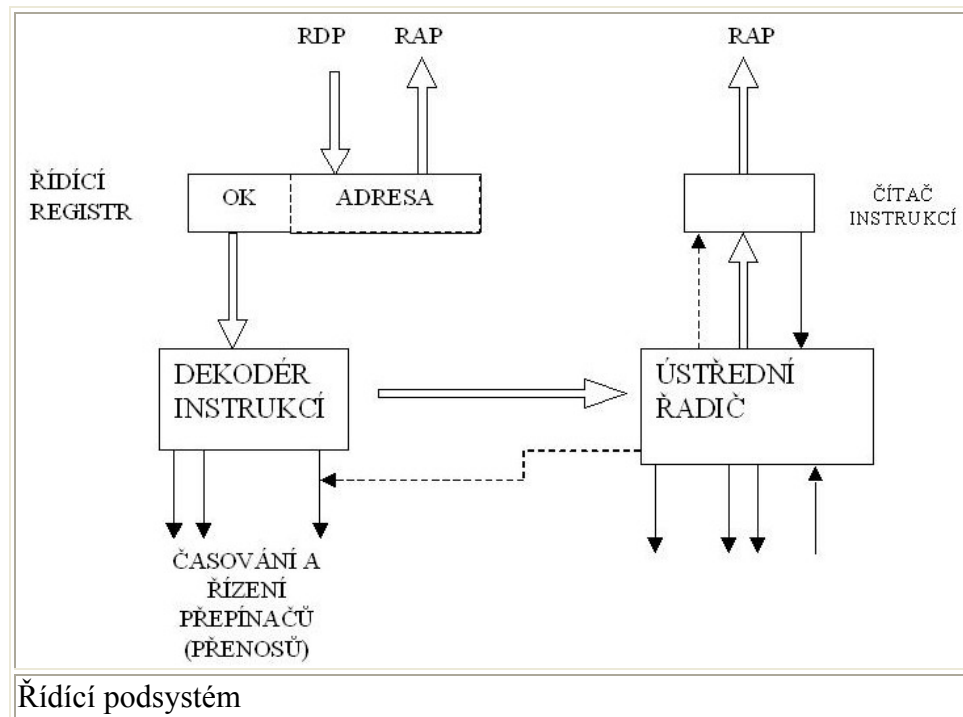
RQ - podíl (quotient)

Operační jednotka obsahuje dále sčítačku, která má i prostředky pro vytváření doplňku, čímž umožňuje odčítání. Přepínač řídí tok informací a umožňuje přímý zápis informace z jednoho registru do druhého, průchod informace za dvou registrů sčítačkou, přesun údaje z paměti do některého registru apod.

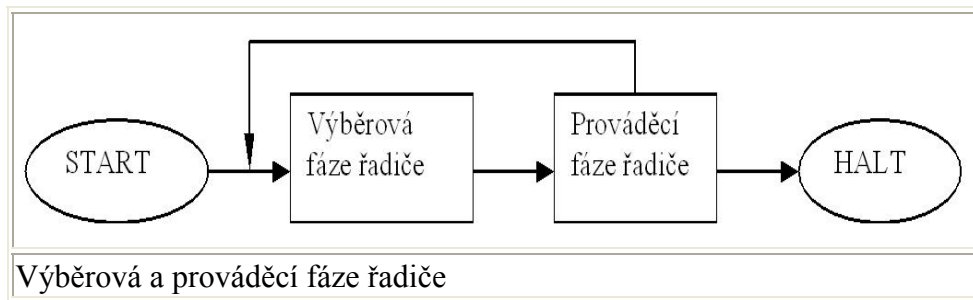
Řadič operační jednotky je autonomní. Dohlíží na činnost operační jednotky. Jakmile se mu předá instrukce, která se má provést, řídí a časově sleduje provádění příslušného algoritmu.

Kromě sériových operačních jednotek se samozřejmě používají sériové paralelní nebo paralelní, které jsou sice rychlejší, ale složitější a tedy i dražší.

Řídicí podsystém (řadič) řídí, resp. dohlíží na veškerou činnost a spolupráci všech podsystémů počítače tak, aby celý systém počítače pracoval podle zadaného programu. Dostává instrukce programu, zpracovává je a transformuje je na posloupnost příkazů pro ostatní části počítače. Těmito příkazy přiděluje úkoly jiným podsystémům. Přitom je stále informován o okamžitém stavu všech podsystémů. Základní funkční jednotky řídicího podsystému jsou uvedeny na obrázku.



Řídicí podsystém pracuje ve dvou fázích: výběrové a prováděcí.



Výběrová fáze (fetch):

Obsah čítače instrukcí se přepíše do adresovaného registru paměti (RAP). V paměti se najde specifikovaná adresa, údaj z tohoto místa se přepíše do registru dat (RDP) a odtud se přepíše do řídicího registru.

Prováděcí fáze (execute):

Vybraná instrukce se analyzuje, čímž se určí, co se má dělat. Pokud se při zpracování požaduje operand, přepíše se adresová část instrukce do adresového registru paměti. Vybraný údaj se pak odesílá registru dat paměti na místo určení, např. do operační jednotky. Jakmile se operand dostane na místo svého určení, řídicí podsystém odešle příslušnému podsystému požadavek na zpracování úkolu a kontroluje jeho provádění. Nakonec připraví adresu další instrukce, tedy zvětší obsah čítače instrukcí o jedničku, a zahájí se nový výběrový cyklus.

Vstupní a výstupní podsystém počítače je spojovacím článkem mezi přídavnými zařízeními a ostatními podsystémy počítače a zahrnuje v sobě logiku řízení vstupů a výstupů. Požadavky na připojení vstupních a výstupních zařízení nejsou vždy známy nebo se nedají dostatečně přesně předvídat při návrhu počítače, a proto se základní vstupní a výstupní funkce z hlediska organizace a připojení přídavných zařízení zajišťují univerzálně.

Základní odlišnosti dnešních počítačů od von Neumannova schématu

- Podle von Neumannova schématu počítač pracuje vždy nad jedním programem. Toto vede k velmi špatnému využití strojového času. Je tedy obvyklé, že počítač zpracovává paralelně více programů zároveň - tzv. multitasking
- Počítač může disponovat i více než jedním procesorem
- Počítač podle von Neumannova schématu pracoval pouze v tzv. diskrétním režimu.
- Existují vstupní / výstupní zařízení (I/O devices), která umožňují jak vstup, tak výstup dat (programu)
- Program se do paměti nemusí zavést celý, ale je možné zavést pouze jeho část a ostatní části zavádět až v případě potřeby

Čím se liší kalkulátor od počítače podle von Neumana



1.3 Historie počítačů

Počítače se rozdělují do tzv. **generací**, kde každá generace je charakteristická svou konfigurací, rychlostí počítače a základním stavebním prvkem. Generace počítačů:



Generace	Rok	Konfigurace	Rychlost (operací/s)	Součástky
0.	1945	Velký počet skříní	Jednotky	Relé
1.	1950	Desítky skříní	100 - 1000	Elektronky
2.	1958	do 10 skříní	Tisíce	Tranzistory
3.	1964	do 5 skříní	Desetitisíce	Integrované obvody
3. ^{1/2}	1972	1 skříň	Statisíce	Integrované obvody (LSI)
4.	1981	1 skříň	desítky milionů	Integrované obvody (VLSI)

1. generace:

První generace počítačů přichází s objevem elektronky, jejímž vynálezcem byl Lee De Forest a která dovoluje odstranění pomalých a nespolehlivých mechanických relé. Tyto počítače jsou vybudovány prakticky podle von Neumannova schématu a je pro ně charakteristický **diskrétní režim práce**. Při tomto zpracování je do paměti počítače zaveden vždy jeden program a data, s kterými pracuje. Poté je spuštěn výpočet, v jehož průběhu již není možné s počítačem interaktivně komunikovat. Po skončení výpočtu musí operátor do počítače zavést další program a jeho data. Diskrétní režim práce se v budoucnu ukazuje jako nevhodný, protože velmi plýtvá strojovým časem. Důvodem tohoto jevu je "pomalý" operátor, který zavádí do počítače zpracovávané programy a data. V tomto okamžiku počítač nepracuje a čeká na operátora.

V této době neexistují vyšší programovací jazyky, z čehož vyplývá vysoká náročnost při vytváření nových programů. Neexistují ani operační systémy.

2. generace:

Druhá generace počítačů nastupuje s tranzistorem, jehož objevitelem byl John Barden a který dovolil díky svým vlastnostem zmenšení rozměrů celého počítače, zvýšení jeho rychlosti a spolehlivosti a snížení energetických nároků počítače. Pro tuto generaci je charakteristický **dávkový režim práce**. Při dávkovém režimu práce je snaha nahradit pomalého operátora tím, že jednotlivé programy a data, která se budou zpracovávat, jsou umístěna do tzv. dávky a celá tato dávka je dána počítači na zpracování. Počítač po skončení jednoho programu okamžitě z dávky zavádí program další a pokračuje v práci.

V této generaci počítačů také začínají vznikat operační systémy a první programovací jazyky, jako jsou COBOL a FORTRAN.

3. generace:

Počítačové systémy, periférie, operační systémy

Počítače třetí a vyšších generací jsou vybudovány na integrovaných obvodech, které na svých čipech integrují velké množství tranzistorů. U této generace se začíná objevovat paralelní zpracování více programů, které má opět za úkol zvýšit využití strojového času počítače. Je totiž charakteristické, že jeden program při své práci buď intenzivně využívá CPU (provádí složitý výpočet), nebo např. spíše využívá V/V zařízení (zavádí data do operační paměti, popř. provádí tisk výstupních dat). Takové programy pak mohou pracovat na počítači společně, čímž se lépe využije kapacit počítače.

S postupným vývojem integrovaných obvodů se neustále zvyšuje stupeň integrace (počet integrovaných členů na čipu integrovaného obvodu). Podle počtu takto integrovaných součástí je možné rozlišit následující stupně integrace:

Označení	Anglický název	Český název	Počet logických členů
SSI	Small Scale Integration	Malá integrace	10
MSI	Middle Scale Integration	Střední integrace	10 - 100
LSI	Large Scale Integration	Vysoká integrace	1000 - 10000
VLSI	Very Large Scale Integration	Velmi vysoká integrace	10000 a více



Integrované obvody je možné vyrábět pomocí různých technologií, z nichž každá má svůj základní stavební prvek a díky němu poskytuje specifické vlastnosti:

- **TTL** (Transistor Transistor Logic): rychlá, ale drahá technologie. Jejím základním stavebním prvkem je bipolární tranzistor. Její nevýhodou je velká spotřeba elektrické energie a z toho vyplývající velké zahřívání se takovýchto obvodů.
- **PMOS** (Positive Metal Oxid Semiconductor): technologie používající unipolární tranzistor MOS s pozitivním vodivostním kanálem. Díky tomu, že MOS tranzistory jsou řízeny elektrickým polem a nikoliv elektrickým proudem jako u technologie TTL, redukuje nároky na spotřebu elektrické energie. Jedná se však o pomalou a dnes nepoužívanou technologii.
- **NMOS** (Negative Metal Oxid Semiconductor): technologie, která využívá jako základní stavební prvek unipolární tranzistor MOS s negativním vodivostním kanálem. Tato technologie se používala zhruba do začátku 80. let. Jedná se o levnější a efektivnější technologii než TTL a rychlejší než PMOS.
- **CMOS** (Complementary Metal Oxid Semiconductor): technologie spojující v jednom návrhu prvky tranzistorů PMOS

i NMOS. Tyto obvody mají malou spotřebu a tato technologie je používána pro výrobu velké části dnešních moderních integrovaných obvodů.

- **BiCMOS** (**B**ipolar **P**ositive **M**etal **O**xid **S**emiconductor): nová technologie spojující na jednom čipu prvky bipolární technologie i technologie CMOS. Používána zejména firmou Intel k výrobě mikroprocesorů.

1.4 Kategorie počítačů

Další typ klasifikace počítačů vychází z výkonu počítačů a z jejich aplikačního nasazení. Definujme si tyto čtyři kategorie počítačů:



mikropočítače

minipočítače

střediskové počítače

superpočítače

Všechny tyto kategorie existují v jednom čase (současné době) vedle sebe. Mikropočítače jsou určeny pro každodenní používání jednému uživateli. Někdy se tato kategorie nazývá také "osobní počítače". Mikropočítače existují díky mikroprocesorům, jejichž nízká cena dovoluje široké použití. Minipočítač sdílí většinou více uživatelů prostřednictvím více terminálů, nebo slouží jako komunikační uzel počítačové sítě apod. Střediskový počítač (Mainframe) svým vysokým výkonem slouží k vědeckotechnickým výpočtům a velkým počtem V/V zařízení pro zpracovávání hromadných dat. Typickými aplikacemi superpočítačů jsou vojenství, meteorologie, seismologie, naftový průmysl, atomová fyzika apod.

2 Číselné soustavy, převody čísel, algebraické výpočty

Cíl:

Cílem této kapitoly studijního textu je seznámení s problematikou číselných systémů, vysvětlení jednotlivých matematických operací v číselných soustavách (jiných než v desítkové).

Po prostudování této kapitoly budete schopni:

- převádět čísla mezi soustavami
- provádět výpočty v jiné než v desítkové soustavě

2.1 Problematika číselných systémů

Klíčová slova:

Číselný systém - číslice = cifry

Arabské číslice: 0,1,2,3,4,5,6,7,8,9

Římské číslice: I, II, III, IV, V, VI, VII, VIII, IX, X, L, C, D, M



Již od prvopočátku lidské společnosti byli první lidé nuceni z praktických důvodů nějakým způsobem zaznamenávat číselné hodnoty. Tím byl položen základ matematiky jako vědy. Podobně jako ostatní vědy i matematika vznikla z praktických potřeb lidí, např. z měření pozemků a obsahu nádob, z počítání času a z mechaniky atd. Podívejme se nyní na historický vývoj jednotlivých soustav použitých v lidských dějinách.



Mezi první nalezené záznamy patří "vroubky" na kostech ze starší doby kamenné. Dokladem je například vlčí kost nalezená v Dolních Věstonicích, která obsahuje sérii 55 vroubků.

- **Nepoziční aditivní soustava** - hodnota použitých symbolů nezávisí na jejich pozici v zápisu
- **Poziční aditivní soustava** - hodnota použitých symbolů závisí na jejich pozici v zápisu



První nepoziční soustavy

Toto byl však pouze počátek. Při malých množstvích lidé vystačili s pouhým použitím jednoduchých vroubků. Kolik bylo vroubků, takový byl počet. Při větším množství bylo nutno zápis zjednodušit tzv. seskupováním tak, že skupina vrypů byla nahrazena jedním znakem. V případě, že hodnota znaku nezávisí na jeho poloze jedná se o **nepoziční aditivní soustavu**.

Typickou představitelkou nepozičních aditivních soustav je **soustava egyptská**. Například egyptský zápis čísla 346 obsahuje 3 znaky pro stovku,

Počítačové systémy, periférie, operační systémy

čtyři pro desítku a šest pro jednotku. Tento způsob zápisu umožňoval pohodlně sčítat i odčítat. Horší už to bylo s násobením a dělením.

Mezi další příklady částečně nepoziční soustavy patří soustava římská:

Prevodní tabulka mezi římskými symboly a dnešním zápisem

číslo	1	5	10	50	100	500	1000
římský zápis	I	V	X	L	C	D	M

Částečně nepoziční soustava je to proto, že je např. rozdíl mezi zápisem XI, který představuje číslo 11, a zápisem IX jenž představuje číslo 9. V římské soustavě totiž platí, jestliže znak s nižší hodnotou následuje za znakem s hodnotou vyšší, hodnoty se sčítají. Je-li pořadí opačné hodnoty se odčítají.



Příklady zápisu pomocí římských číslic:

Římský zápis	Číslo	Výpočet
III	2	1+1
IV	4	5-1
VI	6	5+1
VIII	8	5+1+1+1
IX	9	10-1
XI	11	10+1
XIV	14	10+4
XVIII	18	10+8
XX	20	10+10
XXIX	29	20+9
XL	40	50-10
XC	90	100-10
CD	400	500-100
CM	900	1000-100
MCM	1900	1000+900
MCMLXVII	1967	1000+900+60+7
MCMLXXXIV	1984	1000+900+80+4



1. Zapište číslo 1466 pomocí římských číslic.
2. Co znamená nápis MCMXIV na hrobu neznámého vojáka?
3. Ve kterém roce byl postaven dům, na jehož štítu je nápis MDCLXVI?

Kromě Říma se podobná nepoziční soustava používala od 10 století do prvního století před naším letopočtem v Řecku (**herodiánská symbolika**). Tato soustava pak ustoupila tzv. **jónskému zápisu**, ve kterém se pro zápis čísel využívaly znaky abecedy, které postupně zastupovaly čísla 1 až 9, 10 až 90, 100 až 900. Takto bylo možno zapsat čísla od 1 do 999. Zápis tisíců se prováděl pomocí čárky před znakem.

Tento způsob zápisu se rozšířil spolu s řeckým vlivem v Byzantské říši až do východní Evropy. Lze ho považovat za vrchol snahy efektivně zapsat číslo

Počítačové systémy, periférie, operační systémy

nepozičním desítkovým systémem. Nebyl však schopen obstát při řešení numerických úloh.

Kromě Říma se podobná nepoziční soustava používala od 10 století do prvního století před naším letopočtem v Řecku (**herodiánská symbolika**). Tato soustava pak ustoupila tzv. **jónskému zápisu**, ve kterém se pro zápis čísel využívaly znaky abecedy, které postupně zastupovaly čísla 1 až 9, 10 až 90, 100 až 900. Takto bylo možno zapsat čísla od 1 do 999. Zápis tisíců se prováděl pomocí čárky před znakem.

Tento způsob zápisu se rozšířil spolu s řeckým vlivem v Byzantské říši až do východní Evropy. Lze ho považovat za vrchol snahy efektivně zapsat číslo nepozičním desítkovým systémem. Nebyl však schopen obstát při řešení numerických úloh.

Jak tedy zápis zjednodušit? Vy už to víte. Finta je v zavedení pozice znaku v zápisu. A tak vznikly:



První poziční soustavy

Použití principu poziční soustavy je poprvé doloženo v kulturním odkazu starého Sumeru. Původně se zde používala také soustava nepoziční. Byla to desítková soustava a používala dva znaky ($\bar{Y} = 1$; $\leftarrow = 10$). Pro zápis větších čísel se pak nepoužívaly nové znaky, ale znak pro 1 mohl označovat podle své velikosti 60 nebo 3600. Tato myšlenka, stará nejméně čtyři tisíce let, je pravděpodobně výsledkem používání velkých a malých jednotek v praxi.

Jak ale poznat, jakou velikost autor textu chtěl vyjádřit? A tak dalším vývoji se místo zvětšování velikosti znaku hodnota znaku určovala jeho pozicí v zápisu.



Vznik nejstarší homogenní poziční soustavy je datován 6. až 8. stoletím našeho letopočtu. Touto soustavou je známá desítková indická soustava.

Indičtí matematici měli velkou oblibu v počítání s velkými čísly. Příkladem je například známá úloha o odměně pro vynálezce šachu. A tak museli velice ocenit výhody poziční soustavy.

Indické cifry se sumersko-babylonským pozičním principem a řeckým znakem pro chybějící řád vytvořily systém, který se rozšířil prakticky po celé zemi.

Podle národu, který přispěl k rozšíření tohoto systému v Evropě, je tento systém lidově označován jako "arabské číslice".

Arabské číslice se do Evropy rozšířily už asi v polovině 10 století. Trvalo však ještě dalších 600 let než začala používat ve všech zemích Evropy.

2.2 Desítková soustava

Otázkou zůstává, co rozhodlo ve volbě základu soustavy. Pro nás je desítková soustava samozřejmostí, ale jen pro to, že jsme na ni od malička zvyklí. Ale v

Počítačové systémy, periférie, operační systémy

minulosti existovaly národy, které používaly i jiné základy např. babylónská šedesátková soustava nebo dvacítková soustava Mayů.

Dvacítková soustava starých Keltů se například zachovala ve francouzském číslování, např. osmdesát se ve francouzštině řekne quatre-vingts (čtyři dvacítky) nebo 92 se řekne quatre-vingts-douze (4-20-12). Dvanáctková soustava se dlouho používala v peněžním systému Velké Británie i u nás v počítání na tucty. Na přelomu 20. století bylo u primitivních národů amerického kontinentu objeveno 307 číselných soustav, z nichž pouze 146 bylo desítkových.



Proč tedy používáme právě desítkovou soustavu?

Jisté je, že důležitou úlohu ve volbě soustavy hrál fakt, že člověk má na ruce 10 prstů. I když tento fakt podporuje i jiné soustavy např. pětkovou nebo dvacítkovou. Další výhodou desítkové soustavy je přijatelný počet cifer. Není jich ani moc (60-ková soustava by jich potřebovala 60), ani málo (v tomto případě by zápisy čísel byly dlouhé). Navíc v desítkové soustavě algoritmy základních početních operací nejsou dlouhé.

Kontrolní otázky:

- 1) Jaké znáte typy číselných soustav?
- 2) Ve kterých zemích se tyto soustavy začaly používat?
- 3) Kde byly poprvé použity při výpočtech arabské číslice?
- 4) Jaké jsou výhody desítkové soustavy oproti ostatním?



2.3 Číselný systém, číslo, zápis čísla

V následující kapitole se seznámíme se základními pojmy, které dnešní matematika při studiu číselných soustav používá.



Úvodní pojmy

- **Číslo je řada symbolů.**
- **Každý symbol má definovanou váhu (hodnotu).**
- **Každé číslo je součet matematických výrazů.**
- **Každý výraz je dán součinem číselného symbolu a jeho váhy, přičemž váha je mocninou báze (základu). Mocnina (exponent) začíná nulou a roste po jedné (zprava doleva).**

Desítkové (dekadické) soustavy používají 10 znaků. Tyto znaky označují základní jednotky neboli jednotky nultého řádu. (0-9)

Takže už na základní škole jste se mohli setkat se zápisem:

$$(347,52)_{10} = 3 \cdot 10^2 + 4 \cdot 10^1 + 7 \cdot 10^0 + 5 \cdot 10^{-1} + 2 \cdot 10^{-2}$$

Podobným způsobem se počítá i v jiných soustavách



2.4 Další číselné soustavy:

Ve výpočetní technice se nejčastěji setkáte s těmito soustavami:

- dvojková
- šestnáctková (hexadecimální)
- osmičková (oktanová)

Pro jejich definici je potřeba určit základ a použité číslice:

Dvojková	Oktanová/osmičková	Hexadecimální/ šestnáctková
základ: 2 čísllice: 0,1	základ: 8 čísllice: 0- 7	základ: 16 čísllice: 0 -9,A,B,C,D,E,F

Tímto způsobem bychom si mohli vymyslet jakoukoli soustavu např. trojkovou soustavu a určit si její znaky, např.:

□ = 0	▲ □ = 3
▲ = 1	▲ ▲ = 4
○ = 2	▲ ○ = 5

$(\text{▲} \text{○} \text{□} \text{▲})_3 = 1 \cdot 3^3 + 2 \cdot 3^2 + 0 \cdot 3^1 + 1 \cdot 3^0 = 46_{10}$



Obecný zápis:

$$\text{➤ } A = a_n \cdot k^n + a_{n-1} \cdot k^{n-1} + \dots + a_0 \cdot k^0 + a_{-1} \cdot k^{-1} + \dots + a_{-m} \cdot k^{-m}$$

a_n - cifry

k - základ (např. 10, 2, 16, 8...)

$n, n-1, \dots, 0, 1, \dots, -m$ - mocniny základu

Typ soustavy je zapsán jako dolní index u zadaného čísla!!!



Číslice, která se nachází zcela vpravo je nazývána **nejnižším řádem** nebo nejnižší platnou pozicí.

Číslice **úplně vlevo** - **nejvyšší řád**, nejvyšší platná pozice

Příklady výpočtu hodnoty:

$$(1001)_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = (9)_{10}$$

$$(6372)_8 = 6 \cdot 8^3 + 3 \cdot 8^2 + 7 \cdot 8^1 + 2 \cdot 8^0 = 6 \cdot 512 + 3 \cdot 64 + 7 \cdot 8 + 2 \cdot 1 = (3322)_{10}$$

$$8^0 = 1; 8^1 = 8; 8^2 = 2^6 = 64; 8^3 = 2^9 = 512$$

$$(A13C)_{16} = A \cdot 16^3 + 1 \cdot 16^2 + 3 \cdot 16^1 + C \cdot 16^0 = 10 \cdot 4096 + 1 \cdot 256 + 3 \cdot 16 + 12 \cdot 1 = (41267)_{10}$$

$$16^0 = 1; 16^1 = 16; 16^2 = 2^8 = 256; 16^3 = 2^{12} = 4096$$

Ze dvou přirozených čísel zapsaných v soustavě o témž základu je větší to, v jehož zápisu je více cifer. Mají-li zápisy obou čísel stejný počet číslic, pak je

Počítačové systémy, periférie, operační systémy

větší to číslo, v jehož zápisu číslice nejvyššího řádu označuje větší přirozené číslo. Jsou-li v zápisech obou čísel o stejném počtu číslic stejné všechny číslice řádu vyššího než k-tého a číslice k-tého jsou různé, pak je větší to číslo, v jehož zápisu číslice řádu k-tého označuje větší přirozené číslo.

Vyšší řád (nejbližší vyšší váhu) získáme tak, že nižší váhu vynásobíme základem. Dvojkové číslo roste jako mocnina dvou, dekadické číslo roste jako mocnina deseti.

$$(1000)_2 = (100)_2 * 2$$

$$(1000)_{10} = (100)_{10} * 10$$

Tabulka - váhy v dvojkové soustavě

2^0	1	1
2^1	2	10
2^2	4	100
2^3	8	1000
2^4	16	10000
2^5	32	100000
2^6	64	1000000
2^7	128	10000000
2^8	256	100000000
2^9	512	1000000000
2^{10}	1024	10000000000
2^{11}	2048	100000000000
2^{12}	4096	1000000000000
2^{13}	8192	10000000000000
2^{14}	16384	100000000000000
2^{15}	32768	1000000000000000
2^{16}	65536	10000000000000000

Pro názornost využití v desítkové soustavě :

$$(347,52)_{10} = 3 \cdot 10^2 + 4 \cdot 10^1 + 7 \cdot 10^0 + 5 \cdot 10^{-1} + 2 \cdot 10^{-2}$$

$$752 = 7 \cdot 10^2 + 5 \cdot 10^1 + 2 \cdot 10^0$$

číslice(čís. základ) základ exponent (mocnina báze)

Obdobně platí i pro ostatní číselné soustavy. Pokud má soustava vyšší základ než 10 (počet cifer 0-9), pak se používají za další číslice symboly abecedy (A, B, C, ...), maximálně tudíž můžeme dostat soustavu o základu 36. (Háčkovaná a čárkovaná písmenka a písmenko CH použít nemůžeme)!!

Obdobně řešíme příklady tohoto typu : Převeďte číslo $(A13C)_{16}$ do desítkové číselné soustavy.

Počítačové systémy, periférie, operační systémy

Řešení . **A13C** $= A \cdot 16^3 + 1 \cdot 16^2 + 3 \cdot 16^1 + C \cdot 16^0$
 $= 10 \cdot 4096 + 1 \cdot 256 + 3 \cdot 16 + 12 \cdot 1 = (41267)_{10}$

$(1248)_{10} = (\quad)_{16}$ $(545)_{10} = (221)_{16}$



Tabulka pro převody mezi soustavami:

Desítková	Dvojková	Šestnáctková	Osmičková
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17
16	10000	10	20

Řešený příklad : **Převeďte číslo 90 do dvojkové číselné soustavy.**

ŘEŠENÍ : K cíli vedou dvě cesty. Obě si ukážeme:

90 si vyjádříme jako součet součinů, pomocí rozvoje čísla v soustavě o základu 2
 Tedy : $90 = a_n 2^n + a_{n-1} 2^{n-1} + a_2 2^2 + a_1 2 + a_0$

Kde koeficienty $a_{n-1}, a_{n-2}, \dots, a_1, a_0$ jsou číslice 0 nebo 1, $a_n \neq 0$, $z = 2$.

Tyto koeficienty musíme určit. K tomu nám pomůže vyjádření mocnin čísla 2:
 $2^0=1$ $2^1=2$ $2^2=4$ $2^3=8$ $2^4=16$ $2^5=32$ $2^6=64$ $2^7=128 \dots$

Protože $64 < 90 < 128$, další mocniny neuvádíme, $n=6$. Nyní můžeme číslo 90 vyjádřit následovně:

$$90 = 64 + 16 + 8 + 2 = 1 \cdot 2^6 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^1 =$$

doplníme chybějící součiny, aby rozvoj čísla byl „spojitý“ $= 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot$

$2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 =$ zapíšeme cifry řádu 6,5,...,0, tj. zkrácený zápis, který je řešením: $= (1011010)_2$

$$90 = (1011010)_2$$

Využijeme důkazu věty o jednoznačném vyjádření přirozeného čísla v číselné soustavě o základu z a postupně dostaneme :

$$90 = 2 \cdot 45 + 0$$

$$45 = 2 \cdot 22 + 1$$

$$22 = 2 \cdot 11 + 0$$



Počítačové systémy, periférie, operační systémy

$$\begin{aligned} 11 &= 2 \cdot 5 + 1 \\ 5 &= 2 \cdot 2 + 1 \\ 2 &= 2 \cdot 1 + 0 \\ 1 &= 2 \cdot 0 + 1 \end{aligned}$$

Výsledek získáme tak, že zapíšeme příslušné zbytky zdola nahoru. (Orientační pomůckou nám může být vědomost, že cifra nejvyššího řádu musí být vždy různá od nuly).

$$90 = (1011010)_2$$



Všimněte si:

Šestnáctková - nejvyšší cifra F $(1111)_2$

Osmičková - nejvyšší cifra 7 $(111)_2$

Jaké jsou váhy 6 pozic v daných soustavách:

Dekadická	100000	10000	1000	100	10	1
Dvojková	32	16	8	4	2	1
Osmičková	32768	4096	512	64	8	1
Šestnáctková	1048576	65536	4096	256	16	1

8^5 16^3

Převod z dvojkové soustavy do hexadecimální a osmičkové:

Dvojkové číslo je výhodné rozdělit do skupin po čtyřech. Potom při **hexadecimálním** označení můžeme místo každé čtveřice psát jeden symbol.

101|1010|0110|1110
5 A 6 E - v šestnáctkové

obdobně - do **osmičkové**:

101|101|001|101|110
5 5 1 5 6 - v osmičkové



Vyjádřete následující dvojkové číslo v hexadecimální a dekadické soustavě:

$$(11|1111|0101)_2 = (3F5)_{16} = (1013)_{10}$$

$$512+256+128+64+32+16+4+1=1013$$

$$\text{V osmičkové } (1|111|110|101)_2 = (1765)_8$$

Obdobně:

Dvojková	Hexadecimální	Dvojková
111011	7B	123
01010000	50	80
10110100110111	2D37	11575

Pravidla převádění z dekadické soustavy do dvojkové a hexadecimální:

1. Děleme celé dekadické číslo novým základem
2. Zbytek se stává nejnižším řádem nového čísla

Počítačové systémy, periférie, operační systémy

3. Dělme výsledek předchozího dělení novým základem
4. Zbytek je následující číslicí nového čísla
5. Opakujeme body 3. a 4., dokud neobdržíme nulový výsledek

Převod z hexadecimální do dvojkové:

$$(4E0)_{16} = (0100\ 1110\ 0000)_2$$

Převod z hexadecimální do osmičkové:

$$(4A7)_{16} = (010|010|100|111)_2 = (2247)_8$$



Pravidla pro převádění z hexadecimální do dekadické soustavy:

1. Vynásobme číslici nejvyššího řádu starým základem.
2. K výsledku přičteme následující číslici.
3. Násobme součet starým základem.
4. Opakujeme body 2. a 3.
5. Převod je ukončen, jakmile podle bodu 2. přičteme číslici, která je nejnižším řádem.

Příklad:

$$(1C8)_{16} = (456)_{10}$$

$$\begin{array}{r} 1C8 \\ 1 \\ \hline *16 \\ 16 \\ \hline +12\ (C) \\ 28 \\ \hline *16 \\ 448 \\ \hline +8 \\ \hline (456)_{10} \end{array}$$

$$(3CC)_{16} = (972)_{10}$$

$$\begin{array}{r} 3CC \\ 3 \\ \hline *16 \\ 48 \\ \hline +12\ (C) \\ 60 \\ \hline *16 \\ 960 \\ \hline +12\ (C) \\ \hline (972)_{10} \end{array}$$



$$(3CC)_{16} = (1111001100)_2 = (1714)_8 = (972)_{10}$$

Další příklady:

$$\begin{array}{l} 1) \quad (1248)_{10} = (4E0)_{16} \\ 1248 : 16 = 0 \quad 0 \\ 78 : 16 = 14 \quad E \\ 4 : 16 = 4 \quad 4 \end{array}$$

$$\begin{array}{l} 2) \quad (545)_{10} = (221)_{16} \\ 545 : 16 = 1 \\ 34 : 16 = 2 \\ 2 : 16 = 2 \end{array}$$

$$\begin{array}{l} 2) \quad (47)_{10} = (101111)_2 \\ 47 : 2 = 1 \\ 23 : 2 = 1 \\ 11 : 2 = 1 \\ 5 : 2 = 1 \\ 2 : 2 = 0 \\ 1 : 2 = 1 \end{array}$$

$$\begin{array}{l} 4) \quad (37)_{10} = (100101)_2 \\ 37 : 2 = 1 \\ 18 : 2 = 0 \\ 9 : 2 = 1 \\ 4 : 2 = 0 \\ 2 : 2 = 0 \\ 1 : 2 = 1 \end{array}$$



Další příklad - převod z dekadické soustavy do hexadecimální:

$$(62393)_{10} = (\quad)_2 = (\quad)_{16}$$

Řešení :

$$(62393)_{10} = (1111 \mid 0011 \mid 1011 \mid 1001)_2 = (F3B9)_{16}$$

62393 : 2 = 31196	1
31196 : 2 = 15598	0
15598 : 2 = 7799	0
7799 : 2 = 3899	1
3899 : 2 = 1949	1
1949 : 2 = 974	1
974 : 2 = 487	0
487 : 2 = 243	1
243 : 2 = 121	1
121 : 2 = 60	1
60 : 2 = 30	0
30 : 2 = 15	0
15 : 2 = 7	1
7 : 2 = 3	1
3 : 2 = 1	1
1 : 2 = 0	1

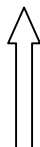


$$(98712)_{10} = (\quad)_{16}$$

Řešení :

$$(98712)_{10} = (18198)_{16}$$

98712 : 16 = 6169	8
6169 : 16 = 385	9
385 : 16 = 24	1
24 : 16 = 1	8
1 : 16 = 0	1



$$(111001100110011)_2 = (\quad)_8$$

Řešení :

$$(111 \mid 001 \mid 100 \mid 110 \mid 011)_2 = (71463)_8$$

7 1 4 6 3

$$(21471)_8 = (\quad)_2 = (\quad)_{16}$$

Řešení :

$$(21471)_8 = (0101 \mid 0011 \mid 1101 \mid 1111)_2 = (53DF)_{16}$$

2.4.1 Algebraická pravidla

Základní pravidla pro sčítání a odčítání s rozdílnými znaménky.

1. Při sčítání dvou čísel se stejnými znaménky, výsledné znaménko je shodné

Počítačové systémy, periférie, operační systémy

2. Když sečítáme dvě čísla s různými znaménky, odečítáme menší od většího (obě čísla bereme bez znamének) a výsledné znaménko bude shodné se znaménkem většího čísla (absolutně)

$$- 12 + (-19) = -31$$

$$12 + 19 = 31$$

$$+12 + (-19) = -7$$

$$- 12 + 19 = +7$$

Sčítání dvojkových a hexadecimálních čísel

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$0 + 1 = 1$$

$$1 + 1 = 10 \text{ - říkáme, že jde o přenos 1 do vyššího řádu}$$

$$1 + 1 + 1 = 11 \text{ - opět přenos 1 do vyššího řádu}$$

$$\begin{array}{r} \text{Př.:} \quad 1001 \quad \quad 0111 \\ \quad \quad \underline{0111} \quad \quad \underline{0101} \\ \quad \quad 10000 \quad \quad 1100 \end{array}$$

přenos do vyššího řádu



Hexadecimální soustava:

$$9 + 5 = E (14)$$

$$9 + 6 = F (15)$$

$$8 + 3 = B (11)$$

$$7 + 9 = 10(16) \text{ - došlo k přenosu}$$

$$A + A = (10 + 10 = 20 = \underline{16} + 4) = 14$$

10 hexadecimálně

$$B + B = (11 + 11 = 22 = 16 + 6) = 16$$

$$F + F = (15 + 15 = 30 = 16 + 14) = 1E$$

$$10 + F = 1F$$

$$8 + A = 12 (8+10=18=16+2)$$

$$A + E = 18 (10+14=24=16+8)$$

$$B + 9 = 14 (11+9=20=16+4)$$

$$C + 7 = 13 (12+7=19=16+3)$$

$$8 + 7 + A = 19 (8+7+10=25=16+9)$$

$$A + B + C = 21 (10+11+12=33=2 \cdot 16 + 1)$$

$$\text{Kontrola: } 21 \cdot 16 = 32 + 1 = 33$$

Osmičková soustava:

$$7 + 1 = 10$$

$$5 + 4 = 11 (9=8+1)$$

$$7 + 7 = 16 (14=8+6)$$

$$6 + 1 = 7$$

$$5 + 7 = 14$$

Odčítání čísel

Pravidla o odčítání vyplývají z pravidel o sčítání

$$\begin{array}{r} 1001 \quad \quad 1001 \quad \quad \text{odečítáme menší od většího (absolutně)} \\ +\underline{0101} \quad \quad -\underline{0101} \end{array}$$

Počítačové systémy, periférie, operační systémy

Příklad:

Vytvoř doplněk k číslu 456

$$\begin{array}{r} 999 \\ -456 \\ \hline 543 \\ +1 \\ \hline 544 \end{array} \quad 544 + 456 = 1000$$

544 = doplněk k číslu 456



Desítkové doplňky čísel:

$$\begin{array}{r} 999 \\ 500 \\ 000 \end{array} \quad \begin{array}{r} 001 \\ 500 \\ \text{nemá doplněk} \end{array} \quad 999 - 000 = 999 + 1 = 1000$$

Doplněk hexadecimálního čísla nazýváme šestnáctkovým doplňkem:

$$\begin{array}{r} 1) \quad 1C8 \quad FFF \\ \quad \quad -1C8 \\ \quad \quad \hline \quad \quad E37 \\ \quad \quad +1 \\ \quad \quad \hline \quad \quad E38 \\ 1C8 + E38 = 1000 \end{array} \quad \begin{array}{r} 2) \quad 4E8 \quad FFF \\ \quad \quad -4E8 \\ \quad \quad \hline \quad \quad B17 \\ \quad \quad +1 \\ \quad \quad \hline \quad \quad B18 \\ 4E8 + B18 = 1000 \end{array}$$

Doplněk dvojkových čísel = dvojkový doplněk:

$$\begin{array}{r} \text{číslo: } 0001 \ 1100 \ 1000 \\ \quad \quad \quad 1111 \ 1111 \ 1111 \\ \quad \quad \quad - \ 0001 \ 1100 \ 1000 \\ \quad \quad \quad \hline \quad \quad \quad 1110 \ 0011 \ 0111 \\ \quad \quad \quad + \quad \quad \quad \quad \quad 1 \\ \quad \quad \quad \hline \quad \quad \quad 1110 \ 0011 \ 1000 \end{array}$$

Doplňkem je číslo 1110 0011 1000

$$\begin{array}{r} \text{Kontrola} \quad 0001 \ 1100 \ 1000 \\ \quad \quad \quad 1110 \ 0011 \ 1000 \\ \quad \quad \quad \hline 1 \ 0000 \ 0000 \ 0000 \end{array}$$

číslo	doplňk
1011 0111	0100 1001
0110 0101	1001 1011

Druhý způsob získávání doplňku dvojkového čísla

- 1) Invertujte každou číslici
- 2) Přičtete 1 do nejnižšího řádu

$$\begin{array}{r} 0001 \ 1100 \ 1000 \\ 1110 \ 0011 \ 0111 \\ \hline 1110 \ 0011 \ 1000 \end{array} \quad \begin{array}{l} \text{číslo} \\ \text{inverze} \\ \text{doplňk} \end{array}$$

Doplňkové sčítání:

- odečítáme-li číslo, provedeme to jako přičtení doplňku odečítaného čísla

Př.: odečtete 456 od 847

1. normální způsob: 847

Počítačové systémy, periférie, operační systémy

$$\begin{array}{r} -456 \\ 391 \end{array}$$

2. pomocí doplňku:

$$\begin{array}{r} 847 \\ +544 \\ \hline 1\ 391 \end{array}$$

přenos do vyššího řádu

Doplňkové sčítání dává vždy stejný výsledek jako odčítání. Výsledek může být ve dvou tvarech:

- normálním
- doplňkovém

(odečítání je vždy ve tvaru normálním). Při doplňkovém sčítání, dojde-li k přenosu do vyššího řádu, je výsledek v normální formě. Pokud nedojde k přenosu do vyššího řádu, je doplněk v doplňkové formě. Aby byl výsledek v normálním tvaru, musí počítač udělat 2 kroky:

1. vytvořit doplněk k výsledku v doplňkové formě
2. změnit znaménko výsledku



Příklad:

$$\begin{array}{r} 456 \\ -847 \\ \hline 609 \end{array}$$

456 456
-847 +153
 609 - nedošlo k přenosu !

následuje 1. krok: $999 - 609 + 1 = 391$

2. krok: - 391

Výsledek je - 391

$$\begin{array}{r} 789 \\ -760 \\ \hline 1\ 029 \end{array}$$

789 789
-760 +240
 1 029 - došlo k přenosu \Rightarrow výsledek je v normální

formě

$$\begin{array}{r} 247 \\ -821 \\ \hline 426 \end{array}$$

247 247
-821 +179
 426 - výsledek je v doplňkové formě

1. krok: $999 - 426 + 1 = 574$

2. krok: Výsledek je -574

Nepřítomnost přenosu indikuje, že výsledek je v doplňkovém tvaru. Abychom získali výsledek v normálním tvaru, musíme vytvořit doplněk výsledku a změnit znaménko.

Doplňkové sčítání ve dvojkové soustavě

$$\begin{array}{r} 1110\ 1001 \\ -0110\ 1011 \\ \hline 1\ 0111\ 1110 \end{array}$$

1110 1001 1110 1001
-0110 1011 1001 0101
 1 0111 1110 - výsledek je v normální formě

$$\begin{array}{r} 0111\ 0111 \\ -1001\ 1101 \\ \hline 1011\ 1010 \end{array}$$

0111 0111 0111 0111
-1001 1101 0110 0011
 1011 1010 - nedošlo k přenosu!

Počítačové systémy, periférie, operační systémy

Doplňěk:

```
1011 1010   číslo
0100 0101
  + 1
-----
0100 0110   doplňěk
```

Výsledek: - 0100 0110

Pojmy k zapamatování:

- Doplněk
- Doplnkový číslo
- Normální tvar výsledku
- Doplnkový tvar výsledku



2.5 Kód BCD

= kód dvojkově desítkové soustavy (binary coded decimal)

= kód čtyřbitový, který se používá pro přímé binární kódování čísel v desítkové soustavě do soustavy dvojkové. Každá desítková číslice je v kódu BCD samostatně vyjádřena jako číslo ve dvojkové soustavě (hexadecimální).

Při kódování jsou jednotlivým bitům zprava doleva přiřazeny váhy:

$$2^0 = 0, 2^1 = 2, 2^2 = 4, 2^3 = 8$$

Desítková číslice je vyjádřena jako součet těchto vah. Kód BCD využívá pouze 10 kombinací čtyř bitů.

Desítkové č.	BCD kód
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Zápis desítkového čísla v kódu BCD:

$$(79523)_{10} = (0111\ 1001\ 0101\ 0010\ 0011)_{\text{BCD}}$$

Použití: v počítači se pomocí kódu BCD uchovávají informace týkající se času a data (popřípadě některé další systémové informace).



Vypočítejte následující příklady a přesvědčete se správnosti řešení v závěru kapitoly

1. Odečtěte:

$$\begin{array}{r} D00A \\ -170B \\ \hline \end{array}$$

2. Převeďte:

$$\begin{array}{l} (B00A)_{16} = (\quad)_{10} \\ (170B)_{16} = (\quad)_{10} \end{array}$$



Shrnutí kapitoly - číselné soustavy

1. číselný systém, číslo, zápis čísla
2. číselné soustavy
3. převádění čísel
 - z dekadické soustavy do dvojkové, hexadecimální, oktanové
 - z hexadecimální do dvojkové, oktanové
 - z hexadecimální do desítkové
 - z dvojkové do desítkové
 - obecně z jakékoliv do desítkové a naopak
4. algebraická pravidla pro sčítání a odčítání
5. doplňkové sčítání
6. kód BCD



Literatura:

B. Pisklák, V. Novotná, Základy elementární matematiky a elementární geometrie, skripta OU
Kolektiv autorů, Číselné soustavy, Kancelářské stroje



Korespondenční úkol č.1:

Vypočítejte následující příklady a výsledky pošlete tutorovi v zadaném termínu.

$$\begin{array}{l} (AA118)_{16} = (\quad)_{10} = (\quad)_2 \\ (AC24)_{16} = (\quad)_{10} \\ (CB6A)_{16} = (\quad)_8 \\ (11100101011101)_2 = (\quad)_{10} \end{array}$$

Sečtěte:

$$CDF15 + 67AD =$$

$$FF + 99 + CA =$$



Korespondenční úkol č.2:

Vypočítejte následující příklady a výsledky pošlete tutorovi v zadaném termínu.

1. Převeďte do dvojkové soustavy $(1548)_{10}$
 $(796)_{10}$
 $(976321)_{10}$
2. Převeďte do sedmičkové soustavy 876
 763
 9810

Počítačové systémy, periférie, operační systémy

3. Převed'te z pětkové do osmičkové soustavy (převodem přes desítkovou...)

$$(443)_5$$

$$(1234)_5$$

4. Převed'te číslo 6148 z desítkové soustavy do soustavy
- dvojkové
 - čtyřkové
 - osmičkové
 - šestnáctkové
 - dvaatřicítkové (prémiový příklad!!!)

Sečtete v šestkové soustavě čísla 4542 a 5354

5. Sečtete ve dvojkové soustavě čísla 10110101110 a 1110110101101
6. Sečtete v hexadecimální soustavě čísla E1EC87 s FE9C5A
7. Zapište číslo (AA425)₁₁ pomocí kódu BCD
8. Převed'te číslo 1010111010111010000101110 z dvojkové soustavy do soustavy:
- osmičkové
 - šestnáctkové
 - čtyřkové
9. Určete rozdíl čísel ve dvanáctkové soustavě : BA94673 – 97BAA94

Korespondenční úkol č.3:

Vypočtete následující příklady a výsledky pošlete tutorovi v zadaném termínu.

$$(AA118)_{16} = (\quad)_{10} = (\quad)_2$$

$$(AC24)_{16} = (\quad)_2$$

$$(CB6A)_{16} = (\quad)_8$$

$$(11100101011101)_2 = (\quad)_{10}$$

Sečtete:

$$CDF15 + 67AD =$$

$$FF + 99 + CA =$$



Výsledky samostatné práce

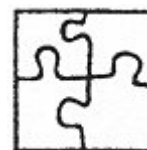
D00A

-170B

B8FF

$$(B00A)_{16} = (53258)_{10}$$

$$(170B)_{16} = (5899)_{10}$$



3 Booleova algebra a základní logické členy a obvody



Matematický prostředek vytvořil George S. Boole jako pomůcku pro znázornění filozofických problémů pomocí matematického aparátu, založeného na dvou pravdivostních hodnotách.

V technice bylo první využití Booleovy algebry při popisu a návrhu reléových obvodů. Značného uplatnění však dosáhla Booleova algebra při návrhu logických obvodů sestavených z hradel „logický součin“, „logický součet“ a „negace“ jimiž lze základní operace Booleovy algebry přímo realizovat.

Booleova algebra nám slouží k matematickému popisu zákonů a pravidel výrokové logiky, která řeší vztahy mezi pravdivými (1) a nepravdivými (0) výroky. Jiná tvrzení nejsou povolena. Pravdivý výrok označujeme logickou hodnotou 1 a nepravdivý výrok logickou hodnotou 0. Nositelem elementární informace o pravdivosti a nepravdivosti výroků je logická proměnná, která může nabývat hodnoty 1 nebo 0.

Booleova algebra studuje dvě proměnné a funkce těchto proměnných. Je to algebra vztahů, nikoliv čísel.

3.1 Logické proměnné a logické funkce

Logická proměnná je veličina, která může nabývat pouze dvou hodnot, označených 0 a 1 (tedy dvojková proměnná), a nemůže se spojitě měnit. Tuto proměnnou označujeme x . Platí:

Logická funkce n proměnných $x_1, x_2, x_3, \dots, x_n$ je funkce, která může nabývat, stejně jako všechny logické proměnné, pouze dvou hodnot.

$$y_1 = x_1, x_2, x_3, \dots, x_n \quad y_2 = a_1, a_2, a_3, \dots, a_n$$
$$V = y_1, y_2$$

Funkce rovnosti platí, když dvě logické proměnné A, B se sobě rovnají, tzn., jestliže $A = 1, B = 1$ nebo $A = 0, B = 0$, což zapisujeme $A = B$. Dvě veličiny $A = a_1, a_2, a_3, \dots, a_n$; $B = b_1, b_2, b_3, \dots, b_n$ se sobě rovnají, když platí $a_i = b_i$ pro všechna i .

3.1.1 Základní logické operátory

V Booleově algebře jsou definovány tři základní operace, jimiž můžeme vyjádřit libovolnou logickou operaci. Jsou to:

Logická negace

Logický součin

Logický součet

Hodnoty proměnné, kterou značíme Y , závislé na jednotlivých kombinacích nezávisle proměnných A, B budeme znázorňovat pro základní logické operace (operátory) pravdivostní tabulkou. Počet řádků v pravdivostní tabulce je dán

Počítačové systémy, periférie, operační systémy

všemi možnými kombinacemi nezávisle proměnných hodnot . Ve sloupci , který je napravo , jsou zapsány stavy hodnot výstupně proměnných Y .

3.1.1.1 Logická negace

Tato operace , která se také velmi často nazývá inverze , dává výsledek zvaný negace . Je aplikována jen na jednu proměnnou . Mění hodnotu nezávisle proměnné na opačnou . Označuje se přidáním „ pruhu “ nad proměnnou x . Jejím výsledkem je hodnota $Y = 1$, jestliže $A = 0$, a naopak $Y = 0$, jestliže $A = 1$. Negace je vyjádřena zápisem „ NE “ (NOT) .

Tabulka funkce negace :

Tato definice vede ke vztahům :

A	\bar{A}
0	1
1	0

$$\bar{0} = 1$$

$$\bar{1} = 0$$

3.1.1.2 Logický součin

Tato operace , nazývána také průnik nebo konjunkce , aplikovaná na dvě proměnné , vytváří součin neboli funkci AND těchto dvou proměnných . Logický součin nabývá hodnotu I jen tehdy , když všechny nezávislé proměnné mají hodnotu I . Jestliže je alespoň jedna hodnota rovna 0 , potom výsledná hodnota se rovná 0 . Značí se symbolem \wedge mezi dvěma proměnnými ($A \wedge B$) . V praxi se však používá zápis :

$$Y = A . B$$

Logický součin je vyjádřen spojkou „ a “ (v angličtině „AND“ nebo „ & “) . Definice logického součinu může být vyjádřena vztahy :

Tabulka funkce logického součinu :

A	B	A . B
0	0	0
0	I	0
I	0	0
I	I	I

$$0 . 0 = 0$$

$$0 . I = I . 0 = 0$$

$$I . I = I$$

3.1.1.3 Logický součet

Tato operace, nazývána také sjednocení nebo disjunkce, aplikovaná na dvě proměnné, vytváří součet neboli funkci OR těchto dvou proměnných. Logický součet má hodnotu 1, jestliže jedna nezávisle proměnná nebo druhá nezávisle proměnná nebo obě mají hodnotu 1. Označuje se symbolem \vee mezi dvěma proměnnými ($A \vee B$).

V praxi se používá ovšem zápis :

$$Y = A + B$$

Logický součet je vyjádřen „nebo“ (v angličtině „OR“).
Definice logického součtu může být vyjádřena vztahy :

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

$$0 + 0 = 0$$

$$0 + 1 = 1 + 0 = 1$$

$$1 + 1 = 1$$

3.1.2 Zákony a pravidla Booleovy algebry

Pro vyhodnocení logických výrazů, různé úpravy a zjednodušení logických výrazů je nutné znát zákony Booleovy algebry :



1. Komutativní zákon :

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

2. Asociativní zákon :

$$A + (B + C) = (A + B) + C$$

$$A \cdot (B \cdot C) = (A \cdot B) \cdot C$$

3. Distributivní zákon:

$$A + B \cdot C = (A + B) \cdot (A + C)$$

$$A \cdot (B + C) = A \cdot B + A \cdot C$$

4. Zákon o agresivnosti prvku 1 a 0 :

$$A + 1 = 1$$

$$A \cdot 0 = 0$$

5. Zákon o neutrálnosti prvku 1 a 0:

$$A + 0 = A$$

$$A \cdot 1 = A$$

6. Zákon o vyloučení třetího :

$$A + \bar{A} = 1$$

$$A \cdot \bar{A} = 0$$

7. Zákon dvojité negace :

$$\overline{\bar{A}} = A$$

Počítačové systémy, periférie, operační systémy

8. Zákon absorpce:

$$A + A = A$$

$$A \cdot A = A$$

$$A + A \cdot B = A$$

$$A \cdot (A + B) = A$$

9. Zákon absorpce negace:

$$A + \bar{A} = I$$

$$A \cdot \bar{A} = 0$$

$$A + \bar{A} \cdot B = A + B$$

$$A \cdot (\bar{A} + B) = A \cdot B$$

$$\bar{A} + A \cdot B = \bar{A} + B$$

$$\bar{A}(A + B) = \bar{A} \cdot B$$

Některé zákony vyplívají přímo z definic základních logických operací – jsou to operace součtu a součinu s hodnotami 0 a I. Ze všeobecných zákonů je nejvýznamnější de Morganův zákon, který se vyjadřuje dvěma rovnostmi.

10. De Morganův zákon:

$$\overline{A + B + C + \dots} = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \dots$$

$$\overline{A \cdot B \cdot C \cdot \dots} = \bar{A} + \bar{B} + \bar{C} + \dots$$

Tento zákon lze tedy vyjádřit tak, že negaci funkce získáme nahrazením každé proměnné její negací a záměnou značek součtu a součinu navzájem. Při použití tohoto zákona je třeba věnovat velkou pozornost implicitním závorkám:

$$A + B \cdot C = A + (B \cdot C)$$

Tedy:

$$\overline{A + B \cdot C} = \bar{A} \cdot (\bar{B} + \bar{C})$$

a neplatí:

$$\overline{A \cdot B + C}$$

3.2 Definice logické funkce

3.2.1 Pravdivostní tabulka

Pravdivostní tabulka je tabulka, do které se zapisuje logická (Booleovská funkce). Pravdivostní tabulka má $r + n$ sloupců a 2^n řádků. Číslo r je počet sloupců výsledných funkcí (obvykle bývá jedna výsledná funkce – tedy jeden sloupec). Číslo n udává počet proměnných. Číslo 2^n udává počet všech možných kombinací proměnných, kde číslo n je počet proměnných. Tyto kombinace reprezentuje počet řádků.

Příklad: Pokud je dána logická funkce, která má tři proměnné a jednu výslednou funkci, tak pravděpodobnostní tabulka bude mít čtyři sloupce ($r + n = 1 + 3 = 4$) a osm řádků ($2^n = 2^3 = 8$).



3.2.1.1 Úplně zadaná funkce

Logická funkce je úplně zadaná, jestliže je známa její hodnota 1 nebo 0 pro všechny možné kombinace hodnot proměnných. Těchto kombinací je pro n proměnných 2^n . Lze tak sestavit pravdivostní tabulku.



Příklad: Je dána funkce tří proměnných $f(A, B, C)$, pravdivostní tabulka bude tedy vypadat následovně:

A	B	C	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

3.2.1.2 Neúplně zadaná funkce

Logická funkce je neúplně zadaná, když její hodnota pro některé kombinace hodnot proměnných je libovolná nebo není určena. S tímto případem se setkáváme, když některé kombinace hodnot jsou fyzikálně nemožné. Hodnotu funkce poté značíme x nebo \emptyset . Pravdivostní tabulka může vypadat následovně:

A	B	C	f
0	0	0	0
0	0	1	1
0	1	0	x
0	1	1	x
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	x

3.2.2 Zápis logické funkce

3.2.2.1 Základní tvary funkce



Logickou funkci můžeme zapsat ve dvou tvarech, nazývaných základní součtový a základní součinný tvar. V praxi se ale často používají názvy úplná disjunktivní normální forma – ÚDNF a úplná konjunktivní normální forma – ÚKNF.

ÚDNF – je to součet základních součinů přímých nebo negovaných proměnných. Každý základní součin (minterm – z ang. minimal polynomial

Počítačové systémy, periférie, operační systémy

term) nabývá hodnoty I pro určitou kombinaci , kdy funkce má hodnotu I , a hodnoty 0 pro všechny ostatní kombinace . ÚDNF vyjadřuje funkci jako součet případů , kdy má hodnotu I .

ÚKNF – je to součin základních součtů přímých nebo negovaných proměnných . Každý základní součet nabývá hodnoty 0 pro určitou kombinaci , kdy funkce má hodnotu 0 , a hodnoty I pro všechny ostatní kombinace . ÚKNF vyjadřuje funkci jako součin případů , kdy má hodnotu 0 .

3.2.2.2 Výpis logických funkcí z pravdivostní tabulky

Tabulka funkce tří proměnných (příklad – libovolná funkce)

A	B	C	f
0	0	0	I
0	0	I	0
0	I	0	I
0	I	I	0
I	0	0	I
I	0	I	0
I	I	0	I
I	I	I	0

a) ÚDNF – základní součtový tvar

Případy, kdy logická funkce je rovna I:
součiny:

Odpovídající

$$\text{kombinace A,B,C} \begin{cases} 000 & \bar{A} \cdot \bar{B} \cdot \bar{C} \\ 010 & \bar{A} \cdot B \cdot \bar{C} \\ 100 & A \cdot \bar{B} \cdot \bar{C} \\ 110 & A \cdot B \cdot \bar{C} \end{cases}$$

Výslednou funkci dostáváme jako součet základních součinů (algebraicky) :

$$f = \bar{A} \bar{B} \bar{C} + \bar{A} B \bar{C} + A \bar{B} \bar{C} + A B \bar{C}$$

b) ÚKNF – základní součinnový tvar

Případy, kdy logická funkce je rovna 0 :

Odpovídající součty :

$$\text{kombinace A,B,C} \begin{cases} 001 & A + B + \bar{C} \\ 011 & A + \bar{B} + \bar{C} \\ 101 & \bar{A} + B + \bar{C} \\ 111 & \bar{A} + \bar{B} + \bar{C} \end{cases}$$

Výslednou funkci dostáváme jako součin základních součtů (algebraicky) :

$$f = (A + B + \overline{C}) \cdot (A + \overline{B} + \overline{C}) \cdot (\overline{A} + B + \overline{C}) \cdot (\overline{A} + \overline{B} + \overline{C})$$

V praxi se častěji používá součtový tvar , tedy ÚDNF .

3.2.2.3 Zápis logické funkce do Karnaughovy mapy

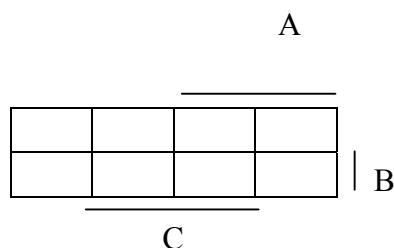
Karnaughova mapa umožňuje přehledný zápis všech hodnot logické funkce . Porovnáme-li pravdivostní tabulky (např. tří proměnných) , zjistíme , že levá strana tabulky je vždy stejná a nepřináší nové informace . Každé kombinaci nezávisle proměnných je v Karnaughově mapě přidělen jeden čtverec , do kterého zapisujeme výstupní hodnotu funkce . Počet čtverců je tedy roven počtu řádků v pravdivostní tabulce . Přiřazení vstupních proměnných jednotlivým řádkům a sloupcům se provádí úsečkami , někdy algebraickým označením . Zápisu funkce do Karnaughovy mapy se také někdy říká grafická metoda .



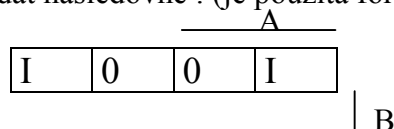
Příklad : Pro příklad použijeme dříve uvedenou pravdivostní tabulku tří proměnných.

A	B	C	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Postup : Nejprve si nakreslíme tabulku (Karnaughovu mapu), která bude mít počet políček (čtverců) stejný , jako počet řádků v pravdivostní tabulce (v našem případě 8) . Potom přiřadíme vstupní proměnné jednotlivým řádkům a sloupcům např. pomocí úseček (tento způsob je více přehledný) . Mapa by měla vypadat následovně :



Poté na základě pravdivostní tabulky , respektive na základě vstupních proměnných , zapíšeme do tabulky hodnotu výstupní funkce . Konečná Karnaughova mapa by měla vypadat následovně : (je použita forma ÚDNF)



I	0	0	I
C			

3.3 Logická funkce n-proměnných

Logická funkce f_n n-proměnných nabývá všech možných hodnot, pro všechny možné kombinace n-proměnných. Počet funkcí je $(2^2)^n$. Toto číslo roste velmi rychle např. pro $n = 3$ je počet 256 možných kombinací.

Funkce rozdělujeme takto:

Funkce jedné proměnné, kde je počet funkcí $(2^2)^1 = 4$

Funkce dvou proměnných, kde je počet funkcí $(2^2)^2 = 16$

Funkce více než dvou proměnných.

3.3.1 Funkce jedné proměnné

Funkce jedné proměnné je kombinace hodnot A. Pro jednu proměnnou jsou funkce uvedeny v následující tabulce:

A	f_0	f_1	f_2	f_3
0	0	0	I	I
I	0	I	0	I

Funkce jsou označeny f_0 až f_3 , kde index představuje hodnotu dvojkového čísla umístěného v odpovídajícím sloupci, přičemž váhy rostou po řádcích směrem nahoru.

Platí: konstanty: $f_0 = 0$ a $f_3 = I$

proměnná sama: $f_1 = A$

negace proměnné: $f_2 = \overline{A}$

3.3.2 Funkce dvou proměnných

Pro dvě proměnné je počet funkcí 16. Jsou dány pravdivostní tabulkou, kde 16 funkcí f_n nabývá všech možných hodnot pro všechny možné kombinace dvou proměnných. Funkce jsou označeny f_0 až f_{15} , kde index představuje hodnotu dvojkového čísla umístěného v odpovídajícím sloupci, přičemž váhy rostou po řádcích směrem nahoru.

A	B	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	0	0	I	I	I	I	I	I	I	I
0	I	0	0	0	0	I	I	I	I	0	0	0	0	I	I	I	I
I	0	0	0	I	I	0	0	I	I	0	0	I	I	0	0	I	I
I	I	0	I	0	I	0	I	0	I	0	I	0	I	0	I	0	I

Počítačové systémy, periférie, operační systémy

Funkce jedné proměnné :

konstanty : $f_0 = 0$ a $f_{15} = 1$

proměnná sama : $f_3 = A$ a $f_5 = B$

negace proměnné : $f_{12} = \bar{A}$ a $f_{10} = \bar{B}$

Funkce odpovídající základním operátorům :

logický součet ve významu slučovacím : $f_7 = A + B$

logický součin : $f_1 = A \cdot B$

Osm nových funkcí :

Součet ve významu vylučovacím neboli součet modulo 2 neboli nonekvivalence , též XOR , běžně zapisována

$$A \oplus B \quad f_6 = A\bar{B} + \bar{A}B$$

Funkce ekvivalence , zapisovaná $A \equiv B$, nebo též

$$A \otimes B \quad f_9 = AB + \bar{A}\bar{B}$$

Funkce ani jeden není nebo také NOR (negace logického součtu) – Piercova funkce

$$f_8 = \bar{A}\bar{B} = \overline{A + B}$$

Funkce alespoň jeden není neboli Schefferova funkce , nazývaná také NAND (negace logického součinu)

$$f_{14} = \bar{A} + \bar{B} = \overline{AB}$$

Negace implikace – inhibice , zábrana $f_2 = A\bar{B}$

Negace obrácené implikace $f_4 = \bar{A}B$

Obrácená implikace $f_{11} = A + \bar{B}$

Implikace $f_{13} = \bar{A} + B$

Implikace

Znalost logických funkcí je nutná zejména při návrhu , minimalizaci a konstrukci logických obvodů . Vynecháním některé funkce může dojít ke konstrukční závadě.

Pomocí funkcí NAND a NOR lze vyjádřit jakoukoliv funkci .

Počítačové systémy, periférie, operační systémy

3.3.3 Funkce více proměnných

Například když si vezmeme funkci tří proměnných . Pro tři proměnné existuje 256 funkcí . Vyjádření takové funkce by bylo již nepřehledné . Pro vyjádření můžeme však využít již známých funkcí dvou proměnných , protože :

$$f(A, B, C) = A \cdot f(I, B, C) + \bar{A} \cdot f(0, B, C)$$

Kontrola : Pro ověření tohoto vztahu stačí položit $A = I$ nebo $A = 0$. Z toho plyne, že

$$f(I, B, C) \text{ a } f(0, B, C)$$

jsou funkce dvou proměnných . Stejná úvaha se může použít pro případ více než tří proměnných .

3.4 Zjednodušování zápisu logické funkce

Zjednodušování , často také minimalizace , logické funkce je určitý postup , kterým lze získat jednodušší vyjádření logické funkce . Minimalizací získáme jednodušší strukturu logické funkce při zachování stejného výsledku .

Minimalizace se proto používá k nejjednodušší technické realizaci , tzn. , že minimalizace je důležitá při konstrukci logických obvodů – stejná funkce může být zhotovena z menšího počtu logických členů = ekonomičnost .

Nejpoužívanější jsou dva způsoby , a to algebraická minimalizace – pomocí Booleovy algebry , nebo grafická minimalizace . Grafických metod je více , například Quinova-McCluskeyho – metoda , ale nejpoužívanější je Karnaughova metoda .



3.4.1 Algebraická minimalizace

Algebraická minimalizace logických funkcí je upravování logického výrazu podle zákonů a pravidel Booleovy algebry . Výsledná funkce je vyjádřena co nejjednodušeji se stejným chováním , což při realizaci znamená jednodušší konstrukci logického obvodu .

Příklad algebraické minimalizace demonstrujeme na jednoduchém příkladě :

A	B	f
0	0	1
0	1	0
1	0	1
1	1	1



Podle již uvedených a tedy i známých pravidel určíme výslednou funkci (ÚDNF) :

$$f = \bar{A}\bar{B} + A\bar{B} + AB$$

Tuto funkci dle Booleových zákonů a pravidel můžeme upravit na tvar:

$$f = \bar{A}\bar{B} + A\bar{B} + AB = A(\bar{B} + B) + \bar{A}\bar{B} = A + \bar{A}\bar{B} = A + \bar{B}$$

Jak vidíme , výsledná zjednodušená logická funkce je omnoho jednodušší než předešlá funkce . Ovšem tato metoda je velice riskantní a úpravy pomocí této metody jsou obtížné . Pro více než tři proměnné je téměř nepoužitelná .

3.4.2 Grafická – Karnaughova metoda

Pro zjednodušení funkce pomocí algebraické minimalizace spojujeme součiny (mintermy) , které se liší v jediné proměnné . Tyto součiny se nazývají sousední .

Příklad: $A B \bar{C} \bar{D} + A B \bar{C} D = A B \bar{C} (\bar{D} + D) = A B \bar{C}$



Karnaughova metoda se díky jasnému geometrickému postupu vyhýbá hledání sousedním součinů složitým algebraickým způsobem , který je navíc velmi nespolehlivý . Tato metoda se používá velmi dobře pro 3 , 4 a 5 proměnných . Příklad : Hledání sousedních součinů pomocí Karnaughovy metody , příklad je pro čtyři proměnné .

		A		
		x		
D		x		B
	x	H	x	
		C		

Jak vidíme , najít sousední součiny x zadaného políčka H je pomocí Karnaughovy metody jednoduché .

3.4.3 Zjednodušení úplně zadané funkce

Do políčka zapíšeme pomocí logických hodnot 1 a 0 pro všechny kombinace proměnných odpovídající hodnoty funkce . Zjednodušování spočívá v hledání pro funkci v disjunktivním tvaru smyčky dvou , čtyř a osmi sousedních políček tak , aby se ze skupiny součinů (mintermů) vyloučila jedna , dvě nebo tři proměnné . Pro vyjádření smyček se používají pouze jen políčka obsahující 1 .

Příklad : Libovolná pravdivostní tabulka úplně zadané funkce čtyř proměnných



A	B	C	D	f
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

Karnaughova mapa této funkce :

		A		
		0	1	
D		1	0	1
		0	1	1
		0	0	1
		1	0	1
		B		
		0	1	
		C		

Úplná disjunktivní normální forma této funkce je :

$$f = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}B\bar{C}D + A\bar{B}C\bar{D} + A\bar{B}CD + A\bar{B}\bar{C}D$$

Z této mapy můžeme udělat tři smyčky :

Smyčka se dvěma políčky

$$\bar{A}\bar{B}\bar{C}D + A\bar{B}\bar{C}D = \bar{B}\bar{C}D(\bar{A} + A) = \bar{B}\bar{C}D$$

Smyčka se čtyřmi políčky na pravé straně

$$A\bar{B}\bar{C}(D + \bar{D}) + A\bar{B}C(D + \bar{D}) = A\bar{B}\bar{C} + A\bar{B}C = A\bar{B}(\bar{C} + C) = A\bar{B}$$

Smyčka s políčky v rozích mapy

$$\bar{A}\bar{B}\bar{D}(\bar{C} + C) + A\bar{B}\bar{D}(\bar{C} + C) = \bar{A}\bar{B}\bar{D} + A\bar{B}\bar{D} = \bar{B}\bar{D}(\bar{A} + A) = \bar{B}\bar{D}$$

Celkový výsledek logické funkce po minimalizaci – zjednodušení je tedy :

$$f = B \bar{C} D + A \bar{B} + \bar{B} \bar{D}$$

Tento způsob upravování je náročný a zbytečně zdlouhavý . Nejpoužívanější způsob je vyloučit proměnné , které mění svůj stav a ponechají se pouze proměnné , které se nemění .

Příklad:



Smyčka se čtyřmi políčky v rozích mapy

A – mění stav , takže se vyloučí

B – zůstává stejná s hodnotou 0 , zapíšeme \bar{B}

C – mění svůj stav , takže se vyloučí

D – zůstává stejná s hodnotou 0 , zapíšeme \bar{D}

Dostáváme tedy součin $\bar{B} \bar{D}$

Jak je vidět , tento součin je naprosto shodný se součinem s předešlé metody. Z toho plyne, že obě metody jsou správné , ale poslední metoda je jednodušší , protože se vyhneme upravování členů .

3.4.4 Zjednodušení neúplně zadané funkce

Pro zjednodušení neúplně zadané funkce použijeme funkci stejnou jako pro zjednodušování úplně zadané funkce . Pro usnadnění smyček zapíšeme X do políček místo hodnoty I , v ostatních případech 0. Zápis bude tedy vypadat následovně:

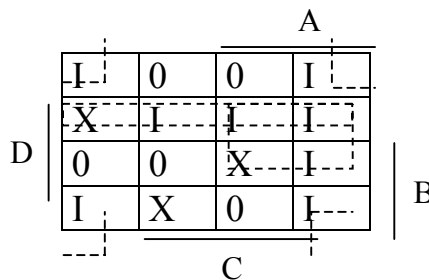
$$f = \bar{B} \bar{D} + \bar{C} D + A D$$

Pravdivostní tabulka neúplně zadané funkce:

A	B	C	D	f
0	0	0	0	I
0	0	0	I	X
0	0	I	0	I

0	0	I	I	0
0	I	0	0	0
0	I	0	I	I
0	I	I	0	X
0	I	I	I	0
I	0	0	0	I
I	0	0	I	I
I	0	I	0	I
I	0	I	I	I
I	I	0	0	0
I	I	0	I	I
I	I	I	0	0
I	I	I	I	X

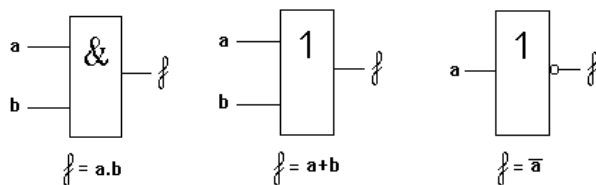
Karnaughova mapa neúplně zadané funkce:



$$f = \bar{C} \bar{D} + \bar{B} D + AD$$

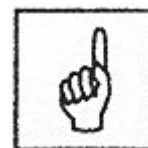
Tyto metody jsou použitelné pro zjednodušování logických obvodů realizovaných pouze pomocí základních logických operátorů – logického součinu a součtu. Pro systémy realizované pomocí složitějších funkcí jsou nepoužitelné.

3.5 Obvodové znázornění Booleovy algebry

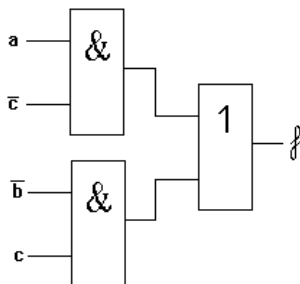


Pravidla pro kresbu značek:

- Vstup je vždy zleva, výstup zprava
- Značky se nesmějí otáčet
- Spoje mají být rovnoběžné s okrajem listu



Př: $f = a \cdot \bar{c} + \bar{b} \cdot c$



3.6 Shefferova algebra

Je vybudovaná na jedné logické funkci = negace logického součinu NAND.

- Pro libovolný počet proměnných $f = \overline{x \cdot y}$

$$\overline{x \cdot x} = \bar{x}$$

$$\overline{x \cdot 0} = 1$$

$$\overline{x \cdot 1} = \bar{x}$$

$$\overline{\overline{x \cdot y \cdot 1}} = \overline{\overline{x \cdot y}} = x \cdot y$$

Pravidla: $\overline{\overline{x \cdot x \cdot y \cdot y}} = \overline{\overline{x \cdot y}} = x + y$

- Pomocí operace NAND lze realizovat všechny operace Booleovy algebry.
- Neplatí zákon komutativní: $\overline{x \cdot y} \neq \overline{y \cdot x}$
- Neplatí zákon asociativní: $\overline{\overline{x \cdot y \cdot z}} \neq \overline{\overline{x \cdot y \cdot z}} \neq \overline{\overline{x \cdot y \cdot z}}$

3.7 Peirceova algebra

Vystavěna na operaci NOR (negace logického součtu) - obdobně jako S-algebra.

Převod minimalizované formy B-algebry na S-algebru:

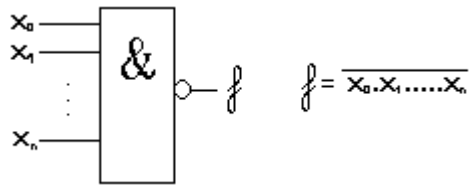
Opakovanou aplikací de Morganových pravidel:

$$f = \overline{\overline{\overline{b \cdot c + a \cdot c + \bar{a} b d}}}$$

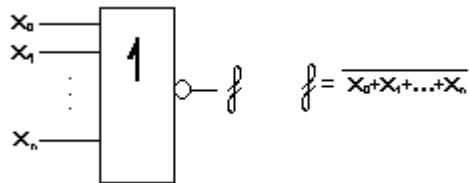
Př.: $= \overline{\overline{b \cdot c \cdot \bar{a} \cdot c \cdot \bar{a} b d}} \quad \overline{a + b} = \bar{a} \cdot \bar{b}$



Obvodové znázornění S-algebry:



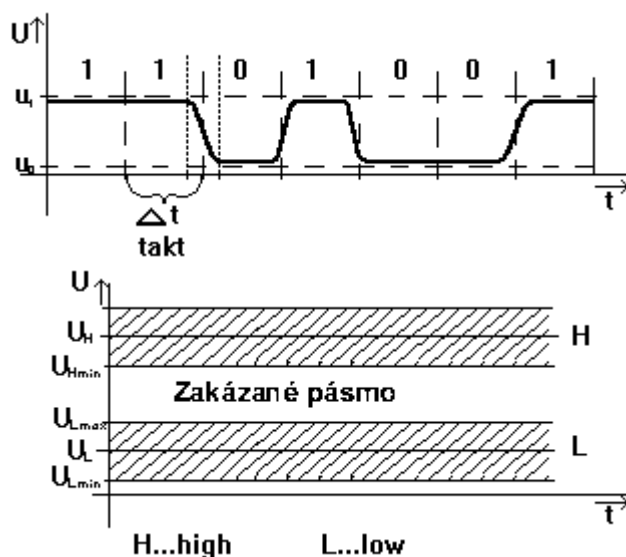
Obvodové znázornění P-algebry:



3.8 Fyzikální podstata signálů

HLADINOVÉ			IMPULSOVÉ	
RELÉ				
 Proud prochází	 Větší proud	 Vyšší napětí	 Přítomnost impulsu	 Kladná polarita
 Proud neprochází	 Menší proud	 Nižší napětí	 Nepřítom. impulsu	 Záporná polar.

AMPLITUDA	KMITOČET	FÁZE
 Vyšší amplituda	 Vyšší kmitočet	 Fáze
 Nižší amplituda	 Nižší kmitočet	 Fáze



Hodnoty jsou stanoveny pro každou výrobní technologii zvlášť.

L ~ 0 H ~ 1pozitivní logika

L ~ 1 H ~ 0negativní logika

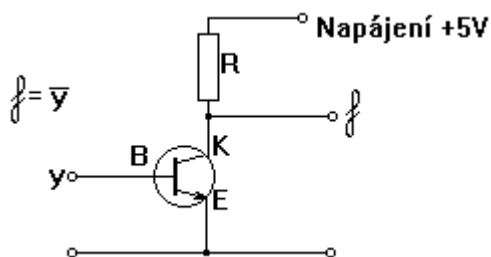
3.9 Technologie TTL (tranzistor-tranzistor logic)

Základní stavební prvek je tranzistor NPN.

Parametry TTL:

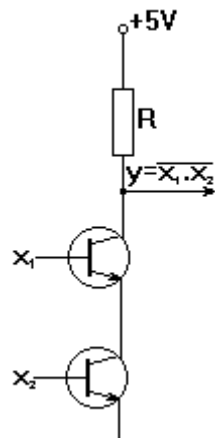
- napájecí napětí + 5V
- L < 0,8V L ~ 0,4V
- H > 2,0V H ~ 2,4V

Invertor v TTL:



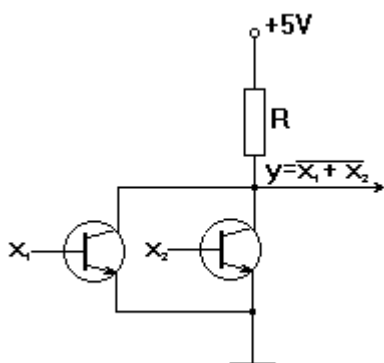
Počítačové systémy, periférie, operační systémy

NAND pomocí dvou tranzistorů:



x ₁	x ₂	NAND	x ₁	x ₂	NOR
0	0	1	0	0	1
0	1	1	0	1	0
1	0	1	1	0	0
1	1	0	1	1	0

NOR:

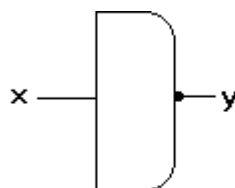
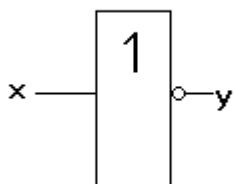


3.10 Základní logické členy

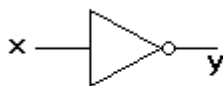
3.10.1 Invertor

USA

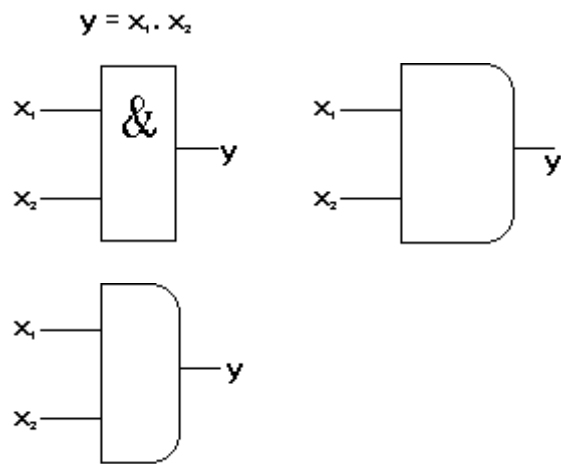
$$y = \bar{x}$$



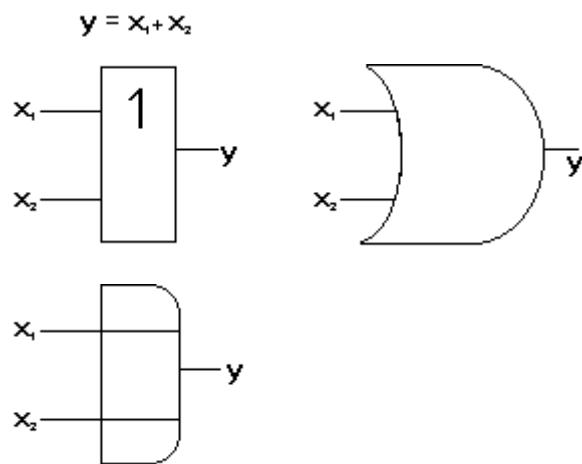
DIN



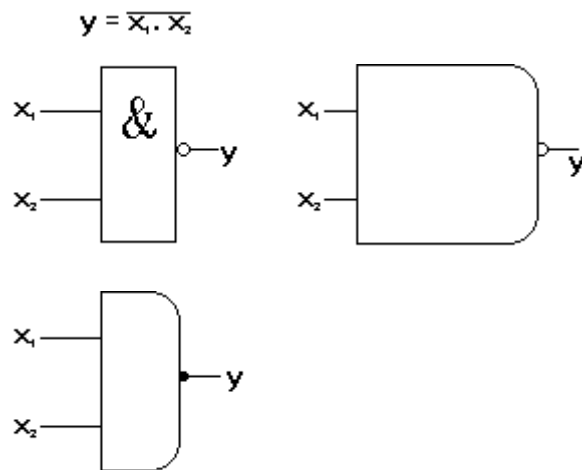
3.10.2 AND



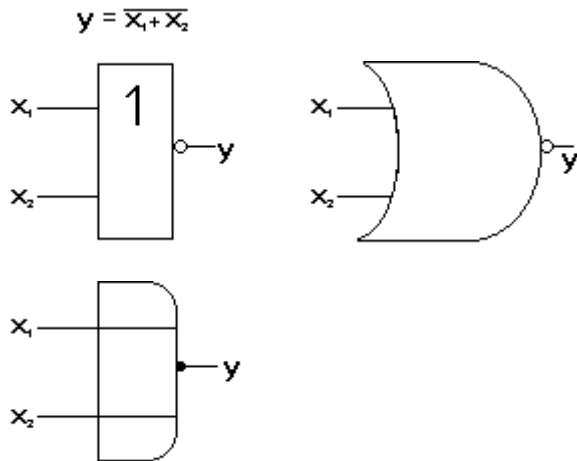
3.10.3 OR



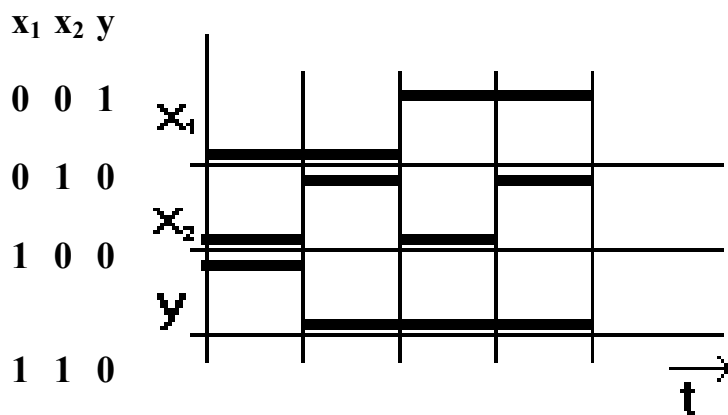
3.10.4 NAND



3.10.5 NOR



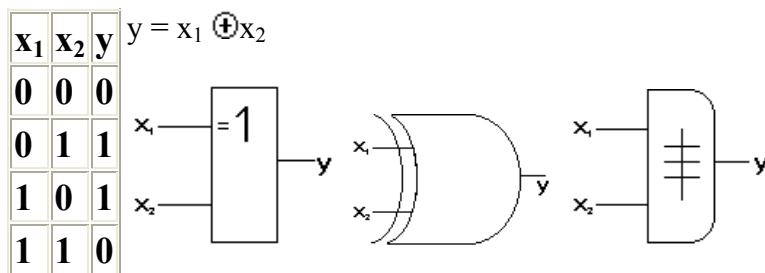
Př.: NOR



3.11 Ostatní logické členy

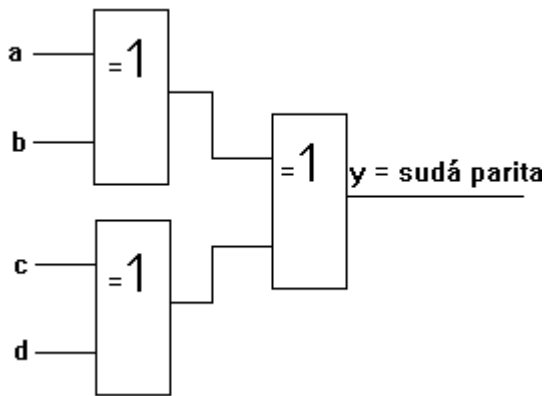
3.11.1 Nonekvivalence – XOR

$\neq \oplus = 1$

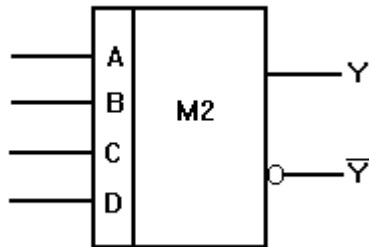




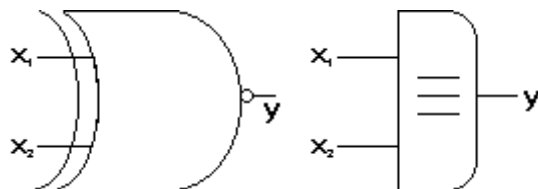
Př. Generátor parity: $y = a \oplus b \oplus c \oplus d = (a \oplus b) \oplus (c \oplus d)$



Schématická značka:



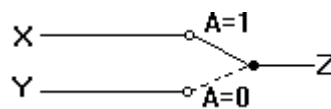
3.11.2 Ekvivalence - NOXOR



3.12 Logické obvody

3.12.1 Multiplexor

$$Z = A \cdot X + \bar{A} \cdot Y$$

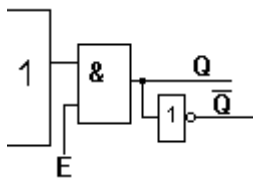
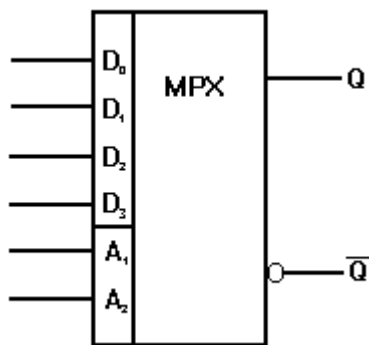
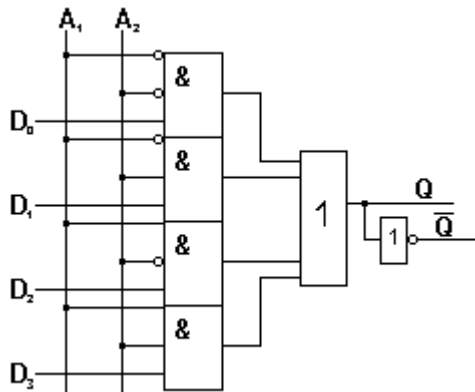


3.12.2 4-vstupý multiplexor

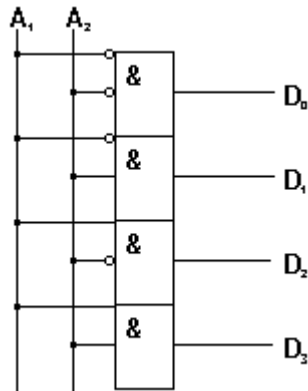
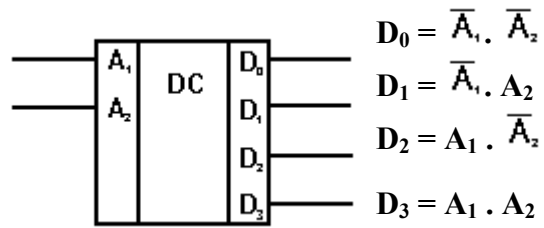
4 datové vstupy, 2 adresové vstupy

Počítačové systémy, periférie, operační systémy

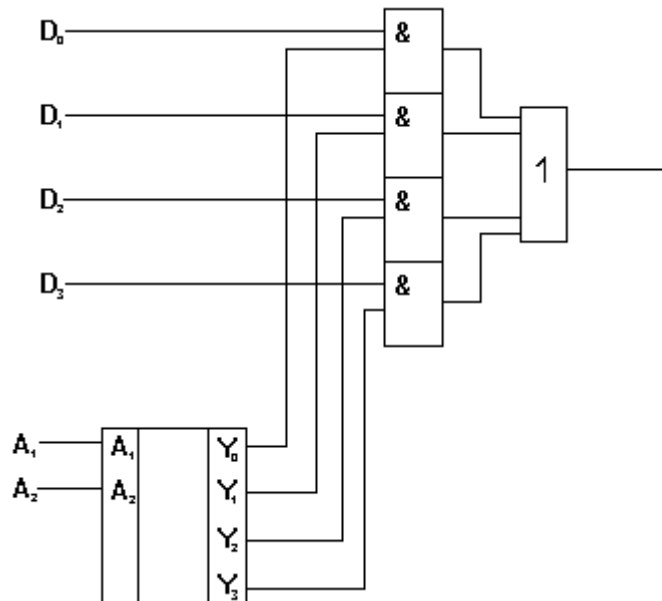
A_1	A_2	Q
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

$$Q = \bar{A}_1 \cdot \bar{A}_2 \cdot D_0 + \bar{A}_1 \cdot A_2 \cdot D_1 + A_1 \cdot \bar{A}_2 \cdot D_2 + A_1 \cdot A_2 \cdot D_3$$


3.12.3 Dekodér



Realizace MPX pomocí dekodéru:



3.13 Sčítačky

3.13.1 Sčítačka MODULO 2

$$\mathbf{x + y = z}$$

1. Tabulka:

x	y	z
0	0	0
0	1	1
1	0	1
1	1	0

2. Rovnice:

$$\mathbf{z = \bar{x} \cdot y + x \cdot \bar{y}}$$

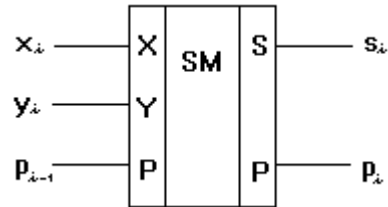
3.13.2 Polosčítačka

x	y	S	P
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S = \bar{x} \cdot y + x \cdot \bar{y}$
 $P = x \cdot y$

3.13.3 Úplná sčítačka

x_i	y_i	p_{i-1}	s_i	p_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



4 Úvod do architektury Windows

Cíl:

Záměrem této kapitoly je seznámení s operačním systémem Windows, s jeho základní architekturou a jeho částmi.

Po prostudování kapitoly budete schopni:

- definovat operační systém
- vyjmenovat části operačního systému
- odpovědět na otázky týkající se součásti OS Windows
- orientovat se v problematice registrů OS Windows

Klíčová slova:

- operační systém
- moduly jádra
- registr

Tento několikastránkový text je opravdu pouze náhledem do složité a obsáhlé problematiky, jakou operační systém Windows 95 nebo Windows 98 a jejich architektura bezesporu je. V první části se čtenář seznámí s obecným dělením operačního systému Windows a jejich složkách. Další část představuje architekturu Windows a popisuje její komponenty (jádro, správce virtuálních zařízení a paměti, VxD apod.). V poslední – třetí – části zlehka nakoukneme do tajemství registrů. Zodpovíme si některé základní otázky týkající se této „tajné“ zbraně operačních systémů Windows.

Každý uživatel Windows 95/98 s nimi může zacházet (a třeba i zdárně), může využívat jejich výhod a komfortu, aniž by měl ponětí o souvislostech konajících se na pozadí, o architektuře systému a hlubším významu konkrétních položek. Ovšem i běžnému uživateli se hodí určitá představa o daném operačním systému, se kterým pracuje. Více však touhu poznat a pochopit „zákulisi“ Windows bude pociťovat, když ona „nádherná paráda“ barevných oken a ikon přestane fungovat nebo začne vykazovat určité chyby a chybová hlášení. Co si pak počít? Nejvhodnějším způsobem řešení je samozřejmě zavolat odborníka, který našemu operačnímu systému „domluví“ a my pak zase na nějaký čas můžeme být s ním (s operačním systémem a s technikem samozřejmě taky) zase kamarádi. Mnoho záludností uživatel může více chápat, více problémů může vyřešit a některým i zabránit, aby vznikly, když bude mít, alespoň určitou představu o fungování používaného operačního systému a tedy i jeho architektuře.



Jak již bylo uvedeno, následující text si nečiní nárok na zevrubné popsání operačního systému Windows 95 či 98 – k tomu slouží obsáhlé publikace od výrobce nebo erudovaných autorů – snaží se však čtenáře stručným, ale výstižným a přehledným způsobem zasvětit do tajů snad prozatím nejvíce používaných operačních systémů Windows 95 a Windows 98. Text je určen jako doplňující studijní materiál k předmětu Úvod do operačních systémů. Protože Windows 95 a Windows 98 jsou si blízké operační systémy (Windows 98 vznikl vylepšením starších „devadesátpětek“) a jsou mezi domácími uživateli v současnosti patrně nejpoužívanějšími operačními systémy (dále v textu budu používat někdy zkratky OS), je architektura OS vysvětlována právě na těchto dvou. Z důvodu větší novosti Windows 98 je mnohdy učivo popisováno na tomto OS – většina však uváděných věcí je obecných pro oba systémy. Jelikož oba OS stále nesou odkaz předešlých operačních systémů (operačních prostředí), jakými byly Windows 3.x nebo DOS, jsou konfrontovány s těmito staršími systémy a tudíž se v textu dočtete, jak tomu bylo – nebylo u nich.

Obecné principy budou platit rovněž pro nové a nejnovější OS, jako kupř. Windows 2000 nebo Windows XP. Ty už však mají svá specifika (stejně jako je již větší rozdíl mezi Windows NT a Windows 95). Tudíž z těchto a rovněž prostorových důvodů i důvodu zachování přehlednosti (aby se čtenář mezi různými OS příliš neztrácel), není obsah zaměřen také na nové operační systémy a odkazy na ně se zde neobjevují.



*Při sestavování textu jsme převážně vycházeli z knihy **Windows 98 - poradce expert autorů Marka Minasiho, Erica Christiansena a Kristiny Shapar** od nakladatelství Grada z roku 1998. Další použitá literatura je uvedena v přehledu na konci práce.*

4.1 Operační systém

Definice **operačního systému** jednotlivých autorů je rozličná. My bychom pro tento úvod mohli použít například tuto:

Operační systém - je souhrn programů, které umožňují uživateli využívat technické a programové prostředky daného počítače. Operační systém ovládá řízení prostředků výpočetního systému, tedy procesor, operační paměť, jednotlivé periférie, atd.

Operační systém můžeme rozdělit do dvou částí:

- a) základ operačního systému
- b) rozšiřující komponenty

Základ operačního systému tvoří programy pro zavedení operačního systému do paměti, programy pro obsluhu periferního zařízení, pro obsluhu souborů dat, pro zavedení uživatelského programu do operační paměti, jeho spouštění apod.

Rozšiřující komponenty jsou programy, které jsou na stejné úrovni s programy uživatelskými, ale jsou dodávány výrobcem operačního systému k zajištění některých důležitých funkcí.

Setkáme se i s následujícími definicemi:

- OS je programové vybavení nezbytné pro provoz počítače.(co je nezbytné ?)
- OS je souhrn programů, které umožňují uživateli využívat technické a programové prostředky počítače.
- OS ovlává řízení prostředků výpočetního systému - procesor, operační paměť, jednotlivé periférie.
- OS je správce prostředků.
- OS je hlavní řídicí program, který spravuje zdroje (prostředky) počítačového systému

Vlastnosti operačních systémů

- jednouživatelský
- multiuživatelský (síťový)
- jednoprogramový
- umožňující multitasking

4.2 Části operačního systému

Stejně jako všechny operační systémy mají i Windows 98 dva hlavní úkoly: umožnit aplikacím využití hardwaru a zajistit hladkou spolupráci aplikací. Podívejme se tedy podrobněji na jednotlivé části obecného operačního systému.



Aplikace:

Aplikační programové rozhraní:

Množina příkazů, jejichž provedení může aplikace žádat na operačním systému; např. „vytvoř soubor“ nebo „přečti vstup z klávesnice“.

Jádro:

Celkový správce systému. Řídí multitasking, správu paměti, přidělování zařízení, a podobně.

Ovladače zařízení

Programy, umožňující jádru vykonávat základní požadavky, např. „přečti polohu myši“, a předávat je odpovídajícím zařízením.

Hardware:

grafická karta, pevný disk, klávesnice, tiskárna aj.

Podívejme se na jednotlivé součásti OS (operačního systému) podrobněji:

4.3 Aplikace

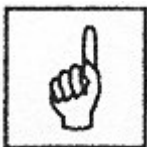


Hlavním důvodem používání OS je spuštění aplikací, které nám pomáhají při plnění našich úkolů. Aplikace nemohou a neumějí přistupovat přímo k hardwaru počítače (přesněji – v mnoha systémech aplikace přistupují přímo k hardwaru – to ale způsobuje velké potíže, a proto používáme Windows 98, abychom omezili problémy známé ze systémů Windows 3.1 a DOS). Chce-li aplikace otevřít soubor, přečíst znak stisknuté klávesy, nebo zobrazit na obrazovce dialogové okno, potom tyto aktivity neprovádí sama; o vykonání takovéto činnosti žádá operační systém.

4.4 Aplikační programové rozhraní API



Ředitelství továrny se běžně nachází jinde než je výrobní hala. Dělník z výrobní haly může komunikovat s členem vedení, ale pravděpodobně ne tak, že jednoduše přijde do budovy a zabuší na ředitelovy dveře. K takové komunikaci je zapotřebí určité „rozhraní“ - v tomto případě sekretářka.



Podobně je operační systém uložen v jedné části paměti a aplikace v části jiné. Přímým skokem aplikace do spuštěného procesu může snadno dojít k havárii celého systému. Tudíž i aplikace musí do OS vstupovat pomocí rozhraní – tzv. **aplikačního programového rozhraní** – zkráceně **API** (Application Program Interface). API je množina běžných operací, které může OS pro aplikace vykonávat – příkladem je otevírání souborů, čtení znaků z klávesnice nebo ukončování programů.

Windows NT, OS/2, Windows 3.x využívaly různé API, a proto se vždy vývojáři museli rozhodovat, na které API se zaměří. Vzhledem k podílu na trhu se většina výrobců softwaru rozhoduje nejprve pro API Windows a teprve později přepracovávají své aplikace pro ostatní OS. Rychlý rozvoj aplikačních programových rozhraní API, který přinesl některým operačním systémům bohatou nabídku aplikací jiným chudý výběr, se ve Windows 95 Windows NT spojil do jediného API pod názvem Win32 API. To je velmi důležité, protože jakmile někdo přijde na trh s novou aplikací pro Windows 95, existuje i nová aplikace pro Windows NT (z počátku existovaly jisté problémy s kompatibilitou, ale po příchodu NT 4 se to hodně zlepšilo).

Windows 98 a NT 5 zdokonalily model ovladačů používaný v NT 4. Nový Win32Driver Model (WDM) zahrnuje některé služby jádra NT do Windows 98 pomocí virtuálního ovladače zařízení ntkernel.vxd. Tzn., že aplikace vytvořené pro Windows 98 budou fungovat i ve Windows NT5.

Kromě Win32 API obsahují Windows 98 i starší API předchozích OS (DOS a Windows 3.x).

4.5 Jádro

Srdce OS obvykle nese název **jádro**, který si vypůjčilo ve světě systému Unix. Jádro je správcem celého systému. Jádro ví o každém programu a o jeho požadavcích na paměť, který program je na popředí a který na pozadí, který proces má vyšší prioritu než ostatní, které programy právě tisknou apod.

4.6 Ovladače zařízení

Jádro neřídí nic přímo, ale pouze zprostředkovaně. Například nikdy nežádá klávesnici o kód stisknuté klávesy. Důvodem takového postupu je existence rozdílů mezi jednotlivými typy funkčně stejných zařízení. (klávesnice není v tomto případě nejvhodnější příkladem, protože jsou všechny víceméně kompatibilní; grafické karty, monitory nebo tiskárny jsou však něco jiného).

Kdysi bývaly OS navrhovány s podporou dvou nebo třech různých zařízení v každé kategorii hardwaru; nebyl to ovšem velký problém, protože stejně neexistovaly více než dvě nebo tři značky. S rostoucím množstvím síťových karet, tiskáren a dalších zařízení začalo být neúnosné začlenit do jádra každého OS podporu pro každý dostupný typ hardwaru – alespoň tedy pokud měly být jeho velikost pod 0,5 GB.

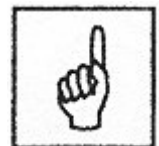
Asi před dvaceti lety začali programátoři navrhovat OS, které měly programy pro řízení jednotlivých typů periférií odděleny od jádra a nazvaly tyto krátké programy **ovladači zařízení**. Výhody ovladačů zařízení jsou zřejmé; mohou být kdykoli nahrazeny novějšími verzemi, třeba bez předchozích chyb, bez nutnosti měnit celý operační systém. Také je takto jednodušší podpora nových typů periférií. Jediným úkolem výrobce hardwaru je dodat spolu s novým zařízením i jeho ovladač; poté, co uživatel nainstaluje nový ovladač, je operační systém schopen okamžitě rozpoznat nový hardware.

Výrobci samozřejmě vytváří ovladače pro DOS a Windows, protože mají největší podíl na trhu. Podpora OS/2 a Windows NT přijde později anebo vůbec ne. To by se však alespoň pro NT mělo změnit s příchodem WDM a sjednocením ovladačů pro Windows 98 a NT 5.

Otázky a úkoly

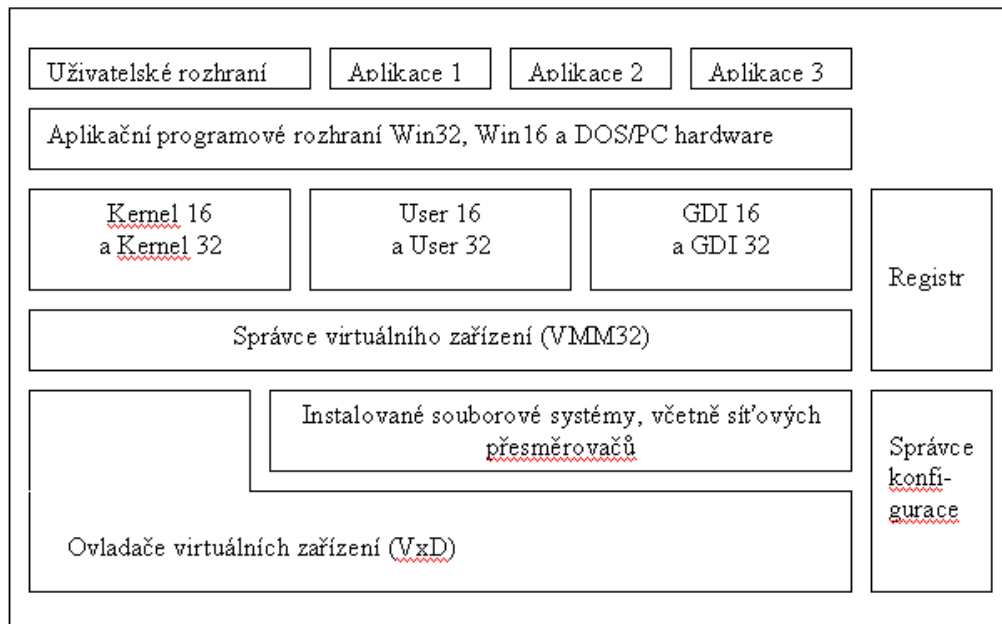
1. Přiřaď k názvům v levém sloupci správné výroky z pravého sloupce.

A) aplikační programové rozhraní	a) mohou být kdykoli nahrazeny novějšími verzemi
B) aplikace	b) správce celého systému
C) jádro	c) množina běžných operací, které může OS pro aplikace vykonávat
D) ovladače zařízení	d) nemohou a neumějí přistupovat přímo k hardwaru počítače



4.7 Součásti Windows 98

Struktura Windows 98 odpovídá struktuře obecného OS, pouze s několika doplňky. Prohlédněte si obrázek 2.



4.8 Moduly jádra

Úkolem mnoha částí Windows není přímá podpora aplikací, ale ovladačů. Samotné programy pro Windows spolupracují se šesti moduly – moduly jádra:

- Kernel16
- Kernel32
- GDI16
- GDI32
- User16
- User32

Kernel16 a Kernel32

Úkolem jádra je řídit systém a udržovat přehled o využívání paměti, periférií, přidělování priorit a podobně.

Jádro není novinkou ve Windows 98; ve skutečnosti bylo součástí několika posledních verzí. Protože ale nejstarší verze Windows byla navržena pro procesory 286, podporovalo pouze 16 bitové instrukce – proto mluvíme o této verzi jako o 16 bitové. Jednou z hlavních příčin komerčního úspěchu Windows 98 je fakt, že podporují procesory 386 a vyšší, a proto mohou být 32 bitové. Z toho důvodu je jádro přepracováno tak, aby mohlo využívat plné síly 32 bitů.

Kvůli podpoře starších aplikací pro Windows obsahují Windows 98 jak staré jádro, označované jako Kernel16, tak i novou verzi Kernel32.

GDI – Rozhraní grafických zařízení

Stejně jako dobrý ředitel, ani jádro nevykonává všechnu svou práci; část úkolů předává jednomu z pomocných programů, nazvanému GDI (Graphical Device Interface). Podobně jako jádro i GDI již existovalo v předchozích verzích Windows, a proto hovoříme o GDI16 a GDI32.

Již z názvu můžeme usuzovat, že GDI se specializuje na úkoly související s grafikou. Kdykoliv vložíte na obrazovku obrázek, je váš požadavek zpracován GDI. To řídí následující oblasti.

- rastrové obrázky
- barvy
- kurzory
- ikony
- písma
- grafické objekty (čáry, křivky apod.)

V další části uvidíme, že GDI musí evidovat každou barvu, kurzor, ikonu apod., a proto potřebuje pro ukládání informací o těchto objektech prostor. Tyto oblasti se nazývají systémové prostředky.

GDI je součástí Windows, která je například aktivována při jakékoli snaze o změnu barvy součástí Windows: okna, tlačítka apod. GDI má na starost písma – používáte-li mnoho znakových sad, „potí“ se právě GDI.



User16 a User 32

Množství operací spojených s činností Windows souvisí s ovládáním oken – klepání na tlačítka, vyvolávání nabídek, přetahování ikon, výběry ze seznamů apod. Všechny tyto činnosti jsou řízeny modelem nazvaným User. Stejně jako předtím používaly starší Windows 16bitovou verzi a novější pracují s 32bitovou verzí. Avšak i tato 32 bitová verze modulu User obsahuje mnoho 16bitového kódu, aby byla kompatibilní se staršími aplikacemi.

4.9 Správce virtuálních zařízení (VMM32) a ovladače VxD

Správce virtuálních zařízení (Virtual Machine Manager, VMM32) je velmi důležitou součástí Windows, která napomáhá jádru vykonávat jeho úlohy, definované v popisu OS. VMM32 není ovšem považována za součást jádra.

Správce virtuálních zařízení

Jedním z úkolů programátorů Windows 95 bylo navrhnout operační systém podporující programy systému DOS. Aplikace pro DOS se chovají jako rozmazlené dítě – chtějí všechn hardware jen pro sebe, protože se o něj neumí podělit. Pokud program pro DOS například nalezne sériovou zásuvku COM1, je docela pravděpodobné, že ji zablokuje pouze pro sebe, a to přestože ji vůbec nepotřebuje. Stejný problém nastává při přidělování paměti.





Programy pro DOS často hlídají i klávesnici a v každém okamžiku sledují stisk nějaké klávesy – pokud stisk klávesy zaregistrují, automaticky ho považují za stisk určený právě pro ně. Aplikace pro DOS nebývají obvykle navrženy pro multitasking.

Většina OS umožňujících multitasking aplikací pro DOS – OS/2, Windows NT, Windows 98 – obchází výše uvedený problém pomyslným rozdělením jednoho systému na několik počítačů, přesněji řečeno **virtuálních zařízení**. Program zodpovídající za řízení všech takových virtuálních počítačů se nejmenuje jinak než Správce virtuálních zařízení. Všichni tito správci využívají velké výhody procesoru 386 (a dalších procesorů navazujících na tuto řadu), nazývané **mapování stránek**.



Tento trik spočívá v následujícím postupu. Řekněme, že pracujete s počítačem se 16 MB paměti; systém DOS ale dokáže využít pouze 1 MB. Tento 1 MB musí ovšem být první megabyte paměti – a proto je tak výhodně mapování stránek. Procesor totiž může použít jakýkoliv volný megabyte paměti – třeba blok adres mezi 3 MB a 4 MB – a pomyslně ho oddělit od zbytku systémové paměti. Následně procesor přidělí tento blok paměti programu pro DOS a sdělí mu: „Zde je k dispozici počítač, který má paměť pouze 1 MB. Musíš mi věřit.“ Kdykoliv potom chce tato aplikace přistoupit například k adrese 7000, správce mapování provede jednoduchý převod adresy: „Jestliže tento program vyžaduje přístup na adresu 7000, musím k ní připočíst základ 3MB.“ Požadavky na přístupy do paměti jsou tímto způsobem zcela transparentně převáděny na odpovídající adresy fyzické paměti.

Správa virtuální paměti

Program zajišťující správu mapování paměti a její přidělování aplikacím Windows 95/98 se nazývá VMM32. Tento program provádí kromě správy virtuálních zařízení i řízení virtuální paměti (správce virtuální paměti je částí správce virtuálních zařízení). **Virtuální paměť** umožňuje OS nabízet aplikacím více paměti, než kolik jí má systém fyzicky k dispozici; pokud např. máte počítač se 16 MB paměti, ale chcete spustit aplikace čítající dohromady 40 MB, systém vám to dovolí.

Správce virtuálního zařízení umožňuje uživateli využívat více paměti tak, že pracuje s diskovým prostorem jako s operační pamětí. Chcete-li spustit program přesahující kapacitu vaší paměti, Správce virtuálních zařízení zkontroluje všechny bloky paměti již dříve přidělené aplikacím a rozhodne se podle stupně jejich využití.

Systémová paměť v počítačích s procesory 386 a vyšších je dělena do bloků o velikosti 4 KB, nazývaných **stránky**. Ta část Správce virtuálních zařízení, která zodpovídá za řízení paměti – nese název **Správce virtuální paměti** – nevyužívá koncept programů a paměti, ale **stránek** a paměti. Má přehled jak o

Počítačové systémy, periférie, operační systémy

počtu stránek, přidělených jednotlivým programům, tak i o jejich umístění v operační paměti, případně na pevném disku.

Například jste před několika hodinami spustili aplikaci Excel, ale již delší dobu jste ji nepoužili. Správce paměti ví, že dlouho nebyly tyto přidělené stránky použity, a proto je přesune z paměti na disk do oblasti určené k ukládání virtuální paměti, nazvané stránkový soubor. Tato operace uvolní část fyzické operační paměti a umožní vám použít více paměti pro jiné aplikace; kromě krátkodobého zvýšení aktivity vašeho pevného disku byste neměli zpozorovat žádný rozdíl.

Když se později rozhodnete použít svůj Excel, budou se Windows snažit přistoupit do paměti na místo, které bylo tomuto programu přiděleno. Jakmile se procesor pokusí přečíst instrukce ze stránky vlastněné programem Excel, objeví, že daná stránka není ve fyzické paměti. Tím se vygeneruje hardwarová chyba, nazvaná chyba stránky (programy není možné vykonávat z pevného disku, ale pouze z paměti). Tato chyba aktivuje správce virtuální paměti, který musí nalézt prostor ve fyzické paměti a nahrát z disku požadované stránky. Nyní zřejmě dojde k problému s nedostatkem fyzické paměti, a proto bude správce virtuální paměti znovu muset rozhodnout, která aplikace má přidělenou paměť a nevyužívá ji.



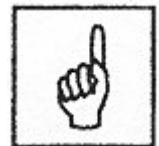
Virtuální paměť umožňoval již systém Windows 3.x, avšak ve statické formě. Mohli jste využívat buď dočasný, anebo trvalý odkládací soubor (nebo také žádný). Jeho velikost však byla neměnná. Windows 95 přinesly **dynamickou virtuální paměť**. Windows zjišťují potřebnou velikost odkládacího souboru podle aktivity disku. Jestliže systém v danou chvíli vyžaduje velký odkládací soubor, systém pro něj na disku vyhradí více místa. Pominou-li důvody pro udržování této velikosti, Windows soubor podle potřeby zmenší. Tuto redukci odkládacího souboru lze postřehnout jako činnost disku ve chvíli, kdy počítač právě nepoužíváte.

Virtuální ovladače zařízení (VxD)

Přestože je schopnost rychle, snadno a účinně mapovat a stránkovat paměť jedním z důležitých základů pro podporu multitaskingu v prostředí DOS, není tato schopnost jedinou potřebnou. Programy pro Dos nejenže si svévolně přidělují paměť, ale také bez ohledu na potřeby dalších programů svévolně blokují další hardware. Klíčem k pochopení významu pojmu „kradení“ hardwaru je výraz vstupní/výstupní adresy.

Vstupní/výstupní adresy neboli **I/O adresy** jsou prostředníkem pro komunikaci programů s hardwarem, například klávesnicí, grafickou kartou, síťovou kartou apod. Jsou podobné adresám paměti pouze s tím rozdílem, že adresy paměti přistupují pouze do paměti, zatímco I/O adresy přistupují k celé skupině zařízení. Např. aplikace, která chce komunikovat s klávesnicí, sleduje I/O adresu 60 – tato adresa je standardní pro klávesnici.

Programy pro DOS působí potíže svým přístupem ke klávesnici (k I/O adrese 60), protože se snaží o převzetí jakékoliv aktivity dříve, než ji převezme jiná aplikace (třeba operační systém). A právě k omezení aplikací pro DOS při



Počítačové systémy, periférie, operační systémy

jejich snaze o přebírání hardwaru existuje virtuální ovladač zařízení – VxD pro Windows 98.

Virtuální ovladač zařízení je program, který je procesorem pověřen spravovat určitou I/O adresu. V našem příkladě je virtuální ovladač klávesnice spojen s adresou 60. Jakkmile potom chce nějaká aplikace přistoupit přímo na adresu 60, virtuální ovladač klávesnice se aktivuje a chová se stejně jako zastupované zařízení.

Virtuální ovladač klávesnice je poměrně snadné si představit: jakmile je aktivován programem pro DOS požadavek na sdělení kódu stisknuté klávesy, prověří nejprve, jedná-li se o program běžící na popředí. Jestliže daný program skutečně běží na popředí, předá mu ovladač vše, co bylo sejmuto z klávesnice. V opačném případě, pokud běží program pro DOS na pozadí, nepředá mu ovladač jako správce klávesnice nic.



Pomocí položky Správce zařízení v Systému okna Ovládacích panelů a dvojitým poklepnutím na určité hardwarové zařízení lze vyhledat informace o virtuálních ovladačích vybraného hardwaru.

Otázky a úkoly



2. Odpověz na otázky: Jaké úkoly plní ?

- a) jádro Kernel 16 (32)
- b) GDI
- c) User 16 (32)
- d) virtuální zařízení, virtuální paměť

3. Najdi v Ovládacích panelech ikonu Systém a zobraz kartu Výkon. Pomocí této karty zjisti velikost paměti RAM a kolik volného místa na pevném disku má Správce virtuální paměti k dispozici pro vytváření odkládacího prostoru virtuální paměti.

4. Máme k dispozici počítač s OS Windows 98 a s velikostí operační paměti RAM 16 MB. Proč se na zapnutém počítači vždy po určité chvíli rozsvítí kontrolka oznamující práci pevného disku, když s počítačem momentálně nepracujeme?

4.10 Registr

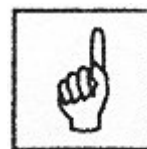
Ve Windows 3.x existovali soubory *.INI, které obsahovali základní nastavení a konfigurační informace. Windows 95 již obsahovaly Registr, který byl určitým zlepšením oproti dosavadnímu přístupu Windows 3.x. Navíc obsahuje informace generované vnitřně neboli dynamicky systémem Windows 98; např. při každém startu systému se vytvoří seznam všech hardwarových součástí počítače (tyto informace jsou samozřejmě nestálé a mohou se měnit při každém startu systému, ale významně napomohou při odstraňování složitých problémů s Windows). Více o registrech v samostatné kapitole.

4.11 Instalované souborové systémy – IFS

Každý OS musí být schopen přistupovat k zařízení na úschovu dat. Např. v systému DOS má každé takové zařízení název: jedno písmeno abecedy následované dvojtečkou (A: - disketová mechanika, C: - systémový disk apod.). Kromě toho musí být data v každém takovém zařízení určitým způsobem organizována. Od počátků systému DOS jsou data organizována stejným způsobem, jakým byl navržen pro správu dat na pružných discích – systémem tabulek FAT. V systému DOS však neexistuje program pro správu souborů FAT, jelikož systém FAT je se systémem DOS tak dokonale propojen, že žádný způsob pro údržbu FAT nebyl implementován.

V novém OS je ale nutné nabídnout mnohem výkonnější a pružnější systém souborů. Například FAT nepracoval dobře v sítích a s CD-ROM; výsledkem bylo jejich složité používání v tomto operačním systému. Cílem, ale nebylo navrhnout nový samostatný systém souborů, který by dobře sloužil všem aplikacím. Lepším řešením je vybudovat takovou otevřenou architekturu systému souborů, která by umožňovala snadné používání a instalaci CD-ROM, sítí, optických disků, velkokapacitních pevných disků apod.

Určitý způsob řešení se objevil ve formě Instalovatelného systému souborů (IFS – Installable File System). Poprvé byl použit v roce 1989 v operačním systému OS/2 verze 1.2, který nabízel svým HPFS alternativu k FAT. Díky své implementaci ve Windows 98 pracuje současný IFS jako zásuvka, do které mohou být připojeny libovolné systémy souborů FAT, přesměrovače souborů v lokálních sítích, i systémy souborů mechanik CD-ROM. Dobrou zprávou pro uživatele však je možnost využít názvů souborů o délce několik set znaků (nikoliv jen 8, jak tomu bylo u starších verzí Windows).



Správce konfigurace

Aby byl OS schopen pracovat spolehlivě, musí mít dokonalý přehled o každém zařízení v počítači. Jak sdělíme operačnímu systému, jaký typ zařízení je v daném počítači? Nabízí se jednoduché řešení: vytvořit pevný seznam zařízení a ovladačů a požadovat, aby ho uživatel v případě jakékoliv změny aktualizoval. Tento způsob používaly všechny předchozí OS, jako DOS, OS/2 nebo Windows NT.

Plug and Play

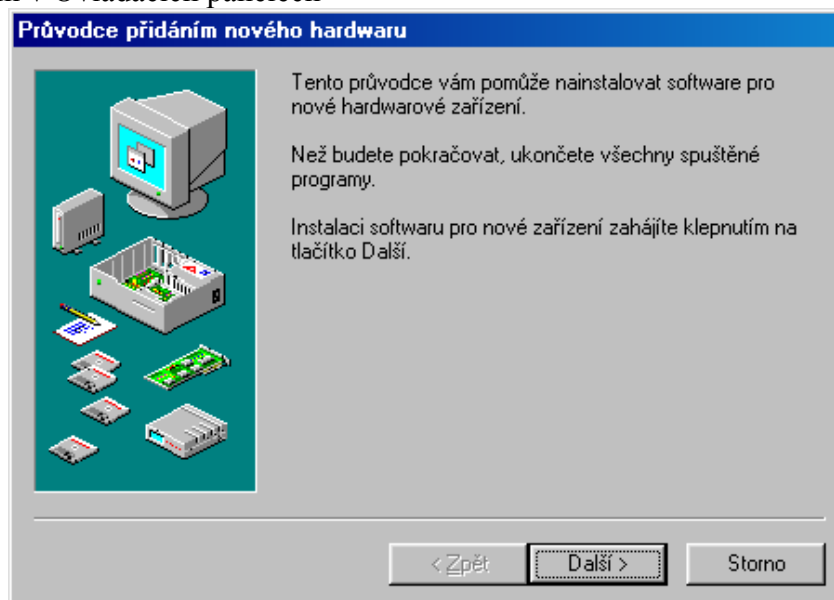
S nástupem technologie Plug and Play ve Windows 95/98 pracuje vše jinak. Na kartách nejsou již žádné přepínače ani propojky, jak tomu bylo u jejich předchůdců předešlých OS. Hardware kompatibilní s Plug and Play je konfigurovatelný pouze softwarově.

Technologie Plug and Play přináší velkou výhodu v tom, že provádí instalaci hardwaru i softwaru automaticky – nebo alespoň tak automaticky, jak je to možné (viz obr. 3). Počítač s technologií Plug and Play nastavuje při každém zapnutí všechna instalovaná zařízení. Při každém spuštění jsou jednotlivým zařízením přidělena vlastní přerušení, takže mezi nimi nemůže dojít k žádnému konfliktu. Výsledné informace o nastavení jsou poté předány systému Windows, který je dále předá ovladačům VxD.

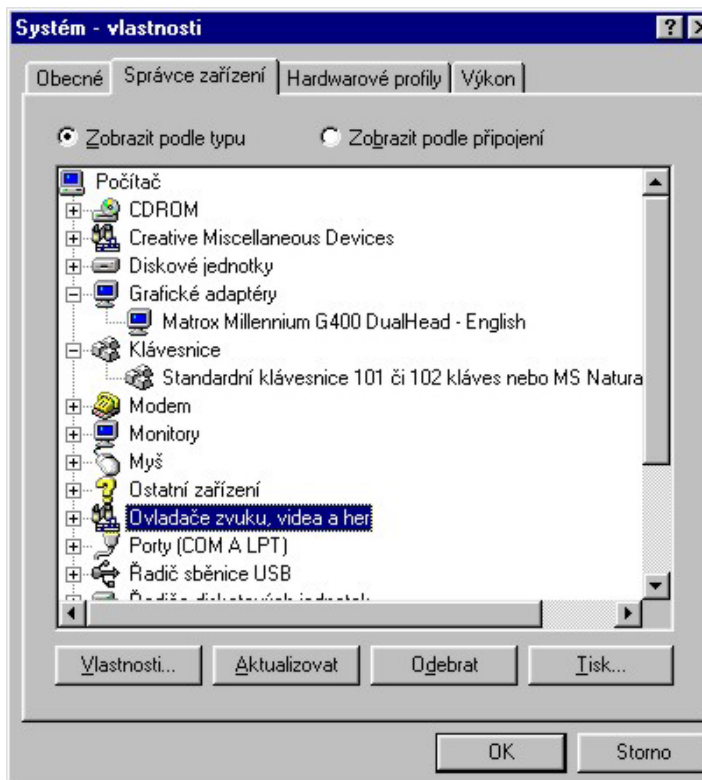
Počítačové systémy, periférie, operační systémy

Ústředním objektem této složité operace je program nazvaný Správce konfigurace. Tato aplikace není pouze nedílnou součástí Windows, ale také součástí systému BIOS počítače kompatibilního s technologií Plug and Play. Využití technologie Plug and Play má však háček, v tom, že kromě OS musí tuto technologii podporovat jak základní deska počítače, tak i jeho komponenty.

Se Správcem konfigurace se budete nejčastěji setkávat prostřednictvím Správce zařízení v Ovládacích panelech



Využití Správce konfigurace pomocí Průvodce přidáním nového hardwaru



Využití Správce konfigurace pomocí Správce zařízení

Otázky a úkoly:

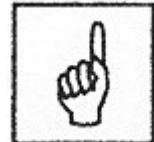
5. Urči dle Správce konfigurace grafickou kartu svého počítače.
6. Jakou výhodu přináší technologie Plug and Play ?



4.12 Registr

Co je to Registr ?

Windows 95/98 si pamatují údaje o hardwaru PC a uživatelích počítače pomocí Registru. **Registr** je v podstatě databázi informací v ASCII kódu, z hlediska uložení na disk rozdělenou na dvě, příp. na tři části:



- soubor **SYSTEM.DAT**, kde jsou zapsány údaje o nastavení použitého hardwaru počítače,
- soubor **USER.DAT**, sloužící k úschově informací o uživatelích, kteří s daným OS na tomto PC pracují – kupř. uživatelská práva, nastavení pracovní plochy apod. (pokud je počítač připojen k síti, může být USER.DAT uložen i na centrálním počítači – serveru)

soubor CONFIG.POL obsahuje řídicí informace pro systémové a uživatelské nastavení (např. lze vytvořit postup, jakým mohou uživatelé nastavovat své displeje nebo sdílená zařízení) – jakékoliv údaje z tohoto souboru mají přednost před nastavením v souborech USER.DAT a SYSTEM.DAT; CONFIG.POL není pro instalaci WINDOWS 98 povinný.

Tyto soubory jsou pro práci Windows klíčové, a proto jsou maximálně chráněny. První dva jsou uloženy v adresáři, do něž byly nainstalovány Windows (nejčastěji C:\Windows), nebo v kořenovém adresáři. Třetí v pořadí se nachází v přihlašovací adresáři síťového serveru.

Některé údaje Registru jsou v každém okamžiku uloženy na disku, zatímco jiné jsou umístěny v operační paměti a dynamicky zobrazují provedené změny. Například pokud změníte nastavení optického disku podporujícího technologii Plug and Play, okamžitě se tyto změny objeví v Registru.

Proč je vlastně Registr uložen ve dvou souborech (USER.DAT a SYSTEM.DAT)?

Z dobrého důvodu. Mnoho uživatelů Windows používá lokální síť, takže mohou přecházet mezi různými počítači připojenými do sítě. Windows 98 umožňují, aby příslušný soubor SYSTEM.DAT byl umístěn na samotném počítači, zatímco soubor USER.DAT na síti. Jakmile se poté, uživatel přihlásí do sítě, Windows 98 najdou na síti jeho soubor USER.DAT a patřičně změní uživatelské nastavení. Každý tedy může své nastavení používat na libovolném počítači.

K čemu je Registr?

Protože Registr obsahuje všechny systémové a uživatelské informace Windows 95/98, lze zálohovat pouze soubory USER.DAT a SYSTÉM.DAT. Díky Registru stačí pouze přenést tyto soubory do jiného adresáře nebo na disketu, vymyslet pro ně nějaký název, a když se něco nepodaří, jednoduše je obnovit.

Co obsahuje Registr?

Registr sám neobsahuje nic – je to pouze uživatelské rozhraní pro databázi s konfigurací systému, které umožňuje zobrazit dané informace v čitelné podobě.

Pokud Registr exportujete vytvoří se textový soubor s příponou .REG, který ovšem jen vytváří propojení s konfiguračními informacemi, a ne přímo s konfigurací. Bez obav bychom mohli na počítači odstranit všechny soubory.REG, protože systém vytvoří z informací v souborech .DAT nový Registr (nový soubor .REG se vytvoří až při dalším exportování Registru). Pokud naopak smažeme nebo přejmenujeme soubory SYSTEM.DAT a USER.DAT, nebude systém při startu schopen načíst Registr a bude muset tyto soubory vytvořit ze záložních kopií obnovovaných při každém úspěšném startu.

Jak to bylo dříve?

Windows 95/98 umožňují spouštěné programů napsaných pro předešlé operační systémy DOS a Windows 3.x. Tyto aplikace však neumějí s Registrem pracovat, a tak musí Windows 95/98 umožnit použití předchůdců dnešní registrační databáze:

- souborů CONFIG.SYS a AUTOEXEC.BAT pro konfiguraci dosových a windowsových (rozuměj verze 3.x) programů
- souborů INI určených pro definici parametrů Windows 3.x

CONFIG.SYS a AUTOEXEC.BAT

Oba jsou první variantou konfiguračních souborů. Byly používány v DOS Windows 3.11, Windows 95/98 je ke své činnosti nepotřebují, nicméně jsou-li v kořenovém adresáři disku přítomny budou Windows 95/98 provádět příkazy v nich zapsané.

CONFIG.SYS byl určen především pro konfiguraci hardwaru PC, sloužil tedy k nahrávání potřebných ovládačů. Ty pracovaly v 16bitovém režimu a pro dnešní Windows jsou brzdou.

V CONFIG.SYS nahraném v kořenovém adresáři disku Windows 95/98 by se neměl nalézt žádný příkaz DEVICE (jímž se ovladače zařízení aktivovaly) – Windows 95/98 používají své virtuální ovladače;

AUTOEXEC.BAT sloužil k nastavení cest, nahrání rezidentních programů a definic systémových proměnných. I tyto úkoly zvládají Windows 95/98 s pomocí Registru a v AUTOEXECu je nepotřebujeme definovat.

V CONFIG.SYS nebyly umístěny pouze příkazy DEVICE, ale ještě další definiční řetězce, nutné k bezchybnému běhu programu DOS.

Správné načtení těchto příkazů do operační paměti ve Windows 95/98

Počítačové systémy, periférie, operační systémy

je zajištěno souborem IO.SYS. Ten se však od původního (dosového) liší.

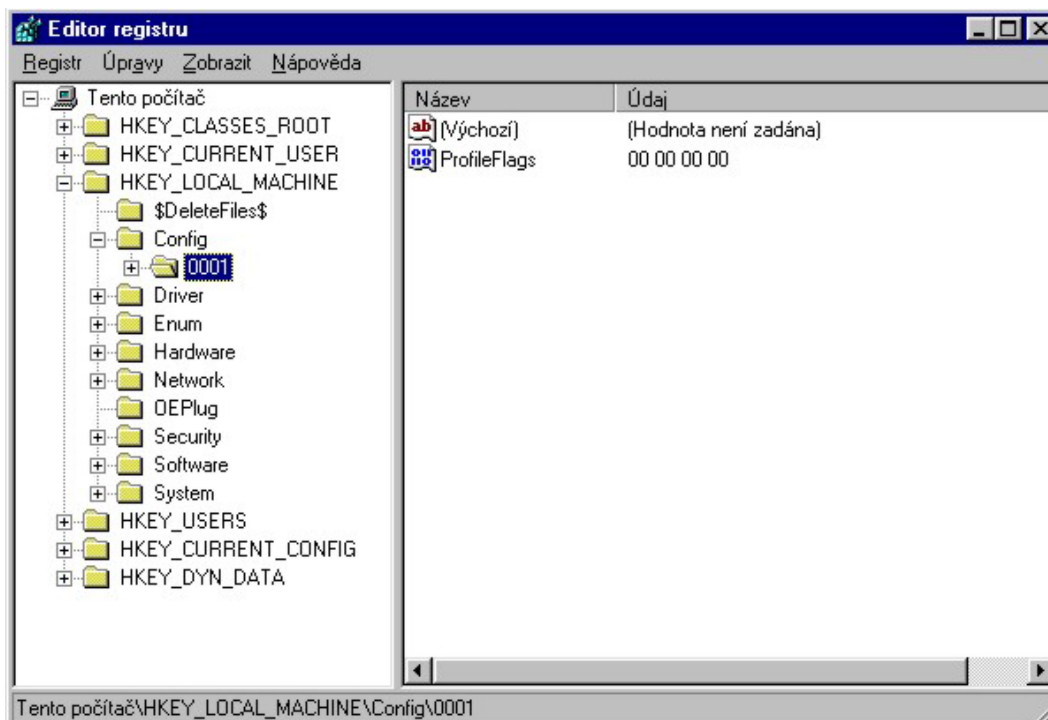
Inicializační soubory Windows 3.x

Jak již bylo uvedeno, ani aplikace napsané pro práci nad Windows 3.x neznají pojem Registr. Kvůli nim jsou v základním adresáři Windows 95/98 (C:\Windows) umístěny inicializační soubory SYSTEM.INI a WIN.INI. Jejich údaje jsou při instalaci Windows 95/98 (přes původní Windows 3.x) nahrány do Registru, ale oba soubory zůstávají na disku – konfigurační data z nich čtou starší aplikace.

Otázky a úkoly

7. Jak bys stručně charakterizoval Registr ?
8. Jaké dva základní soubory Registr zahrnuje ?
9. Co obsahují tyto soubory ?

Editor registru



4.13 Popis registrační databáze

K práci s registračními soubory je určen **Editor registru** – soubor **REGEDIT.EXE**. Protože je programem velmi nebezpečným, nenajdete jej v žádné nabídce, musíte jej spustit přes nabídku Start/Spustit/Regedit nebo si jej můžete najít v adresáři Windows a umístit do nabídky Start, příp. na pracovní plochu (nedoporučuji !!!).



Obrázek ukazuje obrazovku Regeditu. Ta má ve své levé části šest základních položek registrační databáze, pro něž se používá název **klíče** (klíče mohou

Počítačové systémy, periférie, operační systémy

obsahovat podklíče, podobně jako existují adresáře a podadresáře). Každý z nich začíná označením „Hkey_“ a představuje podvětev stromové struktury dat určené k registraci tzv. **hodnot** (např. klíč Mouse je podvětví Hkey_Local_Machine obsahující hodnoty pro nastavení myši). Každá hodnota se skládá ze dvou částí: názvu, který danou hodnotu označuje, a údaje, obsahujícího příslušné konfigurační informace. Hodnoty mohou mít jeden ze dvou formátů:

- binární (nebo hexadecimální vyjádření binárního čísla) – použitý pro většinu hardwarových informací,
- řetězec (běžně čitelný text) – užítý pro většinu softwarových informací.

Funkce podvětví Registru

Jak již bylo uvedeno, šest podvětví Registru si mezi sebe rozdělilo konfigurační informace o počítači a jeho umístění v síti. Jejich význam je následující“

- **Hkey_Classes_Root** - je klíč popisující nastavení programů – např. asociace (jaká přípona patří k programu), informace o OLE apod.
- **Hkey_Current_User** - popisuje uživatele, který je momentálně přihlášen (barvy, zvuky, aplikace aj.).
- **Hkey_Local_Machine** – zde jsou uložena konfigurační data místního počítače (informace o ovladačích, připojení k síti, modemech, přiřazení souborů ...)
- **Hkey_Users** – klíč soustřeďuje údaje o všech uživateli, kteří se mohou k PC přihlásit. Jsou zde obecná nastavení a podklíče pro každého uživatele.
- **Hkey_Current_Config** – ukazuje do podvětve Hkey_Local_Machine\Config, která obsahuje aktuální konfiguraci hardwaru připojeného k počítači.
- **Hkey_Dyn_Data** – obsahuje dynamické informace o zařízeních Plug and Play připojených k počítači. Informace obsažené v této podvětvi jsou uloženy v paměti RAM, takže nemusíte po změně restartovat počítač.

4.14 Úpravy v lokálním Registru

Hlavní problém úprav Registru spočívá ve vyhledávání požadovaných klíčů a hodnot. Operace úprav mohou obsahovat tyto kroky:

- zálohování Registru
- úpravy nastavení
- obnovení Registru

Jelikož práce s Registrem vyžaduje jisté zkušenosti a studium několika mnohem obsáhlejších statí odborné literatury (viz odkaz na literaturu za kapitolou), než přináší tato příručka, **varuji před nějakým konkrétním zasahováním do Registru na základě uvedených informací**, které jsou jen stručným náhledem do jejich problematiky, nikoliv postupem pro práci s Registry. Uvedené informace mají sloužit pro vytvoření představy o smyslu a využití Registru – nejedná se o poučení k práci s ním. Z toho důvodu se

Počítačové systémy, periférie, operační systémy

v textu stručně podíváme na uvedenou první a třetí položku úprav, tj. zálohování a obnovení Registru.

Zálohování Registru

Patřím k dobrým vlastnostem Windows 98, že při startu automaticky zkontroluje Registr, zda není poškozen, a každý den vytvoří záložní kopie systémových souborů. Dojde-li k poškození Registru, Windows 98 jej automaticky obnoví ze zálohy, vytvořené předchozí den. Není-li tato záloha k dispozici, systém se pokusí Registr opravit.

Před jakýmkoliv zásahy do Registru je velmi vhodné exportovat jeho obsah na bezpečné místo (buď na místní disk, nebo pro důkladné zabezpečení přímo na Spouštěcí disketu). V případě neobnovitelného zásahu do nastavení počítače můžete pomocí záložní kopie Registru obnovit původní hodnoty. Pokud kopii nemáte a narušení je příliš vážné, budete muset přeinstalovat celý operační systém.



Zálohování Registru lze provést několika způsoby: pomocí programů SCANREG.EXE či SCANREGW.EXE. Oba programy fungují v podstatě stejně. SCANREG je verze pro DOS, zatímco SCANREGW se spouští ve Windows. Každý z nich má zvláštní funkce.

Dalším způsobem vytvoření záložní kopie Registru je jeho exportování na jiný disk nebo do jiného adresáře. Při exportování Registru se ukládá jeho obsah ve formátu ASCII. V podstatě se převádí obsah souborů USER.DAT a SYSTEM.DAT do formátu, který můžeme prohlížet, prohledávat a ukládat. V tomto formátu nejsou data pro systém čitelná až do okamžiku, kdy je zpětně neimportujeme. Mezitím můžeme dělat do informací v Registru libovolné změny.

Pokud bychom se pokusili obnovit Registr z poškozené verze exportovaného souboru, výsledkem by byla poškozená verze Windows, která nejde spustit. naštěstí Registr se automaticky zálohuje při prvním spuštění systému. Pomocí programu SCANREG lze Registr obnovit z této verze. Musíme však samozřejmě začít s požadovanou úpravou Registru znovu a patrně přijdeme o veškeré změny, které jsme v Registru provedli od prvního spuštění. Vyhneme se, ale kompletní nové instalaci všeho, co jsme dosud složitě nainstalovali.

Obnovení Registru

Žádnou skutečnou změnu nastavení nemůžeme vrátit, pokud nevytvoříme zálohu exportem na jiné místo; jakmile importujeme obsah Registru nebo

Počítačové systémy, periférie, operační systémy

některé podvětvě, budou jakákoliv předchozí nastavení ztracena. Chceme-li uložit kopii konfigurace, exportujeme původní obsah Registru před importováním nového.

Pro obnovení obsahu Registru do původního stavu existují dva způsoby: příkaz Importovat registr v programu Editor registru nebo je možné aktivovat ikonu záložní kopie.

V obou případech zaměníme současnou konfiguraci počítače za nastavení, které jste před časem exportovali. Při exportování dochází k uložení kopií souboru .DAT s aktuálním nastavením. Pokud později obsah Registru importujeme, zaměníme novou aktuální konfiguraci původní exportovanou.

4.15 Slovníček pojmů



Multitasking – režim práce, ve kterém je mikroprocesor schopen zpracovat požadavky od více programů, neboli na počítači běží současně několik programů. K je ho úspěšné realizaci je třeba, aby multitasking zvládl i operační systém a často též aplikační programy. V zásadě existují dva modely multitaskingu:

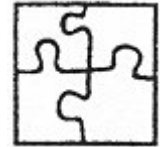
Kooperativní je starší a je prováděn tak, že si přístup k mikroprocesoru postupně „předávají“ jednotlivé aplikace. Pokud nějaký program zhavaruje, nemůže předat řízení mikroprocesoru dál a počítač se „zakousne“.

Preemptivní způsob je novější. Zde je předávání mikroprocesorového času prováděno operačním systémem. Havárie aplikace se neprojeví na práci ostatních programů.

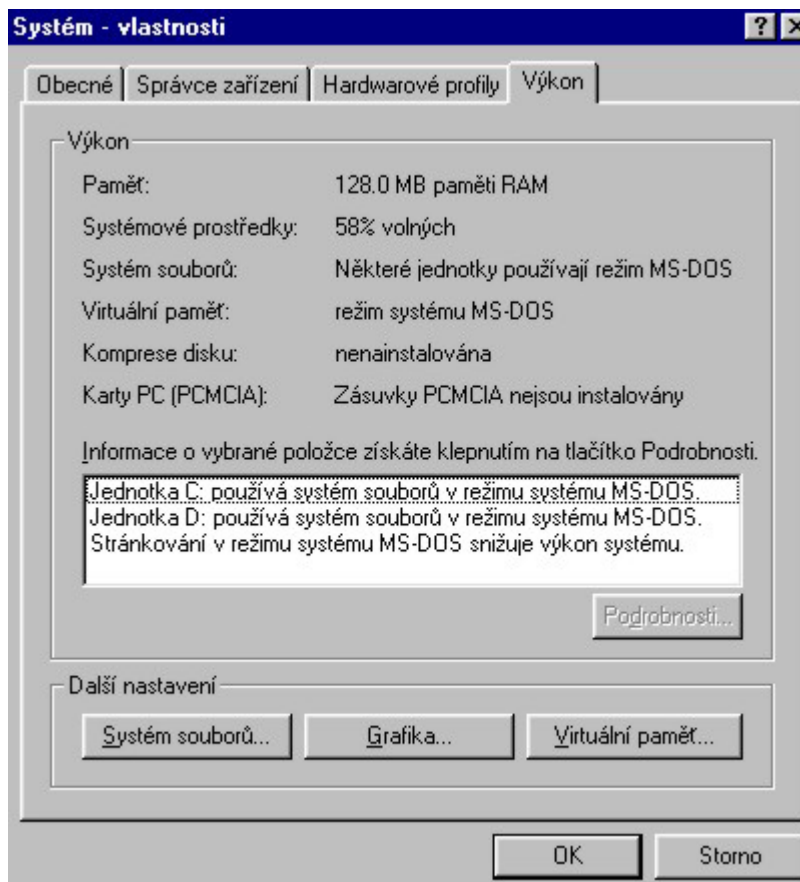
Virtuální počítač (virtual machine, VM) je prostředí v paměti, které z pohledu aplikace vypadá jako samostatný počítač, se všemi prostředky fyzického počítače, které aplikace potřebuje k práci. Windows 98 používají jeden VM, tzv. systémový VM, na němž běží všechny systémové procesy. Každá Win 32 aplikace pak běží na samostatném VM, zatímco aplikace 16bitové Windows sdílejí společný Win16 VM. Každá MS-DOS aplikace má vlastní VM.

OLE (Object Linking and Embedding) – technologie, která umožňuje sdílet data mezi aplikacemi pracujícími s OLE. Pokud používáme aplikaci využívající výhod technologie OLE, můžeme vytvářet složené dokumenty, které mohou obsahovat různé typy informací; data lze zobrazovat a upravovat bez spouštění jiných aplikací. OLE verze 2:0 je obsažena ve Windows 98, která vylepšuje předchozí standart OLE 1.0. Poskytuje služby pro sdílení objektů OLE (jednotek dat) a další funkce potřebné pro manipulaci s těmito daty.

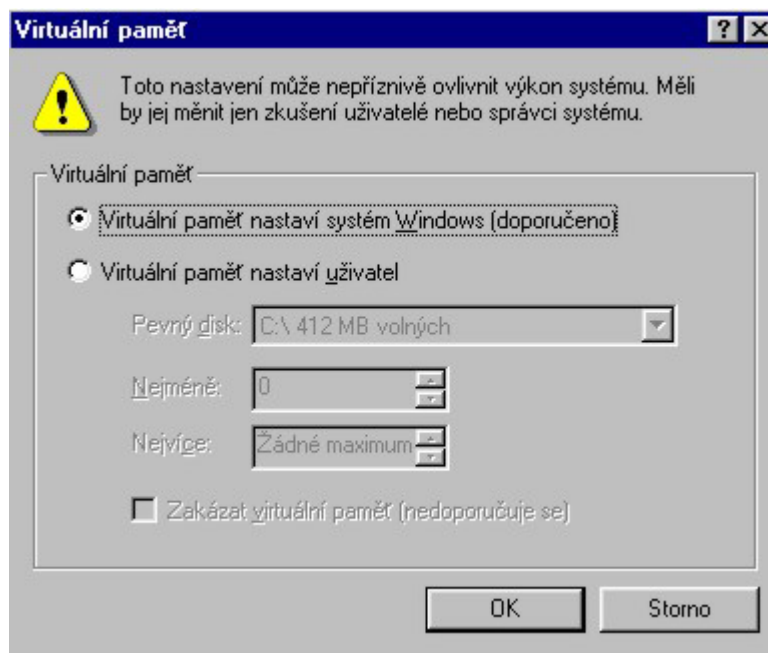
4.16 Řešení a odpovědi



1. *Přiřazení: A – a, B – d, C – b, D – a*
2. *Úkoly komponentů:*
 - a) *Jádro Kernel 16(32) řídí systém a udržuje přehled o využití paměti, periférií, přidělování priorit apod.*
 - b) *GDI se specializuje na úkony související s grafikou prostředí Windows (barvy, písma, obrázky apod.).*
 - c) *User 16 (32) řídí činnosti spojené s ovládáním oken (klepání na tlačítka, výběry ze seznamů, přetahování ikon aj.)*
 - d) *Virtuální zařízení zodpovídá za řízení tzv. virtuálních počítačů. Virtuální paměť umožňuje OS nabízet aplikacím více paměti, než kolik má systém fyzicky k dispozici.*
3. *Velikost operační paměti RAM můžeme určit přes Ovládací panely z karty Výkon – viz obrázek 6 – z něhož lze vyčíst velikost 128 MB (bude se lišit dle užitého počítače). Kliknutím na tlačítko Virtuální paměť... se nám zobrazí okno - viz obrázek 7 – a v něm lze zadat minimální a maximální velikost odkládacího prostoru virtuální paměti z dostupné kapacity volných megabajtů pevného disku (v našem příkladu 479 MB).*



Karta Výkon



Dialogové okno Virtuální paměť

4. *Budete-li používat Windows 98 na počítači s 16 MB RAM zaznamenáte správu virtuální paměti. Šestnáct megabajtů zřejmě nestačí pro umístění celého systému do operační paměti. Kontrolka pevného disku se každých několik sekund rozsvěcuje, protože nepochybně probíhá přesun stránek mezi diskem a pamětí.*
5. *Na počítači, jehož ovladače jsou zobrazeny na obrázku 4 je to Matrox Millennium 6400 Dual-Head.*
6. *Technologie Plug and Play umožňuje automatickou instalaci (nalezení) softwaru a hardwaru.*
7. *Registr je databází informací v ASCII kódu o nastavení místního počítače.*
8. *Dvěmi základními soubory jsou SYSTEM.DAT a USER.DAT.*
9. *SYSTEM.DAT obsahuje údaje o nastavení použitého hardware počítače. USER.DAT uchovává informace o uživateli, kteří s daným počítačem pracují.*

5 Internet a vyhledávací služby

Globální počítačová síť Internet je dnes schopná nabídnout mnoho užitečných služeb. Jejich zvládnutí a využití dnes již není až tak velkým technickým problémem, největší překážkou se stává spíše otázka orientace uživatelům ve značně rozsáhlém a mnohdy nepříliš přehledném světě Internetu.

Cílem této kapitoly je seznámit začínající uživatele s Internetem, principy jeho fungování a možnostmi vyhledávání informací v této celosvětové síti.

V současné době je na trhu mnoho publikací o Internetu, způsobu připojení, tvorbě webových stránek, adresářů stránek.

Další informace je možné nalézt na samotných webových stránkách. Například soustavu přednášek, seriálů a článků Jiřího Peterky ze světa počítačů najdete na adrese: <http://archiv.czech.net/>.



5.1 Co je to Internet

V této části se dozvíte:

- ▶ co byste měli umět a znát, než se pustíte do vyhledávání informací
- ▶ co je to Internet a kdy a proč vznikl
- ▶ jak funguje Internet

. Nároky na uživatele

Vyhledávání na Internetu vyžaduje jen minimální znalosti operačního systému Windows (protože běžné typy internetových prohlížečů pracují právě pod tímto operačním systémem). Práci s Internetem zvládne i naprostý začátečník po velmi krátkém seznámení se s počítačem.

Než tedy začnete pracovat s Internetem, měli byste:

- zvládat ovládání myši
- mít přehled na klávesnici a monitoru
- umět se pohybovat v otevřeném okně pomocí svislých i vodorovných posuvníků
- orientovat se v řádku nabídek a ikon
- znát základy psaní a editace textu



Kdy a jak vznikl Internet

Na počátku Internetu byla americká armáda. V polovině šedesátých let hledali Američané způsob propojení důležitých vojenských pracovišť, které by odolalo i v situaci válečného napadení. Na tuto prvotní síť ARPANET se postupně napojily i civilní instituce, takže vojáci si svou část oddělili a zbytek dali k dispozici univerzitám.

Svou současnou podobu získal Internet až po určité době postupného vývoje. Během této doby se měnily i internetové služby, a s nimi pak i celkový styl práce uživatelů. Dříve existovalo více různých služeb a také více nástrojů a způsobů ovládání, které uživatelé museli zvládnout. Později se ale Internet otevřel komerčnímu využití, mimo akademickou sféru. Přestalo platit, že uživatelé se musejí přizpůsobit Internetu, ale právě naopak, služby se musely přizpůsobit svým „zákazníkům“. Takové zjednodušení práce přilákalo další uživatele, proto má dnes Internet tolik příznivců.

Některé dříve používané služby prakticky zanikly (například služby Gopher a WAIS), a jejich místo zaplnila populární služba World Wide Web, schopná nabídnout jak vysokou uživatelskou atraktivnost (hezké a líbivé stránky), tak i snadnost svého ovládání.

Co je to Internet a jak funguje

Internet je celosvětová počítačová síť složená z menších i velkých počítačových sítí a jednotlivých počítačů.

Internet jsou také lidé, kteří jej vytvářejí i využívají. Většinou se stejným pojmem označují dvě nejzákladnější služby – elektronická pošta a World Wide Web.



Internet jako informační zdroj

Internet jako takový sám o sobě není žádným informačním zdrojem, sám žádné informace neposkytuje, podobně jako třeba veřejná telefonní síť. V souvislosti s Internetem vyvstává řada otázek. Například:

- Kdo je majitelem Internetu?
- Kdo povoluje a schvaluje zveřejnění informací v Internetu?
- Komu je třeba nahlásit zpřístupnění informací?
- Existuje v Internetu centrální evidence informací (obdoba telefonního seznamu s přehledem všech účastníků?)
- Jsou všechny informace v Internetu správné?
- Jak a kde najít hledané informace?



Vlastnické vztahy k Internetu

Internet nemá majitele ani ředitele, neexistuje ani žádná "firma Internet", která by z titulu svého vlastnictvím mohla rozhodovat o tom, co a jak se v Internetu má či nemá stát.

Povolování a schvalování

Neexistuje žádné "povolovací řízení" pro využití Internetu - již jen proto, že není nikdo, kdo by z titulu vlastníka mohl stanovit závazná pravidla pro celý Internet.

Zpřístupnění informací

Počítačové systémy, periférie, operační systémy

Dalším důsledkem vlastnických vztahů k Internetu a neexistence řídicí instituce je i fakt, že nikdo nemá povinnost komukoli oznamovat zpřístupnění nových zdrojů a služeb na Internetu. Proto tedy pro vyhledávání informací platí: pokud v rámci jedné evidence (jedné vyhledávací služby, jednoho katalogu, jednoho seznamu zdrojů apod.) nelze nalézt hledanou informaci, neznamená to, že neexistuje.

Existují například tematicky zaměřené seznamy zdrojů na Internetu, orientované např. na právo, stavitelství atd. nebo obecnější vyhledávací služby zaměřené na vyhledávání lidí (konkrétních uživatelů a jejich adres).

Evidence informací

Neexistuje ani žádná závazná norma či předpis, který by bylo třeba dodržovat, a dokonce ani jednotný návod pro katalogizaci informací.

V tomto ohledu se Internet výrazně odlišuje například od klasických knihoven, které také nabízejí informace, ale v mnohem uspořádanější podobě - mají své abecední, věcné i jiné rejstříky, katalogy, přehledy, ale také své knihovníky.

Naproti tomu v Internetu je typické, že každý uspořádá své informace takovým způsobem, jaký uzná za vhodné a jaký je v jeho silách.

To samozřejmě výrazně ztěžuje jakékoli systematické hledání.

Problém nevhodných informací

Skutečnost, že kdokoli může umístit na Internetu jakékoli informace, je velmi snadno zneužitelná. V jejím důsledku se totiž mohou na Internet dostat i takové informace, které nejsou pravdivé nebo jsou hanlivé, záměrně zkreslené či dokonce protiprávní.

Snadnost zneužití Internetu je proto výrazně větší než u ostatních médií, které jednak fungují na jiném principu - mají odpovědné šéfredaktory a podobně.

Problém kvality a kvantity informací

Další výraznou odlišností Internetu od klasických knihoven je jeho obrovský objem dostupných informací. To opět souvisí s tím, jak snadné, laciné a rychlé je zpřístupnění informací po Internetu. Důsledkem toho se na Internet mohou dostat a dostávají i takové informace, které nejsou příliš důležité, významné, přínosné - tolik "kvalitní".

Obecně lze říci, že kvantita a kvalita informací dostupných na Internetu jsou ve vztahu nepřímé úměrnosti. To pak samozřejmě dále ztěžuje jakékoli vyhledávání, zvláště pro uživatele, kteří nejsou sami odborníky v oblasti, ze které hledají určitou konkrétní informaci.

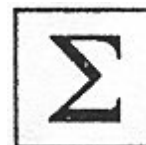
Problém kvality versus kvantity informací na Internetu se snaží řešit i jeden zajímavý trend, který je v poslední době čím dál tím zřetelnější (a také vcelku logický): jde o přechod na placený způsob přístupu k informacím.

Shrnutí

Internet je celosvětová síť složená z počítačových sítí. Nemá majitele ani ředitele. Nikdo nemá povinnost evidence, oznámení zveřejnění informací ani kontroly kvality. Je to téměř naprosto volný prostor bez jakýchkoliv hranic.

Možné přístupy k vyhledávání na Internetu

Možností existuje celá řada, a liší se právě v tom, jakým způsobem se vyrovnávají s problémem velkého objemu dostupných informací a jejich



Počítačové systémy, periférie, operační systémy

relativně nízkou utříděností a nízkou kvalitou. V podstatě lze vymezit tři základní přístupy k vyhledávání informací :

Kvantitativní - vyhledávací služby databázového typu

Kvalitativní - vyhledávací služby katalogového typu

Specializované - seznamy zdrojů



Otázky a úkoly:

- Vysvětli, co je to Internet.
- Kde a proč Internet vznikl?
- Doplň: vlastníkem Internetu je...
- Rozhodni: je Bill Gates ředitelem Internetu? Vysvětli.
- Kdo kontroluje pravdivost informací na Internetu?

5.2 Služby Internetu

Po přečtení této části kapitoly budete vědět:

- ▶ co je WWW a jak funguje
- ▶ co jsou to domovské stránky
- ▶ jak zadat adresu WWW stránky
- ▶ co jsou záložky a jak s nimi pracovat

World Wide Web



WWW je dnes nejpoblárnější služba Internetu, určená pro zpřístupnění informací. Původně byla určena jen pro potřeby sdílení textových informací. Dnes je multimediální.

Pracuje na principu klient/server, což znamená, že pro její fungování jsou zapotřebí WWW servery (výkonné počítače, které uchovávají WWW stránky) a klientské programy (prohlížeče, prostřednictvím kterých uživatelé s touto službou pracují - např. Internet Explorer.)

Hypertext



Služba World Wide Web je vybudována na principu hypertextu. Je to podtržený barevně odlišený text, grafika nebo její část a umožňuje přecházení z jedné stránky na druhou bez ohledu na to, kde se nachází. Odkaz uživatel může snadno určit podle změny ukazatele myši.

Hypertextovým odkazem dnes tedy může být prakticky cokoli - libovolná část textu, i libovolná část obrázku. Dokonce je možné i to, aby různé části jednoho a téhož obrázku představovaly odkazy vedoucí na různá místa. Takovýmto obrázkům říká "klikovatelné mapy", protože nejčastěji to skutečně jsou obrázky představující nějakou mapu, přičemž uživatel si může kliknutím myši zvolit určitou lokalitu na této mapě.

Příklad "klikovatelné mapy":



Právě tímto vzájemným provázáním hypertextovými odkazy vznikla "celosvětová pavučina", kterou si služba WWW dala do svého názvu.

Základní možností hypertextu je postupné rozčlenění informace na menší, snáze přehledné celky (označované jako stránky), a provázání prostřednictvím odkazů.

Přechod mezi stránkami je označován také jako brouzdání (anglicky: browsing) a je pro službu WWW natolik charakteristický, že podle něj byly dokonce pojmenovány i klientské programy sloužící k práci se službou World Wide Web jako browsers = prohlížeče.

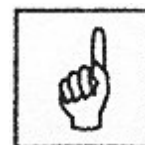
Kromě "brouzdání" se ale uživatel může kdykoli vydat na kteroukoli WWW stránku v rámci celého Internetu - ovšem pouze pokud zná její přesnou adresu (přesněji tzv. URL ukazatel na tuto stránku).

Domovské stránky

Vstupní bod do určité soustavy stránek se nazývá *domovská stránka* (home page). Obsahuje základní orientační prvky - obvyklé uvítání, základní informace o prezentovaném subjektu, a navigaci k dalším dostupným informacím (např. pokračování na dalších stránkách).

Z domovské stránky se takto uživatel může dostat k různým dalším stránkám, se specifickým obsahem. Některé jej mohou zaujmout natolik, že se k nim bude chtít opakovaně vracet. Uživatel, který skutečně přichází poprvé, podruhé, potřetí apod., zde většinou nevádá, že musí postupně "proklikávat" k tomu, co konkrétně hledá - hlavní důraz je zde právem kladen na přehlednost a snadnost orientace.

V případě opakovaného navštívení určité WWW stránky by však tato cesta k informacím na určité konkrétní stránce prostřednictvím postupného "proklikávání" z domovské stránky byla zdoluhavá. Pro takového uživatele je mnohem účelnější zadání *adresy* již jednou navštívené WWW stránky. Každá jednotlivá WWW stránka má totiž svou konkrétní adresu (tzv. ukazatel URL), a takovouto adresu lze prohlížeči kdykoli zadat. Adresy domovských stránek jsou většinou voleny tak, aby byly co možná nejjednodušší, snadno



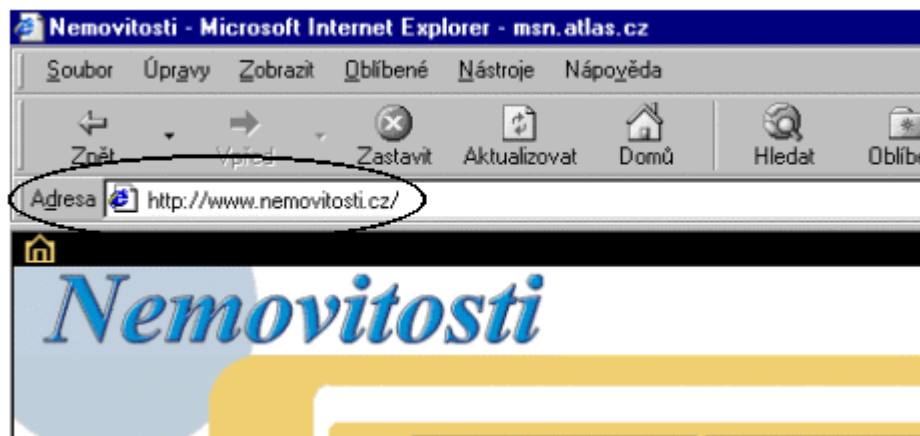
Počítačové systémy, periférie, operační systémy

zapamatovatelné, a někdy je dokonce možné je i uhodnout - například stránky nemovitostí budou mít nejspíš adresu <http://www.nemovitosti.cz/> nebo <http://www.reality.cz/>.



Příklad zadání adresy:

Takové zadávání však v žádném případě není pravidlem a zdaleka nevystihuje všechny stránky zabývající se danou tematikou. Další problém pak nastává v případě složitějšího ukazatele URL, jehož pamatování si a zadávání je pro uživatele komplikací.



Záložky

Většina prohlížečů vychází uživatelům vstříc, a umožňuje jim práci s tzv. záložkami (bookmarks), které tento problém řeší. Princip je velmi jednoduchý: když uživatel navštíví nějakou konkrétní stránku, může svému prohlížeči zadat, aby si její adresu zapamatoval - neboli aby na tuto stránku vložil záložku (podobně jako se vkládají záložky mezi stránky knížek). Například při práci s Microsoft Internet Explorer lze zadat v kategorii oblíbené – Přidat k oblíbeným položkám. Kdykoli později si pak může uživatel vyvolat seznam svých vlastních záložek a kteroukoli z nich si vybrat.

Různé prohlížeče přitom nabízejí různý komfort při práci se záložkami - některé umožňují vytvářet si pouze jednoduchý seznam záložek, zatímco jiné umožňují uživateli vytvářet si celou strukturu "pořadačů", a do nich pak jednotlivé záložky zařizovat.

Důležitou vlastností záložek je skutečnost, že jsou určeny jen pro osobní potřebu konkrétního uživatele.

Shrnutí



WWW je jedna ze služeb Internetu. Uživatel přechází = brouzdá mezi stránkami pomocí hypertextu nebo může přímo zadat adresu stránky. Oblíbené nebo často navštěvované stránky si může uložit jako záložky.

Otázky a úkoly:

- Vysvětli, co je WWW.
- Popiš, na jakém principu je vybudována služba WWW.
- Uveď příklad adresy WWW stránky.
- Vysvětli princip práce se záložkami.
- Co je to brouzdání? Předved' například při hledání vhodného dopravního spojení.
-

5.3 Síťová architektura Internetu

Síťová vrstva TCP/IP

(Transmission Control Protocol/Internet Protocol)

Jde o nejpoužívanější síťovou vrstvu, zajišťující pouze nespojovaný a nespolehlivý přenos, jakési holé minimum, ale snaží se být co nejrychlejší, zajišťuje jednotné přenosové služby nad všemi možnými přenosovými technologiemi nižších vrstev a vytváří tak jakýsi jednotný systém.



Síťový protokol TCP/IP

Protože vrstva TCP/IP jako taková je zajišťována stejnojmenným protokolem, vysvětleme si nyní, co se obecně rozumí pod pojmem síťového protokolu. Síťový protokol je jakási sada pravidel a konvencí pro posílání informací v rámci sítě. Tato pravidla určují obsah, formát, čas a způsob zpracování a řízení chyb zpráv vyměňovaných mezi síťovými zařízeními.

TCP/IP je navrženo a řešeno tak, aby šlo snadno připojovat i dříve samostatné sítě a bylo možné propojit i různé síťové technologie. Svou sadou síťových protokolů tak poskytuje komunikaci v rámci vzájemně propojených sítí tvořených počítači s různou hardwarovou architekturou a různými operačními systémy.

Protokol TCP/IP zahrnuje standardy pro komunikaci počítačů a konvence propojování sítí a směrování provozu. Jeho schopnosti směrování umožňují maximální pružnost v rozsahu celé sítě. V rámci protokolu TCP/IP je mimo jiné k dispozici protokol FTP a Telnet, o kterých se dovíme více v další kapitole.

Postupným připojováním dalších sítí k zárodečnému Arpanetu vzniká vlastní Internet.

Protokol IP (Internet Protocol)

Jde o hlavní přenosový a směrovatelný protokol ze sady protokolů TCP/IP. Snaží se zakrývat specifika přenosových technologií nižších vrstev a fungovat nad nimi optimálně. Zároveň však slouží k adresování protokolu IP, směrování a fragmentaci a opětovnému složení paketů IP.

Koncepce protokolů TCP/IP vznikla zhruba v letech 1977-1978. Samotný Internet pak přešel na TCP/IP k 1.1.1983 a od té doby se nijak principiálně nezměnil. Další vývoj má charakter zdokonalování a obohacování, stávající

Počítačové systémy, periférie, operační systémy

služby se tedy obohacují o další možnosti, jako elektronická pošta, podporu netextových formátů, apod. Na druhou stranu, přibývají i nové služby.

Kritika bezpečnosti Internetu (TCP/IP)

Časté úvahy směřovaly k otázce, zda-li je síť Internet bezpečná z pohledu zneužití, právě v důsledku toho, že protokoly TCP/IP nezajišťují takovou míru bezpečnosti, jakou by si někteří uživatelé představovali.

Faktem je, že při vzniku koncepce TCP/IP nebyl požadavek zabezpečení vznesen. Autoři se tedy soustředili hlavně na efektivnost přenosů a malá míra bezpečnosti spočívá v tom, že přenášená data nejsou šifrována, ani jinak zabezpečena proti odposlechu a zneužití.

V době vzniku Internetu malá úroveň bezpečnosti nevadila. V době však dnešní, kdy se Internet rozšířil zvláště do komerčních zón, stávající úroveň zabezpečení však nestačí. Proto byly navrženy jisté možnosti zvýšení míry zabezpečení, jako jsou šifrování a jim podobné.

Kritika nedostatku adres Internetu (TCP/IP)

Podstata problému spočívala v tom, že autoři TCP/IP zvolili 32-bitový adresový prostor. Pamatovali však i na existenci různě velkých sítí, které potřebují různé počty adres. Proto vznikly třídy A, B, C o kterých se více dovíme níže.

Původní způsob přidělování adres z tohoto adresového prostoru nebyl příliš hospodárný. Autoři ani tak neudělali chybu, jako spíše nedocenili obrovitost svého úspěchu a neodhadli, jak ohromný zájem bude o připojování k Internetu.

Kritika charakteru přenosu Internetu (TCP/IP)

Protože přenosové protokoly TCP/IP jsou orientovány na přenos elektronické pošty, souborů apod., mají tzn. dávkový, neboli nárazový charakter. V praxi jde o to, že negarantují, za jak dlouho jsou data doručena, ani s jakou pravidelností budou doručovány jednotlivé části dat. Tento problém asi nejvíce pociťují multimediální přenosy, zvláště pak přenosy živého zvuku, či obrazu. Zvyšováním přenosových kapacit lze nepříznivý efekt zmírnit, ne však zcela odstranit. Díky tomu se stalo možné například telefonování po Internetu, kdy řešením jsou až specializované přenosové protokoly, podporující přenos v reálném čase.

5.4 Nástroje a služby internetu

Cíl:

- získání vědomostí o nástrojích internetu
- znalost práce s programy **FTP**, **Telnet**

V této kapitole se seznámíme s nástroji a službami poskytované síti Internet. Jak již bylo dříve řečeno, v rámci protokolu TCP/IP je k dispozici protokol FTP a Telnet, které si nyní vysvětlíme v plném rozsahu. Dále se pak seznámíme s problematikou a principem elektronické pošty a diskusních skupin Usenetu.



FTP – protokol, zajišťující přenos dat.
Klient – zajišťuje uživatelské rozhraní.

Server – zajišťuje zpracování dat.

FTP (File Transfer Protocol)

Protokol FTP je členem sady protokolů TCP/IP. Z pohledu uživatele jde o nástroj pracující v textovém režimu, který zajišťuje oboustranný přenos dat, mezi dvěma počítači v síti Internet, přičemž oba počítače musí podporovat příslušné role protokolu FTP, či-li jeden musí být klientem, zajišťující uživatelské rozhraní a druhý serverem, zajišťující zpracování dat FTP. Příkaz **ftp** lze používat interaktivně nebo v dávkovém režimu formou zpracování textových souborů ASCII. Syntaxi **ftp** společně s příslušnými parametry, vlastnostmi a nejčastěji používanými příkazy si zde nyní popíšeme.

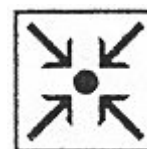
Syntaxe ftp:

```
ftp [-v] [-d] [-i] [-n] [-g] [-s:název_souboru] [-a] [-w:velikost_okna] [-A] [hostitel]
```

Příklad:

Přihlášení k serveru FTP nazvanému ftp.example.microsoft.com, bude vypadat takto: ftp ftp.example.microsoft.com

Přihlášení k serveru FTP nazvanému ftp.example.microsoft.com a spuštění příkazy ftp obsažené v souboru Resynch.txt, bude vypadat takto:
ftp -s:resynch.txt ftp.example.microsoft.com

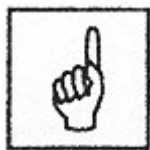


Před parametry příkazu ftp musíte uvádět pomlčku (-).

U přepínačů příkazového řádku se v příkazu ftp rozlišují VELKÁ a malá písmena.

parametr	definice
-v	potlačí zobrazení odpovědi serveru FTP
-d	povolí režim ladění, ve kterém se zobrazují všechny příkazy předávané mezi klientem FTP a serverem FTP
-i	zakáže interaktivní dotazování během přenosu více souborů
-n	potlačí možnost automatického přihlášení po počátečním připojení
-g	zakáže režim expanze názvů souborů. Parametr glob povolí použití hvězdičky (*) a otazníku (?) jako zástupných znaků v názvu místního souboru a v cestě (viz níže, v příkazech FTP)
-s:název_souboru	určuje textový soubor obsahující příkazy ftp. Tyto příkazy jsou po spuštění protokolu ftp automaticky provedeny. V tomto parametru nelze používat mezery. Tento parametr slouží jako náhrada přesměrování (<)
-a	určuje, že k vytvoření vazby datového propojení FTP lze použít libovolné místní rozhraní
-w:velikost_okna	určuje velikost vyrovnávací paměti pro přenos. Výchozí velikost okna je 4096 bajtů
-A	provede anonymní přihlášení k serveru FTP
hostitel	určuje název, nebo adresu IP serveru FTP, ke kterému se chcete připojit. Je-li parametr s názvem hostitele nebo adresou zadán, musí být na řádku uveden jako poslední
/?	zobrazí v příkazovém řádku nápovědu

Přehled příkazů ftp:



syntaxe	definice	definice parametrů
bye	ukončí relaci FTP se vzdáleným počítačem a program ftp	
cd <i>vzdálený_adresář</i>	změní pracovní adresář ve vzdáleném počítači	<i>vzdálený_adresář</i> určuje adresář ve vzdáleném počítači, do kterého chcete přejít
delete <i>vzdálený_soubor</i>	odstraní soubory ve vzdáleném počítači	<i>vzdálený_soubor</i> určuje soubor, který bude odstraněn
dir [<i>vzdálený_adresář</i>] [<i>místní_soubor</i>]	zobrazí seznam souborů a podadresářů, které obsahuje adresář ve vzdáleném počítači	<i>vzdálený_adresář</i> určuje adresář, pro který chcete seznam vytvořit. Není-li adresář zadán, bude použit aktuální pracovní adresář vzdáleného počítače. <i>místní_soubor</i> pak určuje místní soubor, do kterého bude výpis uložen. Pokud není zadán místní soubor, budou výsledky zobrazeny na obrazovce.
get <i>vzdálený_soubor</i> [<i>místní_soubor</i>]	zkopíruje vzdálený soubor do místního počítače pomocí aktuálního typu přenosu souborů	<i>vzdálený_soubor</i> určuje vzdálený soubor, který se má kopírovat <i>místní_soubor</i> určuje název souboru použitý v místním počítači
glob	přepíná expanzi zástupných znaků v názvech místních souborů	
help [<i>příkaz</i>], nebo ? [<i>příkaz</i>]	zobrazí popisy podpříkazů příkazu ftp	<i>příkaz</i> určuje název příkazu, jehož popis chcete zobrazit. Není-li parametr <i>příkaz</i> zadán, zobrazí se seznam všech příkazů
mkdir <i>adresář</i>	vytvoří adresář ve vzdáleném počítači	<i>adresář</i> určuje název nového vzdáleného adresáře
open <i>počítač</i> [<i>port</i>]	připojí počítač k zadanému serveru FTP	<i>počítač</i> určuje vzdálený počítač, ke kterému se chcete připojit <i>port</i> určuje číslo portu TCP, pomocí kterého má být počítač připojen k serveru FTP.
put <i>místní_soubor</i> [<i>vzdálený_soubor</i>]	zkopíruje místní soubor do vzdáleného počítače	<i>místní_soubor</i> určuje místní soubor, který má být zkopírován. <i>vzdálený_soubor</i> určuje název, který má být použit ve vzdáleném počítači
rename <i>název_souboru</i> <i>nový_název</i>	přejmenuje vzdálené soubory	<i>název_souboru</i> určuje soubor, který chcete přejmenovat <i>nový_název</i> nový název souboru
send <i>místní_soubor</i> [<i>vzdálený_soubor</i>]	zkopíruje místní soubor do vzdáleného počítače	<i>místní_soubor</i> určuje místní soubor, který má být zkopírován <i>vzdálený_soubor</i> určuje název, který má být použit ve vzdáleném počítači
status	zobrazí aktuální stav připojení FTP	

Telnet

Telnet – lze tedy chápat jako jakousi bránu prostřednictvím které jednotliví klienti Telnet vzájemně komunikují.

Klient – zajišťuje uživatelské rozhraní.

Server – zajišťuje zpracování dat.



Telnet, podobně jako FTP používá k připojení ke vzdálenému počítači v síti protokol Telnet, který je rovněž součástí sady protokolů TCP/IP.

Klientský software Telnet umožňuje připojení počítače ke vzdálenému serveru Telnet a nabízí tak uživateli na základě vzájemné spolupráce s tímto serverem, či vzdáleným počítačem komunikaci a možnost spouštění jeho aplikací v textovém režimu, prostřednictvím zadávání příslušných příkazů v příkazové řádce klienta Telnet, v okně terminálů.

Server Telnet lze tedy chápat jako jakousi bránu, prostřednictvím které jednotliví klienti Telnet vzájemně komunikují.

Nejčastěji používané příkazy v Telnetu si nyní představíme, společně s jejich definicemi.

Přehled příkazů Telnetu:

syntaxe	definice	definice parametrů
telnet [<i>\\vzdálený_server</i>]	spuštění klienta Telnet a vstup do příkazového řádku programu Telnet	<i>\\vzdálený_server</i> určuje název serveru, ke kterému se chcete připojit
quit , nebo q	ukončení programu Telnet	
open [<i>\\vzdálený_server</i>] [<i>port</i>]	připojení klienta Telnet ke vzdálenému počítači	<i>\\vzdálený_server</i> určuje název serveru, který chcete spravovat <i>port</i> určuje port, který chcete použít
close [<i>\\vzdálený_server</i>]	odpojení klienta Telnet od vzdáleného počítače	<i>\\vzdálený_server</i> určuje název serveru, který chcete spravovat
send [<i>\\vzdálený_server</i>] [<i>ao</i>] [<i>ayt</i>] [<i>esc</i>] [<i>ip</i>] [<i>synch</i>] [<i>?</i>]	odeslání příkazů klienta Telnet	<i>\\vzdálený_server</i> určuje název serveru, který chcete spravovat <i>ao</i> přeruší příkaz výstupu <i>ayt</i> odešle příkaz Are you there <i>ESC</i> odešle aktuální znak escape <i>ip</i> přeruší zpracování příkazu <i>synch</i> provede synchronizační operaci serveru Telnet <i>?</i> umožňují zobrazení úplné syntaxe tohoto příkazu
set	pomocí příkazu set nastavíte typ terminálu pro připojení, zapnete místní odezvu a nastavíte ověřování NTLM a znak escape	
display	zobrazení aktuálního nastavení klienta Telnet	
help , nebo ?	vytiskne informace nápovědy	

Elektronická pošta (e-mail)

Elektronická pošta (e-mail) – umožňuje posílání zpráv v rámci sítě.



Jak už asi víte, v emailová adrese se vyskytuje znak @, který někdy tropí nemalý problém při jeho vytvoření a ne-rozbitím při tom klávesnice :-). Existuje celá řada spolehlivých alternativ jak ho nevytvořit, ale jen jedna, která ho vytvoří kdykoliv, protože k němu přistupuje rovnou pomocí číselného kódu do ASCII tabulky. Postup je následující: stiskněte klávesu Alt a na numerické klávesnici pak napište kód 64.

Elektronická pošta, známá taky pod označením e-mail umožňuje posílání zpráv mezi počítači propojenými sítí. Prostřednictvím elektronické pošty můžeme posílat zprávy svým přátelům a společníkům. V praxi se jedná o zpravidla o bezplatnou službu a doručení do cílové emailové schránky v běžném případě trvá několik desítek sekund, kde pak čeká, dokud se příjemce nepřipojí k internetu a správu si nevyzvedne.

Elektronická pošta, jakož to pošta každá, musí v sobě logicky nést informace o příjemci, v tomto případě emailovou adresu. Ta se skládá ze dvou částí oddělených znakem @, zvaným též jako zavináč. Mimo informací o příjemci, tedy e-mailové adresy a zprávy samotné, může v sobě e-mail nést rovněž jméno příjemce, název zprávy, či přílohu, připojující nejrůznějšími dokumenty v elektronické podobě, jako jím mohou být nejrůznější obrázky, zvuky, programy, či v poslední době tak populární viry.

Syntaxe e-mailu:

Formální podoba a označení elektronické pošty (e-mailu) je mailto:, tedy např. mailto: uzivatelske_jmeno@portal.domena



syntaxe	definice
<i>uzivatelské_jmeno</i>	název emailové adresy, v rámci jednoho portálu je vždy jedinečný
@	znak, parametr, charakterizující elektronickou poštu
<i>portal</i>	označení, či název organizace zprostředkující příslušný e-mail
<i>domena</i>	charakterizující výskyt uzlu, nejčastěji pak udávaný ve zkratce příslušného státu

Zprostředkovatelé emailových služeb, poskytující emailové schránky a jim příslušné adresy, mohou při registraci a provozu vyžadovat uhrazení poplatků s tím spojených, nebo mohou nabízet firemní, či zákaznický, zpravidla

Počítačové systémy, periférie, operační systémy

bezplatný e-mail, určen přednostně tak svým klientům a zaměstnancům, či provozovat e-mail služby zcela bezplatně, kdy jde o tzv. freemaily, spokojující se ve většině případů jen s vaší dobrovolnou registrací, při níž máte i jakousi možnost výběru vlastního názvu neobsazeného uživatelského jména v rámci daného portálu emailové adresy.

Mezi emaily určené svým klientům patří například:

CLICK, provozován společností T-Mobile, znám pod logem



Mezi nejznámější freemaily patří například:

SEZNAM, provozuje firma Seznam a.s., známa pod logem



POST, provozována firmou VOLNÝ, známa pod logem



Usenet

Usenet – diskusní klub, ponechávající příspěvky v jednom místě a jsou tak k dispozici členům ostatním.

Jedná se o přístup k diskusním skupinám Internetu. Usenet je určen k sdílení, či výměně nápadů s jinými lidmi. Na rozdíl od elektronické pošty (emailu), která je privátní záležitostí, jsou diskusní skupiny a konference veřejné, stačí se jen tedy přihlásit do příslušné skupiny, jejíž téma vás zajímá. Každý člen skupiny má pak možnost zasílat své příspěvky k tématu. U diskusních skupin zůstávají na jednom místě (uzlu), který slouží jako jakási „nástěnka“, kde si ji ostatní členové mohou přečíst. Členům elektronické konference se všechny došlé příspěvky zasílají elektronickou poštou.

Syntaxe Usenetu:

Formální podoba a označení Usenetu je **news://**, tedy např.

news://comp.microsoft.windows



Web (World Wide Web)

Pomocí **World Wide Web** (www) sítě si můžeme prohlížet dokumenty, vyhledávat data a přepínat se mezi jednotlivými prostředky internetu.

Samotná webová technologie dovoluje spolupráci mezi členy jednotlivých týmů rozmístěných po celém světě, prostřednictvím systému, který zavádí hypertextové odkazy mezi dokumenty.

Na rozdíl od obyčejných dokumentů, které obsahují jen statické informace, hypertextové dokumenty obsahují vestavěné odkazy, takže čtenář může na další informace jednoduše přeskočit prostým kliknutím myši na slovo, či obrázek. To je důvod názvu těchto dokumentů, kde se nejedná pouze o text, ale o jakýsi nadtext, tedy hypertext. Navíc, odkazy nemusí být pouze na texty, ale i na obrázky, videa, či zvuky.

Vedle elektronické pošty je dnes Web nejpoužívanější prostředek internetu.

Webové servery jsou rozmístěny po celém světě a poskytují informace na jakémkoliv téma, které si jen můžete představit.

5.5 Taxonomie adres a domén sítě Internet

V této kapitole se seznámíme s problematikou internetových adres, domén, s jejími standardy a užití.

HTTP (Hypertext Transfer Protocol)

Protokol používaný k přenosu informací na webu. Adresa HTTP, je typu adresy URL, kterou si vysvětlíme hned jako další.

Syntaxe HTTP:

Má následující formát `http://`. V Microsoft® Internet Explorer se toto označení služby nemusí zadávat. Pokud jej tedy nezadáme, program automaticky předpokládá, že se jedná o adresu `www`, tedy adresu webu.

Zcela zapsaná adresa má pak podobu např. `http://www.osu.cz`.

URL (Uniform Resource Locator)

URL - adresa, která jednoznačně identifikuje umístění v síti Internet

URL je standard internetové adresy, který jednoznačně identifikuje umístění v síti internet. Adrese URL na webu předchází označení **http://**. Adresa URL může obsahovat podrobnější údaje, například název hypertextové stránky, obvykle identifikovaný příponou HTML nebo HTM.

URL nese v sobě tři složky, které slouží k určení, či zpřesnění identifikace zdroje, námi určeného objektu v síti Internet. Je to:

- adresa uzlového počítače internetu, s daným objektem, většinou prezentována adresou
- údaj o konkrétním souboru, v němž se hledaný objekt nachází
- informace o způsobu, jakým se má k objektu přistupovat, respektive kterou z výše popsaných služeb internetu použít.

Setkáte-li se tedy s požadavkem na zadání URL adresy, nejedná se pak o nic jiného, než-li zadání kompletní a správné internetové adresy včetně zkratk služeb, jako například URL `http://www.example.microsoft.com/`

Otázky:

Co je a k čemu slouží URL?

Vymyslete nějakou adresu ve formě URL standardu a zapište ji.

Nyní si uvedeme příklad:

Jedná-li se o soubor `topinky.html`, který se nachází v adresáři WWW, který je zase podadresářem adresáře hypertext, bude internetová adresa vypadat takto:

`http://www.recepty.cz/hypertext/WWW/topinky.html`

Použité služby jsou v tomto případě následující:

Pro načtení se použije protokol `http`. Jedná se tedy o hypertextový dokument. Jednalo-li by se např. o soubory na serveru FTP, bylo by zde uvedeno FTP.

Doménová adresa:



Počítačové systémy, periférie, operační systémy

Dokument se nachází na počítači s doménovou adresou
www/recepty/cz

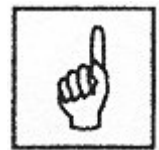
IP adresa a její třídy

Nejprve si vysvětlíme, co je to IP adresa. Jde o 32bitovou adresu sloužící k identifikaci uzlu v rámci propojení sítí IP. Každému uzlu v propojení sítí IP tak musí být přidělena jedinečná adresa IP, která je tvořena identifikátorem sítě a identifikátorem hostitele. Adresa je obvykle reprezentována desítkovými hodnotami jednotlivých oktětů oddělených tečkou, jako například 192.168.7.27.

Předdefinovaná seskupení internetových adres s jednotlivými třídami definujícími síť určité velikosti. Rozsah čísel, která lze přiřadit jako první oktět v adrese IP, závisí na třídě adresy.

Rozdělení adres IP podle tříd:

třídy A v rozsahu od 1.0.0.1 do 126.255.255.254 (hodnoty 1 až 126). První oktět určuje síť a poslední tři oktety určují hostitele v síti. Sítě třídy A jsou tedy nejrozsáhlejší s více než 16 miliony hostitelů na jednu síť.



třídy B v rozsahu od 128.0.0.1 do 191.255.255.254 (hodnoty 128 až 191). První dva oktety určují síť a poslední dva oktety pak hostitele v síti. Sítě třídy B mohou mít v rámci sítě maximálně 65 534 hostitelů.

třídy C v rozsahu od 192.0.0.1 do 223.255.255.254 (hodnoty 192 až 223). První tři oktety určují síť a poslední oktět určuje hostitele v síti. Sítě třídy C mohou mít pak maximálně 254 hostitelů.

Otázky:

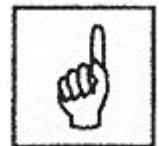
Urči, jak k čemu slouží IP adresa.
Uveďte její příklad



Domény

Doména – repre-zentuje konkrétní IP adresu.

Nejprve si vysvětleme, co to doména je. Doména je základní adresní jednotka na Internetu, která reprezentuje konkrétní IP adresu, například 197.195.97.85. Pro uživatele je tak určitě snadnější přistoupit na www stránky, či FTP rozhraní přes psanou adresu než vyťukáním čtyřech čísel.



Rozdělení domén, společně s syntaxi a příkladem:

http://www.doména 3. úrovně.doména 2. úrovně.doména 1. úrovně

http://www.rozvrh.osu.cz

- *doména 1. úrovně* je nejvyšší, zvaná taky jako top-level doména (TLD), nebo národní doména, udávaná pak nejčastěji ve zkratce příslušného státu, jako v našem příkladě **.cz**
- *doména 2. úrovně* se nachází ihned před národní doménou, tak jak je v našem příkladě **osu.cz**



Počítačové systémy, periférie, operační systémy

- *doména 3. úrovně* – v našem případě **rozvrh.osu.cz**
- *domény dalších úrovní*

Otázka:

Pokuste se určit podobu URL společnosti SONY, se stejnojmennou doménou 2. úrovně v zemích České republiky, Německa, Slovenska a USA v komerční sféře.



Pojmenovávání nejvyšších domén, tedy domén 1. úrovně, bylo zvoleno dle odpovídajícího mezinárodního dvojpísmenného kódu odpovídajícího státu. Výjimkou je USA, která svým třípísmenným kódem označuje typy organizací.

Přehled některých nejvyšších domén přidělených jednotlivým zemím:

nejvyšší doména	stát	nejvyšší doména	stát	nejvyšší doména	stát
.ae	Arabské emiráty	.fr	Francie	.pl	Polsko
.af	Afganistán	.gb	Velká Británie	.pt	Portugalsko
.at	Rakousko	.gr	Řecko	.ro	Rumunsko
.au	Austrálie	.hr	Chorvatsko	.ru	Rusko
.be	Belgie	.hu	Maďarsko	.se	Švédsko
.bg	Bulharsko	.ch	Švýcarsko	.si	Slovinsko
.ca	Kanada	.ie	Irsko	.sk	Slovensko
.cn	Čína	.il	Izrael	.tn	Tunisko
.cz	Česká republika	.it	Itálie	.ua	Ukrajina
.de	Německo	.jp	Japonsko	.uk	Velká Británie
.dk	Dánsko	.lv	Litva	.us	USA
.eg	Egypt	.mn	Mongolsko	.vn	Vietnam
.es	Španělsko	.mx	Mexiko	.yu	Jugoslávie
.fi	Finsko	.no	Norsko	.zm	Zambie

Přehled některých domén organizací užívané spojenými státy americkými:

nejvyšší doména	stát	nejvyšší doména	stát
.com	komerční organizace	.gov	vládní (státní) úřad
.edu	vzdělávací instituce	.mil	vojenská organizace
.net	správcovský uzel	.org	obecně jiná organizace



URL jednoznačně identifikuje objekt, IP adresa pak uzel v rámci sítě IP, či Internetu. Doména pak reprezentuje konkrétní IP adresu.

5.6 Katalogy

Po prostudování této části budete schopni odpovědět na tyto otázky:

- ▶ co jsou katalogy a jak pracují
- ▶ jak hledat v katalogích
- ▶ příklady katalogových služeb

Co jsou katalogy

Katalogy vznikly ze specializovaných *seznamů adres*, obsahující odkazy a komentáře autorů WWW stránek. Dále autoři odkazy uspořádali podle kategorií a poskytli na Internetu ostatním uživatelům. Rozšiřování katalogů a umístování reklam na WWW stránky přináší provozovatelům větší zisky a vyžaduje zaměstnání dalších pracovníků ke zpracování a zařazování dalších stránek... Samotné reklamy však přinášejí další problémy. Jejich obsah je většinou erotického charakteru, vedoucí pak svými odkazy na velké množství pornografických stránek. Nic netušící uživatel (mnohdy zdaleka ne ploletý!) může otevřít slušně vypadající odkaz a nestačí se divit... Stejně tak autor stránek, na které je umístěn tzv. výměnný proužek reklamy nemá možnost zaměření těchto odkazů nijak ovlivnit. Dojem seriózního obsahu stránek pak často vezme za své.

Stávající katalogy jsou neustále aktualizovány. Vznikají další - národní, specializované a jiné katalogy.

Zřejmě nejpodstatnějším rysem katalogů je jejich hierarchická soustava kategorií a podkategorií, kterými uživatelé mohou podle vlastního uvážení procházet a něco konkrétního hledat. To je výhodné zvláště tehdy, pokud uživatel nemá přesnější představu o tom, co vlastně hledá, a chce se pouze inspirovat. Pokud ale přesnější představu má, a chce se co nejrychleji dobrat výsledku, postupné procházení nabídkových stránek by jej mohlo spíše zdržovat. Proto většina katalogů nabízí k využití také jednoduchý mechanismus automatického vyhledávání v rámci nabídkových stránek.

Vyhledávání nových zdrojů

Jednou z možností je ruční vyhledávání o nových zdrojů informací. Provozovatel katalogu si najímá skupinu "brouzdačů", kteří neustále brouzdají Internetem. Právě tito lidé popisují nově nalezený zdroj (což může být jen velmi stručná poznámka, až po mnohem propracovanější popis typu podrobné recenze, včetně poznatků o dostupnosti apod.). Katalogů právě popsaneho typu je pro jejich náročnost poměrně málo. V našem tuzemském Internetu takto funguje zatím jen jediná vyhledávací služba jménem Zmije (<http://www.zmije.cz/>)



Příklady katalogových služeb



Jinou, častější možností, je spoléhání na neformální oznamovací povinnost. Zde provozovatel katalogu pasivně čeká, až mu zřizovatel nového zdroje sám oznámí, co nového je kde k dispozici. Snad nejznámějším příkladem vyhledávací služby katalogového typu je služba YAHOO <http://www.yahoo.com/>, katalogizující odkazy na informační zdroje po celém světě. U nás vznikla obdobně koncipovaná služba Seznam <http://www.seznam.cz/zaměřená> výhradně na tuzemské zdroje, a časem se objevily i některé další české katalogy.

<http://case.rka.cz/>

<http://www.mozek.cz/>

<http://hledej.tu.cz/>

<http://www.nazdar.cz/>

<http://katalog.stranky.cz/>

<http://www.obchody.cz/>

<http://najdi.to/>

<http://www.oko.cz/>

<http://najduto.zde.cz/>

<http://www.opendir.cz/>

<http://web.iol.cz/elchron/alfa.htm>

<http://www.seznam.cz/>

<http://web.navrcholu.cz/>

<http://www.toplist.cz/>

<http://www.atlas.cz/>

<http://www.trifid.cz/>

<http://www.centrum.cz/>

<http://www.uzdroje.cz/>

<http://www.czis.cz/>

<http://www.vokno.cz/>

<http://www.czseznam.cz/>

<http://www.wwwcentrala.cz/>

<http://www.katedrala.cz/>

<http://www.zdroj.cz/>

Shrnutí



Výhody katalogových služeb:

- filtrování (katalogy obsahují jen odkazy na skutečně odpovídající a kvalitní
- stránky; odkaz je zařazen až po prohlédnutí a ohodnocení stránky)
- jednoduchá orientace (systém podkategorií)
- správné zařazování odkazů

Nevýhody katalogových služeb:

- neaktuálnost (časová náročnost kontroly a obnovy odkazů)
- neúplnost (není možné projít všechny stránky)
 - malý objem (pro velký podíl lidské práce nemohou katalogy obsáhnout velký objem informací)



Otázky a úkoly:

Vysvětli, jak se dostanou nové odkazy na zdroje informací do katalogů. Uveď alespoň tři příklady české a jeden příklad zahraniční vyhledávací služby katalogového typu.

Vyhledej v některém českém katalogu seznam vysokých škol v České republice.

5.7 Vyhledávací služby databázového typu

V této části se dozvíte a budete schopni odpovídat na následující dotazy:

- ▶ jak pracují vyhledávací služby databázového typu
- ▶ jak probíhá hledání
- ▶ jak zadat vyhledávací řetězec
- ▶ příklady vyhledávacích služeb databázového typu

Vyhledávací stroje

Největší objem dat dokáží obsáhnout takové formy evidence zdrojů, které se nejčastěji nazývají jako vyhledávací stroje nebo vyhledávací služby databázového typu. Označení vyhledávací stroje vystihuje skutečnost automatického třídění a katalogizace. Při takovém objemu zde již není možná žádná forma lidské práce. Odkazy na dostupné zdroje jsou shromažďovány a bez účasti člověka zaneseny do vhodné databáze.

Vyhledávací mechanismy, fungující na tomto principu, prohledávají všechny WWW stránky prakticky v celém Internetu, a tzv. je indexují. To znamená, že mechanicky registrují výskyt každého jednotlivého slova na dané WWW stránce, aniž by se zabývaly jeho obsahem a významem. Údaj o jeho výskytu si spolu s několika desítkami písmen, které pak slouží k ručnímu dohledávání, a s adresou stránky zanáší do své databáze.

Jak probíhá hledání

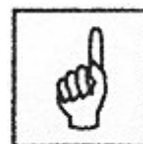
Podstatné je také to, že zde nevznikají žádné nabídky ani kategorie a podkategorie, kterými by se dalo procházet. Všechno je v jedné databázi a uživatelé se k tomu mohou dostat pouze cestou kladení vyhledávacích dotazů, na které databáze odpovídá seznamem vyhovujících odkazů. To je samozřejmě možné a výhodné v situaci, kdy uživatel dosti přesně ví, co hledá, a dokáže to popsát pomocí vhodných klíčových slov.

<http://www.kompas.seznam.cz/>

Příklad zadávání vyhledávacího dotazu:

<http://www.kompas.seznam.cz/>

Příklad pokročilého vyhledávání:



Odkazy Sekce **Fulltext** Firmy

Pokročilé vyhledávání [Jednoduché v](#)

Hledej stránky:

obsahující všechna slova: (AND)

obsahující některá ze slov: (OR)

neobsahující slova: (NOT)

Hledej v doménách: hledaná stránka musí být z domény:

Na stránce zobraz: 25 odkazů

Na svůj dotaz tak uživatel může získat desítky tisíc vyhovujících stránek. Takové množství však není možné postupně projít, a tak musí svůj dotaz postupně upřesňovat. Toto upřesňování je v praxi na celém hledání nejobtížnější. Dosah vyhledávacích služeb databázového typu může být skutečně obrovský: jsou schopné si pamatovat obsah několika desítek milionů WWW stránek, a umožňují v nich vyhledávat na *plnotextovém* (=fulltextovém) principu.

Obecně ale platí, že čím větší je vyhledávací služba (čím více dostupných zdrojů eviduje), tím těžší je správně se jí zeptat.

V režimu jednoduchého vyhledávání se dělá vše proto, aby ani laický uživatel neodešel s nepořízenou. Když uživatel něco konkrétního hledá, musí se obrátit na tuto databázi a položit jí dotaz typu "na kterých stránkách se vyskytuje slovo X (nebo skupina slov XYZ atd.)"? Jako odpověď dostane odkazy na stránky, o kterých vyhledávací mechanismus ví, že se v nich hledané slovo (či slova) vyskytuje, a uživatel si již sám musí "dojít" pro příslušnou stránku, tu si přečíst a sám posoudit, zda je tím co hledal.

Zadávání vyhledávacího řetězce



Nejčastěji to znamená, že uživateli stačí napsat jedno nebo několik klíčových slov, a vyhledávací stroj (databáze) se snaží interpretovat tento dotaz tak, aby výsledek byl co možná nejvíce přijatelný. Konkrétně tak, že když je klíčových slov více, snaží se vyhledávací mechanismus hledat všechna tato klíčová slova. Pokud nic nenajde, přejde na jinou strategii, a snaží se najít alespoň některá ze zadaných klíčových slov. Tím se vyhledávací služba snaží vyjít vstříc uživatelům, kteří nedokáží popsat logickou vazbu mezi jednotlivými klíčovými slovy a tuto vazbu si vlastně snaží domyslet sám.

Nejprve jako logický součin (kdy musí být nalezena všechna klíčová slova současně), a v případě neúspěchu alespoň jako logický součet (stačí alespoň některá klíčová slova). Uživatelé, kteří jsou schopni sami vyspecifikovat logickou vazbu mezi jednotlivými částmi svého dotazu, samozřejmě mají také šanci - v praxi ale velmi záleží na konkrétní vyhledávací službě.

Počítačové systémy, periférie, operační systémy

Některé stačí napsat před příslušné klíčové slovo znaménko "+" (plus), a tím vyjádřit, že dané klíčové slovo se musí vyskytovat (resp. že všechna takto označená klíčová slova se musí v nalezených stránkách vyskytovat současně). Znaménkem "-" (minus) pak lze specifikovat to, že příslušná klíčová slova se vyskytovat naopak nesmí. Jiné vyhledávací služby mají vše vyřešeno tak, že logická vazba mezi jednotlivými klíčovými slovy se zadává prostřednictvím samostatného "listboxu", kde se navolí příslušná možnost - např. "všechna slova" nebo "alespoň jedno slovo". Zadání vyhledávacího řetězce Pro vyhledávání službou databázového typu je nutné znát některá pravidla pro zadávání klíčových slov. Ačkoliv jednotlivé servery mají svá specifika a odlišnosti, obecně lze říci, že platí:
(pro uvedené příklady bylo použito fulltextové hledání v Seznamu)



AND (logický součin)

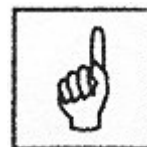
Při zadání AND služba vyhledává stránky, kde se klíčová slova vyskytují současně.

Například: Národní AND divadlo = 5 091 odkazů

OR (logický součet)

Při zadání OR služba vyhledává stránky, kde se vyskytuje alespoň jeden nebo všechny výrazy zadaného termínu.

Například: Národní OR divadlo = 10 809 odkazů



NOT

První výraz v zadaném řetězci v dokumentu má být, druhý ne (někdy se píše AND NOT).

Například: Národní NOT divadlo = 12 848 odkazů

NEAR

Při zadání NEAR služba vypíše stránky s termíny umístěnými do určité vzdálenosti od sebe, přičemž nezáleží na pořadí výrazů v zadaném řetězci. Je jistou obdobou logického součinu, někdy dává dokonce lepší výsledky,

Například: Národní NEAR divadlo = 11 109 odkazů

ADJ

Pracuje podobně jako NEAR, při vyhledávání však dodržuje pořadí slov v zadaném řetězci.

Například: Národní ADJ divadlo = 11037 odkazů

Uvozovky

Vyhledává se celý textový řetězec, ale ne všechny servery užívají této metody.

symboly + a -

Při zadání "+řetězec" služba vyhledává stránky s řetězcem

Při zadání "- řetězec" jsou výsledkem hledání stránky bez zadaného řetězce. Například: +Národní –divadlo znamená výpis stránek s výrazem Národní, ale jedině bez divadla)

+Národní divadlo = 11435 odkazů

-Národní divadlo = 11992 odkazů

+Národní –divadlo = 11640 odkazů

operátor *

Zadání * pracuje jako hvězdičková konvence Windows. Je však nutné zadat nejméně 3 znaky.

Například: Tep* = výrazy s Tep, Teplotechna, Teplice...

velká a malá písmena

Při zadání velkých najde služba řetězce jen s velkými písmeny, při zadání malých písmen najde s malými i velkými.

Například: národní divadlo = 325926 výskytů slov

link:URLtext

Výsledkem hledání jsou stránky obsahující odkaz na zadané URL

Například: 14984 odkazů

title:Text

Výsledkem hledání jsou WWW stránky se zadaným textem v titulku stránky.

text:Text

Hledaný text se musí objevit jen textu stránky, ne v titulku nebo jinde.

domain:jméno_domény

Výsledkem hledání jsou v tomto případě stránky v dané doméně.

Anchor:Text

Při tomto zadání probíhá vyhledávání v odkazech.



Vývoj databázových služeb

Vyhledávací služby databázového typu se za dobu své existence zajímavým způsobem vyvíjely. Zpočátku, kdy jejich kapacita nebyla na dnešní úrovni, tyto databáze nemohly zařazovat plné znění všech WWW stránek, na které v Internetu narazily. Místo toho si všímaly pouze jejich částí. Například tzv. **titulků** (jednořádkových názvů stránky, které jednotlivé browsery vypisují ve svém rámečku úplně nahoře), nebo prvních deseti řádek textu stránky, prvních 20 slov apod. S postupem času ale docházelo ke zvětšování kapacity databází, a v určitém okamžiku se již stalo únosné, aby si tyto služby pamatovaly celý obsah jednotlivých WWW stránek. Přesněji aby si mohly vytvořit indexy z celého a plného znění WWW stránky, pamatovat si je, a pak nabízet možnost plnotextového vyhledávání v rámci WWW stránek, v rozsahu prakticky celého Internetu.



5.8 Příklady některých databázových služeb

První službou, která takovouto možnost nabídla, byla služba Alta Vista (uvedená do provozu v prosinci 1995 firmou Digital, na adrese <http://altavista.digital.com/>). časem se objevily i obdobně fungující služby s "lokálním" dosahem, mapující např. národní prostor českého Internetu (např. služby Atlas, <http://www.atlas.cz/a> Kompas, <http://kompas.seznam.cz/>, mapující jen tuzemské WWW stránky).

Další příklady:

<http://pavouk.uzdroje.cz>

<http://fulltext.centrum.cz/>

<http://hledej.atlas.cz/>

<http://www.kompas.seznam.cz/>

<http://www.megatext.cz/>

<http://www.sherlock.cz/>

Shrnutí

Vyhledávací stroje jsou schopny obsáhnout prakticky všechny informace zveřejněné v Internetu, ale nejsou schopny je třídit podle významu. Vyhledávání probíhá podle zadaných klíčových slov nebo vyhledávacího řetězce a užitím vhodných logických operátorů. Výsledkem hledání pak bývají zpravidla odkazy na tisíce stránek, mezi kterými si uživatel dále vybírá.



Otázky a úkoly:

1. Vysvětli, jak pracuje vyhledávací služba databázového typu.
2. Co jsou to klíčová slova? Uveď příklad.
3. Jaký je rozdíl mezi zadáním logického součtu a logického součinu?
4. Vyzkoušej možnosti vyhledávání zadáním klíčových slov a užitím všech možností logických operátorů. (například Nový Jičín, Lipová-Lázně...)



5.9 Další formy vyhledávání

V této části se dozvíte:

- ▶ co jsou to seznamy zdrojů a jak vznikají
- ▶ co jsou vyhledávací meta-slужby
- ▶ jaké jsou další trendy vývoje vyhledávačů
- ▶ příklady vyhledávacích služeb

Seznamy zdrojů

Prakticky všechny seznamy zdrojů vznikají "ručním sběrem" informací o dostupných zdrojích. Nejčastěji má podobu seznamu záložek, které si podle svého odborného zájmu udržuje určitý konkrétní člověk. Potom tento seznam záložek, sloužící původně pouze jeho osobní potřebě, převede do formy WWW stránky a umístí ji na vhodný WWW server, aby tato stránka byla přístupná všem uživatelům Internetu.

Seznamy zdrojů jsou monotematické, tj. zaměřené na jednu specifickou oblast.

Mezi seznamy zdrojů a katalogy neexistuje ostrá hranice, rozlišení je dáno spíše velikostí a šíří záběru: zatímco seznamy zdrojů nejčastěji obsahují několik desítek, maximálně několik stovek odkazů, v případě katalogů se jedná spíše o tisíce, desítky tisíc, či dokonce statisíce odkazů na existující zdroje.

V dnešním Internetu existuje velké množství nejrůznějších seznamů zdrojů, od nepříliš užitečných experimentů až po opravdu hodnotné a často navštěvované seznamy. Nepříjemným problémem je samotná evidence takovýchto seznamů - vzhledem k jejich počtu nebývá jednoduché je najít. Naštěstí existují i "seznamy seznamů", které se specializují na evidenci různých seznamů zdrojů. Asi nejznámějším "seznamem seznamů" je tzv. *virtuální knihovna WWW* na adrese <http://www.w3.org/vl/>.



Registrace zdrojů

Základním problémem *evidence zdrojů* je neexistence "registrační povinnosti". Takovéto zaregistrování nového zdroje u nejrůznějších katalogů i dalších druhů vyhledávacích služeb je samozřejmě v zájmu toho, kdo nový zdroj zřizuje. Na druhé straně ale nikdo nemá tuto povinnost, a proto se může stát (a skutečně stává), že nejrůznější katalogy "neznají" mnohé zajímavé zdroje. Současně s tím je nutné si uvědomit, že nejrůznějších katalogů a dalších vyhledávacích služeb existuje celá řada, a i dost svědomitý zřizovatel nového zdroje je nemusí znát všechny, takže svůj zdroj registruje jen u některých.

Pro koncové uživatele z těchto faktů vyplývá jeden podstatný závěr: pokud některý katalog nějaký konkrétní zdroj nezná, ještě to zdaleka neznamená, že takovýto zdroj neexistuje. Hledáte-li něco konkrétního, má smysl se obracet na různé katalogy i další vyhledávací služby, a pak si přebrat výsledky.

Existují specializované registrační služby, které nové zdroje zaregistrují u více katalogů a vyhledávacích služeb současně, ale ani tyto registrační služby nemohou být úplné.

Výsledkem je dosti značná různorodost v obsahu jednotlivých katalogů a vyhledávacích služeb, umocněná dále skutečností, že každá z nich může fungovat trochu jiným způsobem

Vyhledávací meta-slужby

Všechny vyhledávací služby databázového typu nabízí možnost plnotextového vyhledávání v rámci WWW stránek, ale jen některé z nich nabízí možnost vyhledávání i jinde - například v diskusních skupinách síťových novin, elektronických konferencích, archivech souborů apod. Pro takovéto více specializované vyhledávání, například vyhledávání lidí a jejich elektronických adres, vyhledávání firem, dokumentů, souborů v tzv. FTP archivech apod. dnes existují specializované služby, dostupné ze světa Web-u prostřednictvím vhodných bran. Zde pak také platí, že s jedním dotazem je vhodné se obracet na více vyhledávacích služeb.

Jednou z možností, jak efektivně položit stejný dotaz více různým vyhledávacím službám, je použití tzv. vyhledávacích meta-slужeb - jen je jich relativně málo, a nejsou příliš známé.

Jsou to nadstavbové služby, které samy o sobě nic nevyhledávají, ale pouze "znají" skutečné vyhledávací služby a dokáží mezi ně rozdělovat jednotlivé dotazy. Vše pak může fungovat takovým způsobem, že uživatel adresuje svůj dotaz jedné vyhledávací meta-slужbě. Ta jej přijme a sama se rozhodne, kterým "skutečným" vyhledávacím službám jej předá k zodpovězení. Uživatel přitom může vyhledávací meta-slужbě účinně pomoci při sestavování vyhledávacího plánu tím, že jí naznačí podstatu svého dotazu - zda například jde o hledání osoby, dokumentu, firmy apod.

Meta-slужby právě popsaného typu dnes existují ve dvou základních variantách - první počítá s provozováním meta-slужby někde "uvnitř" Internetu, na vhodném uzlovém počítači, kterému pak uživatelé zasílají své jednotlivé dotazy. Od tohoto uzlu pak odchází několik konkrétních dotazů na konkrétní vyhledávací služby, a odpovědi pak směřují zpět do zmíněného uzlu. Ten z odpovědí sestaví jeden výsledný celek (například odstraní případné duplicitní odpovědi), a odešle jej zpět původnímu tazateli.

Počítačové systémy, periférie, operační systémy

Druhá varianta předpokládá provozování meta-slужby přímo na počítači uživatele. Výhodou je zde maximální komfort a možnost vyptat se uživatele na podstatu jeho dotazu, tak aby bylo možné sestavit co možná nejefektivnější vyhledávací plán.

Příkladem takovéto služby je vyhledávací centrála Archon.

<http://www.archon.cz>

Další trendy vývoje

Příjemnou vlastností drtivé většiny vyhledávacích služeb je fakt, že pro své uživatele fungují zdarma. Náklady na jejich provoz je nejčastěji hrazen příjmem z tzv. reklamních proužků, které umisťují na své stránky. Vzhledem k poměrně vysoké návštěvnosti vyhledávacích služeb to může být atraktivní i pro zadavatele reklamy, a tak je tento způsob financování obvykle schopen příslušné služby "uživit". Problémy vyvstávají však v souvislosti s obsahem těchto reklamních proužků. Je většinou nevhodný pro mládež, která Internet často navštěvuje a ani autor stránek tento obsah zpravidla nemůže ovlivnit. Vyhledávací služby pak samozřejmě mají samy zájem na co největší návštěvnosti svých stránek (což jim umožňuje zvyšovat cenu za reklamu na svých stránkách). Jedním ze způsobů, jak si svou návštěvnost zvýšit, je i poskytování dalších informačních služeb, než jen "pouhé" vyhledávání. Příkladem může být poskytování zpravodajských přehledů, předpovědí počasí, programů televize a rozhlasu, různých aktuálních zajímavostí apod. Tohoto typu jsou například služby Dogpile, MetaFind či SavvySearch.

Příklady vyhledávacích služeb

Seznam

První český vyhledávací server <http://www.seznam.cz/> vznikl r. 1996 jako typický představitel katalogů - po vzoru americké služby Yahoo (proto se mu jistý čas dokonce přezdívalo "české Yahoo"). Časem si ale tento katalog pořídil "doplňek" ve vyhledávací službě databázového typu, fulltextový vyhledávač jménem Kompas <http://kompas.seznam.cz/>. Databáze Kompasů obsahuje automaticky nasbírané dokumenty z domény .cz, podporuje logické spojky u jednoduchého vyhledávání (jako zřejmě jediný český nástroj), podporuje českou diakritiku. Obě vyhledávací služby, seznam a Kompas, pak pojí nejen společný zřizovatel a provozovatel, ale především jejich vzájemná provázanost - hledáte-li něco u jedné z těchto služeb a neuspějete-li, je vám nabídnuto vyhledání téhož i ve druhé službě. Seznam i Kompas se soustředují jen na tuzemské odkazy.

Nevýhodou je nepříliš pružné zařazování nových stránek do databáze (21dnů) a zdlouhavé prohledávání. Vyhledávací služba hledá jen v titulcích stránek a jejich popisích, ani s operátory nejsou lepší výsledky. Lepší je přechod na služby Kompas. Velmi dobrá je služba vyhledávání e-mailů - jednoduché i pokročilé, obsahuje rozsáhlou databázi adres.

Atlas

Další významnou tuzemskou vyhledávací službou je služba jménem Atlas, na adrese <http://www.atlas.cz>. Ta vznikla již jako integrovaná vyhledávací služba,



jejímž hlavním prvkem je vyhledávač databázového typu, doplněný nejen katalogem, ale i dalšími specializovanými vyhledávacími službami (pro vyhledávání lidí a jejich emailových adres, a pro vyhledávání uživatelů dostupných pro on-line komunikaci prostřednictvím Internetu). Zařazení nových stránek do databáze trvá Atlasu několik dnů. Při vyhledávání užívá logické operátory i spojení v uvozovkách.



Shrnutí kapitoly:

Seznamy zdrojů jsou obdobou katalogů, jsou však zpravidla zaměřeny na určitou oblast a nejsou tak obsáhlé. Meta-slужby jsou specializované a dotaz zadávají jiným vyhledávacím službám.



Otázky k procvičení:

1. Vysvětli rozdíl mezi seznamy zdrojů a katalogy.
2. Kolik odkazů můžeš získat po zadání dotazu seznamu zdrojů?
3. Jak probíhá evidence zdrojů?
4. Co vyhledávají meta-slужby?
5. Uveď příklady českých a zahraničních vyhledávacích služeb.

5.10 Cvičení

V této kapitole si můžete na základě otázek, vycházejících z příslušných kapitol prověřit, procvičit a případně zpětnou vazbou prohloubit své vědomosti.



Testové otázky a úkoly k textu

- 1) Stručně popište, co je to Internet.
- 2) Vyjmenujte, jakými nejruznějšími možnostmi se může realizovat přenos dat.
- 3) Definiujte, co je to Server.
- 4) Vyjmenujte, jaké nároky jsou kladeny na Server.
- 5) Charakterizujte síťovou vrstvu TCP/IP.
- 6) Určete název sítě, která je zárodečnou sítí, sítí Internet.
- 7) Charakterizujte síťový protokol TCP/IP, jeho vlastnosti a užití.
- 8) Vyjmenujte protokoly obsaženy v rámci protokolu TCP/IP.
- 9) Určete jednu z možností, jak zvýšit míru zabezpečení sítě Internet.
- 10) Objasněte dávkový, či nárazový charakter přenosu.
- 11) Určete příslušné postavení rolí Klienta a Servera v síti.
- 12) Charakterizujte FTP nástroj a jeho možnosti užití.
- 13) Uveďte syntaxi FTP společně s nejdůležitějšími příkazy a jejich definicemi.
- 14) Charakterizujte nástroj Telnet a uveďte jeho užití.
- 15) Uveďte nejdůležitější příkazy Telnetu, společně s jejich definicemi.
- 16) Uveďte podobu formální syntaxe elektronické pošty (emailu).
- 17) Určete, všechny nezbytné informace, které musí email obsahovat, aby byl doručen do příslušné schránky.
- 18) Vysvětlete využití přílohy u emailu.
- 19) Co to jsou freemaily a vyjmenujte alespoň jednoho zástupce.
- 20) Charakterizujte, k čemu se používá Usenet.

Počítačové systémy, periférie, operační systémy

- 21) Určete úhlavní rozdíl mezi elektronickou poštou (emailem) a diskusní skupinou Usenet.
- 22) Co nám umožňuje Web (World Wide Web)?
- 23) Urči, k čemu slouží HTTP protokol?
- 24) Co je a k čemu slouží URL?
- 25) Vymyslete nějakou adresu ve formě URL standardu a zapište ji.
- 26) Urči, jak a k čemu slouží IP adresa.
- 27) Uveď příklad IP adresy.
- 28) Vyjmenuj třídy IP adres a urči jejich úhlavní rozdíl.
- 29) Definuuj pojem doména.
- 30) Urči a popiš jednotlivé umístění úrovní domén.
- 31) Pokuste se určit podobu URL společnosti SONY, se stejnojmennou doménou 2. úrovně v zemích České republiky, Německa, Slovenska a USA v komerční sféře
- 32) Jak nazýváme programy, které slouží k zobrazení internetových stránek, či obecně Webu?
- 33) Jmenujte alespoň jeden z nich
- 34) Vysvětlete funkci domovské stránky, nebo-li homepage a popište, nebo předved'te její způsob nastavení.
- 35) Vysvětlete jednotlivé způsoby, jak lze uložit obsahy stránky www, binární soubory, tedy soubory ke stáhnutí a obrázky
- 36) Určete, k čemu jsou využívány vyhledávače a jmenujte alespoň jeden
- 37) Rozdělte vyhledávače podle forem vyhledávání
- 38) Vyjmenujte logické operátory a vysvětlete jejich význam
- 39) Pomocí vyhledavače Seznam se pokuste nalézt na internetu hlavní stránku Ostravské University

5.11 Závěr

Internet představuje velmi rychle se rozvíjející informační médium. Jeho prostřednictvím lze získat informace prakticky o každém tématu. Problémem je obrovské množství informací a nesnadná orientace v této „celosvětové pavučině“. Neexistence jednotného řídicího centra Internetu se snaží řešit vyhledávací služby. Vycházejí vstříc uživatelům, kteří se stránkách Internetu snaží získat informace o daných tématech.

Vyhledávací služby jsme rozdělili do tří základních kategorií. Uvedené příklady služeb jsou rozdělené do jednotlivých sekcí podle převažujícího podílu vyhledávání, neboť není možné vyznačit ostré hranice mezi nimi. Srovnání jednotlivých typů vyhledávacích služeb najdeme v následující tabulce:



Počítačové systémy, periférie, operační systémy

	Seznamy zdrojů	Katalogy	Vyhledávací služby databázového typu
Obsahové zaměření	Monotematické	Univerzální	Univerzální
Nabídkové stránky s možností procházení	Ano (jedna stránka nebo několik málo stránek)	Ano (celá hierarchická struktura nabídkových stránek)	Ne
Možnost automatického vyhledávání v nashromážděných informacích	Ne	Ano (jako doplněk/alternativa k brouzdání nabídkovými stránkami)	Ano (jako jediný a hlavní způsob práce)
Počet evidovaných zdrojů (zmapovaných stránek)	Desítky, max. stovky	Desítky tisíc, max. stovky tisíc	Desítky milionů

6 Komprimace dat

Cíl:

Po nastudování této kapitoly budete vědět:

- které programy považujeme za komprimační
- jaké jsou principy komprimace dat
- jak si vybrat pro naši potřebu ten nejvhodnější komprimační program.

A navíc proč vůbec data komprimovat.

Klíčová slova:

- bezztrátová komprimace
- ztrátová komprimace

Většina z nás, uživatelů počítačů, se s nimi dříve či později setká. Dat je pořád víc a víc a větších, disky a diskety nestačí, Internet je ucpaný. Ano, řeč je o komprimačních, archivačních nebo také „pakovacích“ programech. Komprimační programy představují nejlevnější řešení našeho problému, protože data jednoduše stlačí do menšího objemu. Těchto programů existuje spousta, ale jejich funkce je, jak již bylo naznačeno, stejná: zmenší velikost uložených datových souborů. Nejednoho uživatele, který při své práci pociťuje nutnost používat komprimační programy, trápí nezodpovězené otázky typu:

Proč pakovat soubory (archivovat)?

Který program je nejlepší?

Co je základem komprimace a jaké algoritmy se používají?

Jak se liší komprimace textových, binárních a grafických souborů?

Co s komprimací zvuku a videa?

Na tyto i jiné otázky se Vám pokusí odpovědět studijní text, který právě držíte v ruce. Studijní materiál je zaměřen na vysvětlení základních pojmů z oblasti komprimace a archivace dat a rozšíří Vaše znalosti o bezztrátové komprimaci. Studijní text je rozdělen do krátkých kapitol, na jejichž konci je vždy provedeno shrnutí nejdůležitějších poznatků. Každá kapitola je doplněna kontrolními otázkami, jejichž řešení naleznete v předchozím textu a dále úkoly, jejichž řešení se nachází v závěru studijního textu. Na tuto k materiál pak navazuje druhý díl týkající se ztrátové komprimace.



6.1 Komprimační programy

Komprimační programy jsou programy schopné zakódovat datové soubory do podoby, kdy je velikost těchto souborů menší ve srovnání s jejich velikostí před komprimací. Komprimační program musí být samozřejmě schopen dekomprimovat data do původní podoby.

Tak jako ostatní software i komprimační programy procházejí svým vývojem a dnes nám nabízejí mnohem větší uživatelský komfort a více funkcí, než je uvedeno v naší definici. Jednou z nich je například možnost vytváření archivů dat složených z komprimovaných souborů, které spolu logicky souvisejí. Odtud také pochází často užívaný termín archivační programy. Rozšiřující funkce však nejsou v komprimačním programu nezbytně nutné. Program označovaný za komprimační musí být schopen data alespoň komprimovat a dekomprimovat.

A proč vlastně data komprimovat? Vždyť v dnešní době existuje velké množství hardwaru, který je určen pro zálohování dat. Připomeňme jenom ZIP mechaniky, páskové zálohovače se sekvenčním přístupem atd.. Většina uživatelů však tyto možnosti nevyužívá a důvodem jsou vysoké pořizovací náklady a navíc skutečnost, že dat určených pro zálohování není zase tak mnoho. Na druhé straně pak uživatelé, kteří potřebují zálohovat velké množství dat narážejí na nedostatek místa i na vysokokapacitních zálohovacích mechanikách. Prvním důvodem, který svědčí pro komprimaci, je tedy zálohování dat při současné úspoře místa na paměťovém médiu. Dalším důvodem pro komprimování dat je možnost při zálohování vytvářet archivy dat složené z komprimovaných souborů, které spolu logicky souvisí. Soubor je na pevném disku zaznamenáván do clusterů (alokačních jednotek). Cluster je nejmenší přidělená jednotka paměti na nosiči. Jeho velikost bývá různá, a to podle konstrukce a celkové kapacity média. Typická velikost clusteru je 2 kB, což znamená, že libovolně malý soubor zabere minimálně 2 kB. Proto je vhodné soubory sdružovat do komprimovaných archivů. Navíc se ještě zvýší přehlednost uložení souborů na pevném disku.

Jedním z dalších důvodů komprimace je potřeba dálkového přenosu dat.

Komprimovaná data se samozřejmě v počítačových sítích přenášejí rychleji a tak při přenosu několikamegabytového souboru po telefonních linkách za pomoci modemu můžeme předchozí komprimací ušetřit až několik desítek minut. A to nemluvíme o ušetřených poplatcích za telefon. Při kratším přenosu se také minimalizuje nebezpečí přerušení spojení během přenosu.

K výše uvedeným důvodům mluvícím jasně pro komprimaci můžeme přidat možnost rozdělení komprimovaných souborů do bloků, které mají velikost shodnou s kapacitou běžně používané diskety. Prakticky všechny dnes používané komprimační programy nabízejí tuto funkci a často se jedná o jedinou možnost, jak přenést například 3MB soubor z jednoho počítače na druhý.

Shrnutí:

Komprimační programy jsou programy schopné zakódovat datové soubory do podoby, kdy je velikost těchto souborů menší ve srovnání s původní velikostí. Zároveň musí být schopny data dekodovat do původní podoby.



Počítačové systémy, periférie, operační systémy

Pro komprimaci dat svědčí tyto důvody:

- úspora místa na paměťovém médiu
- zvýšení přehlednosti uložených souborů
- zvýšení rychlosti dálkového přenosu dat
- ušetření finančních prostředků při dálkovém přenosu dat
- možnost rozdělení komprimovaných souborů do bloků požadované velikosti

Otázky:

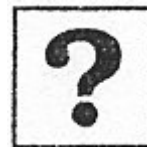
Vymezte pojem komprimační program.

Představuje označení komprimační program totéž co archivační program?

Je možné MS Word považovat za komprimační program?

Jaké znáte komprimační programy?

Který komprimační program používáte a proč?



6.2 Parametry komprimačních programů a jejich testy

Tato krátká kapitola nám pomůže nalézt odpověď na základní otázku každého uživatele komprimačních programů: „Jaký komprimační program je nejlepší?“ Seznámíte se také s pojmy kompresní poměr, rychlost komprese a dekomprese.

Při řešení tohoto problému je důležité si uvědomit, že komprimovaná data jsou různého charakteru, a proto vyžadují rozdílný přístup ke kompresi. Je zřejmé, že textová data se svým charakterem budou lišit od grafických a rozdílnost těchto dat bude upřednostňovat v tom kterém případě konkrétní komprimační algoritmy. Většinou lze předpokládat, zda datový soubor bude obsahovat dlouhé texty stejných dat (například v grafických souborech) a pak je možné použít algoritmy vyhledávající dlouhé řady dat, které jsou posléze nahrazeny jediným symbolem. Jindy jsou zase známy pravděpodobnosti, se kterými se v datech vyskytují jednotlivé symboly. Pak je jednou z možností použít frekvenčně závislé kódování založené na přepisu častěji užívaných prvků abecedy na kratší bitové posloupnosti a méně často vyskytující se prvky na delší bitové posloupnosti. Z výše uvedených skutečností vyplývá, že neexistuje jednoznačná odpověď na otázku, který komprimační program je nejlepší.

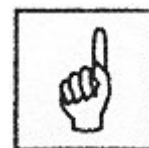
Každý komprimační program je založený na jednom algoritmu a ten bývá pro posloupnosti dat určitého typu vhodnější a pro jiné pak méně vhodný. Každý komprimační algoritmus je navržen tak, že hledá a využívá pro kompresi dat určitý řád v uložených datech. Tímto řádem může být opakování sekvence znaků, identifikace dlouhých bloků stejných dat a další.

O vhodnosti použitého komprimačního programu vypovídá test, který ovšem sleduje tzv. objektivní výkonové ukazatele – parametry:

- rychlost komprese
- rychlost dekomprese
- kompresní poměr

Rychlost komprese (komprimace) je dána časem potřebným pro sbalení do komprimované podoby. Čas potřebný pro kompresi různými komprimačními programy se u rozsáhlých souborů liší o celé minuty, což není zanedbatelné.

Rychlost dekomprese je pak čas potřebný k rozbalení komprimovaného souboru do původní podoby. Tento parametr je ze všech tří nejméně důležitý,



protože doba potřebná k dekompresi bývá téměř u všech komprimačních programů kratší, než doba potřebná ke kompresi. Proto tento parametr nebývá omezujícím faktorem při výběru komprimačního programu.

Kompresní poměr je nejdůležitější a nejsledovanější. Bývá udáván v několika různých formách, které při správné interpretaci vystihují stejnou skutečnost. Kompresní poměr bývá udáván jako:

- 1) **Poměr mezi velikostí komprimovaného a původního souboru.** Jedná se o procentuální vyjádření velikosti komprimovaného souboru vzhledem k původnímu souboru. Čím je menší hodnota procentuálního vyjádření, tím je výsledek komprimace lepší
- 2) **Někdy je kompresní poměr nahrazen doplňkem předcházejícího údaje do 100%.** V tomto případě naopak čím vyšší je hodnota procent, tím je výsledek komprimace lepší.
- 3) **Kompresní poměr se také vyjadřuje převráceným zlomkem: velikost dekomprimovaných dat ke komprimovaným**



Pokud kompresní program zkomprimuje soubor o původní velikosti 200kB na 50kB, je kompresní poměr 25%

Druhá forma zápisu by byla charakterizován kompresním poměrem 75%.

Poslední možností zápisu by byl kompresní poměre 4:1.

V praxi se můžeme setkat se všemi výše uvedenými typy kompresních poměrů. Další parametr udávající výkon kompresních programů vychází z požadavku nejlepšího kompresního poměru za přiměřený čas. Jedná se o poměr rychlost komprimace / kompresní poměr. Tento parametr často nebývá v testech komprimačních programů zmiňován.

Na soubor parametrů, které charakterizují výkon určitého komprimačního programu ve vztahu k určitému souboru má kromě použitého komprimačního algoritmu vliv mnoho dalších faktorů. Především se jedná o typ hardwaru, na kterém byl test proveden. Z tohoto hlediska je nutné brát v úvahu například :

- taktování procesoru
- přístupovou dobu pevného disku a jeho fragmentaci
- velikost a přístupovou dobu paměti RAM
- používání vyrovnávací paměti cache

typ operačního systému pod kterým byly komprimované soubory testovány Tyto parametry obvykle neovlivní dosažený kompresní poměr, ale projeví se na všech ostatních parametrech. Pokud má být test komprimačních programů opravdu použitelný, měl by kromě názvu a verze testovaných komprimačních programů obsahovat minimálně následující údaje:

- typ testovaných dat (textová, grafická,..)
- celková nekomprimovaná velikost původního souboru
- údaj o uspořádání souborů do stromové struktury
- typ počítače, na kterém byl test proveden
- operační systém, na kterém byl test proveden

Při volbě komprimačního programu se nemusíme řídit jen objektivními výkonovými ukazateli. Neméně důležitým hlediskem je i komfort ovládání těchto programů a další archivační funkce. Například prohlížení textových a grafických souborů uložených v komprimovaných archivech, možnost oddělení

Počítačové systémy, periférie, operační systémy

manipulace s jednotlivými soubory archivu. Většina moderních komprimačních programů uživateli také nabízí volbu, zda chce vytvořit archiv s nejlepším kompresním poměrem, nejkratší dobou komprese atd..

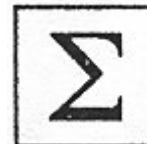
Shrnutí:

Vhodný komprimační program vybíráme podle typu dat určených ke komprimaci. Data různého charakteru totiž vyžadují rozdílný přístup ke kompresi a každý kompresní algoritmus je navržen tak, že hledá a využívá pro kompresi dat určitý řád v uložených datech. O vhodnosti zvoleného programu objektivně vypovídají parametry výkonu:

- rychlost komprese
- kompresní poměr

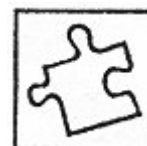
První dva parametry jsou ovlivňovány typem použitého hardware.

Jedním z hledisek výběru kompresního programu by měl být uživatelský komfort ovládání těchto programů.



Úkoly:

- 1) Zapište kompresní poměr ve všech formách pro případ, kdy je soubor o původní délce 400 kB zkomprimován na 160 kB.
- 2) Při komprimaci je dosaženo 30% kompresního poměru. Co tento údaj znamená?
- 3) Interpretujte zápis kompresního poměru 5:1



6.3 Základní druhy a přístupy v kodování

Než ukončíme obecné povídání o základních principech komprimování, je nezbytné se zmínit o druzích a přístupech v kodování. Po prostudování následujícího textu zjistíte, jaký je rozdíl mezi ztrátovou a bezztrátovou, fyzickou a logickou, adaptivní a neadaptivní komprimací. Zároveň si osvojíte tyto pojmy.



Komprimační algoritmy můžeme rozdělit podle různých kritérií. V podstatě existují dva zcela rozdílné principy komprese dat: bezztrátová a ztrátová.

Bezztrátová komprimace se používá tam, kde si nelze dovolit jakoukoliv ztrátu dat – u binárních a textových souborů. Ztráta i jen jediného bitu u binárního souboru vede v naprosté většině případů ke katastrofě v podobě další nepoužitelnosti tohoto souboru. U textových a grafických situace tak kritická není. Bezztrátovou komprimací využívají například tyto programy: ZIP, RAR, ARJ atd.. Touto komprimací se budeme podrobněji zabývat v následujících kapitolách.

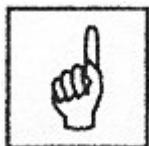
Ztrátová komprimace se používá u souborů, u nichž nevádí ztráta dat.

Jsou to redundantní soubory (soubory s nadbytečným množstvím informací).

Jedná se především o obrázky, zvuková data a video. Obrázky mohou být uloženy v různých formátech a vy jste si určitě už všimli, že obrázek BMP

zabírá na paměťovém médiu více místa, než obrázek ve formátu JPEG, PCX, GIF atd.. Tyto formáty v sobě totiž zahrnují již kompresi dat. Většina obrázků převedených do digitální formy obsahuje nadbytečné informace, které není lidské oko schopno postihnout. Lidské oko má omezenou rozlišovací schopnost ve vztahu k barevné hloubce a k obrysovým detailům. Barvy blízko ležících bodů průměruje. I když ztrátová komprimace vede ke „ztrátě“ nadbytečných dat, je kvalita obrázků měřená lidským zrakem stále stejná. Náročnost obrázků na paměť lze touto metodou snížit i 30x. Podobný způsob komprimace se používá i zvuků a videa. U videa například často postačí uchovávat rozdíly dvou následujících obrázků a z nich zpětně sestavovat celý film.

Kromě speciálního softwaru pro ztrátovou komprimaci využívají nedokonalosti lidských smyslů i některá hardwarová zařízení. Ta fungují tak, že zvukovou a obrazovou informaci, kterou lidské smysly nejsou schopny rozeznat neukládají ani nepřenášejí – ztrácejí ji cestou. Již nyní se ukazuje, že ztrátová komprese obrazu a zvuku je velice důležitá a v budoucnosti bude ještě významnější, díky masovému nástupu digitálních obrazových technologií (videokonference, obrazový telefon...). Ty totiž budou správně fungovat jedině při vysoké rychlosti přenosu a to bude zase možné jen při použití ztrátové komprimace přenášených dat. Hardwarovými aspekty komprimace se však tento text nezabývá.

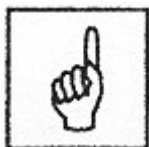


Výše uvedené rozdělení komprimací je zásadní, ale přesto existují ještě další kritéria dělení komprimačních algoritmů.

Všechny kompresní algoritmy převádějí data do kompaktnější formy, která má stejnou informační hodnotu jako původní data. Tuto skutečnost zajišťuje použití logické nebo fyzické komprimace. Rozdíl mezi nimi spočívá v tom, zda komprimační algoritmus přihlíží k logické hodnotě komprimovaných dat a nebo ne.

Logická komprimace používá logické substituce sekvence znaků jinou, úspornější řadou. Například ODS – Občanská demokratická strana, ZŠ – základní škola atd..

Fyzická komprese probíhá bez zřetele na logiku dat, se kterými pracuje. Vytváří se při ní nová sekvence znaků, jejíž vztah k původním datům lze rozpoznat výhradně s použitím dekompresního algoritmu. Bez jeho znalosti nelze data dekomprimovat. Všechny dále popsané kompresní algoritmy v tomto textu využívají fyzickou komprimaci dat.



Další kritérium rozdělení je založeno na porovnávání množství práce, která se podle algoritmu vykoná při kompresi a dekompresi dat.

Pokud je doba potřebná pro kompresi a dekompresi dat přibližně stejná, jedná se o **symetrickou kompresi**.

V opačném případě se jedná o **kompresi asymetrickou**. Některé algoritmy jsou záměrně konstruovány jako asymetrické. Používanější jsou algoritmy, jejichž práce při komprimaci je delší než při dekomprimaci. Tyto algoritmy se používají ke komprimaci souborů, se kterými často pracujeme. Algoritmy, které naopak využívají delší práce při dekomprimaci, se používají méně často. Používají se především při vytváření záložních kopií dat v pravidelných intervalech, u nichž se předpokládá, že se již s největší pravděpodobností nebudou používat.

Počítačové systémy, periférie, operační systémy

Podle přizpůsobení se charakteru dat se komprimační algoritmy dělí na adaptivní a neadaptivní.

Neadaptivní algoritmy jsou určeny výhradně pro komprimaci specifického druhu dat a obvykle obsahují předdefinované slovníky nebo řetězce znaků, jejichž pravděpodobnost výskytu v souborech dat je vysoká. Konkrétním příkladem je Huffmanovo kódování, které bude popsáno dále. Použití neadaptivního algoritmu na vhodný typ dat je velmi účinné co do dosaženého kompresního poměru i co do času potřebného ke kompresi a dekompresi dat.

Adaptivní algoritmus je schopen dosáhnout určité nezávislosti na komprimovaných datech a neobsahuje žádné statické slovníky řetězců. Adaptivní algoritmus si buduje tyto slovníky dynamicky pro každý komprimovaný soubor dat znovu v průběhu kódování. Příkladem takového algoritmu je Lempel-Ziv-Welch algoritmus, jehož popis je také součástí jedné z dalších kapitol. Adaptivní algoritmy mají širší použití, ale mají také nižší rychlost ve srovnání s neadaptivními algoritmy.



Shrnutí:

Existují dva rozdílné principy komprimace dat: bezztrátová a ztrátová. Bezztrátová komprimace se používá u binárních a textových souborů, u nichž nelze připustit ztrátu dat. Ztrátová komprimace se používá pro komprimaci obrázků, zvuku a videa, protože u těchto souborů nevádí ztráta některých dat. Komprimační algoritmy lze dělit i podle jiných kritérií na:

Fyzické a logické - podle přihlídnutí k logické hodnotě dat.

Symetrické a asymetrické – podle práce potřebné ke kompresi a dekompresi dat.

Adaptivní a neadaptivní – podle schopnosti přizpůsobit se charakteru dat, se kterými pracují.



Otázky a úkoly k zamyšlení:

- 1) Vyjmenujte základní druhy komprimací.
- 2) Vysvětlíte, proč ztrátová komprimace není vhodná pro komprimaci binárních a textových souborů.
- 3) Vysvětlíte, proč je nevhodné používat bezztrátovou komprimaci pro kompresi zvuku a videa.
- 4) Vysvětlíte rozdíly mezi adaptivní a neadaptivní komprimací, zhodnoťte jejich účinnost.
- 5) Kdy je vhodné pro komprimaci používat asymetrické algoritmy?
- 6) Z jakého hlediska vychází rozdělení komprimačních algoritmů na fyzické a logické?
- 7) U kterého typu komprimačních algoritmů je nutná znalost dekomprimačního algoritmu?
- 8) Zkuste zařadit podle výše uvedených kritérií vám známé komprimační programy.



Pokud se vám podařilo zodpovědět všechny výše uvedené otázky, orientujete se dobře v základní problematice komprimačních programů a můžete přistoupit k dalším částem studijního textu, který vám zprostředkuje konkrétnější



informace týkající se podstaty základních bezztrátových komprimačních algoritmů. V opačném případě se ještě jednou vraťte k úvodním kapitolám.

6.4 Bezztrátová komprimace

Následující kapitola vás srozumitelnou formou seznámí se základními charakteristikami nejužívanějších bezztrátových komprimačních algoritmů a jejich principem. Zároveň je uvede do souvislostí s vámi používanými komprimačními programy. Jednotlivé metody jsou popsány a vysvětleny v samostatných kapitolách a doplněny konkrétními příklady.

Bezztrátovou komprimaci (lossyless compression) využívají všechny komprimační programy určené pro komprimaci souborů. Předpokládá se totiž, že pro většinu souborů je nepřípustná jakákoliv ztráta dat v důsledku komprimačního a dekomprimačního procesu. Popis konkrétních komprimačních algoritmů začneme nejjednodušším, a to RLE algoritmem.

RLE (Run – length Encoding)



Run - length - Encoding označovaný zkratkou **RLE** je bezztrátový algoritmus, který je do češtiny někdy překládán jako proudové kódování. Jedná se o algoritmus, který je použitelný pro jakýkoliv druh dat, ale na jejich charakteru závisí dosažený kompresní poměr. Ve většině případů RLE nevykazuje tak dobrých kompresních poměrů jako složitější algoritmy, ale jeho výhodou je snadná instalace, jednoduchost a z toho vyplývající poměrně vysoké rychlosti komprese a dekomprese.

Základním principem **komprimace RLE** algoritmu je zhuštění opakovaných znaků, které se v souboru vyskytují hned za sebou. Řetězec opakujících se znaků se nazývá proud a ten je komprimován do formy jednoho paketu RLE. Paket RLE vždy obsahuje dvě informace:

- proudové číslo udávající počet znaků proudu snížený o jedničku
- proudovou hodnotu, která se shoduje s hodnotou opakujícího se znaku v proudu.



Soubor obsahuje následující sekvenci znaků : AAAkkRRRRRm. Zakóduje se do podoby 3A1k4R0m. V uvedené posloupnosti znaků komprimační algoritmus rozpozná 4 proudy a přiřadí jim čtyři pakety. Jsou to 3A, 1k, 4R, 0m. První údaj v paketu je proudové číslo a za ním je proudová hodnota.

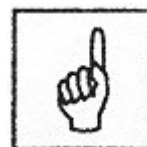
Na příkladu vidíte, že zmenšení velikosti souboru bylo dosaženo díky proudům, které obsahují více než dva znaky. Pokud se proud skládá jen z jednoho znaku, pak je komprimovaný paket dokonce větší než původní proud dat. Z této skutečnosti vyplývá, na jaká data bude tento algoritmus použitelný a na jaká nikoliv. RLE algoritmus není vhodný pro komprimování textových souborů a většiny binárních souborů, dále se nehodí pro komprimaci fotografií. Metoda RLE je vhodná pro komprimaci jednoduchých obrázků s malou barevnou hloubkou maximálně 256 barev. Takovéto obrázky vznikají například v programu Paintbrush. U nich je možné dosáhnout kompresního poměru 70%. (to odpovídá komprimaci dat na 30% původní velikosti).

Počítačové systémy, periférie, operační systémy

Protože RLE má praktický význam pro komprimaci jednoduchých obrázků s malou barevnou hloubkou, zaměříme se při dalším popisu algoritmu výhradně na tato grafická data. Existují tři přístupy k praktické realizaci algoritmu RLE ve vztahu k bitovým mapám jednoduchých obrázků. Ve všech jsou obvykle data předlohy proudově kódována sekvenčním postupem, který přistupuje k datům jako k jednorozměrnému poli dat. Bitová mapa se začíná číst a komprimovat od levého horního rohu až do pravého dolního rohu. Při komprimaci je vhodné udržovat informaci o koncích jednotlivých řádků (sloupců) bitové mapy. Tyto značky lze použít, pokud je třeba rozbalit jen část obrázku.

Praktické přístupy se dělí podle typu základních datových prvků tvořících proudy:

- bitová úroveň
- bytová úroveň
- pixelová úroveň



Bitová úroveň rozeznává jen dva proudy znaků: proud 0 a proud 1. Paket je tvořen jediným bytem logicky rozděleným do dvou částí. Nejvýznamnější bit každého bytu určuje proudovou hodnotu a sedm dalších proudové číslo. Již dříve jsme si řekli, že proudové číslo je sníženo o jednu. Důvodem je skutečnost, že délka proudu musí být vždy v rozsahu 1 – 128 znaků, 7 bitů lze totiž zapsat číslo 0 – 127. V případě, že původní nekomprimovaná data obsahují proud delší než 128 znaků, musí být při komprimaci rozdělen na více proudů.

Tato metoda je prakticky účinná pouze pro monochromatické obrázky.

Pokud budeme byt s hodnotou 151 interpretovat jako paket kódování RLE na bitové úrovni, zastupuje tento paket proud jedniček dlouhý 24 znaků. Číslo 151 v desítkové soustavě má totiž ve dvojkové soustavě vyjádření 10010111. Nejvyšší bit je jednička a nižších sedm bitů představuje číslo 23 v desítkové soustavě.



Bytová úroveň kóduje proudy opakujících se bytových hodnot a nevíšmá si dělení na bity nebo 16bitových (32bitových) slov. Paket se skládá ze dvou bytů: první udává proudové číslo v rozsahu 0 – 255 a druhý byt proudovou hodnotu ve stejném rozsahu. Bývá doplněn o schopnost ukládat přesné nekódované bytové proudy do kódovaného toku dat. Metoda RLE na úrovni bytů je vhodná pro obrázky, které jsou uloženy ve formě 1 byt na pixel.

U pixelové úrovni je jediný pixel reprezentován více než jedním bytem . Informaci o počtu bytů je možné získat v hlavičce grafického souboru. Velikost paketu RLE se liší podle počtu bytů na pixel. Jestliže obrázek podporuje schéma „3 bytes per pixel“ (tři byty na pixel), bude paket RLE čtyřbytový. První byt udává proudové číslo, další tři proudové hodnoty.

Tato modifikace je vhodná pro složitější obrázky. Některé obrázky používají pro zobrazení pixelu počet bitů, který neodpovídá celým bytům a pak je možné RLE paket vytvářet na bitové úrovni a tím zlepšit kompresní poměr.



Počítačové systémy, periférie, operační systémy

Existují i další modifikace metody RLE. Jednou z nich je modifikace, která se snaží vyhnout záporné kompresi nastávající při velkém počtu jednoznakových proudů v datech předlohy. RLE paket v tomto případě je 3 bytový: první byt tvoří příznakovou hodnotu, druhý byt proudové číslo, třetí proudovou hodnotu. Pokud komprimační program zjistí jedno-, dvou- nebo tříbytový proud, zapíše se hodnoty přímo do komprimovaného toku dat. Při dekomprimaci se pak podle příznaku v 1. Bytu rozhodne, zda se paket dekóduje podle údajů proudového čísla a hodnoty nebo se v opačném případě přímo přidá do nekomprimované podoby souboru. Nevýhodou je vyšší kompresní poměr u některých dat. Další problémy pak nastávají, jestliže nekódovaný tok dat obsahuje hodnotu, která se shoduje s hodnotou vyhrazenou pro příznak.

Další modifikace RLE komprese představuje forma vertikálních replikačních paketů nebo také pakety s opakovanými vzorkovými řádky. Tato forma se hodí pouze na druhy dat, u nichž se předpokládá opakování celých řádků. U této modifikace dosahujeme zlepšeného kompresního poměru vyjádřením toho, zda se opakuje celý předchozí řádek.

Předpokládejme, že jeden řádek obrázku je složen z dat uložených do 512 bytů a všechny pixely jsou stejné. Při dvoubytovém schématu RLE se řádek zakóduje do 8 bytů (maximální kódovatelný proud 128 bytů). Pokud má stejnou barvu prvních 100 řádků předlohy, zakóduje se těchto 100 řádků do 800 bytů.

Použijeme-li metodu vertikálních replikačních paketů stačí na 2. až 100. řádek jeden paket a na celou komprimaci tedy 10 bytů. Tato metoda je využívána grafickými formáty WordPerfect Graphics Metafile (WPG) a GEM Raster (IMG).

Shrnutí:

RLE představuje jeden z nejjednodušších komprimačních algoritmů.

Základním principem komprimace je to, že se každá sekvence stejného znaku původních dat uloží ve formě nesoucí informaci, o který znak se jedná (proudová hodnota) a o počtu opakování znaku (proudové číslo).

Nevýhody:

Úzká oblast dat, na kterých tato metoda dosahuje dobré kompresní poměry. Komprimace RLE je vhodná pouze pro jednoduché obrázky s malou barevnou hloubkou. Nehodí se ani pro složitější obrázky ani pro textové nebo binární soubory.

Výhody:

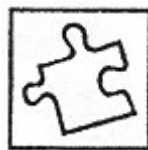
Jednoduchost algoritmu a jeho snadné použití vedoucí k vysoké komprimační a dekomprimační rychlosti.

Otázky:

- 1) Vysvětlíte základní princip RLE kódování.
- 2) Uveďte pro jaký typ dat je tato komprimační metoda vhodná a proč.
- 3) Uveďte pro jaký typ dat naopak tato metoda vhodná není a proč.
- 4) Popište tři konkrétní způsoby realizace RLE algoritmu.
- 5) Který formát obrázků využívá tohoto algoritmu?

Úkol:

- 4) Zakódujte řetězec znaků **BBBBBaaaPPPkl** RLE algoritmem. Jaká proudová čísla a proudové hodnoty budou mít jednotlivé pakety?



Lempel-Ziv-Welch algoritmus

Cílem následující kapitoly je vysvětlit základ nejrozšířenějšího komprimačního algoritmu, který používají komprimační programy (např. ARJ, PKZIP, ZOO, LHA) i různé grafické formáty obrázků. Se známete se také s pojmy jako LZ77, LZ78, které označují další formy tohoto algoritmu.

Základním principem Lempel-Ziv- Welch algoritmu je vyhledávání stejných posloupností bytů. Z odkazů na tyto posloupnosti algoritmus buduje datový slovník. Schéma komprimace je následující:

Jestliže se posloupnost bytů ve slovníku nevyskytuje je přidána do vytvářeného slovníku a nezměněna zapsána do komprimovaného toku dat. Jestliže se posloupnost bytů ve slovníku již nachází, zapisuje se do komprimovaného toku dat pouze zástupná slovníková hodnota, která odpovídá nalezené vstupní posloupnosti. Zástupná hodnota je menší než původní, a proto dochází ke kompresi.

Dekomprese je inverzní proces, algoritmus čte komprimovaný tok dat a vytváří si datový slovník posloupností bytů. Pro potřeby dekomprimace není bezpodmínečně nutné uchovávat datový slovník vytvořený při komprimaci. Existují dvě základní schémata tohoto komprimačního algoritmu. Původně byly vytvořeny algoritmy LZ77 a LZ78 pány Abrahamem Lempelem a Jakobem Zivem. Modifikací LZ78 pro potřeby hardwarových zařízení vznikla dnešní podoba Lempel-Ziv-Welch algoritmu.

Komprimační část algoritmu LZ77 se pokouší hledat nejdelší opakující se posloupnosti znaků. Pokud se taková posloupnost znaků nalezne, zapíše se jen pomocí odkazu na předcházející výskyt řetězce. Svůj dynamický slovník vytváří pomocí odkazů do již zkomprimovaného textu.

Řetězec „leze po železe“ je zakódován do podoby: leze po že[10,4]. Znaky [10,4] považujeme za schématicky zapsaný ofset, který udává, že z předcházejících deseti znaků má dekodér vybrat čtyři.

Podstatou tohoto algoritmu je použití „posuvného okna“, které obsahuje koncovou část již přečteného a zkomprimovaného textu. V tomto okně algoritmus vyhledává co nejdelší řetězec odpovídající řetězci na vstupu. V kladném případě jej zakóduje v podobě odkazu, který musí obsahovat ukazatel na začátek podřetězce a jeho délku. Aby kompresní algoritmus dosáhl co nejlepšího kompresního poměru, musí být okno přiměřeně veliké (většinou několik kB). Posuvné okno má dvě části: prohlížecké okno a aktuální okno. Na počátku algoritmu odpovídá posuvné okno aktuálnímu, v posuvném okně se pak najde co nejdelší počáteční podřetězec (předponu) řetězce z aktuálního okna začínající v prohlížeckém okně. Ten se zakóduje v podobě ukazatele na počátek podřetězce v prohlížeckém okně a jeho délku. Dvojici znaků: ukazatel, délka se od jednotlivých znaků na výstupu rozeznává nejčastěji prostřednictvím zápisu LZSS: <1, ukazatel,délka >, <0, jednotlivý znak>. Od výše uvedeného algoritmu se odvozuje celá řada modifikací nazývaná třída algoritmů LZ77.

Dekomprimace zkomprimovaného souboru touto metodou je rychlá a jednoduchá- Když ukazatel najde ofset udávající ukazatel a délku řetězce, tento řetězec zkomprimuje na výstup.



Metoda LZW, která patří do třídy algoritmů LZ78 používá odlišně slovníkové odkazy. Vytváří dynamický slovník opakujících se řetězců v průběhu komprimace a různé modifikace se liší ve způsobu vytváření tohoto slovníku. Podstata slovníkové komprese spočívá v tom, že algoritmus postupně rozpoznává a ukládá do tabulky řetězce znaků a tyto řetězce nahrazuje ve výstupním textu přirozenými čísly z předem definovaného intervalu.

Definice intervalu je závislá na charakteru dat. Např. při kódování ASCII je prvních 255 čísel vyhrazeno pro zobrazení samostatných znaků z původního souboru a další čísla se přidělují nalezeným řetězcům. Přitom se vytváří tabulka již rozeznáných řetězců, která je v paměti. Když je slovník zaplněn dříve než je přečten celý soubor, je smazán a začíná se vytvářet znovu. Smazáním slovníku se neztrácí žádná informace, jen se algoritmus připraví o možnost využití smazaných odkazů. Při kompresi se často zjišťuje, zda je již nějaký řetězec ve slovníku, a proto je součástí algoritmu i některá efektivní vyhledávací metoda.



Pro větší názornost a pro lepší pochopení algoritmu si uveďme příklad konkrétního zakódování vstupního řetězce znaků: WEB/WEB/WEB! Běh algoritmu začíná s prázdným slovníkem a řetězcem w obsahujícím první znak W zdrojového souboru. Po přečtení dalšího znaku E zjistí, zda se řetězec W+E vyskytuje ve slovníku. Pokud ne, zapíše se nový odkaz na řetězec do slovníku. Pokud řetězec obsahuje jediný znak, bude do slovníku zanesen jediný znak. Do slovníku se vlastně zapisují čísla větší než 255. Postup vytváření výše uvedeného řetězce je zachycen v tabulce.

Řetězec	Přečtený znak	Výstup	Nová položka ve slovníku
		W	
W	E	W	(256)=WE
E	B	E	(257)=EB
B	/	B	(258)=B/
/	W	/	(259)=/W
W	E		
WE	B	(256)	(260)=WEB
B	/		
B/	W	(258)	(261)=B/W
W	E		
WE	B		
WEB	!	(260)	(262)=WEB!
!	(eof)	!	

Dekompresní část algoritmu čte postupně kódy komprimovaného souboru, zapisuje příslušné řetězce na výstup a přidává nové řetězce do slovníku. Do slovníku je vždy přidán řetězec reprezentovaný předcházejícím kódem a první znak z řetězce s aktuálním kódem. Pro případ, že by byl přečten kód, kterému

Počítačové systémy, periférie, operační systémy

ještě nebyl přiřazen řetězec, je do slovníku přidán řetězec skládající se z předcházejícího řetězce a jeho posledního znaku. Slovníky vznikající při komprimaci i dekomprimaci jsou stejné.

Protože LZW algoritmy nemusí zdrojová data ani zkomprimovaná data uchovávat a nezpracovávají data po blocích, využívají se také u takových zařízení, jimiž data jen procházejí a je žádoucí, aby byla zpracovávána rychle.

Shrnutí:

Skupina komprimačních programů známá pod označením Lempel-Ziv-Welch (LZW) algoritmus je velice rozšířená jak při bezztrátové komprimaci grafických, tak i binárních nebo textových dat. Tento typ komprese používá například grafický formát GIF, TIFF a programy ARJ, PKZIP. LZW je také součástí modemových standardů pro přenos dat.

Základním principem tohoto kompresního algoritmu je vyhledávání stejných posloupností bytů v datovém souboru. Zápisem odkazů na tyto posloupnosti dat algoritmus buduje datový slovník. Každá další výskyt takové posloupnosti bytů se zakóduje jedním ze dvou postupů:

Odkazem na předcházející výskyt posloupnosti

Použitím slovníkového odkazu spojeného s příslušným řetězcem.

Výhody:

Velmi dobrý kompresní poměr.

Rychlá komprese i dekomprese a nevelké nároky na paměť.

Adaptivní metoda vytvářející dynamický substituční slovník. Tento slovník není nutné ukládat pro potřeby dekomprese.

Možnost kontinuálního vysílání zkomprimovaných dat bez nutnosti čekání na dokončení komprimace celého bloku dat.

Nevýhody:

Rychlý nárůst slovníkových kódů odkazujících na řetězce původního souboru, což vede někdy k zaplnění paměti určené pro slovníky. Z toho vyplývá nutnost smazání slovníku a vytváření slovníku nového, který nemůže využívat původní odkazy. U větších slovníků pak zase narůstá doba potřebná k vyhledávání řetězce v tabulce.

Otázky:

- 4) Vysvětlete podstatu LZW algoritmu.
- 5) Pro které typy souborů je tento komprimační algoritmus vhodný?
- 6) Které grafické formáty využívají LZW algoritmus?
- 7) Proč se tento algoritmus stal součástí modemových přenosových standardů?
- 8) Které komprimační programy vycházejí z tohoto algoritmu?
- 9) Shrňte výhody a nevýhody LZW algoritmu.

Huffmanovo kódování a jemu podobné algoritmy

V závěrečné kapitole vysvětlíme podstatu posledního významného algoritmu, který řadíme mezi bezztrátové a tím je Huffmanovo kódování. Zároveň uvedeme další s ním příbuzné algoritmy jako jsou Shannon-Fanovo a nejnovější aritmetické kódování. Tak jako v předchozích kapitolách vymezíme oblast použití těchto komprimačních algoritmů a posoudíme jejich výhody a nevýhody.



Huffmanovo kódování

Je zástupcem nejznámější skupiny algoritmů, které pracují na základě četností znaků v kódovaných datech. Z toho je zřejmé, že tyto algoritmy jsou určeny především pro komprimování textových souborů.

Základní myšlenkou je uplatnění následujícího postupu: algoritmus zjistí pravděpodobnosti výskytů jednotlivých znaků a každému přiřadí jedinečný kód.

Ten má různou bitovou délku. Přiřazení jednotlivých kódů musí respektovat požadavek na přiřazení bitově kratších kódů častěji se vyskytujícím znakům a bitově delších kódů znakům méně častým. Pak algoritmus postupně načítá znaky souboru a přiřazuje jim předem přiřazené kódy a zapisuje je na výstup. Jednotlivé varianty tohoto algoritmu se liší především způsoby přiřazování jednoznačných kódů znakům vstupního souboru.

Kódy přiřazené podle četnosti výskytu jednotlivým znakům souboru jsou definovány binárním stromem. Vytvoření binárního stromu vysvětlíme na konkrétním příkladu.



Komprimujeme soubor , který obsahuje pouze následující znaky: A, 8, 0, K, R. Při analýze pravděpodobnosti výskytu těchto znaků byly zjištěny tyto údaje:

Znak A se v souboru vyskytuje s pravděpodobností 50%.

Znak 8 se v souboru vyskytuje s pravděpodobností 12,5%.

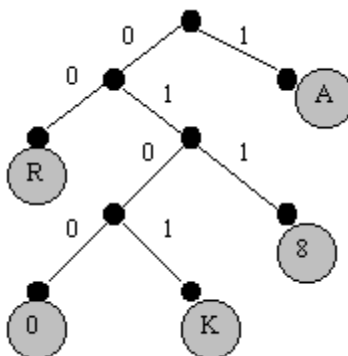
Znak 0 se v souboru vyskytuje s pravděpodobností 6,25%.

Znak K se v souboru vyskytuje s pravděpodobností 6,25%.

Znak R se v souboru vyskytuje s pravděpodobností 25%.



Binární strom vytváříme od koncových listů stromu, tj. od znaků s nejmenší pravděpodobností. Vybereme dva znaky s nejmenší pravděpodobností a uděláme z nich dva koncové listy stromu. Pak vybereme ze zbývajících zase znak s nejmenší pravděpodobností, uděláme z něj list a tak postupujeme dále až ke kořeni stromu. Každý uzel binárního stromu představuje součet pravděpodobností všech listů ve struktuře pod ním. Kódy jednotlivým znakům se přiřazují postupem po stromové struktuře, při pohybu od uzlu k uzlu zaznamenáváme jedničku vždy při cestě vpravo dolů a nulu při cestě vlevo dolů.



Podle binárního stromu přiřadíme následující kódy: A-1, 8-011, 0-0100, K-0101, R-00.

Uvedeným způsobem je možné binárních stromů vytvořit více, ale výsledná komprimovaná délka souboru bude vždy stejná, protože počet bitů přiřazených jednotlivým znakům bude ve všech případech stejný. Dekompresi vyžaduje znalost původního binárního stromu, ale je jednoduchá. Algoritmus čte kód bit po bitu a podle hodnoty bitu postupuje po binárním stromu od kořene až po list představující znak.

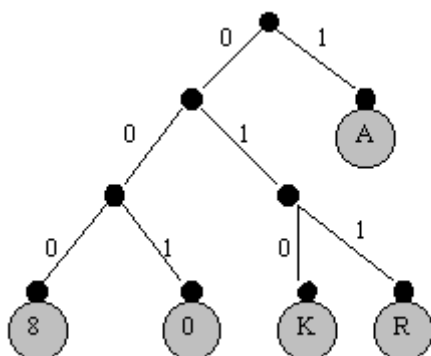
Nevýhodou tohoto algoritmu je skutečnost, že binární strom je možné bez problémů sestavit jen v případě, že pravděpodobnosti výskytu všech znaků vstupního souboru jsou mocninou čísla $1/2$. Jenom v tomto případě totiž každé vyšší patro stromu obsahuje uzel s dvojnásobnou pravděpodobností výskytu proti listům umístěným o patro níž. Jen při dodržení této zásady je při komprimaci dosaženo ideálního kompresního poměru. Ve skutečnosti tato podmínka není splněna v podstatě nikdy, a proto se pravděpodobnosti výskytu znaků zaokrouhlují. Při zaokrouhlování však může dojít k poměrně velkému nepřesnostem. Například znak s pravděpodobností 22% musí být zaokrouhlen na 12,5%. Tyto nepřesnosti pak zhoršují kompresní poměr a algoritmus tvorba binárního stromu se stává složitější.

Shannon-Fanovo kódování

Je podobné Huffmanovu kódování. Rozdíl mezi nimi spočívá v tvorbě binárního stromu. Konstrukce binárního stromu je v tomto případě jednodušší a dá se shrnout do dvou kroků:

- Soubor symbolů rozděl na dvě skupiny se stejnou nebo co nejpodobnější pravděpodobností výskytu znaků v obou skupinách. První skupině se přiřadí symbol binární nuly a druhé binární jedničky. V předchozím případě by po takovém rozdělení první skupina obsahovala znak A a druhá skupina znaky 8, 0, K, R.
- Opakuj první krok na všech dosud vytvořených skupinách tak dlouho, až každá skupina bude obsahovat jen jediný znak.

Při vytváření stromu v Shannon-Fanově kódování se postupuje naopak - od kořene k listům a v některých případech může použít několik bitů navíc, zatímco Huffmanova metoda při správném použití generuje kódy optimální.



V Shannen-Fanově variantě je kód příslušející ke znaku A reprezentován jedním bitem a všem ostatním znakům je přiřazen kód tříbitový. Tak se vlastně smaže rozdíl v pravděpodobnosti výskytu znaku 5% a 22%.



Aritmetické kódování



Aritmetické kódování odstraňuje nedostatek Huffmanova kódování, které zajišťuje optimální kompresní poměr pouze pro pravděpodobnosti jednotlivých znaků vyjádřených jako mocniny $1/2$. Aritmetické kódování odstraňuje tento nedostatek tak, že **celou zprávu kóduje jako jediné kódové slovo, jako číslo z intervalu $\langle 0,1 \rangle$** . Na začátku kódování uvažujeme celý interval. Jak se zpráva prodlužuje, zužujeme postupně interval a dolní a horní mez nově vytvářeného intervalu se k sobě přibližují. Nakonec stačí zapsat jediné číslo z výsledného intervalu a to samo o sobě reprezentuje celou zprávu. **Algoritmus komprese můžeme zapsat do několika kroků:**

- Zjištění pravděpodobnosti výskytu jednotlivých znaků ve zdrojovém souboru.
- Rozdělení intervalu na podintervaly, jejichž vzájemný poměr velikostí odpovídá poměru pravděpodobností jednotlivých znaků.
- Uložení tohoto základního intervalu.
- Vlastní komprese víceznakové zprávy. Ta bude probíhat tak, že se nejprve vybere první znak vstupního souboru a ten zúží interval $\langle 0,1 \rangle$ na podinterval příslušející tomuto znaku. Tento podinterval bude dělen stejně jako první podle načteného znaku. Postup se opakuje tak dlouho, až bude načten poslední znak.
- Posledním bodem je vybrání kteréhokoliv zlomku náležejícího do výsledného intervalu a jeho převedení do binární formy.



Pomocí aritmetického kódování zakódujte tříbytovou zprávu obsahující znaky „**XXY**“.

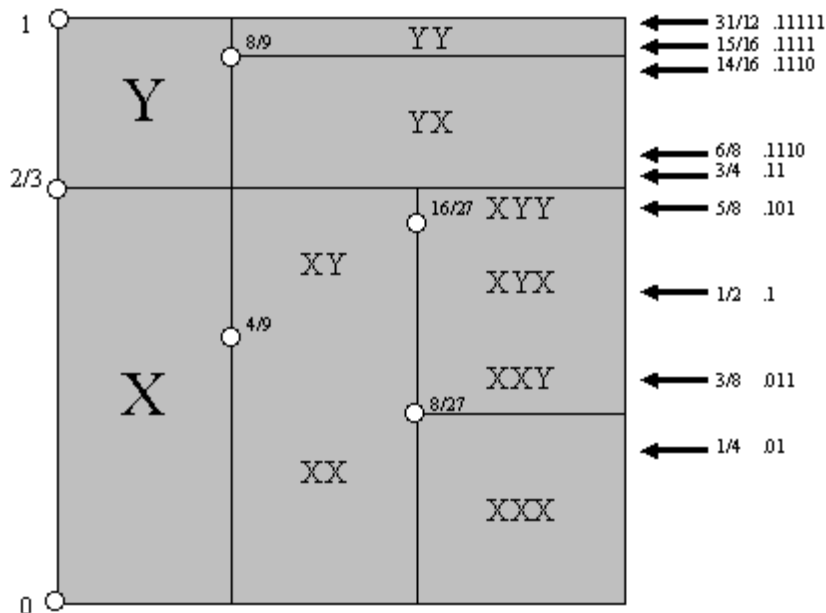
Podle prvního bodu zjistíme, že zpráva obsahuje pouze dva znaky.

Pravděpodobnost výskytu znaku **X** je $2/3$, pravděpodobnost znaku **Y** je $1/3$.

V dalším kroku rozdělíme v tomto poměru interval $\langle 0,1 \rangle$ a zapamatujeme si toto rozdělení.

Načteme první znak souboru a zjourníme původní interval na podinterval $\langle 0,2/3 \rangle$. Ten opět rozdělíme v poměru 2:1 a dělicím bodem bude v tomto případě bod $4/9$. Načteme další znak, zase **X**, který podinterval na $\langle 0,4/9 \rangle$. Ten zase rozdělíme v poměru 2:1, dělicím bodem je $8/27$. Posledním znakem je písmeno **Y** a to zase zúží interval na $\langle 8/27;4/9 \rangle$. Ještě musíme vybrat hodnotu z intervalu, která převedena do binární formy bude reprezentovat vstupní řetězec. Vybereme číslo $3/8$.

V následujícím diagramu jsou zobrazeny všechny možnosti kódování našeho řetězce:



Dekomprimace bude vypadat následovně: Načteme kód 3/8, který reprezentuje vstupní hodnotu znaku. Tento zlomek patří do intervalu $\langle 0, 2/3 \rangle$, kterému odpovídá znak **X**. Zapišeme hodnotu na výstup. Nová hodnota kódu bude $(3/8 - 0)/(2/3) = 4/9$. $4/9$ náleží opět do intervalu $\langle 0, 2/3 \rangle$. Druhý znak na výstupu bude zase **X**. Další hodnota kódu bude $(4/9 - 0)/(2/3) = 2/3$. Zlomek $2/3$ leží v podintervalu $\langle 2/3; 1 \rangle$ a odpovídá znaku **Y**. Víme, že původní řetězec měl tři znaky, a proto dekomprimace končí.



Pro potřeby dekomprimace je nutné předat dekomprimační části algoritmu také poměry pravděpodobností jednotlivých znaků a počet znaků původní zprávy. Dekomprese pak začíná načtením rozdělení intervalu $\langle 0, 1 \rangle$ na podintervaly podle poměru pravděpodobností výskytu jednotlivých znaků a počtu znaků. Další postup bude odpovídat následujícím krokům:

- Vybereme podinterval intervalu $\langle 0, 1 \rangle$, tak aby kód náležel do tohoto intervalu.
- Na výstup zapišeme znak náležející svou pravděpodobností do tohoto intervalu.

Převádění aritmetického kódování do počítačové praxe je náročné a potýká se s řadou problémů, z nichž nejzávažnější je potřeba konverze reálných čísel na bitové vyjádření. Tyto operace musí být emulované softwarově a důsledkem je poměrně dlouhá doba zpracování. Dalším problémem je zjištění pravděpodobnosti výskytu jednotlivých znaků, což se řeší sekvenčním načítáním celého souboru a vypočtením pravděpodobností z absolutního počtu výskytů. Pokud se bude jednat o textové soubory, je možno využít statických tabulek frekvencí jednotlivých znaků v různých jazycích.

Shrnutí:

Huffmanovo kódování, Shannon-Fanovo kódování a aritmetické kódování tvoří rodinu příbuzných komprimačních algoritmů založených na využití pravděpodobností výskytu jednotlivých znaků nebo jejich posloupností



Počítačové systémy, periférie, operační systémy

v komprimovaných datech. Žádná z těchto metod se příliš nepoužívá samostatně, ale dosahuje velmi dobrých výsledků v kombinaci s některou slovníkovou metodou (LZ77, LZ78, LZW). Samostatně má toto kódování praktický význam především při použití na textových souborech.

Aritmetické kódování svým principem obchází nevýhodu Huffmanova a Shannon-Fanova kódování, která dosahují optimálního kompresního poměru pouze při hodnotách pravděpodobností výskytu jednotlivých znaků, které jsou celočíselnou mocninou čísla $1/2$.

Výhody:

Velmi dobré kompresní poměry ve spojení se slovníkovou metodou.

Poměrně jednoduchý algoritmus dekomprimace.

Nevýhody:

Huffmanovo a Shannon-Fanovo kódování je nejefektivnější při hodnotách pravděpodobností výskytu jednotlivých znaků, které jsou celočíselnou mocninou čísla $1/2$.

Aritmetické kódování má vysoké nároky na technické vybavení počítače.

U aritmetického kódování je nutná softwarová manipulace s čísly o desetinném rozvoji na 20-30 desetinných míst, přitom je nutné vyvarovat se jakéhokoli zaokrouhlování – relativně dlouhá komprese.

Otázky:



- 1) Vysvětlíte podstatu Huffmanova kódování.
- 2) Čím se liší Huffmanovo kódování od Shannon-Fanova kódování?
- 3) Pro jakou komprimaci dat se tyto algoritmy využívají?
- 4) Vysvětlíte podstatu aritmetického kódování .
- 5) Shrňte nevýhody aritmetického kódování.

Korespondenční úkoly:



Zpracujte krátké pojednání srovnávající komprimační algoritmy bezztrátové komprese. Zhodnoťte také výhody a nevýhody těchto algoritmů a jejich využití ve známých komprimačních programech. Soubor zkomprimujte v programu WinZIP.

Vyberte libovolný textový a grafický soubor, zkomprimujte je ve dvou odlišných komprimačních programech a porovnejte jejich výkonové parametry.

Závěr:



Dospěli jste úspěšně k závěru tohoto studijního materiálu, který se vám snažil přiblížit problematiku komprimace datových souborů různého typu. Jeho cílem bylo seznámit vás s vymezením pojmu komprimační program, s rozdělením komprimačních programů podle obecných kritérií a především pak podrobněji objasnit podstatu nejpoužívanějších algoritmů bezztrátové komprese. Doufám, že vám pomohl nalézt odpověď na otázky spojené s komprimací dat a že vás posunul z pozice uživatele „začátečníka“ k pozici uživatele zkušenějšího a poučeného.

Řešení a odpovědi na otázky:

1. otázka

- a) Kompresní poměr je 40 %.
- b) Kompresní poměr je 60%.
- c) Kompresní poměr je 5:2.

2. otázka

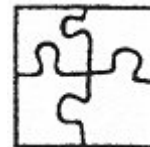
30% kompresní poměr znamená, že původní soubor byl zkomprimován na 30% své původní délky.

3. otázka

Původní délka nekomprimovaného souboru je 5x větší.

4.. otázka

Soubor se zakóduje do podoby 4B2a2P0k0l. V uvedené posloupnosti znaků komprimační algoritmus rozpozná 5 proudů a přiřadí jim 5 paketů. První údaj je proudové číslo a druhý proudová hodnota.



6.5 Ztrátová komprimace

Cíl:

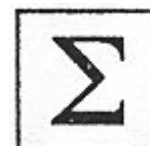
Cíl této kapitoly směřuje k tomu, aby si zájemce osvojil základní pojmy týkající se ztrátové komprimace, získal přehled o metodách a standardech, pochopil možnosti využití ztrátové komprimace v praxi. Cílem není podrobný popis jednotlivých algoritmů a schémat.

Malé opakování:

Použití a přenos běžně používaných informací, jako je zvuk, obraz a video v digitálním tvaru vyžaduje použití rozsáhlých pamětí a velkých šířek přenosového pásma. To vše má za následek zvyšování nákladů na uchovávání a přenos informací v této podobě. Máme-li zájem o výhody digitálního tvaru informace (lépe se uchovává) máme k dispozici právě technologii komprimace. Komprimace umožňuje efektivní digitální reprezentaci zdrojových signálů jako je text, zvuk, obraz i video, použitím redukováného počtu prvků digitální informace (bitů), než má originál. Pokud má být použít komprimace efektivní, musí umožňovat reprodukci komprimované informace v požadované kvalitě. Požadovaná kvalita může být v intervalu od požadavku úplné rekonstrukce původní informace (např. textové), až po omezení přenášeného frekvenčního pásma (zvuková informace), počtu zobrazovaných barev (obrazová informace, video) nebo počtu rozlišitelných zobrazovacích prvků informace (obraz, video, dynamika zvukového signálu).

Tedy zjednodušeně - komprimace je metodou efektivní reprezentace zdrojového signálu v digitálním tvaru nejmenším možným počtem bitů při zachování požadavku přijatelné ztráty věrnosti.

Metoda komprimace může být **bezztrátová** (je možné dokonale rekonstruovat původní signál zdroje informací) anebo **ztrátová**. Při ztrátové komprimaci vycházíme obvykle z modelu vnímání informace uživatelem a podle požadavků na kvalitu, resp. rozsahu vnímání, redukuje se objem přenášené



Počítačové systémy, periférie, operační systémy

informace. V tomto případě zpětná rekonstrukce signálu není možná. Na druhé straně tyto metody umožňují dosáhnout zajímavých komprimačních poměrů, z tohoto důvodu se používají pro paměťově náročná média jako je video nebo zvuk.

Metody ztrátové komprese obrazu

JPEG

Jedna z nejpokrokovějších metod ve ztrátové kompresi redundantních grafických předloh. Standardizační skupina: Joint Photographic Experts Group komise pro standardy pracující pod International Standard Organization (ISO).

Oficiální název : digitální komprimace a kódování obrázků se spojitým tónováním.

Nejedná se o samotný přesně definovaný algoritmus. Je to sada kompresních metod, kterou je možno přizpůsobit konkrétním požadavkům uživatele.

Komprimační technologie je založena na faktu, že lidské oko dokáže hůře rozeznat malé změny barev blízkých bodů, než rozdíly v intenzitě a jasu, proto je vhodnější pro barevné obrázky, než pro různé stupně šedi.

Vlastnosti:

- je vhodný ke kompresi složitých barevných grafických předloh (fotografie s velkou barevnou hloubkou - typicky 24 bitů na pixel (16 mil. barev), kompresní poměr je pak běžně 20:1, i bez vizuální újmy na kvalitě obrázku. uvedený kompresní poměr se týká pouze digitálních fotografií (tedy obrázků s rozmanitou skladbou co se týká barevnosti i drobných detailů), horších výsledků se dosahuje pro černobílé fotografie - lze rozlišit pouze odstíny šedi.

- není vhodný pro jednoduché vektorové obrázky, kde dochází ke střídání málo barev s jednobarevným pozadím (např. Malování ve Windows) zde JPG dosahuje nejhorších výsledků (zanechává na jednobarevném pozadí viditelné artefakty, které dále degradují obrázek).

Pro jednoduché vektorové obrázky je vhodnější použít některý z grafických formátů - které využívají jednoduché bezztrátové metody komprese. Většinou pracují s barevnou hloubkou do 8 bitů na pixel (256) barev .

- není vhodný pro černobílé kresby, protože nemají žádnou z grafických vlastností , které jsou podmínkou úspěšné komprese. - vyžaduje alespoň 16 odstínů šedé (4 bity na pixel).

Např. GIF je bezztrátový až do 256 odstínů šedé, zatímco JPG nikoli.

Ale barevná fotografie exportovaná například do formátu GIF tedy nejdříve ztratí většinu své

barevné hloubky , přičemž dosáhne kompresního poměru 3:1, GIF používá bezztrátovou kompresní metodu Lempel-Ziv- Welsh (LZW) , která zvýší kompresní poměr průměru na konečnou hodnotu 5:1 . Ve srovnání s kompresním poměrem 20:1 je to velice málo a ještě dochází ke ztrátám barevné informace.

Kompresní poměry od 30:1 mívají nevhodnou kvalitu - vhodné pouze jako náhledy pro knihovny.

Počítačové systémy, periférie, operační systémy

- Q faktor možnost volby nastavení kvality obrázku po kompresi. Hodnoty se pohybují od 1 do 100. Čím menší je hodnota Q faktoru, tím je lepší kompresní poměr, ale za cenu horší kvality obrázku. Cílem je rozumný kompromis.
- vysokých kompresních poměrů dosahuje díky redundantnosti dat.
- nebezpečí degradace obrázku (hromadění "ztrátovosti") při opakované kompresi a dekompresi lze snížit tak, že při potřebě udělat úpravy v takovém obrázku (JPEG) jej stačí vyexportovat se stejným Q faktorem, s jakým byl obrázek uložen dříve.

Co je příčinou zhoršení kvality obrázku při malé hodnotě Q faktoru? (Ztrátovost metody JPEG.)



Druhy obrázků JPEG

JPEG základní

Základní verze kompresního schématu JPEG funguje tak, že celý obrázek je dekomprimován z formátu JPEG během jediného skenu od horního k dolnímu okraji obrázku.

JPEG progresivní

Progresivní JPEG je určen speciálně pro publikování na stránkách WWW celý obrázek se zobrazuje v několika po sobě následujících skenovacích přebězích, během nichž se zlepšuje kvalita obrázku-výhoda k prvnímu náhledu na celý obrázek postačí zpracování malého zlomku celkových dat kódujících původní grafickou předlohu. Je výhodný pro dial-up připojení k serveru WWW. Jako určitá nevýhoda se jeví fakt, že při každém přeběhu musí počítač provést prakticky stejné množství matematických operací, jako při jediném přeběhu u základního JPEG-. Při dnešních výkonech počítačů je limitujícím faktorem přenosová rychlost modemů. Pokud se data na dekodéru objevují dostatečně rychle, pak lze dekodér upravit tak, že přeskakuje některé zobrazovací přeběhy. Tzn., že ani majitelé rychlých připojení k I. Nebudou trpět větší matematickou náročností při dekódování předlohy.

Kompresní algoritmus JPEG

Skládá se z následujících částí:

Transformace barev

Snímky se převedou do barevného modelu YUV (nebo YcbCr atd.), kde je samostatně uchovávána jasová (Y) a dvě barvonosné složky (U a V). Tento mód je pro další zpracování daleko vhodnější než obvykle v počítači užívaný mód RGB. Důvodem je již zmíněná vlastnost lidského oka lépe rozeznávat změny v intenzitě (jasu) a nikoli malé změny barvy.

Redukce barev

Pro zmenšení objemu obrazových dat se používá tzv. podvzorkování barev. Jasové složky se ponechávají beze změny, ale barevné složky jsou redukovány v poměru 2:1 v horizontálním nebo 1:1 (beze změny) ve vertikálním směru.



Počítačové systémy, periférie, operační systémy

Snížení počtu barev je prováděno průměrováním na menší počet hodnot. Podvzorkování barev je vhodné především u složitých barevných předloh.

Diskrétní kosinová transformace

Obraz se rozdělí do čtvercových oblastí 8×8 bodů a na každou takovou oblast se aplikuje diskrétní kosinová transformace. Tato transformace v podstatě převede amplitudovou informaci na informaci frekvenční, u které lze jednoduše rozlišit, které části informace jsou pro danou oblast obrazu dominantní (nižší frekvence), a které části popisují pouze jemné detaily (vyšší frekvence). Získáme tak opět matice 8×8 . Na rozdíl od původních matic, kde byla obrazová informace rozložena rovnoměrně (pro každý pixel jsme znali jeho barevnou hodnotu), po aplikování transformace se informace s největším vlivem dostaly do levého horního rohu a se stoupající vzdáleností od něj důležitost informací klesá. Tím se dominantní a důležitější složky obrazu reprezentované nižšími frekvencemi oddělí od drobných detailů (vyšších frekvencí) a obraz se tak připraví k další redukci vypuštěním těchto drobných detailů.

Kvantifikace

Až do této fáze se v úpravách obrazu jednalo o vratné operace a nedocházelo (kromě podvzorkování barev) ke ztrátě informací. Nyní přichází fundamentální krok: provedením kvantifikace dochází k nevratným změnám, proto je komprese JPEG nazývána kompresí ztrátovou. Každá ze 64 frekvencí jednotlivých bloků po diskrétní kosinové transformaci se vydělí kvantifikační koeficienty (které jsou součástí předem připravených kvantifikačních matic) a zaokrouhlí na celá čísla. Podle volby kvantifikační matice můžeme ovlivnit stupeň komprese a tím i výslednou kvalitu obrázku.

Kódování

Po provedení kvantifikace dostaneme řádkou matici, která kromě několika koeficientů v levém horním rohu obsahuje samé nuly. Tuto matici převedeme do posloupnosti čísel, přičemž postupuje po diagonálách od levého horního k pravému dolnímu rohu. Takto získaná posloupnost se zakóduje Huffmanovým nebo aritmetickým kódováním (viz bezztrátová komprimace). V části kódování nedochází tedy nedochází k další ztrátě informací.

Konečná fáze

Pod pojmem konečné fáze se rozumí přidání hlavičky ke komprimovaným datům. Aby dekompresor mohl správně rekonstruovat původní grafická data, je nutné zahrnout do hlavičky kvantifikační tabulku a tabulku Huffmanových kódů.

Dekomprese

Při dekompresi JPEG se používá inverzního postupu. Vzhledem k tomu, že inverzní postup není z matematického hlediska o moc jednodušší, nebývá doba potřebná pro dekompresi o mnoho kratší než při původním kódovacím procesu.

Dokážete odpovědět?

1. Jak se liší základní a progresivní JPEG ?
2. Popište kompresní algoritmus JPEG.
3. Je pravda že JPEG je velmi vhodný pro jednoduché vektorové obrázky ?



Shrnutí

Metoda kódování grafických předloh JPEG je vhodná pro kompresi složitých obrázků s velkou barevnou hloubkou. Jedná se o ztrát. kompresi, která ke zlepšení kompresního poměru využívá redundanci grafických dat způsobenou nedokonalým vjemem lidského oka. Pro odlišení drobných detailů (jež lze při komprimaci "ztratit") od fundamentálních tvarů v obrázcích se používá diskretní kosinová transformace.

Výhody:

Mnohem lepší kompresní poměr než u bezztrátové komprese. Díky výbornému kompresnímu poměru velice vhodný formát pro publikaci obrázků na stránkách WWW. Progresivní forma standardu JPEG je vhodná i pro uživatele s pomalým připojením přes modem.

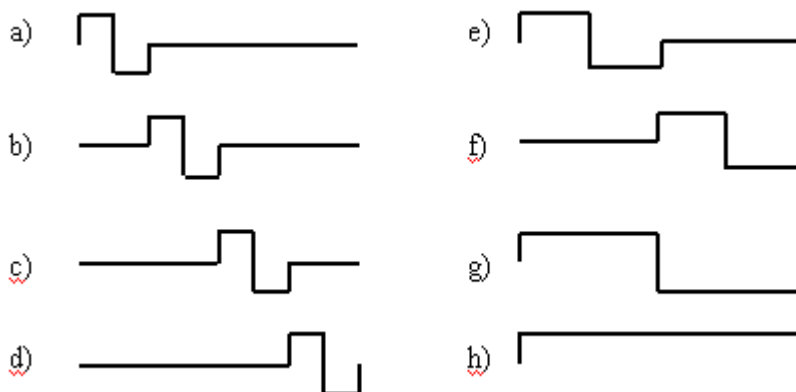
Nevýhody:

Nevhodný pro jednoduché barevné předlohy s rovnými ostrými barevnými hranami a velkými jednobarevnými plochami. Poměrně značná složitost komprimačního a dekomprimačního algoritmu promítající se do poměrně dlouhé doby zpracování.



Vlnková transformace (wavelet transformation)

- obecně se předpokládá, že se jedná o příští standard ve ztrátové kompresi obrazu
 - metoda komprese obrazu je založena na **matematické transformaci obrazu** (je méně náročná na paměť)
 - Obnovení původního obrazu se děje inverzní matematickou transformací.
- Zjednodušeně lze k výkladu použít tzv. Haarovu transformaci. Je to matematická transformace do jiného ortogonálního prostoru (podobně jako diskretní Fourierova transformace), kde bázovými funkcemi je množina funkcí, které se podobají obdélníkovým pulsům viz obrázek.



Množina funkcí pro osmiprvkovou 1D Haarovu transformaci

Popis

a),b),c)d) jsou určeny pro zakódování nejmenších detailů v nejjemnějším rozlišení obrazu. Pomocí dalších dvou funkcí - označených e),f) se budou kódovat hrubší detaily obrazu. Funkce označená g) se použije pro kódování



Počítačové systémy, periférie, operační systémy

nejhrubších změn při malém rozlišení obrazu a za pomoci poslední funkce h) se bude kódovat průměrná hodnota celého obrazu.

Nyní předpokládejte, že chceme udělat diskrétně vzorkový obraz spojité funkce. Lze to provést skenováním této funkce za použití postupného "vyclonění" funkcí $[\sin(x)/x]$.

Stejného výsledku lze dosáhnout, pokud začneme s vysokým rozlišením již diskrétně vzorkovaného obrazu funkce, který prakticky splývá s funkcí spojitou. Diferencováním obrazů při různých rozlišeních lze sestavit množinu obrazů funkce ukazující skokové změny při různých rozlišeních. Zjistným složením všech takto vytvořených obrazů potom získáme původní funkci (nebo grafický obrázek).

Jedná se vlastně o expanzi dat: pokud začínáte s obrázkem 1024x1024 bodů, dostanete př. obrázek 64x64 bodů nejhorším rozlišením a další diferencované obrázky o velikosti 128x128 bodů, 512x512 bodů a poslední o rozměrech 1024x1024 bodů.

Pokud se pozorně zamyslíte nad systémem, kterým byla tato množina obrázků konstruována, uvědomíte si, že všechny obrázky budou obsahovat znatelnou redundanci - většina hodnot v obrázcích se bude pohybovat okolo nuly. Lze tedy (na základě znalostí použitého systému, jakým jsme sestavili jednotlivé obrázky) sestavit množinu funkcí na kterékoli úrovni rozlišení tak, že jejich celková hodnota ze všech vyšších úrovní bude nula. Tím jsme sestavili ortonormální množinu bázevých funkcí pro transformaci obrazu. Matematická transformace uskutečněná za pomoci této množiny transformačních funkcí se nazývá vlnková transformace (wavelet transformation).

Taková transformace se trochu podobá Haarově transformaci s tím rozdílem, že obdélníkové funkce byly nahrazeny symetrickými funkcemi, které v nekonečnu spojitě limitují k nule. (Haarovy funkce díky svému obdélníkovému tvaru dosahují nuly ostře a nikoli postupně.)

Fraktální komprese (Fractal Image Compression)

- fraktální komprese obrazu patří mezi další rozvíjející se komprimační techniky. Jedná se také o ztrátovou komprimaci redundantních dat, která je vhodná (jako všechny ztrátové algoritmy) pro složité předlohy s velkou barevnou hloubkou
- fraktální geometrii lze ve stručnosti jedinou větou charakterizovat jako geometrii, která se svými tvary a hraničními čarami nepodobá klasické geometrii rovných čar a hladkých povrchů, ale spíše geometrii stromů, mraků či hor
- Počátky fraktální geometrie, na jejímž základě se fraktální komprese obrazu vyvinula, spadají do roku 1977.

Rozšiřující učivo

- Matematický popis takové geometrie se opírá o teorii iteračních funkcí (Iterated Function Theory) a především systémy iteračních funkcí (Iterated Functions Systems), které podal Michael Barnsley v bibli fraktální geometrie Fractals Everywhere .Kniha obsahuje i důkaz teorému známého jako Callage Theorem, který říká , jak musí vypadat sytém iteračních funkcí , aby tyto funkce mohly reprezentovat obrázek.
- Od této chvíle se myšlenky začaly obracet směrem k reprezentaci" přírodě se podobajících obrázků " za pomoci systému iteračních funkcí a k otázce, zda by tato metoda nemohla sloužit ke kompresi takových grafických předloh.k tomu bylo nejprve nutné vyřešit problém zpětné rekonstrukce obrazu ze systému iteračních funkcí. Barnsley, vyzbrojen znalostí svého Callage Theoremu , publikoval řešení v roce 1988.
- Řešení však mělo stále několik drobných nevýhod: Zakódování barevného obrázku trvalo přibližně 100 hodin, dekódování okolo 30ti minut a oba tyto procesy byly natolik složité, že u nich byla nezbytná asistence lidského operátora - proces se stále nedařilo zautomatizovat.



Řešení na sebe nechalo čekat až do roku 1992, kdy Arnaut Jacquin publikoval první praktický algoritmus fraktální komprese grafických dat.

Tento algoritmus lze metaforicky přirovnat ke kopírce , která pracuje jako klasická kopírka s několika odlišnostmi. Odlišnosti: v kopírce systém čoček způsobuje , že výstupem z kopírky jsou vícenásobné kopie originálu.. Systém čoček zmenšuje originál.

V kopírce funguje zpětná vazby tak, že výstup z kopírky se zároveň stává vstupem pro další stupeň "kopírování". Původním vstupem může být cokoli.



Skalární a vektorová kvantifikace

- není samostatnou formou ztrátové komprese , většinou se používá ve spojení s jinými metodami.
- skalární kvantifikace spočívá v tom, že skalární hodnota je reprezentována fixní podmnožinou bitů (Např. u obrázku s barevnou hloubkou 16 bitů na pixel reprezentujeme každý pixel pouze významnější osmicí bitů, dochází ke komprimaci obrazu v poměru 2:1 za cenu ztráty barevné informace.)
- vektorová kvantifikace se zakládá na reprezentaci malých polí hodnot. (např. barevný obrázek může být reprezentován 2D polem tripletů - hodnot RGB. Ve většině obrázků nevyužívají tyto triplety celý barevný prostor RGB, ale mají tendenci koncentrovat barvy do určitých částí obrázků. Pro obrázek lesa se vybere relativně malá oblast a v ní aproximovat každý triplet jednou reprezentativní barvou -zelenou. Triplet tak degeneruje ze tří bajtů na jediný bajt, za cenu rozumné ztráty barevné informace.

Metody ztrátové komprese videa a audiosignálu

MPEG

Moving Picture Expert Group, komise, která se zabývá vývojem standardů pro komprimaci videa a připojeného audiosignálu. Skupina pracuje pod organizací ISO.

Normy pro kompresi videa (podle různých oblastí využití.)

a) MPEG - 1

Tato nejpoužívanější norma je navržena s ohledem na technologii Cd tak, že nejvyšší datový tok je až 1,5Mb/s. v principu je norma MPEG - 1 definována až do velikosti obrázku 4095x4095x60 (60 snímků za sekundu)

Pracuje s neprokládanými celými snímky.

Skládá se ze čtyř částí:

- IS 11172-1 popisuje synchronizaci a multiplexaci videa i zvukového signálu
- IS 11172-2 popisuje kompresi neprokládaného videesignálu
- IS 11172-3 popisuje kompresi připojeného audiosignálu
- IS 11172-4 popisuje testování shody přenesených dat s původnímu

b) MPEG - 2

Tato forma je navržen s ohledem na využití v dálkových a satelitních přenosech signálu při zachování televizní kvality. Umožňuje rozlišení až 1638x1638 bodů, omezení je aby výška i šířka snímku byly dělitelné 16, pro lepší rozdělení na oblasti při komprimaci.

Vedle neprokládaných snímků dovoluje také použití prokládaných snímků - výhoda pro použití u televizních přijímačů, které právě prokládání obrazu používají.

c) MPEG - 3

Původně myšlena jako podpora HDTV , tedy televize s vysokým rozlišením, ale toto pokrývá po úpravách i MPEG -2. Od normy se upustilo.

d) MPEG - 4

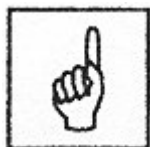
Na rozdíl od rostoucích požadavků na datový tok v předcházejících normách je norma

MPEG - 4 definována pro přenos videa a připojeného audiosignálu po pomalých linkách s rychlostí od 4800 do 64000 bitů/s, tedy převážně po modemech .Tato rychlost je velice malá a MPEG - 4 proto vychází z rozlišení 176x144 bodů, při 10 snímcích za sekundu.

Kompresi a přenos dat, videosekvencí a audiosignálů je možno řešit buď softwarově nebo hardwarově.

Softwarové řešení je vždy levnější, má však značně vysoké nároky na výpočetní výkon počítače. Je poměrně vhodné pro amatérské využití, především tam, kde se pracuje s menšími objemy dat.

Hardwarové řešení je rychlejší, nevyžaduje tak výkonný počítač, jediným jeho nedostatkem je relativně vyšší cena. Pro profesionální využití je tato investice nutná protože zatím co hardwarová komprese videa je možná v reálném čase,



Počítačové systémy, periférie, operační systémy

softwarová komprese videosouboru zabere i na výkonném počítači často mnohonásobně delší dobu, než je délka přehrání výsledného videosouboru.

Shrnutí MPEG:

MPEG je ztrátový kompresní standard pro uchovávání a přenos digitalizovaných videosekvencí snímků a připojeného audiosignálu .



Existují tři normy

- MPEG - 1 vytvořený s ohledem na přenosové rychlosti a využití v souvislosti s technologií CD
- MPEG - 2 pro využití při dálkových a satelitních přenosech vysokou rychlostí. Na rozdíl od MPEG - 1 je schopen pracovat s prokládaným videosignálem.
- MPEG - 4 navržen pro použití na pomalých (především modemových linkách)

Zatímco standard MPEG - 1 je zatím a možná i pro budoucnost vyhovující, pro MPEG - 2 stále není nalezen nejvýhodnější algoritmus a tato norma není ještě plně standardizována.

Pro přenos audio signálu v normě MPEG - 1 slouží tři "vrstvy" nazývané Layer 1 až 3. I komprese zvuku je ztrátová a lze při ní dosáhnout kompresních poměrů až 100:1. U nejsložitějšího záznamu pomocí vrstvy Layer 3 lze dosáhnout kompresního poměru až 1:12 při zachování kvality běžného audio CD.

Výhody:

Možnost ztrátové komprese přináší velkou výhodu digitalizovaného videa a audia proti oproti běžným analogovým technologiím.

Při kompresi videosnímků je možné použít tzv. pohybové vektory udávající, jak se změnil snímek oproti jinému snímku sekvence. Není tedy nutné každý snímek videa kódovat jako statický - to je výhoda, která umožňuje dosáhnout ještě lepších kompresních poměrů.

Nevýhody:

Samotná komprimace videa je složitým procesem jak na paměť, tak i na početní výkon počítače. Ačkoliv dekódování (přehrávání) je mnohonásobně jednodušší, i zde jsou paměťové nároky a především početní výkon počítače limitující pro čistě softwarové zpracování dat ve formátu MPEG.

K dosažení profesionálních výsledků v této oblasti je nezbytné využít přídatné HW zařízení (speciální karty pro přehrávání videa).

7 Počítačové viry

Cíl:

Po prostudování kapitoly budete vědět, co jsou počítačové viry, jaké je jejich rozdělení, jaké jsou principy virů.

Klíčová slova:

- počítačové viry
- infiltrace
- zdrojový text

Cílem studijního textu je, vzhledem k zadanému rozsahu této práce, pouze stručné seznámení s podstatou počítačové infekce, se zaměřením na počítačové viry, jejich druhy, dělením podle různých kritérií a jejich projevy v praxi. Mou snahou je především zpřehlednit danou problematiku. Určitě zde nenaleznete návody, jak virus napsat nebo naopak vyléčit, i když mi nedá, abych v závěru pro zájemce neuvedla pár zdrojů. Požadavky na dosažené vědomosti čtenářů z oblasti výpočetní techniky nejsou náročné (základní vědomosti o HW, SW a sítích). V případě potřeby lze využít slovník pojmů, který naleznete na konci této kapitoly. Věřím, že ti, kteří doposud jen přijímali existenci „nějakých virů“ jako fakt a nutné zlo, získají o nich po přečtení alespoň přehled.



7.1 Úvod

V první řadě je třeba vysvětlit, že při infekci počítače nemusí jít nutně o virus. Tento pojem se nesprávně uchytil natolik, že řada lidí označuje slovem „virus“ prakticky cokoli, bez ohledu na to, že se ve skutečnosti jedná například o červa (viz. infiltrace).

Počítačové viry a škodlivý software vůbec, na "řádný" softwar působí tak, že se bez ohledu na vůli uživatele množí na disku, připojuje se k programům, které se pak místo vlastní funkce věnují replikaci viru, a podle míry zlomyslnosti svého autora páchají i jinou neplechu - od nevinného blikání kontrolky na klávesnici až po úplné zničení všech dat na disku počítače, dříve i poškození některého hardwaru.

Dnes prý po světě běhá přes deset tisíc počítačových virů. Skrývají se i ve zdánlivě obyčejných textových dokumentech. Můžete je dostat na disketě i elektronickou poštou. Některé se mění doslova pod rukama a jiné parazitují na programech, jejichž původním účelem byla ochrana proti virům. Snad jediné, co počítačové viry nedokáží, je "přeskočit" z elektronického mozku PC do biologického mozku uživatele. A ukážeme si, že ani tohle možná nelze tvrdit moc kategoricky (viz. spamming).

7.2 Trochu historie

Vývoj počítačové infekce je záležitostí všehovšudy posledních 12 let. Cesta k úvahám o teoretické možnosti programů, které by byly schopny samostatného rozmnožování a infekčního šíření v paměti a na discích počítače, však byla na světě přinejmenším od roku 1976. Zřejmě vůbec první úspěšné pokusy o napsání infekčního, replikujícího se programu (viru) lze datovat až do let 1986-1987. Ačkoli už první napsané viry měly na svědomí několik menších epidemií, ještě v roce 1988 někteří významní odborníci o existenci virů pochybovali.

Jedním z prvních napsaných virů byl virus zvaný Jerusalem podle místa, kde byl nalezen a asi i napsán. Dává o sobě občas vědět dosti nepříjemným způsobem - vždy v pátek třináctého maže programy, které jsou toho dne používány. Téhož roku se v Holandsku objevil nový virus jménem Datacrime, který vyhrožoval mazáním disků 12. října téhož roku.

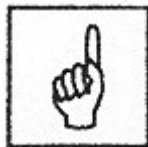
Hrozba škodlivých kódů se vyvíjí souběžně s vývojem počítačového prostředí. Vzniku nových typů virů a prostředků infekce hojně přispívá nárůst používání Internetu, expanze mamutích společností a jejich sítí a jev makrovirů.

7.3 Infiltrace a jejich typy



Než se začneme věnovat samotným virům, ráda bych se zastavila u pojmu infiltrace. Je třeba si uvědomit, že počítačové viry nejsou jediné, které nám ztrpčují život. Skutečné počítačové viry (jak už jsem se zmínila v úvodu) jsou jen součástí tzv. škodlivého softwaru, který se liší svými vlastnostmi a projevy, díky čemuž můžeme provádět jeho klasifikace. Také je třeba zdůraznit, že jmenovaná infiltrace by neměla bez přispění člověka žádnou šanci na šíření. Jejich aktivaci totiž dochází pouze v případě, že je uživatel spustí. Tvrzení, že k aktivaci a následné infekci počítače dojde při pouhém vložení diskety do mechaniky, popřípadě načtení diskety, jsou tedy zcela nesmyslná.

Počítačovou infiltrací nazveme jakýkoliv neoprávněný vstup do počítačového systému, a tím i do jeho souborů, programů atd.



Mezi infiltrace řadíme:

- hackerství
- trojské koně
- červy
- spamming
- viry

7.4 Hackerství a průniky

Jedná se v podstatě o zakázané přihlášení se (login) do počítače a jeho neoprávněné použití, často spojené se zcizením informací. Ještě nebezpečnější jsou tzv. průniky do počítačových bankovních systémů, počítačových sítí telefonních ústředí, databank apod., nejčastěji za účelem finančního zisku.

S touto činností souvisí tzv. backdoors:

- V překladu něco jako „zadní vrátka“.
- Chování je opět velice podobné trojanu s tím rozdílem, že se na sebe nesnaží vůbec upozorňovat. Tiše po aktivaci/spuštění napadeného souboru „zaleze“ do systému a čeká, až se hacker (osoba, která „útočí“ na uživatelův počítač) prostřednictvím sítě Internet připojí na postižený počítač.

Takto může hacker snadno získávat data, vymazávat mu soubory, vypínat operační systém Windows, hrát si se „šuplíkem“ CD-ROM mechaniky atd. Ke své činnosti má tedy naprosto volnou ruku a může provádět prakticky cokoli!



- Backdoory lze rozdělit na dvě části, klientskou a serverovou. Serverová je ta část, která se usadí v počítači postiženého uživatele. Pomocí klientské části, kterou vlastní hacker, lze serverovou část a tak zároveň i PC na dálku ovládat.

Klient naváže nejčastěji spojení se serverem prostřednictvím protokolu TCP/IP, na kterém je založena síť Internet. Každý počítač má v této síti přiděleno jedinečné číslo – IP adresu (v praxi to lze přirovnat k poštovní adrese). Pokud zná hacker IP adresu počítače, na kterém běží serverová část, už mu nic nebrání, aby mohl tento počítač ovládat.



Mnohem více jsou ohroženi ti uživatelé, kteří jsou připojeni statickou IP adresou, prostřednictvím pevné linky, bezdrátově apod. V případě připojení do Internetu prostřednictvím modemu je IP adresa přidělována poskytovatelem dynamicky, což znamená, že je pokaždé jiná a to stěžuje hackerům práci.

7.5 Trojské koně = Trojan Horses

- „Nejretardovanější“ forma infiltrace, uživatel se musí opravdu hodně snažit, aby se do jeho počítače dostal.
- Trojský kůň (někdy označován jako trojan) je většinou program, který se na první pohled chová jako zcela legální program, ve skutečnosti však tajně provádí škodlivé operace. Příkladem zcela typického trojského koně (jejich éra je však nenávratně pryč) je falešná verze antiviru McAfee VirusScan. I když byl trojský kůň vzhledově podobný

Počítačové systémy, periférie, operační systémy

(stejný výstup na monitor) jmenovanému antiviru, ve skutečnosti pouze mazal bezbranné soubory na disku.

- Důležitou skutečností je, že trojský kůň NENÍ narozdíl od viru schopen replikace (množení) a nepřipojuje se k hostiteli - souboru. Trojský kůň se tak nejčastěji vyskytuje na počítači pouze v jednom exempláři - souboru, který v sobě neobsahuje nic jiného, než jmenovaného trojana (tj. platí soubor = trojan).
- Z předcházejícího bodu je zřejmé, že jedinou metodou, jak se trojského koně zbavit, je smazání dotyčného souborů.
- Trojanů je sice mnoho, díky nízké inteligenci se s nimi běžně nesetkáme.

7.6 Červi - Worms

- Červ je na rozdíl od trojského koně sebereplikující program.
- Je soběstačný, samostatný, nevyžaduje žádný hostitelský program, je schopen šířit své funkční kopie nebo jejich části do jiných počítačových systémů.
- Program typu červ sám vytváří své kopie a způsobuje, že jsou spuštěny.
- Červi ke svému šíření využívají nejčastěji síťových služeb. Například elektronickou poštu. E-mail obsahuje většinou přílohu se souborem po jehož spuštění dojde k aktivaci červa. Ten se pak nejčastěji ubytuje v počítači a ve vhodném okamžiku odešle další takto postižené e-maily ostatním uživatelům na jejich e-mailové adresy, které si uživatel eviduje ve svém adresáři kontaktů.



Možná se ptáte, proč vlastně uživatel takový soubor v e-mailu spustí... je to jasné, koho by nelákal soubor s názvem Anna Kurniková či Pamela Anderson, který je navíc v těle zprávy podpořen lákavým textem. Běžný uživatel se těší na obrázek, ale ve skutečnosti jde o červa, který jen čeká na to, až ho někdo spustí.

- Vzhledem k tomu, že červi se šíří především prostřednictvím elektronické pošty, rychlost šíření je obrovská. Důkazem může být například červ I_Love_You (VBS/Loveletter.A), který se dokázal po celém světě rozšířit doslova za pár hodin.
- Červy lze obecně považovat za jistou podskupinu virů, nikoliv však opačně.

7.7 Spamming

Jedná se o šíření nevyžádaných mailů, k nimž patří i poplašné zprávy zvané hoaxy. Tyto důrazně varují před nebezpečnou počítačovou infekcí, která však ve skutečnosti neexistuje.

Text hoaxy obsahuje většinou tyto body:

- Charakteristika vymyšleného viru, zvláště pak údajný způsob šíření.
- Popis ničivých účinků viru.

Počítačové systémy, periférie, operační systémy

- Varování pochází z důvěryhodných zdrojů. Pisatel poplašné zprávy se odvolává na renomované firmy (*IBM a FBI varují* nebo *Microsoft upozorňuje* atd...)
- Výzva k dalšímu rozeslání. Mnoho nezkušených uživatelů velmi ochotně na výzvu reaguje. Nepravdivé informace se pak šíří po síti geometrickou řadou.

Hoax je vlastně jeden z několika typů obtěžujících nevyžádaný e-mailů. Dalšími typy jsou například Čínské modlitby - řetězové dopisy štěstí (obdoba klasické dopisové pyramidy), mobil zdarma, návody a nabídky na rychlé zbohatnutí, žertovné řetězové maily (viz SNĚHOVÁ KOULE), živelné rozesílání reklamy atd.

Čím spamming škodí?

- Na první pohled je tento druh infiltrace zcela neškodný. Avšak v okamžiku, kdy nezkušení uživatelé začnou na základě výzvy takového hoaxu mazat v systému soubory, které jsou „zaručeně“ virem, pak už může dojít ke škodám značného rozsahu.
- Zatěžuje server a linky.
- Zveřejňují se seznamy e-mailových adres (především způsobem *předit dál, všem apod.*).

Možná jsme věnovali otázce spammingu více místa, než bylo žádoucí, ale jak už bylo uvedeno, jedná se o velmi častý jev a nežádoucí maily vám skutečně mohou znepříjemnit a zkomplikovat práci s vaším PC. Příkladem je celá řada a každý, kdo využívá e-mail se určitě s něčím podobným setkal. A jak se proti nim bránit? NEČÍST, MAZAT!!!

O spammingu se můžete více dočíst na webu: www.antispam.cz.



7.8 Počítačové viry

Nejznámější a nejčastější formou infiltrací jsou počítačové viry. Název je odvozen z podobnosti chování těchto programátorských produktů s biologickými originály.

- Počítačový virus má schopnost vlastního množení a infikování dalších systémů bezvědomí uživatele.
- Aby byl schopen sebereplikace, musí být připojen k hostitelské proveditelné jednotce-*hpj* - (spustitelné soubory s příponou COM, EXE, SCR, VBS..., boot sektory atd.).

Počítačové systémy, periférie, operační systémy

- Po spuštění takovéto *hpj* (spuštění programu spustitelným souborem, start počítače - zavedení programu z boot sektoru), se zároveň provede kód viru . Současně dochází k replikaci viru připojením své vlastní kopie k jiným dalším takovým objektům.
- Virus je tedy program, který je schopen zapisovat sám sebe do nejrůznějších míst systému.
- Škála destruktivní činnosti je samozřejmě velmi široká a bude záviset nejen na skupinovém typu viru, ale i na jeho konkrétním typu, mnohdy variantě či mutaci.

7.9 Obecné vlastnosti viru:

- musí se skrývat, aby nebyl odhalen
- z toho důvodu má malou velikost
- je naprogramován většinou v Assembleru

Nejobvyklejší ničivé akce virů :



- vymazání souborů
- přeformátování disku
- modifikace dat
- přepis tabulky rozdělení disku (FAT)
- zničení této tabulky
- označování sektorů za vadné
- přepsání boot sektoru atd.

Protože chce virus setrvat v systému co nejdéle nepozorován, využívá některé techniky, které mu to umožňují. Jednou z těchto technik je i technika sebeidentifikace.

Technika sebeidentifikace je postup, kterým virus určuje, zda jím *hpj* byla či nebyla již napadena. Tato procedura obvykle zahrnuje vyhledávání určité hodnoty na známém místě v proveditelné jednotce.

Schopnost sebeidentifikace je nutná, jestliže se virus chce vyhnout několikanásobnému infikování jednotlivých proveditelných jednotek (to má pak za následek neustále prodlužování souboru a snazší identifikaci viru).

Vysvětlili jsme si, že VIRUS není každá počítačová infekce, ale pouze jedna z možných infiltrací.

Infiltrace je jakýkoliv neoprávněný vstup do počítačového systému, do jeho souborů, programů, atd.

Mezi infiltrace jsme zařadili hackerství, trojské koně, červy a počítačové viry.



Otázky:

- 1) definuj virus
- 2) urči rozdíl mezi červem a virem
- 3) popiš princip trojského koně
- 4) vysvětli pojem hoax
- 5) vysvětli pojem sebeidentifikace a uveď, kde se používá



7.10 Druhy virů

Třídění virů do různých skupin je důležité především z hlediska boje proti nim. Protože existuje mnoho různorodých virů nejrůznějších projevů, je třídění obtížné a nejednotné. Existuje řada hledisek pro klasifikaci virů, tím se liší i výsledky daného třídění.

Klasifikace podle způsobu umístění do paměti

- rezidentní viry
- nerezidentní viry



Klasifikace podle napadených oblastí

- bootové viry
- souborové viry
 - prodlužující viry
 - mezerové viry
 - přepisující viry (virony)
 - duplikující viry
 - kernel viry
 - multiparitní viry
- retroviry
- klastrové viry
- makroviry

Klasifikace podle chování a možnosti detekce

- - viry typu stealth a sub-stealth
- - polymorfní viry
- - tunelující viry
- - pomalé/rychlé infektory
- - obrněné viry
- - sparse viry

Mutační generátory virů

7.11 Klasifikace podle způsobu umístění do paměti

Rezidentní viry

- setrvává nastálo ilegálně v paměti
- při prvním spuštění infikovaného souboru (pokud se jedná o souborový virus) nebo při prvním zavedení systému z infikovaného boot sektoru (pokud se jedná o bootový virus) se stane rezidentním v paměti, a odtud potom provádí svoji škodlivou činnost
- virus zůstává v paměti dokud není systém vypnut
- většina virů se umísťuje až na vrchol systémové paměti, ale pod dosovskou mez 640 KB

Počítačové systémy, periférie, operační systémy

- existují i viry, které využívají pro ukrytí kusu svého kódu nízkou systémovou paměť, paměť videokarty atd.
- při své instalaci do paměti jsou velmi těžko detekovatelné;
- se zapnutím systému jsou okamžitě schopny infekce souborů či boot sektoru nebo tabulky
- rozdělení

Nerezidentní viry - viry přímé akce

- nemusí být nutně nastálo umístěny v paměti, stačí, když jsou aktivovány společně s hostitelským programem
- pak přebírají řízení jako první, provedou svoji činnost, nejčastěji replikaci a předají řízení zpět hostitelskému programu (replikací zde většinou rozumíme například napadení všech vhodných souborů, postupně nebo naráz, v aktuálním adresáři, či napadení souborů uvedených v proměnné PATH)
- nerezidentní viry jsou většinou souborové viry

7.12 Klasifikace podle napadených oblastí

Bootové viry – boot-sektorové viry



Jsou to viry, které infikují proveditelný kód nacházející se v určitých systémových oblastech disku (boot-sektory disket, tabulka rozdělení pevného disku, boot-sektor pevného disku).

Typické schéma činnosti jednoduchého boot viru

- prvním krokem je "nabootování" (tedy zavedení operačního systému) z diskety nakažené boot virem; v tomto okamžiku je vir zaveden do paměti a je mu předáno řízení
- disketa nemusí být systémová; typickým zdrojem takovéto nákazy bývá disketa pro přenos dat, která zůstala omylem v mechanice při startu počítače
- v další fázi vir obsadí dostatečný velký kus paměti, a překopíruje se do ní
- aby kopie viru plnila svou funkci, musí vir následně změnit adresy některých systémových služeb (typicky diskové operace, někdy také časovač a jiné) tak, aby jejich vyvoláním došlo nejprve k aktivaci rezidentní části viru, která pak rozhodne, co se opravdu provede
- každý boot vir je rezidentní
- následuje kontrola, zda je nakažen boot sektor pevného disku, ze kterého se běžně zavádí systém, pokud ne, vir jej okamžitě nakazí
- vir vyhledá na napadené disketě původní boot sektor, který načte do paměti, a spustí jej
- od této chvíle počítač pokračuje v běžném zavádění systému s tím, že vir je nadále aktivní v paměti připraven změnit běh věcí příštích
- pokud je po zavedení systému provedena jakákoli operace přistupující na disketu, je vir před provedením této operace aktivován a v případě, že tato disketa ještě není napadena, infikuje

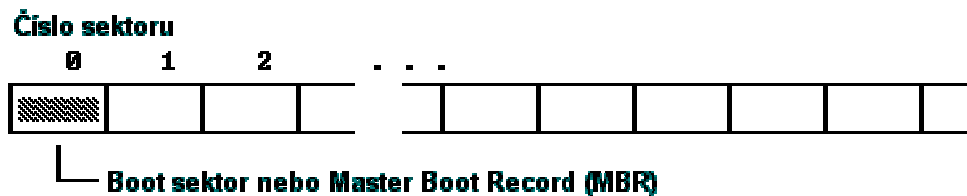
Počítačové systémy, periférie, operační systémy

- pokud jsou splněny zadané podmínky, provede vir připravenou akci (výpis textu, formátování disku apod.).

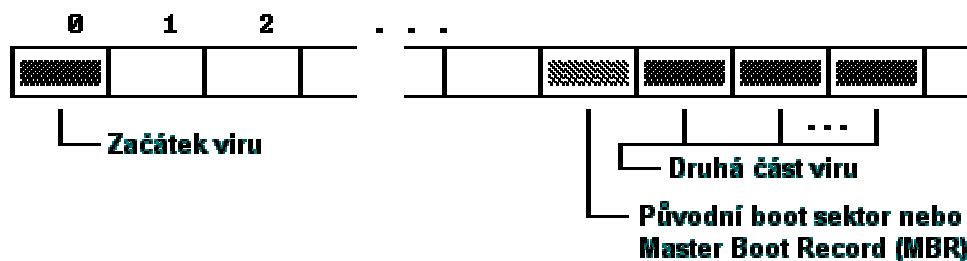
Viry této kategorie v podobě, jak byly právě popsány, jsou poměrně zranitelné. Jejich přítomnost je často patrná ze zmenšení dostupné operační paměti o velikost, kterou zabral vir (velikost paměti lze zjistit např. programem CHKDSK či MEM). Pokud se vir neobtěžuje své umístění operačnímu systému ohlásit (i takové jsou), tak riskuje, že bude přepsán někým jiným a bude následovat operačního systému.



Pevný disk před infekcí boot virem...



..a po infekci:



Možná místa pro umístění původního boot sektoru a druhé části boot viru:

- sektory v prázdných klastrech
- sektory v používaných klastrech
- sektory v systémových oblastech (hlavní adresář...)
- sektory, které se nacházejí mimo aktivní rozsah disku (na to má vliv partition table)

Souborové viry

Druhou a zřejmě nejrozšířenější skupinou jsou viry souborové. Jak již z názvu vyplývá, jejich hlavním hostitelem jsou soubory. Spustitelné soubory mohou mít několik formátů. V operačním systému DOS rozeznáváme dva základní, COM a EXE.

Spustitelný soubor typu COM

Počítačové systémy, periférie, operační systémy

- nejjednodušší formát spustitelného souboru o maximální délce 64 KB (=délka paměťového segmentu)
- nemá žádnou hlavičku a tak je jeho infekce naprosto jednoduchá: při infekci tohoto souboru si virus obvykle uschová do svého těla první tři bajty COM souboru a místo nich vloží instrukci skoku (JMP), která se v budoucnu postará o spuštění připojeného těla viru
- při spuštění infikovaného souboru dojde nejprve k aktivaci viru, který v operační paměti obnoví původní 3 bajty a díky tomu dojde i ke spuštění původního programu, který byl v souboru před infekcí

Spustitelný soubor typu EXE

- - struktura spustitelného souboru typu EXE je mnohem složitější než v případě COM souboru
- - skládá se ze dvou částí, z hlavičky (header) a vlastního kódu programu

Pro zajímavost uvádím celou strukturu hlavičky:



Offset:	Popis:
00h	Signatura souboru (znaky MZ)
02h	Velikost poslední stránky v B
04h	Délka souboru v 512 B stránkách
06h	Počet položek v relokační tabulce
08h	Velikost hlavičky v 16 B paragrafech
0ah	Minimální potřebná paměť v paragrafech
0ch	Maximální potřebná paměť v paragrafech
0eh	Počáteční hodnota SS
10h	Počáteční hodnota SP
12h	Kontrolní součet
14h	Počáteční hodnota IP
16h	Počáteční hodnota CS
18h	Offset první relokační položky
1ah	Úroveň překrývání

- i v případě EXE souboru si musí virus uschovat do svého těla několik důležitých údajů (offset 0eh až 11h) a některé další musí vypočítat (offset 02h až 05h)
- při spuštění infikovaného souboru opět dojde nejprve k aktivaci viru, který v operační paměti obnoví původní uschované údaje v hlavičce, díky čemuž dojde i ke spuštění původního programu.

Jak je vidět, struktura souborů COM a EXE je rozdílná, stejně jako je rozdílný způsob infekce těchto souborů. Virus tak musí být schopen rozeznat oba typy spustitelných souborů. Virus může tento problém vyřešit následovně:

- typ spustitelného souboru odvodí přímo z jeho názvu (např. AHOJ.COM -> jde o COM)

Počítačové systémy, periférie, operační systémy

- typ spustitelného souboru odvodí podle jeho obsahu (EXE soubory mají v prvních dvou bajtech znaky "MZ"). Evidentní je, že druhý způsob je bezpečnější.

Souborové viry mohou infikovat i batové soubory (BAT), ovladače (SYS) apod. Navíc jsou souborové viry zaměřeny na různé operační systémy (Windows 3.x, Windows95/NT, OS/2, Macintosh, Unix...). Mechanismus činnosti těchto virů je však ve všech případech podobný. Souborové viry můžeme dále dělit podle metody infekce. Toto dělení si nyní ukážeme.



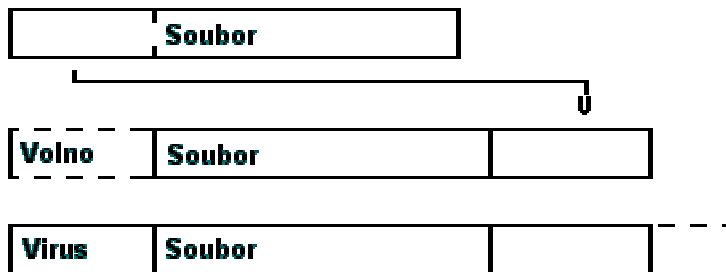
Prodlužující viry též Parasitic viruses - parazitické viry

- souborové viry, které při infekci změní (většinou prodlouží) obsah cílového souboru
- nepoškodí obsah cílového souboru (narozdíl od přepisujících virů)
- parazitické viry se dokážou umístit před původní program (prepending), nebo za něj
- (appending), popřípadě do středu souboru (inserting)
- nejčastěji se však parazitické viry připojují na konec souboru

Souborový parazitický virus na začátku souboru (prepend)

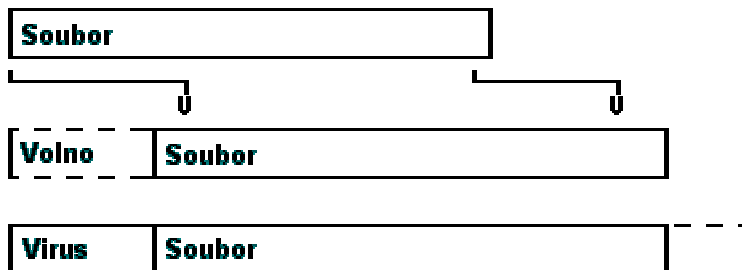
1. metoda

("vyseknutí" části původního programu a následné umístění na konec souboru - na konci uschované údaje)



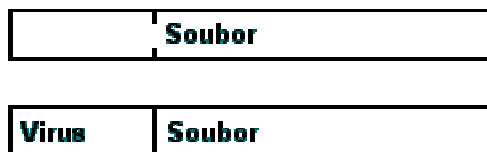
2. metoda

(posun celého původního programu doprava)



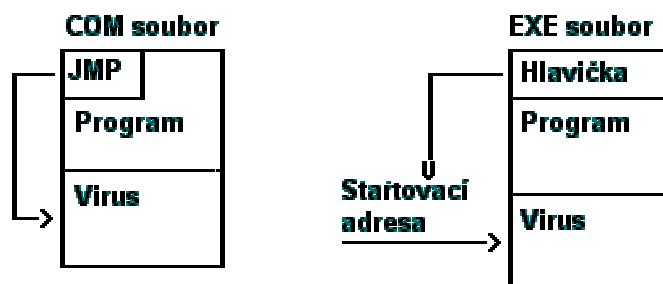
3. metoda

(přepsání části původního programu - trvalé znehodnocení původního programu - na konci uschované údaje)



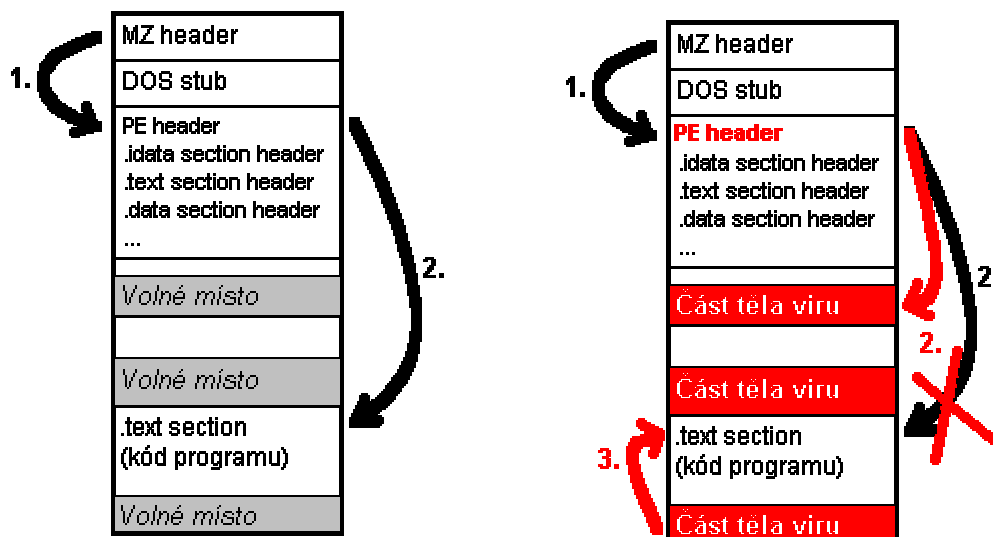
Souborový parazitický virus na konci souboru (append)

Tento způsob je nejpoužívanější a bližší informace jsou nahoře.



Mezerové viry – Cavity virus

Metoda (oblíbená zvláště ve Windows), při které se tento souborový virus "rozleje" do nevyužitých míst (a těch je hodně) hostitelského souboru (především COMMAND.COM). Napadený soubor má pak stejnou velikost jako původní originál. Typickým příkladem viru z této kategorie může být virus Win32/CIH (Černobyl).



Přepisující viry - Overwriting viruses – (Virony)

- souborové viry, které přepíší obsah cílového spustitelného souboru vlastním kódem (tělem), a tak zničí původní obsah souboru
- takto infikovaný soubor je již nefunkční a nemůže být opraven

Počítačové systémy, periférie, operační systémy

- po jeho spuštění dojde pouze k aktivaci viru, který se ve většině případů rozmnoží do dalších spustitelných souborů (COM, EXE...). Snad ve všech případech se jedná o nerezidentní viry (viry přímé akce)
- na masové rozšíření nemají prepisující viry žádnou šanci

Duplikující viry

- napadají soubor typu exe tak, že vytvoří nový soubor se stejným jménem, ale s příponou .com, do kterého umístí své tělo
- při volání původního souboru se podle dosovských priorit dává přednost com-souboru
- dojde ke spuštění viru a ten spustí původní program
- příkladem tohoto viru je AIDSII



Multiparitní viry

- viry, které se chovají jako bootové i jako viry souborové
- jsou poměrně mladé, nejznámější je například One_Half

Retroviry

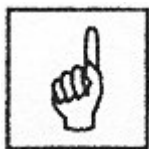
- cílem je obejít či znemožnit činnost konkrétního antivirového programu
- napadenou oblastí jsou tedy soubory příslušející antivirovým programům
- existují viry, které vypínají rezidentní hlídací programy (VirStop, Vshield, TbScanX apod.), jako například virus Tequila

Makro viry - Macro viruses

- makroviry patří k nejrozšířenější skupině virů i když jich je o hodně méně než klasických virů; jak již z názvu vyplývá, pro své šíření využívají makra
- makra jsou programy, které si uživatel může sám vytvořit pro usnadnění práce v některých aplikacích; typickou aplikací může být například Microsoft Word, či Microsoft Excel
- makrojazyk, ve kterém lze makra vytvářet je v těchto produktech natolik dokonalý, že v něm lze vytvořit i šířící se program (makrovirus); celý chod makroviru je závislý na existenci auto-maker (samospouštěcí makra)
- auto-makra jsou speciální makra, která se spouštějí při určité operaci (například při otevření dokumentu, uzavření dokumentu apod.) a ne na přímý pokyn uživatele (tedy manuální spuštění makra - klávesovou zkratkou apod.); pokud tedy dokáže makrovirus tyto auto-makra využít, může se úspěšně šířit (připojí se k dalšímu dokumentu opět ve formě maker)
- že makra (ať makroviru, či makra uživatele) jsou společně uloženy s dokumentem (v případě Wordu) či s tabulkou (v případě Excelu) v jednom souboru; přestupným místem makroviru se stává globální šablona, která se v případě Wordu jmenuje NORMAL.DOT; tato šablona se automaticky otevírá při každém spuštění Wordu, takže pokud ji makrovirus napadne, má kontrolu nad Wordem hned po jeho startu
- nutné je též poznamenat, že makrojazyky některých produktů se mohou lišit, proto například makrovirus pro textový editor AmiPro není schopen provozu pod editorem Word apod.

Schéma infikovaného a neinfikovaného souboru

Neinfikovaný soubor (dokument, tabulka)	Infikovaný soubor (dokument, tabulka)
Hlavička souboru	Hlavička souboru
Systémová data (adresář, FAT)	Systémová data (adresář, FAT)
Text	Text
Fonty	Fonty
Makra (pokud jsou)	Makra (pokud jsou) Makra viru
Další data	Další data



Drtivá většina makrovirů se dokáže šířit v aplikacích společnosti Microsoft.

Makroviry v dnešní době:

- **Stealth** - tyto makroviry maskují svoje makra, aby se tak bránily proti snadnému odhalení. Běžný makrovirus lze například jednoduše odhalit pomocí menu Tools/Macro, v němž uvidíme jednotlivé makra viru. Stealth virus tomu však dokáže zabránit, i když možná ještě podezřelejším způsobem. Menu Tools/Macro totiž úplně odstraní, nebo ho znefunkční... Jistou výjimkou jsou makroviry označené jako "Class". Ty svoje tělo ukládají do speciálního modulu ThisDocument, popřípadě ThisWorkBook a tak se dokážou vyhnout detekci přes menu Tools/Macro i bez použití "stealth" technik. Metodu "Class" lze využívat až v jazyce VBA5.
- **Polymorfní** - tyto makroviry dokážou modifikovat strukturu vlastního těla. Polymorfní makroviry pak nelze detekovat podle sekvencí, či podle klasických kontrolních součtů (CRC).
- **Multipartitní** - tato skupina makrovirů se dokáže šířit několika způsoby. Například makrovirus Shiver je napsán tak, že se dokáže šířit jak v programu Word, tak i v programu Excel. Makrovirus však může vypouštět například i souborový viru (WM/Navrhar), čímž se makrovirus stává opět multipartitním.
- **Multiplatformní** - některé makroviry se dokážou šířit pod různými systémy, kde se některé produkty, především společnosti Microsoft používají (PC, MAC...).



7.13 Klasifikace podle chování a možnosti detekce

Tunelující viry

- „tunelování“ je nejnámější technikou, kterou se virus brání

Počítačové systémy, periférie, operační systémy

- spočívá ve schopnosti vyhledávat původní adresy systémových, nejčastěji diskových služeb a ty pak používat při práci s disky ve snaze obejít případný antivirový software
- jiný způsob aktivního boje proti antivirovým programům spočívá přímo v manipulaci s jeho rezidentní částí; některé antiviry totiž komunikují se svými rezidentními hlídači pomocí několika snadno přístupných služeb (které jsou nadto v mnoha pramenech dokumentovány) a obsahují i funkci "dočasná deaktivace" hlídače
- pro vir tedy není nic snazšího, než se dotázat (jak jinak, než pomocí zmíněných služeb) na přítomnost takovýchto rezidentních hlídačů v paměti, a pakliže tyto úslužně odpoví "ano, jsem tady", pomocí známé služby je deaktivovat
- vir může také obsahovat část kódu, která manipuluje s datovými soubory antivirů (to však jsou spíše retroviry), nejčastěji s databází sekvencí virů nebo kontrolních součtů. Jejich změnou nebo smazáním může dosáhnout toho, že zůstane při antivirové kontrole neodhalen.

Mutační generátory virů

- Nejde o viry jako takové, ale o kódy, které mohou být přidány k jakémukoliv viru a tím se z něj stane polymorfní virus. Také se jim říká polymorfní či mutační motory.
- Tyto viry jsou velmi špatně detekovatelné. Navíc, je-li takový polymorfní motor vhodně připojen k původnímu viru, bude přeměňovat virový kód po každé nové infekci.
- Nejznámější je Mutating Engine (MtE), napsaný osobou, která si říká Dark Avenger. Další ze známých je generátor TPE – Trident Polymorphic Engine.
- Nutno podotknout, že s těmito prostředky už si klasické skenery neumějí poradit.

Tato kapitola byla věnována klasifikaci virů z hlediska způsobu umístění do paměti, podle napadených oblastí, podle chování a detekce a na mutační generátory virů.

Podrobněji jsme se věnovali souborovým, bootovým virům a makrovirům, které tvoří skupinu virů podle napadené oblasti.

Viry souborové jsme rozčlenili na dalších šest skupin virů a to z hlediska jejich projevů a replikací na prodlužující, mezerové, přepisující duplikující, kernelové a multiparitní.

Podle chování a možností detekce jsme rozdělili viry na viry typu stealth, polymorfní, tunelující, pomalé/rychlé infektory, obrněné viry.





Otázky k textu:

- 1) uveď rozdíl mezi rezidentními a nerezidentními viry
- 2) popiš podrobněji princip a projevy bootových virů
- 3) které soubory nejčastěji napadají souborové viry a jakým způsobem infekce probíhá
- 4) čím se liší mezerové viry od virů parazitních
- 5) co jsou „virony“
- 6) stručně charakterizuj retroviry
- 7) proč jsou makroviry takovou hrozbou, v čem spočívá jejich nebezpečí



Úkoly k zamyšlení:

1. Je možné infikovat počítač načtením „zavirované“ diskety? Svou odpověď zdůvodni.
2. Je formátování disku spolehlivým způsobem, jak odstranit viry ?
3. Může se vir rozšířit i na disketu chráněnou proti zápisu?
4. Může vir poškodit nebo zničit hardware počítače?

7.14 Slovník pojmů

Assembler

Programovací jazyk nejnižší úrovně. Program je zapisován přímo pomocí instrukcí procesoru (jejich zkratk) a může tak maximálně využít vlastností počítače. Převážná většina počítačových virů je vytvořena právě v tomto jazyce. Termín assembler také označuje překladač zmíněného jazyka do kódu proveditelného procesorem. S příchodem OS Windows 9x/2000/NT se ke slovu dostávají i vyšší programovací jazyky (C++, Delphi, Visual Basic atd.).

Autoexec.bat - soubor

Textový soubor obsahující seznam příkazů, které operační systém provede automaticky při svém startu. Lze jej mimo jiné využít k automatickému spouštění antivirové kontroly, stejně jako k automatickému zavirování počítače hned po startu (v případě napadení programu, který je z tohoto souboru spuštěn).

BIOS - "Basic Input Output System"

Pod operačním systémem (MS-DOS, Windows 9x) se nachází ještě jedna vrstva programů tvořících tzv. BIOS. Tyto programy jsou tak důležité, že jsou uloženy v paměti ROM. BIOS transformuje požadavky od programů na sekvence pro řízení hardwaru (disky, tiskárny, monitory...).

CARO (Computer Anti-Virus Researchers Organisation)

Organizace CARO vznikla v roce 1991 a založili ji pánové Fridrik Skulason (Virus Bulletin, Alan Solomon (S&S International) a Vesselin Bontchev (Univerzita v Hanburgu). Rozhodli se totiž, že pojmenování nových virů bude mít jistá pravidla. Základní pojmenování by mělo vypadat takto:

Family_Name.Group_Name.Major_Variant.Minor_Variant[:Modifier]
Musejí být používány pouze alfanumerické znaky [A-Za-z0-9_\$\$%&!'"#-], mezery se mají vytvářet podtržítkem (_). Např.: Dark_Avenger. Jednotlivé části pojmenování viru musí být dlouhé maximálně 20 znaků. Nesmí být používány názvy firem, jména atd..... Nejbliže k pojmenování CARO má jednoznačně antivirus F-Prot, na druhém konci pak stojí PC-Cillin.



CMOS

Paměť CMOS obsahuje velmi důležité informace o perifériích a částech počítače. Její obsah je zálohován baterií. Virus může údaje v této paměti poškodit a tak znemožnit start počítače. Náprava je velice jednoduchá, PC v takovém případě vypíše něco ve smyslu "CMOS checksum error". Přesto je umožněn vstup do setupu biosu, kde stačí např. obnovit a uložit defaultní hodnoty. Po dalším restartu PC bude již vše v pořádku.



Config.sys - soubor

Textový soubor obsahující konfigurační pokyny pro operační systém. Zde se také nacházejí příkazy pro zavádění potřebných ovladačů. Config.sys má podobný účel jako Autoexec.bat, který se však zavádí po startu počítače později.

CRC - kontrolní součet

Číselná hodnota vypočítaná podle daného algoritmu na základě obsahu souboru (případně libovolných dat). Případná změna výchozích dat se projeví jiným výsledkem kontrolního součtu. Ve své původní podobě se jedná o prostý součet všech bajtů souboru; téměř všechny programy však používají nejrůznější modifikace tohoto algoritmu s cílem zvýšit spolehlivost a snížit napodobitelnost tohoto součtu. Kontrolní součty bývají používány pro přesnou identifikaci viru a jsou též důležitou součástí kontroly integrity.

Debugger

Ladící prostředek (program) určený k vyhledávání chyb při vývoji programu. Umožňuje mimo jiné sledovat vykonávání programu po jednotlivých instrukcích, což jej činí neocenitelným pomocníkem při analýze počítačových virů. Mnoho virů (přesněji jejich autorů) je si této slabiny vědomo, proto používají programové konstrukce, které mají takové sledování (trasování) programu zkomplikovat.

Dekryptor

Úsek kódu viru, který zajišťuje převedení zakódované části viru do původní, spustitelné podoby. Nejčastěji se vyskytuje na začátku viru. Může, ale nemusí mít konstantní podobu v různých generacích téhož viru.

Dropper

Dropper je program, který byl navržen s cílem instalovat virus do systému. Podstatné je, že virus je v tomto programu obsažen tak, že nemůže být detekován virovými skenery. Jinými slovy, dropper není program infikovaný virem a velmi často jsou jako droppery používány trojské koně. Dropper může svou funkci splnit tím, že virus instaluje do paměti a nebo přímo infikuje nějakou proveditelnou jednotku při instalaci viru.

Emulace kódu

Postup, při kterém nejsou jednotlivé instrukce programu prováděny přímo procesorem, ale jsou zpracovávány speciálním programem. Takto lze kromě jiného velmi bezpečně sledovat provádění kódu a jeho vliv na počítač (s možností zabránit nebo změnit provedení nevhodné instrukce); toto nebezpečí je ovšem vykoupeno poměrně značným zpomalením takto emulovaného kódu oproti přirozenému zpracování procesorem.

Falešný poplach

Situace, kdy antivirový program označí zdravý soubor či jinou oblast za napadenou virem. K tomu může dojít díky náhodné podobnosti kódu neškodného programu s částí skutečného viru; falešné poplachy se vyskytují například u programů, které se nechovají (obdobně jako viry) k systému zcela



korektně. Rozhodnutí o tom, zda je soubor skutečně napaden, nebo se jedná o falešný poplach je nejlépe přenechat odborníkům. Příliš vysoké procento falešných poplachů může jinak účinný antivir znehodnotit.

Fronta instrukcí procesoru (prefetch queue)

V době vykonávání jedné instrukce procesorem je několik následujících instrukcí již načteno v procesoru, kde čekají ve "frontě" na své dokončení. Změna těchto instrukcí v operační paměti již nemá na jejich provádění vliv, což může způsobit na pohled nepochopitelné chování programu. Délka této fronty instrukcí (tj. počet a velikost instrukcí, které jsou v této frontě uloženy) se u různých typů procesorů liší.

Germ virus

Jako germ virus se označuje virus ve své nulté generaci. Germ virus vzniká například po jeho kompilaci ze zdrojového kódu (ASM) do spustitelného tvaru (COM, EXE) (po vyléčení má soubor délku 0). Germem je třeba i třibajtový COM soubor napadený virem testujícím délku své oběti a který se normálně takto krátkým souborům vyhýbá. Obdobně může být germem i COM virus přeložený ze zdrojáku do EXE tvaru. Zde je to však technicky dost obtížné odlišit, zda se jedná o Dropper či Germ.

Hoax

Hoax je emailová zpráva, která většinou upozorňuje na velmi nebezpečné, ve skutečnosti neexistující viry. Nezkoušený uživatel pošle tuto zprávu dalším lidem, aby je upozornil před nebezpečím (které ve skutečnosti neexistuje...). Je opravdu hrozné, že se hoax někdy šíří rychleji než skutečný virus :(Pro detailní informace doporučuji navštívit www.hoax.cz.

Identifikace viru

Identifikaci viru se většinou rozumí spolehlivé rozpoznání viru včetně jeho přesného určení. Takováto přesná identifikace má význam pro uživatele, který může lépe zjistit na čem je, tak pro vlastní antivir, který může poměrně bezpečně provádět další akce, jako je například léčení.

Instrukce

Instrukce jsou elementární příkazy pro práci procesoru.

Intended virus

Obvykle jsou takto označovány viry, které by měly dělat něco, co ve skutečnosti dělají špatně, nebo vůbec. Můžou za to samotní autoři virů, kteří se často snaží vytvořit něco, na co nemají.

In the Wild (ItW) - "viry v divočině"

Seznam In the Wild vytváří asi 46 specialistů z celého světa. Do tohoto seznamu jsou zapisovány viry, které se v jednotlivých oblastech světa nejvíce vyskytují (jsou hlášeny jednotlivými pozorovateli). Podle tohoto seznamu lze zjistit, jaké viry nás mohou nejvíce ohrozit. In-the-Wild list lze najít na adrese: <http://www.wildlist.org/WildList>.

Operační systém (OS - operating system)

Operační systém je software (program) umožňující uživateli komunikovat s počítačem/"železem" (hardware).

Sekvence - Signatura - Řetězec - String

Skupina bajtů vyskytujících se v tělech virů, podle kterých je skener hledá.

Trasování

Stav, kdy je kód prováděn po jednotlivých strojových instrukcích, přičemž po každé instrukci je možno analyzovat její vliv na stav počítače. Trasování

Počítačové systémy, periférie, operační systémy

programu lze provádět ručně (např. pomocí debuggeru), nebo automaticky (tak to dělají některé antiviry).

Trigger

Anglický termín pro spouštěcí podmínku, jejíž splnění má za následek provedení nějaké speciální, často škodlivé, činnosti viru (do té doby se vir snažil být co nejméně nápadný). Typickým příkladem takové podmínky je dosažení nějakého data.

Zdrojový text programu - "Zdroják"

Výchozí tvar programu, kde jsou jednotlivé požadované činnosti popsány v textové formě pomocí smluvených konvencí daného programovacího jazyka. Zdrojový text nemůže být přímo spuštěn. Nejprve musí být překladačem převeden do binární podoby.



8 Konfigurace počítače PC

Základní konfigurace počítače PC se skládá z následujících částí:



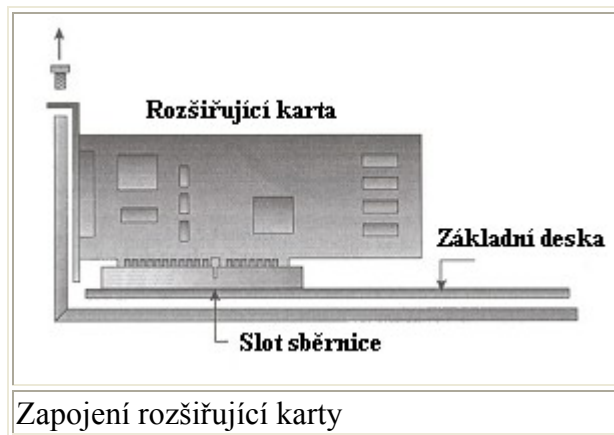
- **Základní jednotka:** část je uzavřena do skříně počítače, obsahuje (nebo může obsahovat) tyto komponenty:
 - Základní deska
 - Procesor
 - Numerický (matematický) koprocessor
 - Operační paměť
 - Cache paměť
 - CMOS paměť
 - Mechaniky pružných disků
 - Pevné disky
 - Rozhraní pevných disků
 - Interní mechanika CD-ROM
 - Interní mechaniky jiných diskových médií
 - Videokarta (grafická karta)
 - Zvuková karta
 - I/O karta
 - Síťová karta
 - Další zařízení
- **Monitor**
- **Tiskárna**
- **Klávesnice**
- **Myš**
- **Externí mechaniky diskových médií**
- **PCMCIA zařízení**
- **Scanner**
- **Další zařízení**

8.1 Základní deska (mainboard, motherboard)

Deska plošného spoje tvořící základ celého počítače. Základní deska obsahuje:

- Procesor (mikroprocesor)
- Patiči pro numerický koprocessor (popř. osazený koprocessor)
- Obvody čipové sady
- Rozšiřující sběrnici (bus)
- Paměti
- Vyrovnávací cache paměť
- Sloty umístěné na rozšiřující sběrnici pro připojení rozšiřujících karet

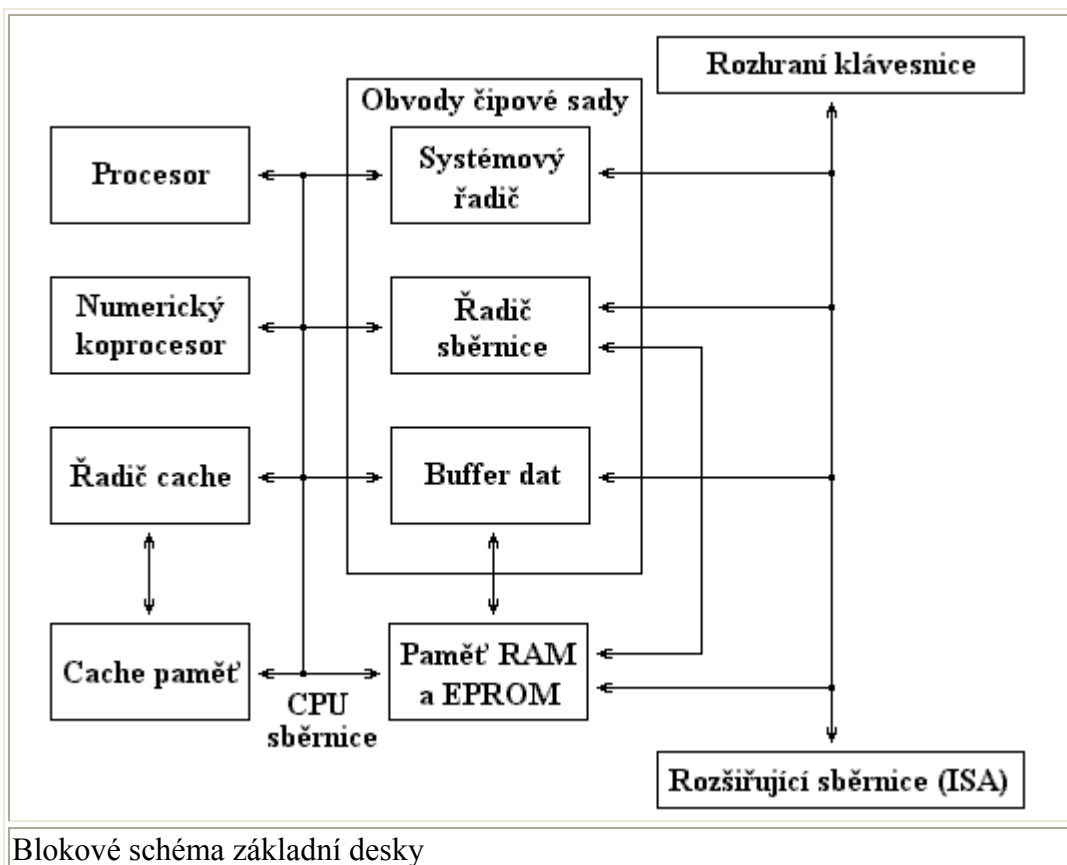
Počítačové systémy, periférie, operační systémy



- CMOS paměť
- Hodiny reálného času
- Akumulátor zálohující CMOS paměť

Vzhledem k tomu, že u novějších procesorů (80486 a vyšší) je již numerický koprocessor integrován přímo na čipu procesoru, není nutné, aby základní deska obsahovala patici pro jeho zapojení. Základní deska dále může obsahovat:

- Vstup / výstupní porty (I/O - Ports)
- Řadič pružných disků
- Rozhraní pevných disků
- Videokartu (videoadaptér)



Blokové schéma základní desky

Zařízení jako jsou procesor, numerický koprocessor, řadič cache paměti, paměti a obvody čipové sady jsou společně propojeny pomocí tzv. **systémové sběrnice (CPU bus)**, která umožňuje jejich rychlou vzájemnou komunikaci. Čipová sada je tvořena obvody s následující funkcí:

- **systémový řadič:** obvod, který řídí společnou činnost jednotlivých obvodů základní desky a realizuje následující funkce:
 - generuje hodinové signály
 - vytváří adresy pro paměti RAM
 - generuje řídicí signály pro paměťový subsystém
 - zabezpečuje RESET systému po připojení elektrického napájení nebo stisku tlačítka RESET
- **řadič sběrnice:** zabezpečuje komunikaci mezi systémovou sběrnicí a rozšiřující sběrnicí, dále obsahuje rozhraní reproduktoru a rozhraní paměti EPROM
- **buffer dat:** obvod, který slouží k zachycování dat a jejich přepínání mezi jednotlivými datovými sběrnicemi osobního počítače

8.2 Procesor (mikroprocesor)



- Procesor je integrovaný obvod zajišťující funkce CPU
- Tvoří "srdce" a "mozek" celého počítače
- Do značné míry ovlivňuje výkon celého počítače (čím rychlejší procesor, tím rychlejší počítač)
- Bývá umístěn na základní desce počítače
- Obsahuje rychlá paměťová místa malé kapacity nazývané **registry**
- Základní parametry procesoru:

Parametr	Popis	Jednotka	Rozsah
Rychlost	Počet operací provedených za jednu sekundu	Hertz [HZ]	4,77MHz – 1,5 GHz
Efektivita mikrokódu	Efektivita, se kterou jsou napsány jednotlivé mikroprogramy provádějící jednotlivé instrukce procesoru. Je to počet kroků potřebných pro provedení jedné instrukce (např.: vynásobení dvou čísel)		
Numerický koprocessor	Přítomnost (nepřítomnost) speciální jednotky pro přímé provádění výpočtů v pohyblivé desetinné čárce.		
Počet instrukčních kanálů	Udává maximální počet instrukcí proveditelných v jednom taktu procesoru	Číslo	1 - 4
Šířka slova	Maximální počet bitů, které je možné zpracovat během iediné	bit	16 - 32

Počítačové systémy, periférie, operační systémy

	operace		
Šířka přenosu dat	Maximální počet bitů, které je možné během jediné operace přenést z (do) čipu	bit	8 - 64
Interní cache paměť	Kapacita rychlé interní cache paměti integrované přímo na čipu procesoru	Byte	0 - 512 KB
Velikost adresovatelné paměti	Velikost paměti, kterou je procesor schopen adresovat (používat)	Byte	1 - 4096 MB

Procesory INTEL

Osmdesátá léta tohoto století zaznamenala zřejmě nejvýraznější pokrok v dosavadním vývoji výpočetní techniky. Velkým dílem k tomu přispěl i vznik a rozvoj procesorů firmy INTEL a osobních počítačů firmy IBM.

První procesor vznikl v roce 1969 v americké firmě Intel (Integrated Electronics), která tehdy vyvíjela obvody pro elektronické kalkulátory japonské firmy Busicom. Protože šlo o to, aby návrh byl cenově efektivní, zvítězila myšlenka sestavit univerzální procesor, kterým by se redukovala složitost japonského návrhu. Tehdy vznikl 4bitový jednočipový procesor, který později dostal označení 4004.



V roce 1974 firma Intel představila již 8bitový procesor s označením **8080**, který se stal základem prvních osmibitových mikropočítačů (jeho ekvivalent byl použit v našem počítači IQ 151). Jako inovovaný typ tohoto procesoru byl nakonec na trh uveden procesor **Intel 8085**, který však nezaznamenal větší komerční úspěch.

Firma Intel začíná dále pracovat na poli modernějších 16bitových procesorů a v roce 1977 dokončuje vývoj svého prvního 16bitového procesoru **Intel 8086**. Tento procesor je plně kompatibilní s svým předchůdcem 8080. Bylo tedy možné používat velké množství programů původně určených pro mikropočítače osazené procesorem 8080 i na počítačích s novým procesorem 8086. Krátce po vyrobení procesoru 8086 přichází firma Intel na trh s procesorem označeným 8088, který je z pohledu uživatele plně slučitelný s 8086, ale vykazuje menší výkon. Procesor 8088 využila firma **IBM** (International Business Machines) pro svůj nový mikropočítač pojmenovaný **IBM PC** (Personal Computer, který byl na trh uveden v roce 1981. IBM PC má 128 kB (256 kB) operační paměti, černobílý monitor a dvě mechaniky pružných disků. O dva roky později je tento typ počítače rozšířen o pevný disk a jeho operační paměť je zvýšena na 640 kB. Takto vzniklý nový počítač je prodáván pod označením **IBM PC/XT** (Extended Technology).

Dalšími následníky procesorů 8086 a 8088 byly **80186** a **80188**. Tyto procesory byly plně kompatibilní se svými předchůdci. Měly několik drobných vylepšení své architektury, díky kterým vykazovaly o něco vyšší výkon. Procesory 80186 a 80188 nezaznamenaly však žádného většího rozšíření.

Počítačové systémy, periférie, operační systémy

V roce 1982 začíná firma Intel vyrábět nový procesor 80286. Procesor 80286 je plně kompatibilní s předešlými procesory. Mezi jeho hlavní přínosy patří podpora pro paralelní zpracování více programů. V roce 1984 se začínají prodávat první počítače vybudované na tomto procesoru. Tyto počítače nesou označení **IBM PC/AT** (Advanced Technology). Tyto typy počítačů jsou již standardně vybavovány pevným diskem a minimální kapacitou paměti 1 MB. Jako další procesor firmy Intel je vyroben v roce 1985 32bitový procesor s označením **Intel 80386**. Jedná se opět o procesor plně kompatibilní s předcházejícími procesory, poskytuje vyšší výkon a větší programátorské možnosti.

Zmodernizováním tohoto procesoru a integrací jednotky pro výpočty v pohyblivé desetinné čárce přímo na čip procesoru vzniká nový typ procesoru označený jako Intel 80486. Tento procesor je dále následován výkonnějším procesorem **Intel Pentium** a posléze procesorem **Intel Pentium Pro, Pentium II, Pentium III a nyní Pentium IV**.

Poznámka: firma Intel v současnosti ovládá se svými procesory asi 80% trhu veškerých procesorů

8.3 Paměti



Paměť počítače je zařízení, které slouží k ukládání programů a dat, s nimiž počítač pracuje. Paměti lze rozdělit do tří základních skupin:

- **registry:** paměťová místa na čipu procesoru, která jsou používána pro krátkodobé uchování právě zpracovávaných informací
- **vnitřní (interní, operační) paměti:** paměti osazené většinou na základní desce. Bývají realizovány pomocí polovodičových součástek. Jsou do nich zaváděny právě spouštěné programy (nebo alespoň jejich části) a data, se kterými pracují.
- **vnější (externí) paměti:** paměti realizované většinou za pomoci zařízení používajících výměnná média v podobě disků či magnetofonových pásek. Záznam do externích pamětí se provádí většinou na magnetickém nebo optickém principu. Slouží pro dlouhodobé uchování informací a zálohování dat.

Základní parametry pamětí jsou:

- **kapacita:** množství informací, které je možné do paměti uložit
- **přístupová doba:** doba, kterou je nutné čekat od zadání požadavku, než paměť zpřístupní požadovanou informaci
- **přenosová rychlost:** množství dat, které lze z paměti přečíst (do ní zapsat) za jednotku času
- **statičnost / dynamičnost:**
 - **statické paměti:** uchovávají informaci po celou dobu, kdy je paměť připojena ke zdroji elektrického napětí
 - **dynamické paměti:** zapsanou informaci mají tendenci ztrácet i v době, kdy jsou připojeny k napájení. Informace v takových pamětech je nutné tedy neustále periodicky oživovat, aby nedošlo k jejich ztrátě.

Počítačové systémy, periférie, operační systémy

- **destruktivnost při čtení:**
 - **destruktivní při čtení:** přečtení informace z paměti vede ke ztrátě této informace. Přečtená informace musí být následně po přečtení opět do paměti zapsána.
 - **nedestruktivní při čtení:** přečtení informace žádným negativním způsobem tuto informaci neovlivní.
- **energetická závislost:**
 - **energeticky závislé:** paměti, které uložené informace po odpojení od zdroje napájení ztrácejí
 - **energeticky nezávislé:** paměti, které uchovávají informace i po dobu, kdy nejsou připojeny ke zdroji elektrického napájení.
- **přístup**
 - **sekvenční:** před zpřístupněním informace z paměti je nutné přečíst všechny předcházející informace
 - **přímý:** je možné zpřístupnit přímo požadovanou informaci
- **spolehlivost:** střední doba mezi dvěma poruchami paměti
- **cena za bit:** cena, kterou je nutno zaplatit za jeden bit paměti

Následující tabulka ukazuje výše popsané tři typy paměti a jejich parametry.

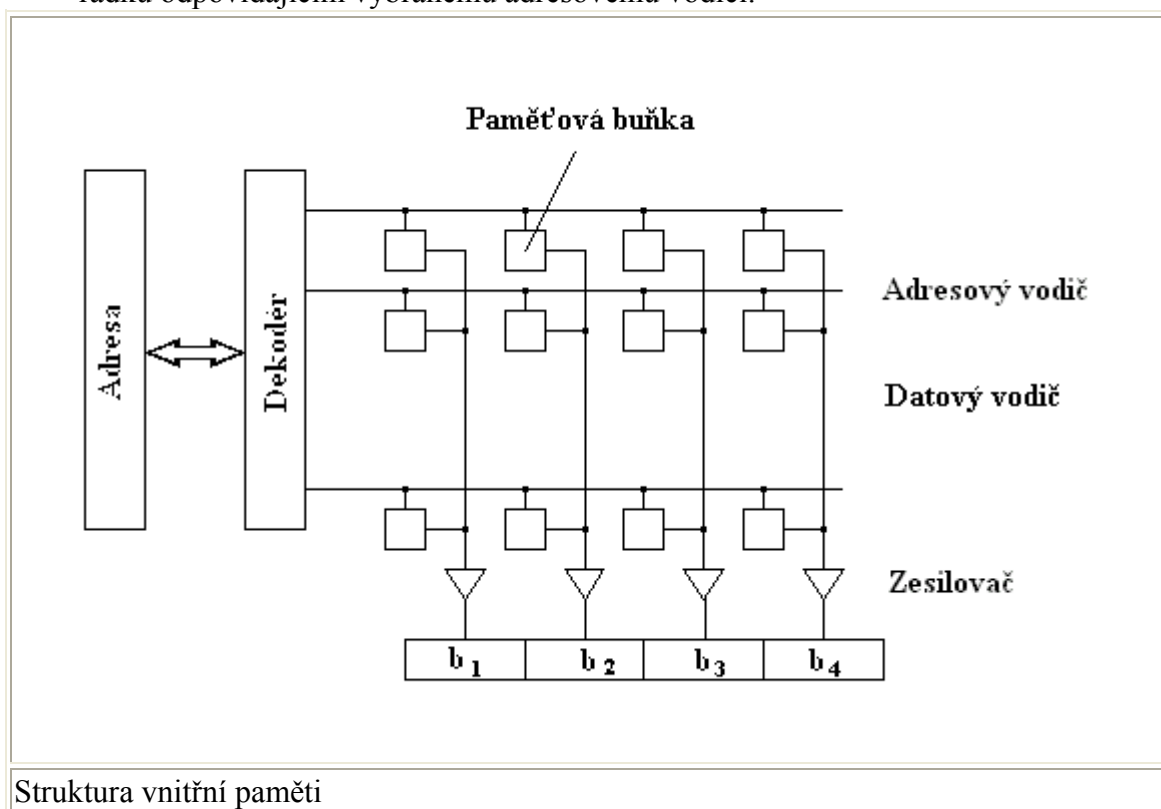
	registry	vnitřní paměti	vnější paměti
kapacita	velmi malá (jednotky bytů)	vyšší (řádově 100 kB - 100MB)	vysoká (řádově 10 MB - 10 GB)
přístupová doba	velmi nízká (velmi rychlá paměťová místa)	vyšší (řádově 10 ns)	vysoká (řádově 10 ms - 10 min)
přenosová rychlost	vzhledem k malé kapacitě se většinou neuvažuje	vysoká (řádově 1 - 10 MB/s)	nižší než u vnitřních pamětí (řádově 10 MB/min - 1 MB/s)
statičnost / dynamičnost	statické	statické i dynamické	statické
destruktivnost při čtení	nedestruktivní	destruktivní i nedestruktivní	nedestruktivní
energetická závislost	závislé	závislé	nezávislé
přístup	přímý	přímý	přímý i sekvenční
spolehlivost	velmi spolehlivé	spolehlivé	méně spolehlivé
cena za bit	vzhledem k nízké kapacitě vysoká	nižší než u registrů a vyšší než u vnějších pamětí	vzhledem k vysoké kapacitě nízká

Vnitřní paměti

Interní paměti jsou zapojeny jako matice paměťových buněk. Každá buňka má kapacitu jeden bit. Takováto buňka tedy může uchovávat pouze hodnotu logická jedna nebo logická 0. Při přístupu do paměti (čtení nebo zápis) je vždy udána adresa paměťového místa, se kterým se bude pracovat. Tato adresa je přivedena na vstup dekodéru. Dekodér pak podle zadané adresy vybere jeden z

Počítačové systémy, periférie, operační systémy

adresových vodičů a nastaví na něm hodnotu logická 1. Podle toho, jak jsou zapojeny jednotlivé paměťové buňky na příslušném řádku, který byl vybrán dekodérem, projde resp. neprojde hodnota logické jedničky na datové vodiče. Informace je dále na koncích datových vodičů zesílána zesilovačem. V případě, že hodnota logická jedna projde přes paměťovou buňku, obdržíme na výstupu hodnotu bitu 1. V opačném případě je na výstupu hodnota bitu 0. Zcela analogický je postup i při zápisu hodnoty do paměti. Opět je nejdříve nutné uvést adresu paměťového místa, do kterého se bude zapisovat. Dekodér vybere adresový vodič příslušný zadané adrese a nastaví na něj hodnotu logická 1. Dále se nastaví hodnoty bitů b_1 až b_4 na hodnoty, které se budou do paměti ukládat. Tyto hodnoty jsou potom uloženy do paměťových buněk na řádku odpovídajícím vybranému adresovému vodiči.



Struktura vnitřní paměti

Vnitřní paměti je možné rozdělit do následujících základních skupin:

- ROM
- PROM
- EPROM
- EEPROM
- Flash
- RAM
 - DRAM
 - SRAM

Paměti ROM (Read Only Memory)

Paměti ROM jsou paměti, které jsou určeny pouze pro čtení informací.

Informace jsou do těchto pamětí pevně zapsány při jejich výrobě a potom již

Počítačové systémy, periférie, operační systémy

není možné žádným způsobem jejich obsah změnit. Jedná se tedy o statickou, energeticky nezávislou paměť určenou pouze ke čtení.

Paměti PROM (Programable Read Only Memory)

Paměť PROM neobsahuje po vyrobení žádnou pevnou informaci a je až na uživateli, aby provedl příslušný zápis informace. Tento zápis je možné provést pouze jednou a poté již paměť slouží stejně jako paměť ROM. Paměti PROM představují statické a energeticky nezávislé paměti.

Paměti EPROM (Eraseable Programable Read Only Memory)

Paměť EPROM je statická energeticky nezávislá paměť, do které může uživatel provést zápis. Zapsané informace je možné vymazat působením ultrafialového záření. Tyto paměti jsou realizovány pomocí speciálních unipolárních tranzistorů, které jsou schopny na svém přechodu udržet elektrický náboj po dobu až několika let. Tento náboj lze vymazat právě působením UV záření. Paměti EPROM jsou charakteristické malým okénkem v pouzdře integrovaného obvodu obsahujícího tuto paměť. Pod okénkem je umístěn vlastní paměťový čip a to je místo, na které směřuje při vymazávání zdroj UV záření. Při práci bývá tento otvor většinou přelepen ochranným štítkem, aby nedocházelo ke ztrátám informace vlivem UV záření v ovzduší. Zapojení jedné buňky paměti EPROM je podobné jako u paměti EEPROM (viz dále).

Paměti EEPROM (Electrically EEPROM)

Tento typ paměti má podobné chování jako paměti EPROM, tj. jedná se o statickou energeticky nezávislou paměť, kterou je možné naprogramovat a později z ní informace vymazat. Výhodou oproti EPROM pamětem je, že vymazání se provádí elektricky a nikoliv pomocí UV záření, čímž odpadá nepohodlná manipulace s paměti při jejím mazání.

8.3.1 Paměti Flash

Flash paměti jsou obdobou paměti EEPROM. Jedná se o paměti, které je možné naprogramovat a které jsou statické a energeticky nezávislé. Vymazání se provádí elektrickou cestou, jejich přeprogramování je možné provést přímo v počítači. Paměť typu Flash tedy není nutné před vymazáním (naprogramováním) z počítače vyjmout a umístit ji do speciálního programovacího zřízení.

8.3.2 Paměti RAM

Paměti RAM jsou určeny pro zápis i pro čtení dat. Jedná se o paměti, které jsou energeticky závislé. Podle toho, zda jsou dynamické nebo statické, jsou dále rozdělovány na:

- **DRAM** - Dynamické RAM
- **SRAM** - Statické RAM

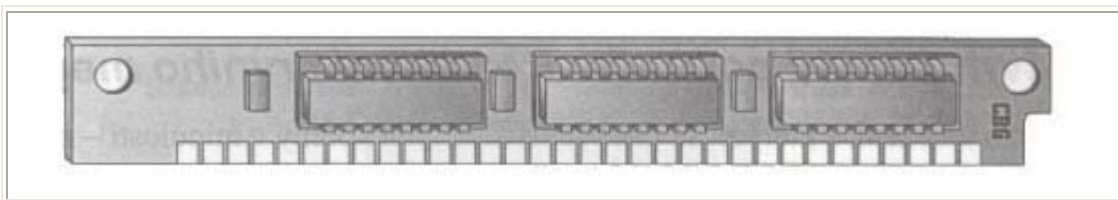
8.3.3 Organizace pamětí v PC

Paměti jsou integrovány na miniaturních deskách plošného spoje označovaných jako **SIMM**(Single Inline Memory Module), které jsou potom jako celek osazovány do odpovídajících konektorů na základní desce (popř.

Počítačové systémy, periférie, operační systémy

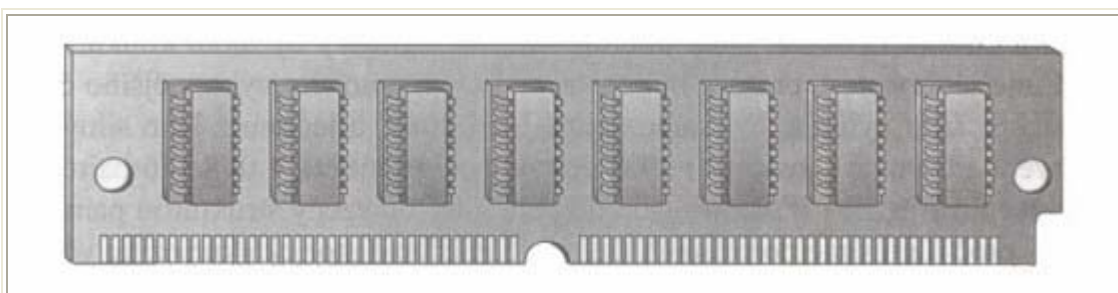
jiných zařízení využívajících ke své činnosti paměť). Tyto moduly jsou vyráběny ve dvou variantách:

- **30-pin SIMM:** používaný u většiny počítačů s procesory 80286, 80386SX, 80386 a některých 80486. Mají 30 vývodů a šířku přenosu dat 8 bitů (bezparitní SIMM) nebo 9 bitů (paritní SIMM). Jsou vyráběny s kapacitami 256 kB, 1 MB a 4 MB.



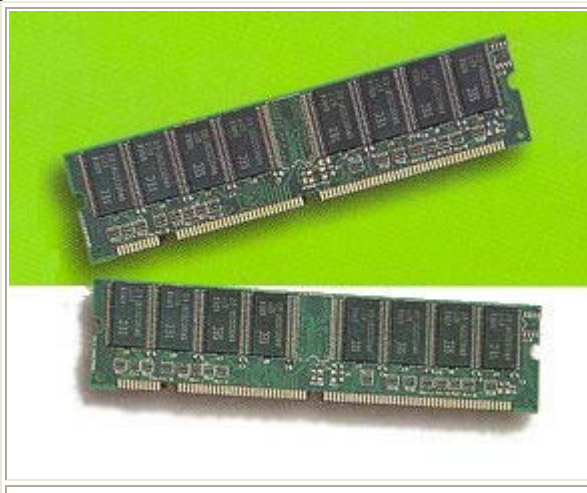
Paměťový modul SIMM (30-pin)

- **72-pin SIMM (PS/2 SIMM):** používaný u počítačů s procesory 80486 a vyššími. PS/2 SIMMy mají 72 vývodů, šířku přenosu dat 32 bitů (bezparitní SIMM) nebo 36 bitů (paritní SIMM - pro každý byte jeden paritní bit). Jsou vyráběny s kapacitami 4 MB, 8 MB, 16 MB, 32 MB.



Paměťový modul SIMM (72-pin)

Posledním dnes vyráběným typem paměťových modulů jsou paměťové moduly typu **DIMM** (Dual Inline Memory Module). Jedná se podobně jako v případě modulů SIMM o malou desku plošného spoje s osazenými paměťovými obvody. Moduly DIMM mají 168 vývodů a šířku přenosu 64 bitů. Vyrábějí se s kapacitami 16 MB, 32 MB, 64 MB, 128 MB a 256 MB.



Paměťové moduly DIMM

Stav čekání (Wait State)

Paměť musí být schopna reagovat na požadavky procesoru během dvou taktů hodin (takt hodin je převrácená hodnota frekvence procesoru). Kromě přístupové doby mají paměti DRAM používané jako operační paměť ještě tzv. **nabíjecí dobu**.

Příklad: Mějme počítač s procesorem o frekvenci 66 MHz => 1 takt hodin je 15 ns => 2 takty hodin jsou 30 ns => jsou potřeba paměti DRAM s přístupovou dobou 30 ns. Takto rychlé paměti DRAM však neexistují. DRAM paměti jsou vyráběny s přístupovou dobou 60 - 70 ns. Proto je nutné v tomto případě při každém přístupu do paměti přidat dva čekací takty (celkem 4 takty = 60 ns), kdy procesor nebude dělat nic, ale bude čekat na pomalejší operační paměť. Pokud tuto úvahu provedeme opačným směrem, tj. 60 ns = frekvenci 16,6 MHz, zjistíme, že tímto řešením sice počítač bude pracovat, ale jeho výkon je při přístupech do paměti degradován na procesor s frekvencí 16,6 MHz.

Řešení:

- Rychlejší paměti DRAM? Neexistují.
- Použít místo paměti DRAM paměti SRAM? Příliš drahé.

Proto v současných moderních počítačích se používají tzv. cache paměti.

8.3.4 Cache paměti

Cache paměť je rychlá vyrovnávací paměť mezi rychlým zařízením (např. procesor) a pomalejším zařízením (např. operační paměť). V dnešních počítačích se běžně používají dva druhy cache paměti:

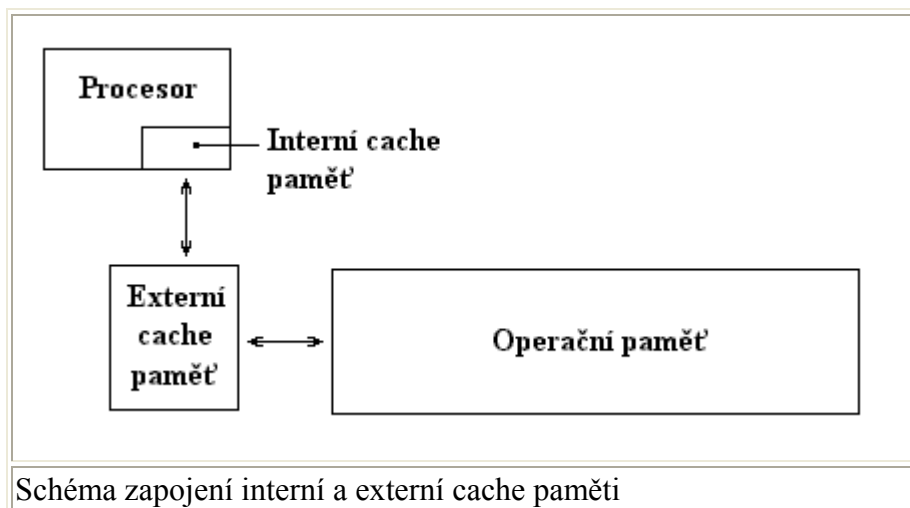
- **externí (sekundární, L2) cache:**

Externí cache paměť je paměť, která je umístěna mezi pomalejší operační paměť a rychlým procesorem. Tato paměť je vyrobena jako rychlá paměť SRAM a slouží jako vyrovnávací paměť u počítačů s výkonným procesorem, které by byly bez ní operační paměti velmi zpomalovány. První externí cache paměti se objevují u počítačů s procesorem 80386. Jejich kapacita je 32 kB popř. 64 kB. S výkonnějšími procesory se postupně zvyšuje i kapacita externích cache pamětí na 128 kB, 256 kB, 512 kB. Externí cache paměť je osazena na základní desce počítače (výjimku tvoří procesory Pentium Pro a Pentium II, které mají externí cache paměť integrovanou v pouzdře procesoru). Její činnost je řízena řadičem cache paměti.

- **interní (primární, L1) cache:**

Interní cache paměť je paměť, která slouží k vyrovnání rychlosti velmi výkonných procesorů a pomalejších pamětí. Tento typ cache paměti je integrován přímo na čipu procesoru a je také realizován pomocí paměti SRAM. Interní cache paměť se objevuje poprvé u procesoru 80486 s kapacitou 8 kB. Takovýto procesor musí mít v sobě integrován také řadič interní cache paměti pro řízení její činnosti.





Práce cache paměti vychází ze skutečnosti, že program má tendenci se při své práci určitou dobu zdržovat na určitém místě paměti, a to jak při zpracování instrukcí, tak při načítání (zapisování) dat z (do) paměti. Je-li požadována nějaká informace z paměti, je nejdříve hledána v cache paměti (interní, pokud existuje, a následně v externí). Pokud požadovaná informace není přítomna v žádné z cache paměti, je zavedena přímo z operační paměti. Kromě momentálně požadované informace se však do cache paměti zavede celý blok paměti, takže je velká pravděpodobnost, že následně požadované informace již budou v cache paměti přítomny. Pokud dojde k zaplnění cache paměti a je potřeba zavést další blok, je nutné, aby některý z bloků cache paměť opustil. Nejčastěji se k tomuto používá **LRU** (Least Recently Used) algoritmu, tj. algoritmu, který vyřadí nejdéle nepoužívaný blok.

Cache paměti bývají organizovány jako tzv. **asociativní paměti**. Asociativní paměti jsou tvořeny tabulkou (tabulkami), která obsahuje vždy sloupec, v němž jsou umístěny tzv. **tagy** (klíče), podle kterých se v asociativní paměti vyhledává. Dále jsou v tabulce umístěna data, která paměť uchovává, a popř. další informace nutné k zajištění správné funkce paměti. Např.:

- informace o platnosti (neplatnosti) uložených dat
- informace pro realizaci LRU algoritmu
- informace protokolu **MESI** (Modified Exclusive Shared Invalid), který zajišťuje synchronizaci dat v cache pamětech v případě, že cache paměti je v počítači více (u interních cache paměti v okamžiku, kdy počítač obsahuje více procesorů).

Poznámka: Kromě externích a interních cache paměti je možné se setkat i se specializovanými cache paměťmi umístěnými mezi operační paměť a některé pomalejší zařízení (pevný disk, apod.).

8.3.5 CMOS paměť

Paměť s malou kapacitou sloužící k uchování údajů o nastavení počítače a jeho hardwarové konfiguraci. Tato paměť je energeticky závislá, a proto je nutné ji zálohovat pomocí akumulátoru umístěného většinou na základní desce, aby nedošlo ke ztrátě údajů v ní uložených.

V CMOS paměti bývají většinou uloženy:

Počítačové systémy, periférie, operační systémy

- informace o typech a kapacitách jednotek pružných disků
- informace o typech, kapacitách a parametrech pevných disků
- typ používané video karty
- kapacita operační paměti
- nastavení parametrů cache paměti
- pořadí jednotek pro zavádění operačního systému
- povolení / zakázání různých funkcí základní desky:
 - využívání interní a externí cache paměti
 - antivirová ochrana systémových oblastí disků
 - prohození pořadí jednotek pružných disků
 - stínování určitých částí paměti (zavádění programového vybavení z pomalejší ROM paměti do rychlejší paměti RAM)
 - činnosti rozhraní pružných disků, pevných disků
 - činnosti vstup / výstupních portů
- nastavení rychlosti repetice klávesnice
- nastavení parametrů přenosu informací z pevných disků
- nastavení parametrů pro režim s úsporou elektrické energie
- nastavení přiřazení IRQ úrovní
- nastavení hesla k programu SETUP, popř. k celému počítači

Tyto parametry se nastavují většinou pomocí programu zvaného SETUP. SETUP bývá uložen nejčastěji v permanentní paměti počítače, která bývá realizována jako EPROM (u starších počítačů) nebo jako Flash (u novějších počítačů). Špatné nastavení výše zmíněných parametrů může způsobit výrazné snížení výkonu celého počítače, až nefunkčnost některých jeho částí, popř. nefunkčnost celého počítače.

Vzhledem k tomu, že tyto informace jsou pro počítač velmi důležité a jejich špatné hodnoty mohou být příčinou výše zmíněných problémů, není žádoucí, aby k nim měl přístup kdokoli. Proto při přístupu do programu SETUP a tím i ke změnám parametrů v CMOS paměti je možné požadovat heslo.

8.4 Čipové sady

Čipová sada (Chip set) je sada integrovaných obvodů speciálně zkonstruovaná pro práci s konkrétním typem procesoru. Obvody čipové sady realizují funkce, jako např.

- řízení činnosti paměti DRAM i SRAM
- řízení činnosti jednotlivých sběrnic
- komunikace mezi sběrnicemi

8.5 Sběrnic (bus)

Pod pojmem sběrnice obecně rozumíme soustavu vodičů, která umožňuje přenos signálů mezi jednotlivými částmi počítače. Pomocí těchto vodičů mezi sebou jednotlivé části počítače komunikují a přenášejí data.

Zařízení jako jsou procesor, koprocesor, cache paměť, operační paměť, řadič cache paměti a operační paměti a některá další zařízení jsou propojena tzv. **systémovou sběrnicí (CPU bus)**.

Počítačové systémy, periférie, operační systémy

Osobní počítače musí být navrženy tak, aby bylo možné jejich snadné rozšiřování o další zařízení (zvukové karty, síťové karty, řadiče disků apod.). Takovéto rozšiřování je velmi často uskutečňováno pomocí tzv. **rozšiřující sběrnice** počítače (častěji označované pouze jako sběrnice), na kterou se jednotlivá zřízení zapojují. Tato rozšiřující sběrnice a zapojovaná zařízení musí tedy splňovat určitá pravidla. Takže ve výpočetní technice je pojem sběrnice také chápán jako standard, dohoda o tom, jak vyrobit zařízení (rozšiřující karty), která mohou pracovat ve standardním počítači.

Podle způsobu práce a zapojení rozlišujeme několik základních typů sběrnic:

- **synchronní sběrnice:** sběrnice pracující synchronně s procesorem počítače. Platnost údajů na sběrnici jednoznačně určuje hodinový signál. Tímto způsobem dnes pracuje převážná většina všech sběrnic.
- **pseudosynchronní sběrnice:** dovoluje zpozdít přenos údajů o určitý počet hodinových period.
- **multimaster sběrnice:** dovoluje tzv. **busmastering**, jedná se o sběrnici, která může být řízena několika zařízeními, nejen procesorem.
- **lokální sběrnice:** spočívá ve vytvoření technické podpory toho, že se náročné operace s daty realizují rychlou systémovou sběrnici. Tato systémová sběrnice se prodlouží a umožní se tak přístup na ni i ze zásuvných modulů dalších zařízení. O rozvoj lokálních sběrnic se nejvýrazněji zasloužili výrobci videokaret, pro něž byly dosavadní sběrnice pomalé. Nevýhodou lokálních sběrnic je o něco vyšší cena samotné základní desky s lokální sběrnici a také zařízení pro ni určených.



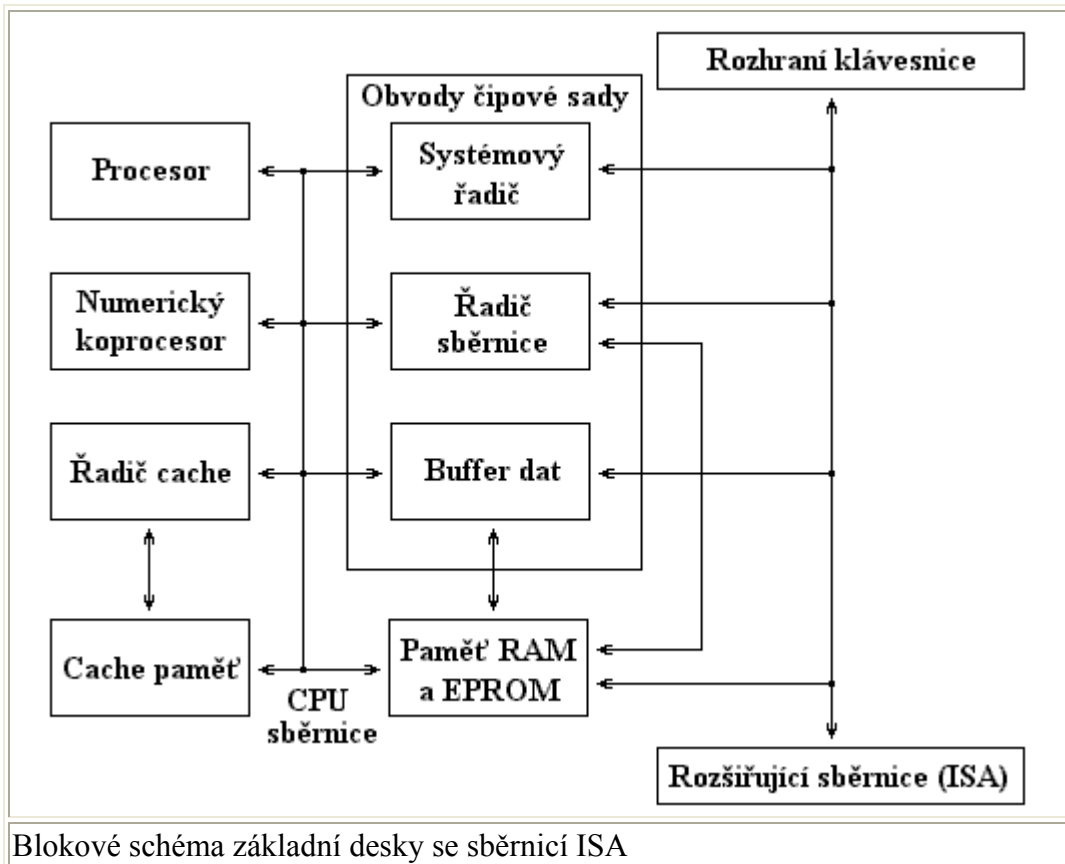
Mezi základní parametry každé sběrnice patří:

Parametr	Význam	Jednotka
Šířka přenosu	Počet bitů, které lze zároveň po sběrnici přenést	bit
Frekvence	Maximální frekvence, se kterou může sběrnice pracovat	Hz
Rychlost (propustnost)	Počet bytes přenesených za jednotku času	B/s

Sběrnice pro PC

8.5.4 Sběrnice ISA (AT bus)

S příchodem procesoru 80286 se používá typ sběrnice označovaný jako **ISA** (Industry Standard Architecture). Tento typ rozšiřující sběrnice je vyroben s 16bitovou datovou sběrnici a 24bitovou adresovou sběrnici.



8.5.5 Sběrnice MCA (MicroChannel)

Sběrnice **MCA** (MicroChannel Architecture) je novým typem sběrnice, který byl vyvinutý pro novou řadu počítačů firmy IBM s označením IBM PS/2. Hlavním cílem IBM bylo zrychlit přenos dat uvnitř počítače a snížit hladinu šumu na sběrnici.

Obrovskou nevýhodou a patrně i důvodem, proč se sběrnice MCA nerozšířila, je její nekompatibilita s ISA a to, že počítače PS/2 neměly osazenu pro zpětnou kompatibilitu i sběrnici ISA.

8.5.6 Sběrnice EISA

Sběrnice **EISA** (Extended Industry Standard Architecture) byla vyrobena 9 firmami (AST Research, Compaq, Epson, NEC, Olivetti, Tandy, Wyse a Zenith) jako odpověď na sběrnici MCA. Záměrem bylo poskytnout sběrnici s vyšším výkonem, ale takovou, která by byla kompatibilní se sběrnicí ISA.

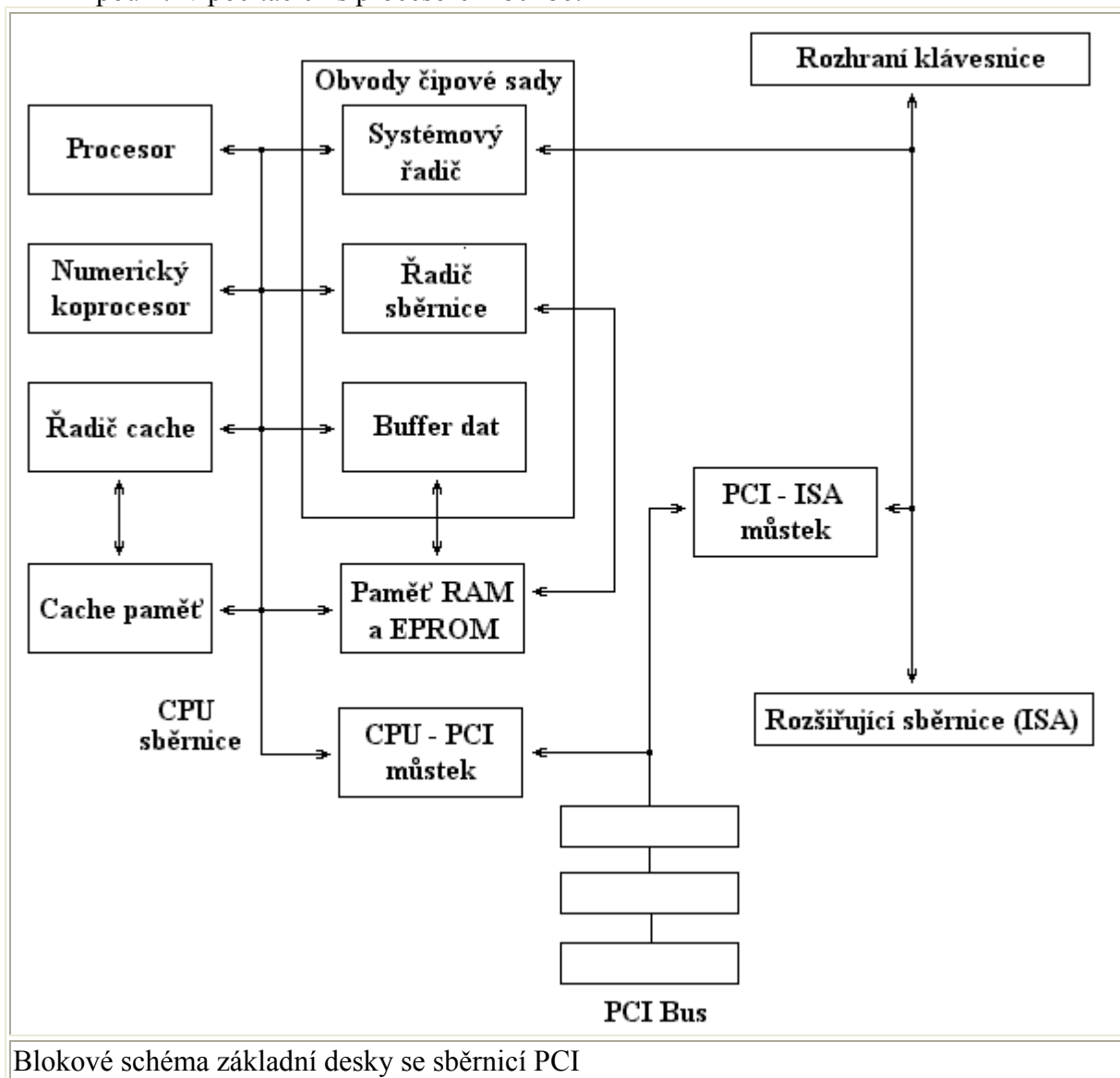
8.5.7 Sběrnice VL bus

Sběrnice **VL bus** (VESA Local Bus) byla navržena v roce 1992 konsorciem **VESA** (Video Electronic Standards Association) a jedná se o klasickou lokální sběrnici. Šířka přenosu dat i adresy je 32 bitů. VL bus podporuje maximálně 3 přídavné sloty. Čím vyšší je počet karet zasunutých na sběrnici VL bus, tím nižší je maximální frekvence, se kterou může sběrnice pracovat. Teoretická mez VL busu je 50 MHz. Prakticky je možné, aby pracovala s frekvencí 33 MHz při třech osazených přídavných kartách.

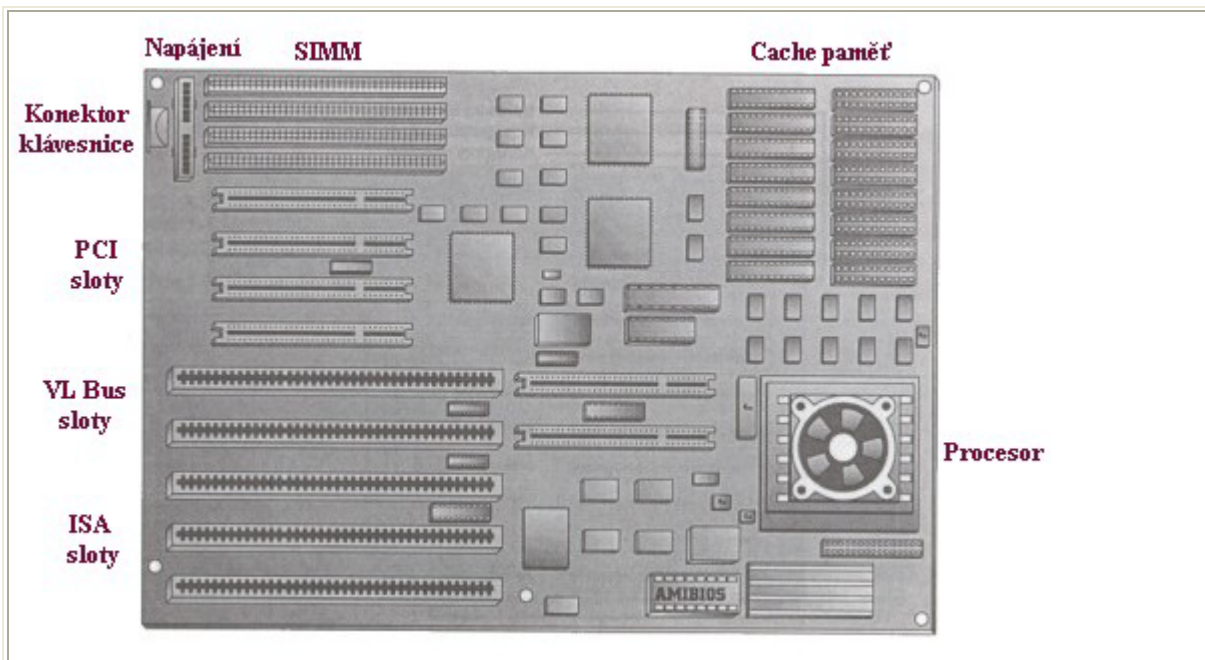
8.5.8 Sběrnice PCI

Sběrnice **PCI** (Peripheral Component Interconnect) je zatím posledním typem sběrnice pro počítače PC. Jedná se o rychlou sběrnici vyrobenou firmou Intel pro počítače s procesory Pentium.

Sběrnice PCI je první sběrnici s šířkou přenosu 64 bitů a využívá tak plně 64bitové datové sběrnice Pentia. Dovoluje však i přenos o šířce 32 bitů pro použití v počítačích s procesorem 80486.



Blokové schéma základní desky se sběrnici PCI

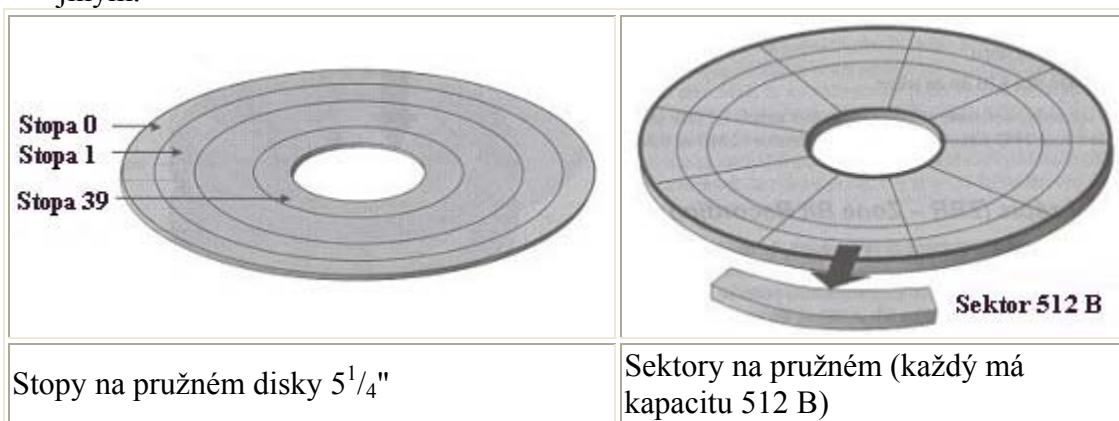


Základní deska se sběrnicí PCI a VL-Bus

8.6 Pružné disky (floppy disky, diskety)

Pružné disky patří mezi přenosná média pro uchování dat. Pružný disk je tvořen plastovým kotoučem, na jehož povrchu je vrstva oxidu železa. Celý kotouč je potom uzavřen v obdélníkovém pouzdře, vystlaném hebkým materiálem, které jej chrání před nečistotou a mechanickým poškozením a ve kterém se kotouč při práci otáčí. V tomto obalu je vyříznutý tzv. **čtecí otvor**, kterým přistupuje čtecí a zapisovací hlava k médiu.

Záznam dat na médium je prováděn magneticky. Jednotlivá data jsou zapisována do soustředných kružnic, tzv. **stop** (**track**), na obě strany diskety. Každá stopa je rozdělena ještě na tzv. **sektory** (**sector**), jež tvoří nejmenší úsek média, na který je možné zapisovat. To znamená, že při zápisu na pružný disk jsou data zapisována po sektorech a poslední sektor již nemusí být plně zaplněn. Nová data mohou být zapisována opět od začátku dalšího sektoru. Z toho vyplývá, že ne všechny sektory jsou v případě plně nahané diskety zcela zaplněny. Vlastní zápis na pružný disk bývá prováděn s kódováním MFM, zatím se neobjevily (a asi ani neobjeví) pružné disky se kódováním RLL popř. jiným.

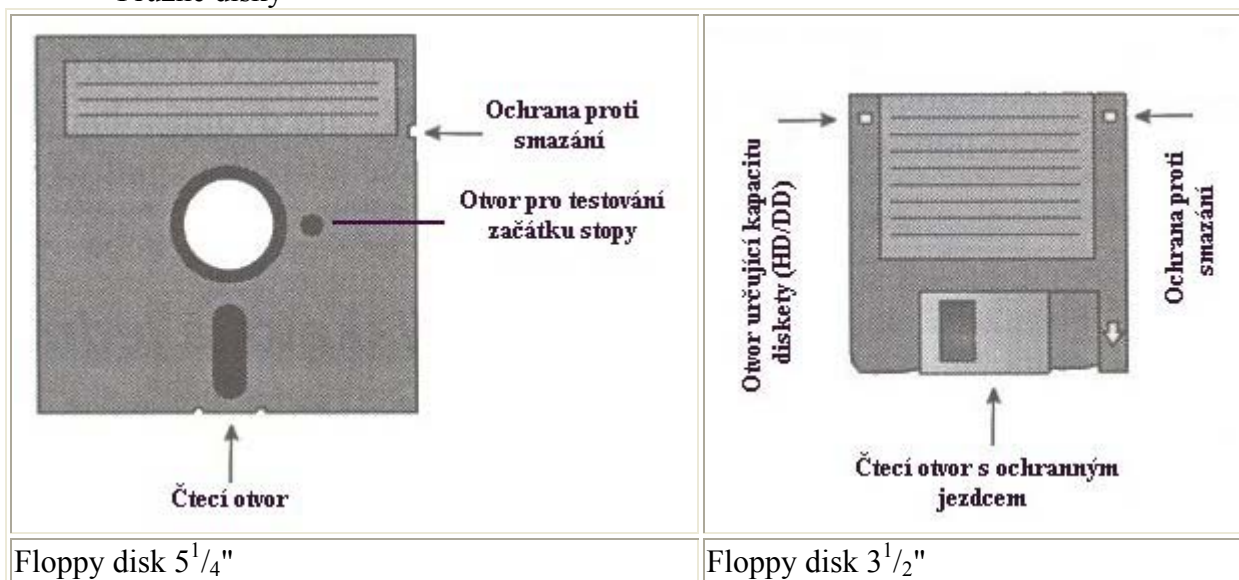


Počítačové systémy, periférie, operační systémy

Základními parametry disket jsou jejich velikost, hustota záznamu dat a z toho vyplývající kapacita:

Velikost	Hustota	Stopy	Sektory	Strany	Kapacita sektoru	Kapacita diskety
5 ¹ / ₄ "	DD	0-39	1-9	0-1	512 B	360 kB
5 ¹ / ₄ "	HD	0-79	1-15	0-1	512 B	1,2 MB
3 ¹ / ₂ "	DD	0-79	1-9	0-1	512 B	720 kB
3 ¹ / ₂ "	HD	0-79	1-18	0-1	512 B	1,44 MB

Pružné disky



Zkratky DD a HD ve sloupci hustota značí po řadě Double Density a High Density, tj. disketu s dvojitou a vysokou hustotou záznamu. Pro vyjádření hustoty záznamu se také někdy používá jednotka **TPI** (**T**racks **P**er **I**nch), která udává počet stop na jeden palec. Diskety 5¹/₄" HD mají hustotu záznamu 96 TPI a u disket 3¹/₂" HD je hustota 135 TPI.

8.6.4 Mechaniky pružných disků

Mechaniky pružných disků jsou zařízení pro čtení a zapisování na pružné disky. Je možné je rozdělit podobně jako pružné disky podle velikosti (5¹/₄", 3¹/₂") a podle hustoty záznamu (DD, HD).

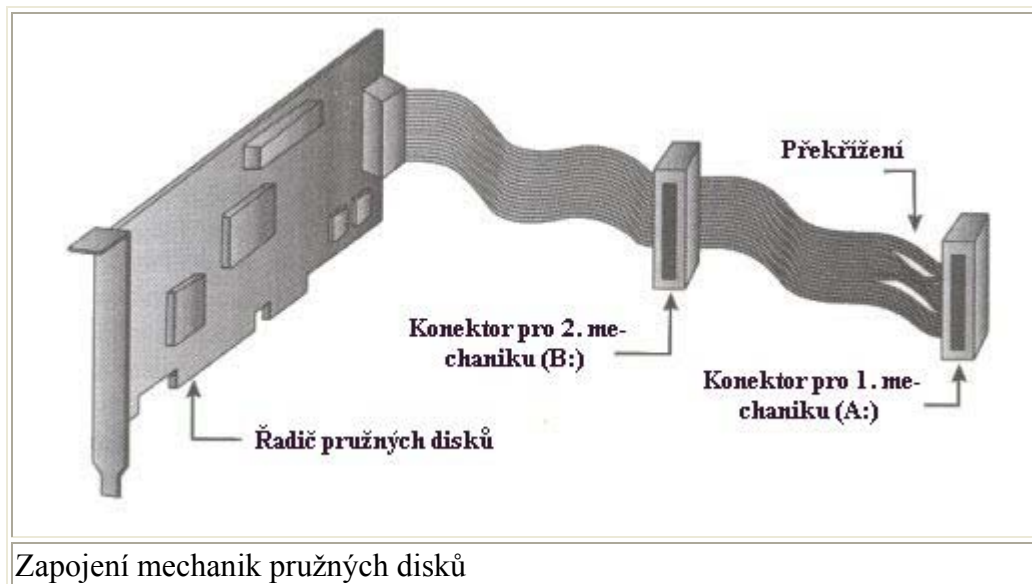
Standardní řadič podporuje připojení max. 2 mechanik pružných disků.

Připojení disketových mechanik k řadiči je provedeno pomocí plochého kabelu se 34 vodiči. Tento kabel má zpravidla 5 konektorů:

- 1 pro připojení k řadiči
- 2 pro připojení mechaniky 5¹/₄"
 - 1 pro případ zapojení jako první mechaniky (v MS-DOSu A:)
 - 1 pro případ zapojení jako druhé mechaniky (v MS-DOSu B:)
- 2 pro připojení mechaniky 3¹/₂" (analogicky jako u mechanik 5¹/₄").

Počítačové systémy, periférie, operační systémy

Propojení řadiče s druhou disketovou mechanikou je provedeno přímo (1:1), tj. kontakt 1 je na řadiči spojen s kontaktem 1 mechaniky, kontakt 2 s kontaktem 2 atd. Propojení první mechaniky již není (1:1), ale propojující kabel je překřížen. Podle tohoto překřížení je tedy rozlišeno, která mechanika je první a která je druhá.

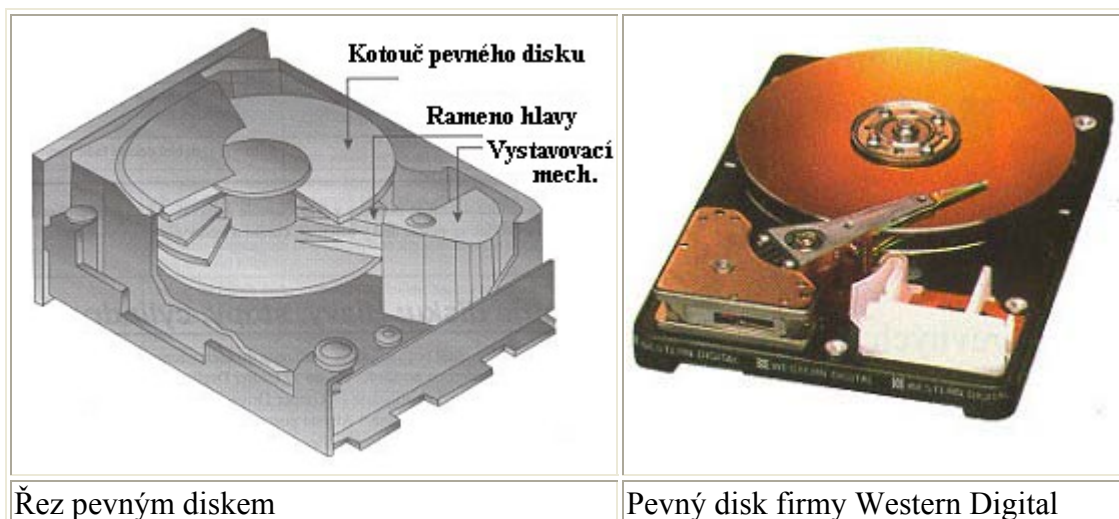


Vlastní čtení popř. zápis z pružného disku v mechanice probíhá ve třech krocích.

1. vystavení čtecích (zapisovacích) hlav na požadovanou stopu pomocí krokového motorku.
2. pootočení na příslušný sektor
3. zápis (čtení) sektoru

8.7 Pevné disky (Winchester disky, hard disky)

Pevné disky jsou média pro uchování dat s vysokou kapacitou záznamu (řádově stovky MB až desítky GB). V současnosti jsou pevné disky standardní součástí každého PC. Jedná se o pevně uzavřenou nepřenositelnou jednotku. Uvnitř této jednotky se nachází několik nad sebou umístěných rotujících kotoučů (disků). Tyto disky se otáčejí po celou dobu, kdy je pevný disk připojen ke zdroji elektrického napájení nezávisle na tom, zda se z něj čte (na něj zapisuje). Rychlost otáčení bývá 3600 až 7200 otáček za minutu. Díky tomuto otáčení se v okolí disků vytváří tenká vzduchová vrstva, na níž se pohybují čtecí/zapisovací hlavy. Vzdálenost hlav od disku je asi 0,3 až 0,6 mikronu



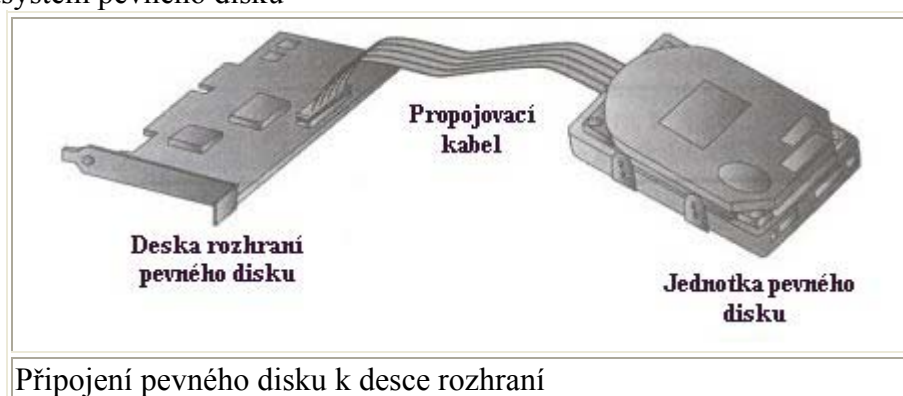
Řez pevným diskem

Pevný disk firmy Western Digital

Podsystém pevného disku se skládá z:

- diskových jednotek
- desky rozhraní pevných disků
- příslušných kabelů propojujících diskové jednotky s deskou rozhraní

Podsystém pevného disku



Připojení pevného disku k desce rozhraní

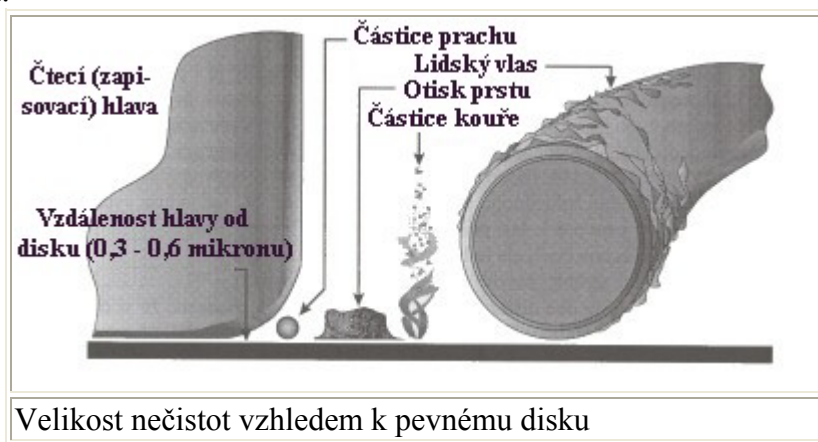
Základní parametry pevných disků jsou uvedeny v následující tabulce:

Parametr	Vysvětlení	Rozsah
Velikost	Průměr disků použitých ke konstrukci pevného disku	2"; 3 1/2"; 5 1/4"
Počet cylindrů	Počet stop na každém disku	300 - 3000
Počet hlav	Odpovídá počtu povrchů, na které se provádí záznam	2 - 256
Počet sektorů	Počet sektorů na každé stopě	8 - 64
Mechanismus vystavení hlav	Mechanismus, pomocí kterého se vystavují čtecí/zapisovací hlavy na patřičný cylindr. U starších typů pevných disků bývá realizován pomocí krokového motorku a u novějších disků pomocí elektromagnetu	Krokový motorek / elektromagnet
Přístupová doba	Doba, která je nutná k vystavení čtecích / zapisovacích hlav na požadovaný cylindr	8 - 65 ms

Počítačové systémy, periférie, operační systémy

Přenosová rychlost	Počet bytů, které je možné z disku přenést za 1 sekundu	700 - 5000 kB/s
Typ rozhraní	Určuje, jaký typ desky rozhraní musí být v počítači osazen, aby bylo možné tento pevný disk připojit	ST506, ESDI, IDE, EIDE, SCSI
Metoda kódování dat	Způsob, kterým jsou data při zápisu na disk kódována	MFM, RLL, ARLL, ERL
ZBR	Metoda, která dovoluje zapisovat na stopy, které jsou vzdálenější od středu pevného disku (jsou větší), vyšší počet sektorů	ANO / NE

Vzhledem k velmi vysoké hustotě záznamu je skutečně nutné, aby jednotka pevného disku byla pevně uzavřena, protože i velmi malá nečistota způsobí její zničení.

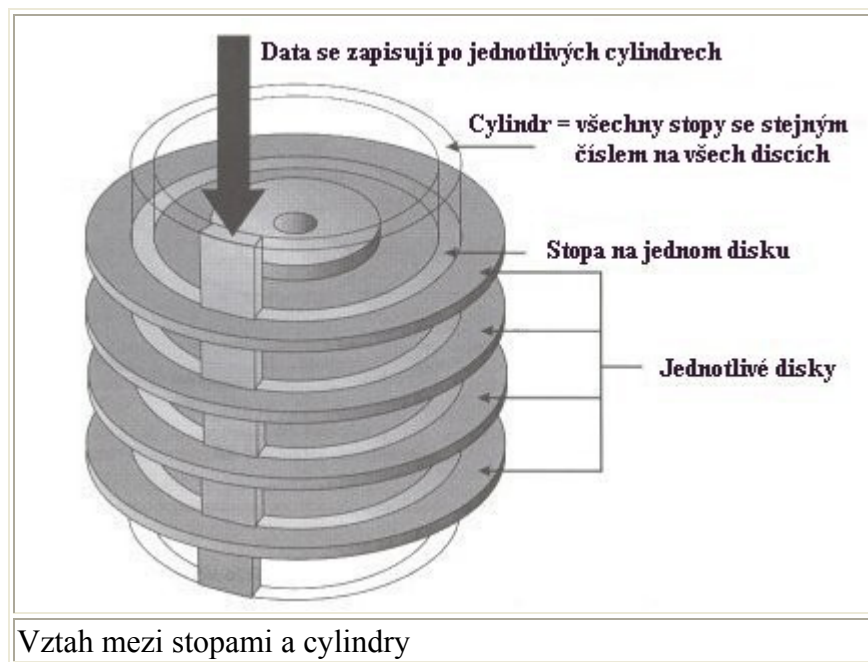


8.7.4 Geometrie pevných disků

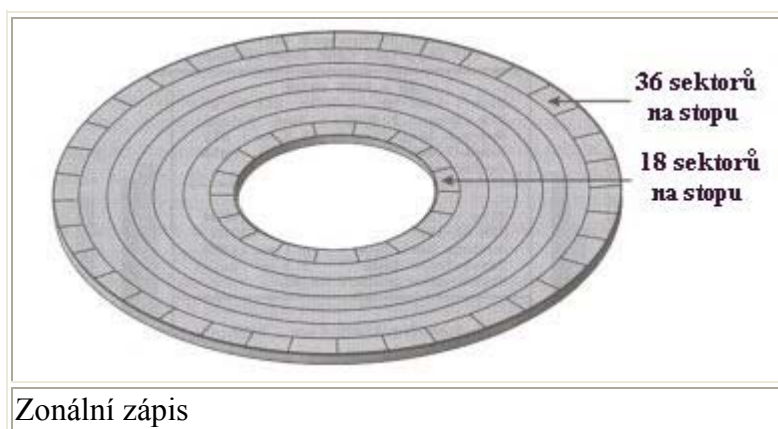
Všechny jednotlivé disky, ze kterých se celý pevný disk skládá, jsou podobně jako u pružného disku rozděleny do soustředných kružnic nazývaných **stopy (tracks)** a každá z těchto stop je rozdělena do **sektorů (sectors)**. Množina všech stop na všech discích se stejným číslem se u pevných disků označuje jako **válec (cylinder)**.

Geometrie disku udává hodnoty následujících parametrů:

- **Hlavy disku (heads):** počet čtecích (zapisovacích) hlav pevného disku. Tento počet je shodný s počtem aktivních ploch, na které se provádí záznam. Většinou každý jednotlivý disk má dvě aktivní plochy a k nim příslušné čtecí (zapisovací) hlavy.
- **Stopy disku (tracks):** počet stop na každé aktivní ploše disku. Stopy disku bývají číslovány od nuly, přičemž číslo nula má vnější stopa disku.



- **Cylindry disku (cylindry):** počet cylindrů pevného disku. Tento počet je shodný s počtem stop. Číslování cylindrů je shodné s číslováním stop.
- **Sektory (sectors):** počet sektorů, na které je rozdělena každá stopa. U většiny pevných disků je podobně jako u pružných disků počet sektorů na všech stopách stejný. Tento způsob do jisté míry plytvá médiem, protože vnější stopy jsou delší a tudíž by se na ně mohlo umístit více sektorů. Existují však i pevné disky, u nichž se používá tzv. **zonální zápis** označovaný jako **ZBR (Zone Bit Recording)**. Jedná se metodu zápisu na pevný disk, která dovoluje umístit na vnější stopy pevného disku větší počet sektorů než na stopy vnitřní. ZBR tedy lépe využívá záznamové médium, ale způsobuje podstatně složitější přístup k datům. Sektory bývají číslovány od jedničky.

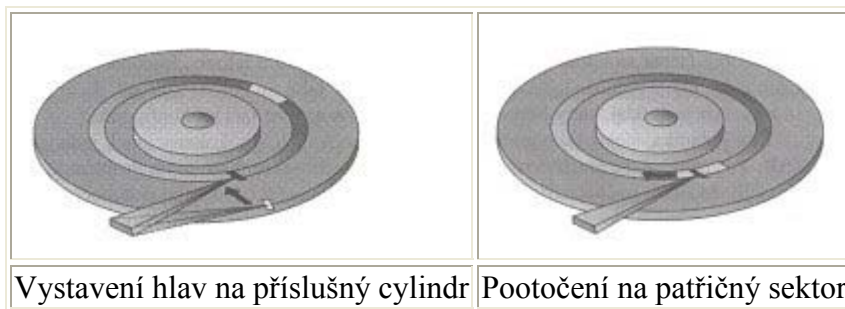


Zápis (čtení) na (z) pevný disk probíhá podobně jako u pružného disku na magnetickou vrstvu ve třech krocích:

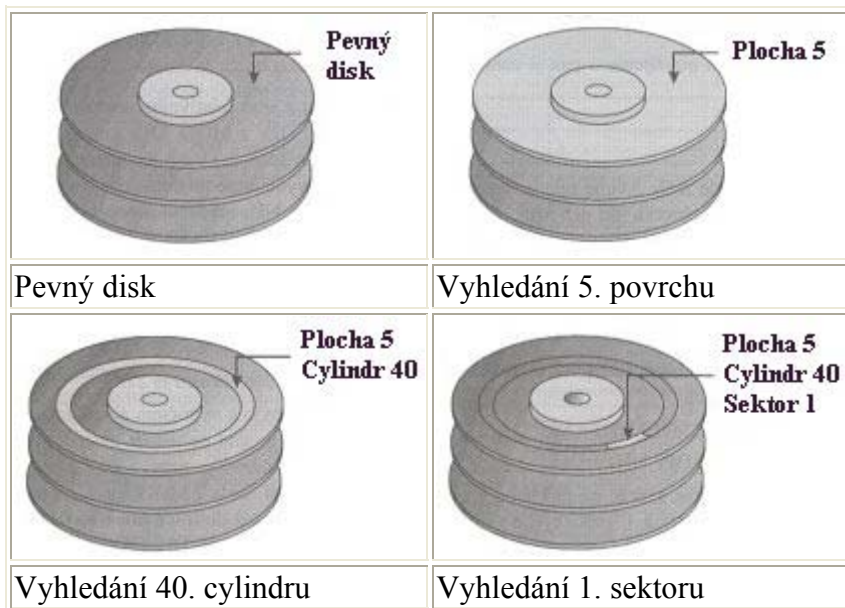
Počítačové systémy, periférie, operační systémy

- vystavení zapisovacích (čtecích) hlav na příslušný cylindr pomocí krokového motorku (dříve) nebo elektromagnetu (dnes)
- pootočení disků na patřičný sektor
- zápis (načtení) dat

Data jsou na pevný disk ukládána tak, že nejdříve je zaplněn celý 1. cylindr, potom 2. cylindr a tak dále až po poslední cylindr. Tento způsob dovoluje, aby se čtecí (zapisovací) hlavy podílely na čtení (zápisu) paralelně. Ukládání dat po jednotlivých discích by bylo podstatně pomalejší, protože v daném okamžiku by vždy mohla pracovat právě jedna hlava. Fáze zápisu (čtení) na (z) pevný disk:



Fáze vyhledání 1. sektoru ve 40. cylindru na 5. povrchu pevného disku

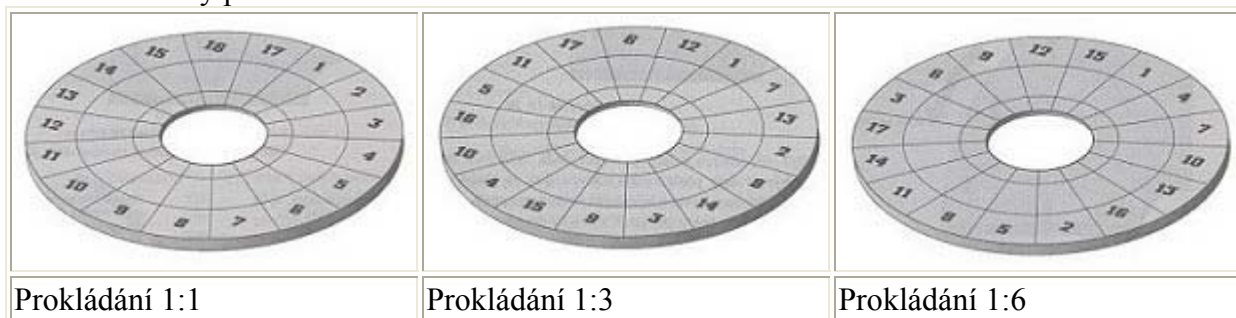


Protože rychlost otáčení pevného disku je poměrně vysoká, může se stát, že poté, co je přečten (zapsán) jeden sektor a data jsou předána dále, dojde k pootočení disků, takže čtecí (zapisovací) hlavy se nenacházejí nad následujícím sektorem, ale až nad některým z dalších sektorů. Nyní by tedy bylo nutné čekat další otáčku, než čtecí (zapisovací) hlavy budou nad požadovaným sektorem, a pak by se situace znovu opakovala. Protože tento způsob by velmi zpomaloval práci pevného disku, zavádí se tzv. **faktor prokládání pevného disku**. Jedná se o techniku, při které nejsou data zapisována (a posléze čtena) do za sebou následujících sektorů, ale jsou během jedné otáčky disku zapisována vždy do

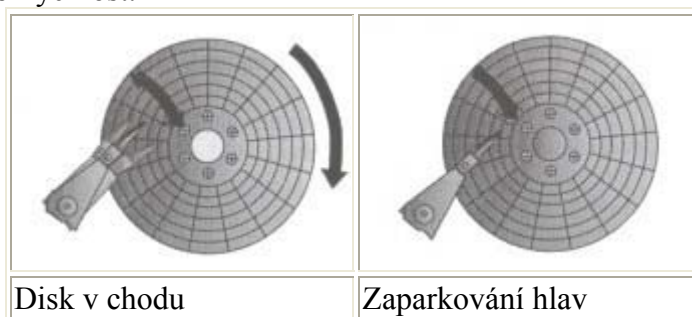
Počítačové systémy, periférie, operační systémy

každého n-tého sektoru (faktor prokládání 1:n). Číslo n je voleno tak, aby po přečtení a zpracování dat z jednoho sektoru byla čtecí (zapisovací) hlava nad dalším požadovaným sektorem.

Faktory prokládání



Při vypnutí počítače (a tím i pevného disku) se pevný disk přestává otáčet. Tím přestává existovat tenká vrstva, na které se pohybují čtecí (zapisovací) hlavy a vzniká riziko jejich pádu na disky. Tento pád by totiž mohl jednotlivé disky poškodit. Proto v okamžiku, kdy má pevný disk ukončit svou činnost, je nezbytné, aby čtecí (zapisovací) hlavy byly přemístěny do zóny, která je speciálně uzpůsobena k jejich přistání. U starších pevných disků bylo nutné vždy před vypnutím počítače provést pomocí nějakého programu tzv. **zaparkování diskových hlav**, tj. jejich přemístění na patřičné místo. Nové pevné disky již využívají tzv. **autopark**, který je založen na tom, že po vypnutí pevného disku se pevný disk ještě chvíli setrvačností otáčí a tím vyrobí dostatek energie nutné pro přemístění hlav do parkovací zóny. Pro tuto parkovací zónu bývá většinou vyčleněna nejvnitřnější stopa disku, protože je na ní nejnižší rychlost.



8.7.5 Rozhraní pevných disků

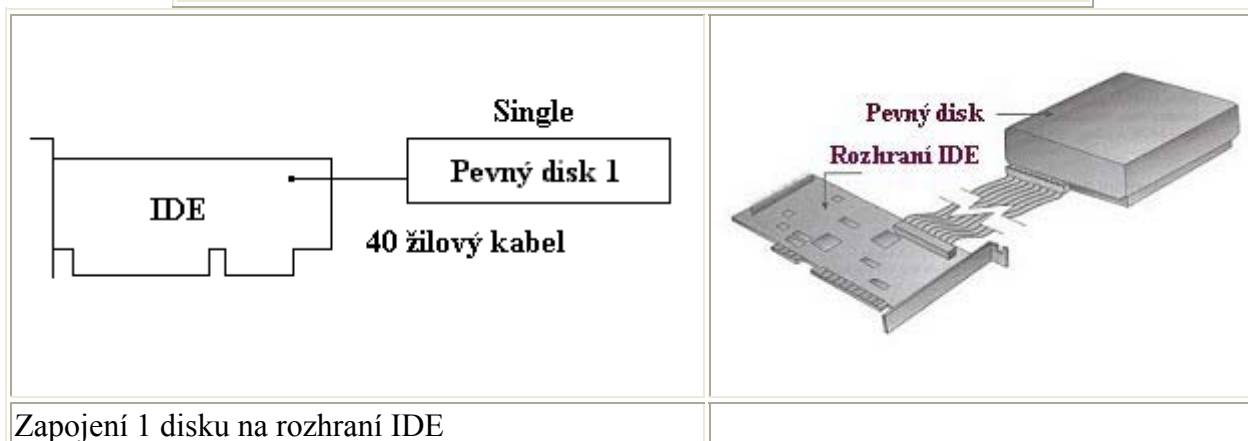
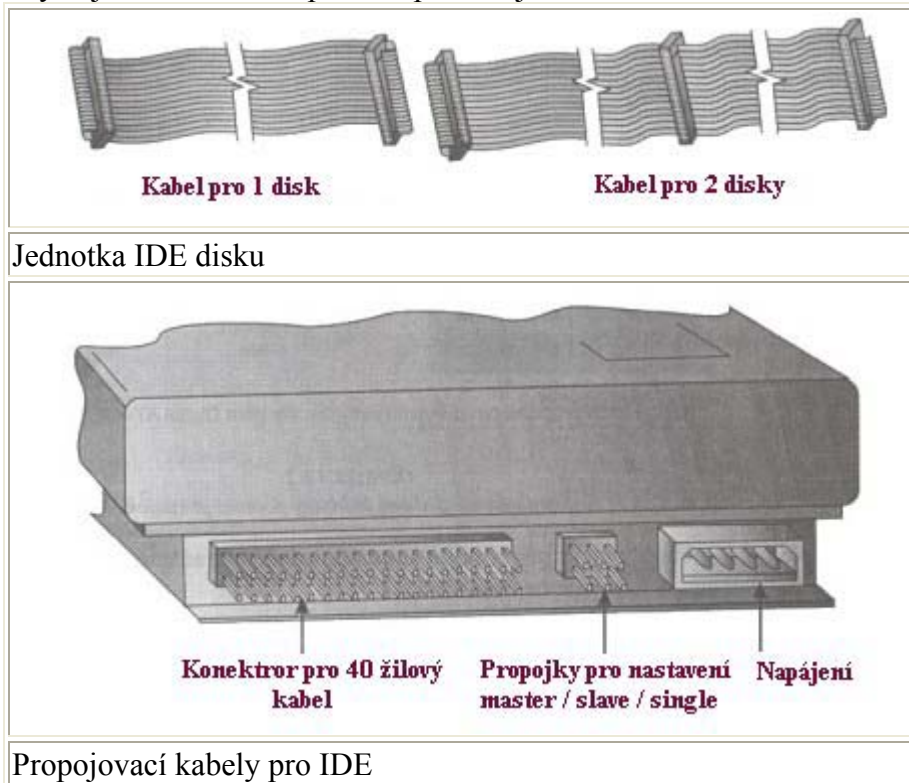
Rozhraní pevných disků jsou zařízení, která zprostředkovávají komunikaci mezi pevným diskem a ostatními částmi počítače. Rozhraní pevného disku určuje způsob komunikace a tím typ disku, který je možné k němu připojit.

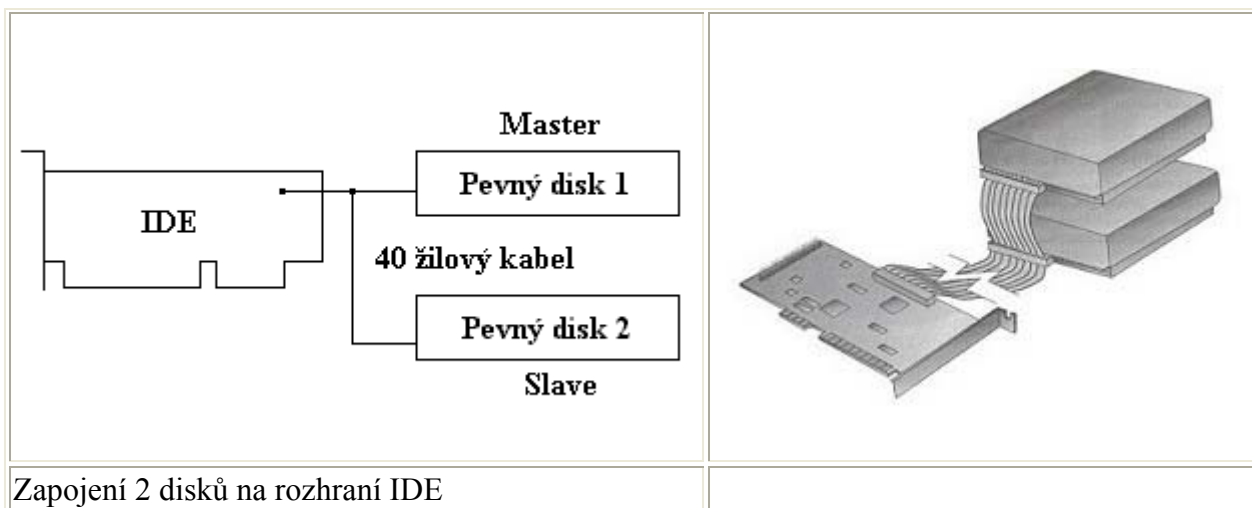
8.7.5.1 Rozhraní IDE

Rozhraní **IDE** (**I**ntegrated **D**evice **E**lectronics) nazývané též nesprávně AT-Bus bylo navrženo v roce 1986 firmami Western Digital a Compaq jako následník rozhraní ST506. Cílem bylo navrhnout levné rozhraní, které by poskytovalo vyšší výkon než předcházející dvě rozhraní. Jedním z limitujících faktorů jak u rozhraní ST506, tak u rozhraní ESDI byl propojovací kabel. Čím je delší kabel, tím nižší je maximální přenosová rychlost a tím vyšší je hladina šumu. Tato úvaha vedla k závěru, že hlavní řídicí jednotka disku byla umístěna

Počítačové systémy, periférie, operační systémy

přímo na pevný disk (tím se zkrátí kabel na minimum) a vlastní rozhraní už slouží pouze jako prostředník mezi diskem a sběrnicí. Díky tomuto řešení se podstatně snížila hladina šumu a je možné umístit na jednu stopu vyšší počet sektorů (26 až 35). Teoretická hranice přenosové rychlosti je 8 MB/s a prakticky se pohybuje asi v rozmezí od 700 kB/s do 1400 kB/s. Zapojení diskových jednotek IDE se provádí pomocí jednoho 40 žilového kabelu.





Rozhraní IDE dovozuje programově zjistit informace o geometrii připojených disků a je možné k němu připojit maximálně dva pevné disky. Protože každý z disků má svou řídicí jednotku umístěnou přímo u sebe, je nutné v případě zapojení dvou disků tyto disky nastavit pomocí propojek (jumperů) tak, aby jeden z nich byl jako **master** (hlavní) a druhý jako **slave** (podřízený). Operační systém se pak bude zavádět z disku označeného jako master. Doporučuje se, aby jako master byl nastaven novější disk, protože je možné předpokládat, že jeho elektronika bude lepší než elektronika staršího disku. V případě zapojení jednoho disku je nutné tento disk nastavit jako **single** (jediný). Toto nastavení bývá někdy shodné jako nastavení pro master. Připojování jiných zařízení než jsou pevné disky není oficiálně podporováno. Vzhledem k jednoduchosti rozhraní IDE bývá velmi často toto rozhraní integrováno na jedné desce společně s I/O porty.

Při komunikaci s pevným diskem má rozhraní IDE následující omezení:

- 4 bitů pro adresaci povrchu disku (maximálně 16 povrchů)
- 10 bitů pro adresaci cylindru (maximálně 1024 cylindrů)
- 6 bitů pro adresaci sektoru (maximálně 64 sektorů)

Při zápisu 512 B do jednoho sektoru je takto kapacita omezena na 512 MB (0,5 GB).

8.7.5.2 Rozhraní EIDE

Rozhraní **EIDE** (**E**nanced **I**ntegrated **D**evice **E**lectronics) je stejně jako jeho předchůdce navrženo firmou Western Digital. Vychází ze standardu IDE, zachovává kompatibilitu zdola a odstraňuje následující nedostatky rozhraní IDE: dovozuje zapojení až čtyř zařízení

- dovozuje zapojení i jiných zařízení než jsou pevné disky (např. CD-ROM, páskové mechaniky atd.)
- při práci s diskem používá adresovací metodu **LBA** (**L**inear **B**lock **A**ddress), která eliminuje omezení kapacity disku na 512 MB. Při adresaci LBA je rezervováno:
 - 4 bitů pro povrch (maximálně 16 povrchů)
 - 16 bitů pro cylindr (maximálně 65536 cylindrů)
 - 8 bitů pro sektor (maximálně 256 sektorů)

Počítačové systémy, periférie, operační systémy

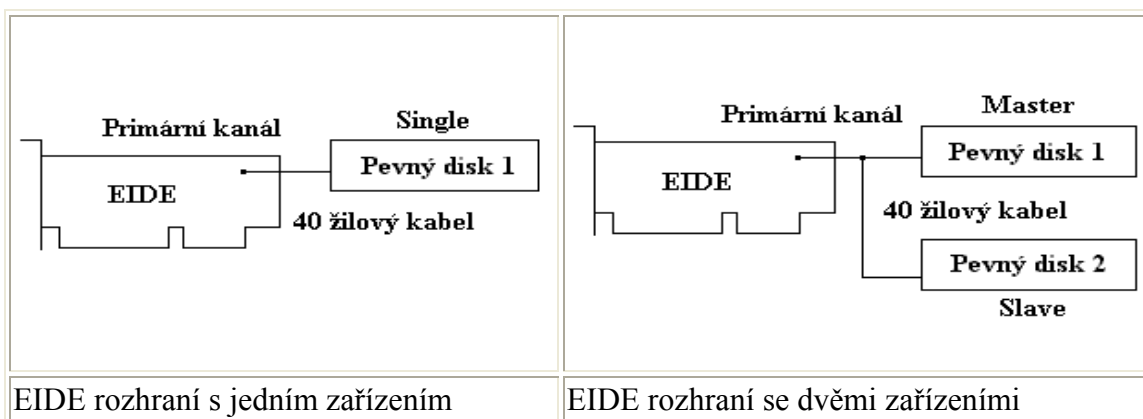
Při kapacitě 512 B na jeden sektor pak dostáváme maximální velikost disku 128 GB. Tato kapacita je však omezena možnostmi BIOSu na 8 GB.

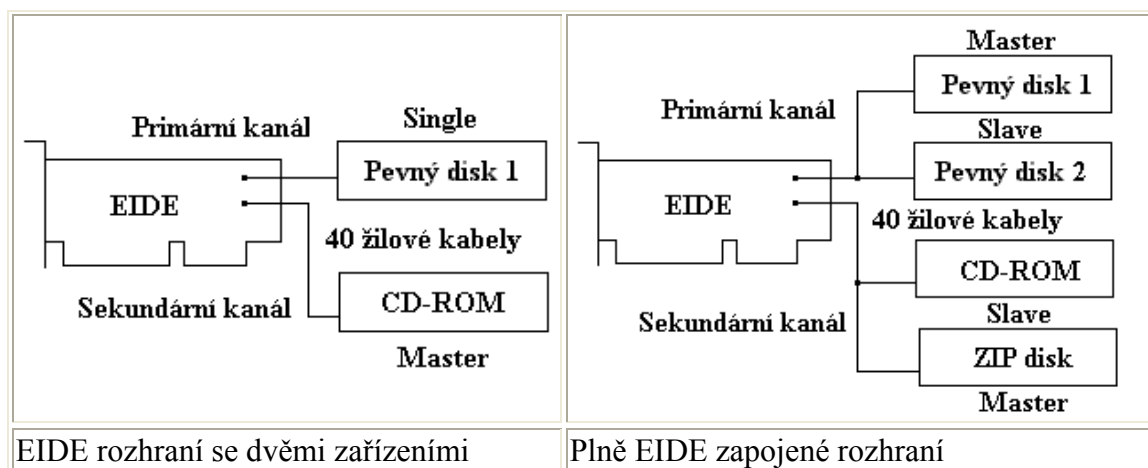


- poskytuje vyšší přenosovou rychlost a může komunikovat buď prostřednictvím režimu **PIO** (Processor Input Output), nebo prostřednictvím **DMA** (Direct Memory Access) režimu.
 - **PIO**: režim, při kterém je přenos dat řízen procesorem. Tento režim se postupně vyvíjel a poskytoval stále větší rychlost:
 - **PIO 0**: maximální přenosová rychlost je 2-3 MB/s
 - **PIO 1**: maximální přenosová rychlost je 5,22 MB/s
 - **PIO 2**: maximální přenosová rychlost je 8,33 MB/s
 - **PIO 3**: pro VL-Bus a PCI maximální přenosová rychlost je 11,1 MB/s
 - **PIO 4**: maximální přenosová rychlost je 16,6 MB/s
 - **PIO 5**: maximální přenosová rychlost je 20 MB/s
 - **DMA**: režim, ve kterém se pro přenos dat nevyužívá procesor:
 - **DMA 0**: maximální přenosová rychlost je 2,08 MB/s
 - **DMA 1**: maximální přenosová rychlost je 4,17 MB/s
 - **DMA 2**: maximální přenosová rychlost je 8,33 MB/s
 - **DMA Multiword 0**: maximální přenosová rychlost je 4,17 MB/s
 - **DMA Multiword 1**: maximální přenosová rychlost je 13,3 MB/s
 - **DMA Multiword 2**: maximální přenosová rychlost je 16,6 - 22 MB/s

Jednotlivá zařízení připojená k EIDE rozhraní jsou zapojena na dva kanály:

- **primární** (primary IDE) **sekundární** (secondary IDE)



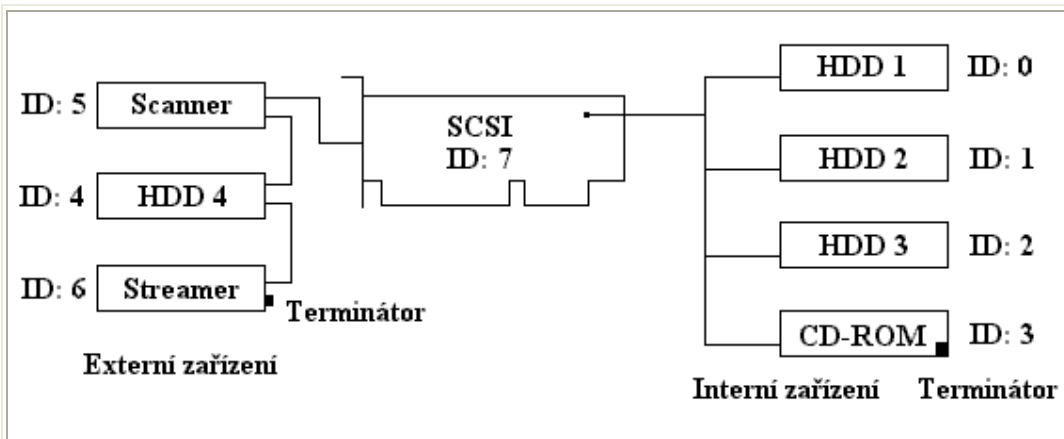


Na každý kanál je možné připojit maximálně dvě zařízení pomocí 40 žilového kabelu, který je shodný s kabelem IDE. Na obou kanálech je potom u jednotlivých zařízení nutné nastavit správným způsobem propojky do pozic master/slave/single. Nastavování se provádí podle stejných pravidel jako u IDE rozhraní. Operační systém se standardně zavádí ze zařízení master (single) na primárním kanálu. Při zapojování zařízení se nedoporučuje na jednom kanále kombinovat rychlé zařízení (např. pevný disk) s pomalejším zařízením (např. CD-ROM), protože pak dochází ke zpomalování celého kanálu a tím i pevného disku.

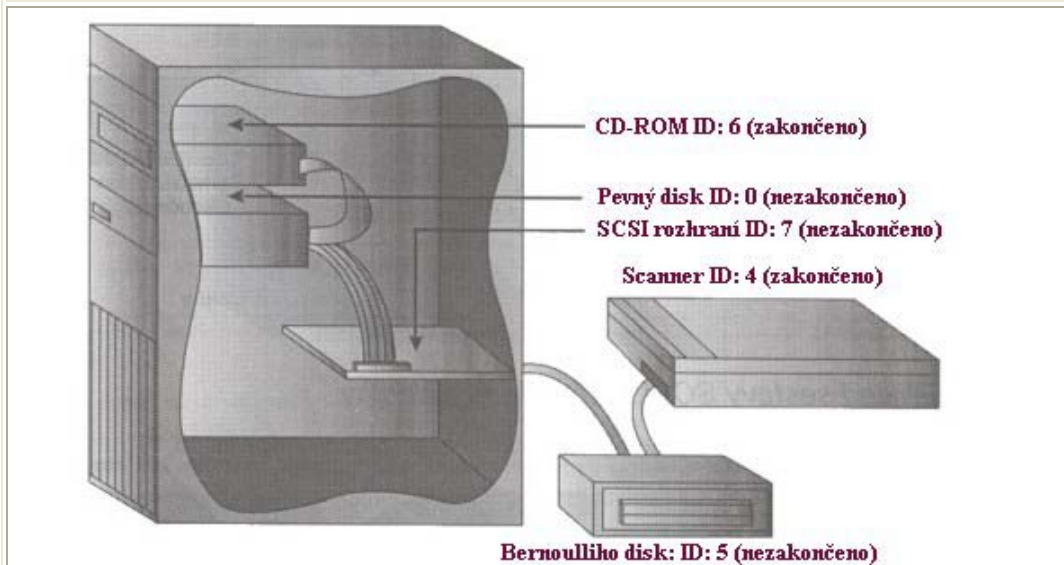
8.7.5.3 Rozhraní SCSI

Rozhraní **SCSI** (Small Computer Systems Interface) bylo vyvíjeno zhruba ve stejné době jako rozhraní ESDI. Cílem SCSI bylo vytvořit standardní rozhraní poskytující sběrnici pro připojení dalších zařízení. SCSI dovoluje připojit ke své sběrnici až 8 různých zařízení, z nichž jedno musí být vlastní SCSI rozhraní. Mezi další velké výhody patří možnost připojení nejen interních zařízení, jako tomu bylo u všech předchozích rozhraní, ale i zařízení externích. SCSI není pevně vázáno na počítač řady PC, ale je možné se s ním setkat i u jiných počítačů (např.: MacIntosh, Sun, Silicon Graphics).

Jednotlivá zařízení jsou propojena pomocí 50 vodičové sběrnice a nesou jednoznačnou identifikaci v podobě **ID čísla** (v rozmezí 0-7). ID 7 bývá většinou nastaveno na SCSI rozhraní a ID 0 bývá zařízení, ze kterého se zavádí operační systém. Sběrnice musí být na posledních zařízeních ukončena tzv. **terminátory** (zakončovací odpory), které ji impedančně přizpůsobují a zabraňují tak odrazu signálů od konce vedení. Tyto terminátory jsou buď součástí zařízení, nebo lze použít externí terminátory.



Zapojení zařízení na rozhraní SCSI



Zapojení SCSI rozhraní

K SCSI rozhraní je možné připojovat celou řadu různých zařízení, jako jsou např. pevné disky, CD-ROM mechaniky, páskové jednotky (streamery), scannery, magnetooptické disky, Bernoulliho disky atd. Externí zařízení mají dva konektory :

- vstupní: směrem od řadiče
- výstupní: směrem k dalšímu zařízení

Délka celé sběrnice by u SCSI-1 neměla přesáhnout 25 m.

Jako rozšíření předchozího SCSI-1 vzniká rozhraní SCSI-2, které je též komerčně nazýváno jako Fast SCSI. SCSI-2 je zdola kompatibilní s původním SCSI-1, má však vyšší přenosovou rychlost (až 10 MB/s) a přísnější nároky na kabeláž (celá délka sběrnice může být maximálně 3 m). Dalším rozšířením rozhraní SCSI je rozhraní označované jako SCSI-3, které dovoluje připojit až 32 zařízení s ID v rozmezí 0-31.

8.8 Videokarty

Počítače řady PC používají podobně jako většina počítačů k zobrazení

Počítačové systémy, periférie, operační systémy

informace vakuovou obrazovku, která je součástí monitoru. Videokarty (grafické karty, grafické adaptéry) jsou zařízení, která zabezpečují výstup dat z počítače na obrazovku monitoru.

Videokarta má vliv na to, jaký software může uživatel na počítači provozovat a jak rychle se data na obrazovku přenášejí. Většina videokart dovoluje práci ve dvou základních režimech:

- **textový režim:** režim, který umožňuje zobrazovat pouze předem definované znaky, jako jsou písmena (A, a, B, b, C, c, ...), číslice (1, 2, 3, ...), speciální znaky (&, ^, %, ...) a pseudografické znaky (symboly pro vykreslování tabulek). Tyto znaky jsou přesně definované pomocí matic bodů a je možné je zobrazovat pouze jako celek.
- **grafický režim:** režim, ve kterém jsou informace zobrazovány po jednotlivých obrazových, bodech tzv. **pixelech** (Picture Element). Tento režim již nepoužívá předem definované znaky, ale může z jednotlivých pixelů vykreslit prakticky "libovolnou" (závisí na možnostech konkrétní karty) informaci.

Základní parametry každé videokarty jsou

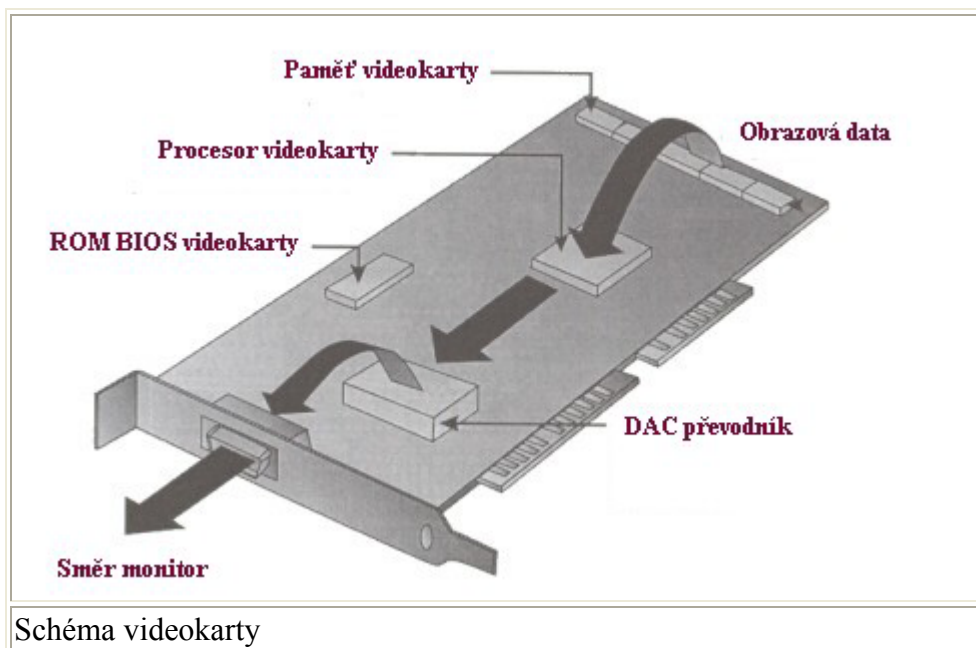
Parametr	Vysvětlení
Rozlišení v textovém režimu	Počet znaků, které je možné v textovém režimu zobrazit na jednom řádku, a počet řádků, které je možné umístit na obrazovku
Matice znaku	Počet bodů (ve vodorovném a ve svislém směru), ze kterých se může skládat jeden znak v textovém režimu
Rozlišení v grafickém režimu	Počet pixelů, které je možné v horizontálním a ve vertikálním směru zobrazit
Počet barev (barevná hloubka)	Počet barev, které je možné zároveň zobrazit. Udává se většinou pouze pro grafický režim.
Rychlost	Počet pixelů, které videokarta dokáže vykreslit za jednotku času. Udává se pouze v grafickém režimu.

Grafické karty, které jsou schopny zobrazit maximálně dvě barvy, jsou označovány jako **monochromatické** (černobílé).

Moderní videokarty se skládají z následujících částí:



- procesor
- paměť
- DAC převodník
- ROM BIOS



Při práci zapisuje procesor počítače obrazová data do videopaměti. Takto zapsaná data jsou potom čtena procesorem videokarty, který na jejich základě vytváří digitální obraz. Digitální obraz je poslán na vstup DAC (Digital Analog Convertor) převodníku, který z něj vytváří analogový obraz nutný pro moderní monitory, řízené spojitě (analogově) měnící se hodnotou signálů tří základních barev (Red - červená, Green - zelená, Blue - modrá)

Typy videokart

8.8.4 Grafická karta VGA

Grafický adaptér VGA (Video Graphics Array) firmy IBM byl vyroben v roce 1987 původně pro řadu počítačů IBM PS/2. Jedná se o kartu, která je schopna v textovém režimu zobrazovat 80 x 25 znaků a jeden znak je definován v matici 9 x 14 bodů. Znaky v textovém režimu mohou být zobrazovány v 16 barvách. V grafickém režimu dokáže tato videokarta zobrazit maximálně 640 x 480 bodů v 16 barvách. Tento typ videokarty vyžaduje oproti předchozím kartám nový typ monitoru, který není řízen digitálním sledem signálů, ale spojitě (analogově) měnící se hodnotou signálu každé ze základních barev (Red - Červená, Green - Zelená, Blue - Modrá).

8.8.5 Grafická karta SVGA

Videokarta SVGA (Super Video Graphics Array) je dnes nejpoužívanější typ grafické karty. Jejím nejdůležitějším prvkem je procesor, který do značné míry ovlivňuje její výkon. U modernějších typů videokart je tento procesor schopen realizovat (buď sám nebo za pomoci nějakého dalšího obvodu) některé často používané grafické operace. Takováto videokarta bývá nazývána také jako **akcelerátor** a umožňuje podstatně vyšší výkon, protože není nutné, aby každý pixel, který se má zobrazit na obrazovce, byl vypočítán procesorem počítače. Procesor počítače tak pouze vydá příkaz kartě, co má vykreslit (linku, kružnici, obdélník), a vlastní výpočet jednotlivých zobrazovaných pixelů provede k tomuto účelu specializovaný procesor videokarty. Kromě těchto jednoduchých

Počítačové systémy, periférie, operační systémy

operací je možné, aby procesor videokarty prováděl i složitější operace používané při práci s 3D grafikou (např. zakrývání neviditelných hran, stínování apod.) nebo operace spojené s přehráváním videosekvencí.

Videokarty tohoto typu se pak nazývají **3D akcelerátory** a **multimediální akcelerátory**. Procesor videokarty je propojen pomocí sběrnice s videopamětí. Šířka této sběrnice bývá (32b, 64b, 128b). Paměť na videokartě může být následujících druhů:

- **DRAM** (Dynamic RAM) popř. **EDO DRAM** nebo **SDRAM**: paměť, do které může v daném okamžiku buď procesor počítače zapisovat, nebo z ní může procesor karty číst. Tato paměť je levnější, ale poskytuje nižší výkon.
- **VRAM** (Video RAM): paměť mající možnost dvou vstupů a výstupů. Tato paměť, která bývá také označována jako dvoubranová (dvouportová), dovoluje, aby v jednom okamžiku do ní procesor počítače zapisoval a zároveň procesor videokarty z ní četl. Tento druh paměti je dražší, ale poskytuje vyšší výkon.
- **SGRAM** (Synchronous Graphic RAM): podobně jako paměť DRAM, ale navíc má podporu blokových operací, tj. má rychlejší operace, jako jsou například přesun bloku dat z jedné části paměti do druhé, naplnění části paměti stejnou hodnotou apod.
- **WRAM** (WindowRAM): dvoubranová paměť podobně jako VRAM s podporou blokových operací.



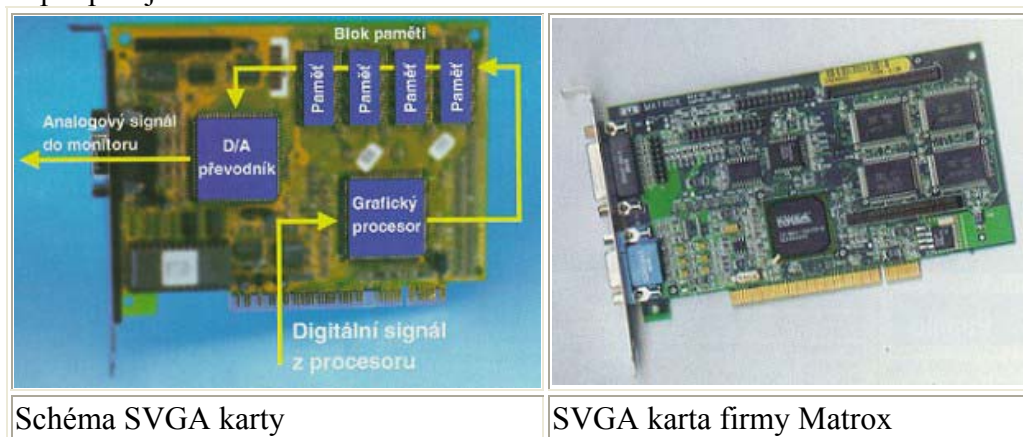
V závislosti na kapacitě této paměti, tzv. **video paměti**, a procesoru, který tato karta používá, je možné zobrazovat následující režimy:

Kapacita video paměti	Maximální rozlišení	Počet barev
256 kB	640 x 480	16
	800 x 600	16
512 kB	1024 x 768	16
	800 x 600	256
1 MB	1600 x 1200	16
	1280 x 1024	16
	1024 x 768	256
2 MB	800 x 600	65536
	640 x 480	16,7 mil.
	1600 x 1200	256
	1280 x 1024	256
3 MB	1024 x 768	65536
	800 x 600	16,7 mil.
	1600 x 1200	256
	1280 x 1024	65536
	1024 x 768	16,7 mil.

Počítačové systémy, periférie, operační systémy

4 MB	1600 x 1200	65536
	1280 x 1024	16,7 mil.
6 MB	1600 x 1200	16,7 mil

Super VGA vyrábí v dnešní době mnoho výrobců (Diamond, Matrox, ATI a další). Při této výrobě však došlo ke ztrátě vzájemné kompatibility v režimech s vyšším rozlišením. Posledním naprosto standardním režimem, který je na všech SVGA kartách kompatibilní, je režim VGA 640 x 480 v 16 barvách. Režimy s vyšším rozlišením již bývají nekompatibilní a vyžadují speciální programové ovladače určené pro práci s tímto typem videokarty. Z důvodu této nekompatibility byl později zaveden standard, který byl nazván **VESA** (Video Electronics Standard Association). Tento standard dnes většina videokart podporuje přímo svým hardwarem, jiné karty jej podporují pouze softwarově pomocí programů dodaných výrobcem videokarty a některé starší karty jej nepodporují vůbec.



Vzhledem k tomu, že dnes většina počítačů pracuje pod operačním systémem provozovaným v grafickém režimu (Windows, OS/2, X Window System), jsou na výkon videokarty kladeny vysoké nároky. Proto je více než vhodné, aby do počítačů s výkonnými procesory byly osazovány výkonné akcelerátory určené pro sběrnici PCI popř. **AGP** (Accelerated Graphic Port - speciální typ sběrnice vyvinutý firmou Intel určený pouze pro videokarty). V opačném případě je totiž možné, že by videokarta značně degradovala výkon celého počítače.

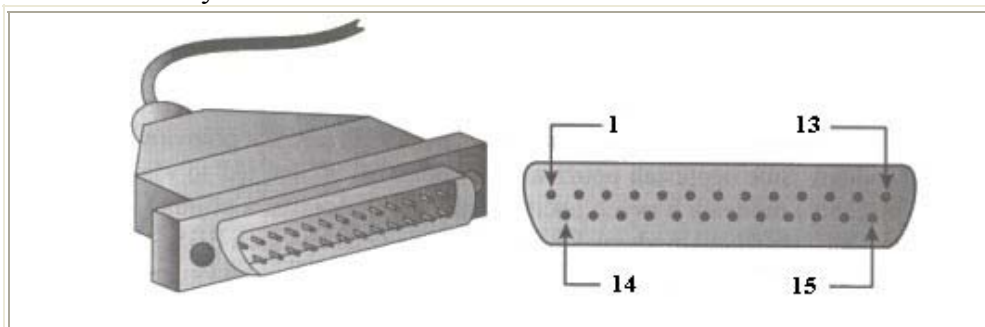
8.9 IO karta

IO karta (Input/Output) je deska obsahující porty pro připojení periferních zařízení. Port je místo spojení procesorové jednotky s komunikačním kanálem a slouží k připojení dalších periferních zařízení. Standardní I/O karta obsahuje:

- **1 paralelní port:** bývá označován jako **LPT1** a slouží např. pro připojení tiskárny, ZIP disku, propojení dvou počítačů. Informace jsou přes paralelní port přenášeny paralelně, tzn. že je vždy zároveň přenášena určitá sada bitů.
- **2 sériové porty:** bývají označovány jako **COM1**, **COM2** a slouží pro připojení počítačové myši, tiskárny, propojení dvou počítačů. U sériového portu jsou informace přenášeny sériově, tj. jednotlivé bity jsou posílány jednotlivě za sebou.
- **1 game port:** slouží k připojení křížového ovladače pro hry (joystick)

Počítačové systémy, periférie, operační systémy

Paralelní port je vyveden z počítače prostřednictvím 25 kolíkové zásuvky typu Canon. Sériové porty bývají většinou vyvedeny pomocí 9 kolíkové a 25 kolíkové zástrčky Canon.

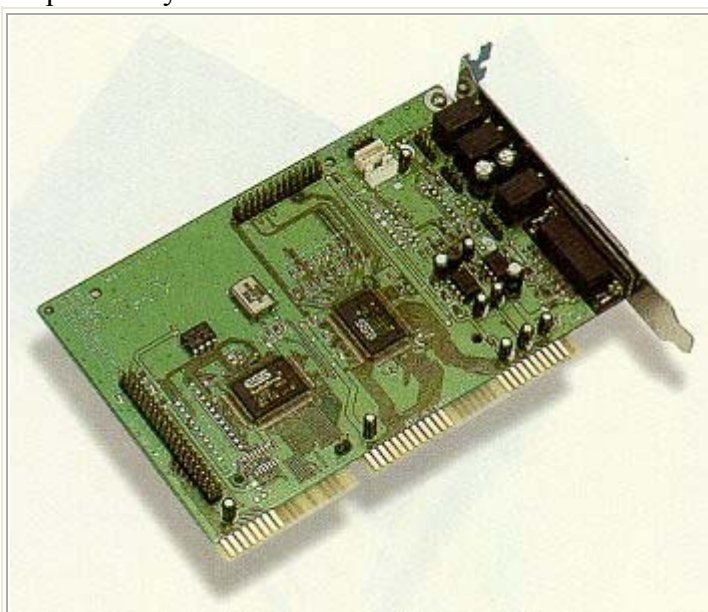


25 kolíkový konektor Canon

Vzhledem k jednoduchosti I/O karty je možné se dosti často setkat s kombinovanými kartami obsahujícími I/O kartu společně s řadičem pružných disků a rozhraním IDE (EIDE) Nové základní desky určené pro procesory Pentium a vyšší mají již I/O kartu integrovánou přímo jako svou součást. V případě potřeby je možné, aby v jednom počítači byla osazena více než jedna I/O karta a počítač tak měl více portů. Při jejich osazování je však nutné dbát na jejich správné nastavení, aby nedošlo ke konfliktu prostředků (IRQ, I/O Adresy), které tyto karty využívají, a tím i k jejich špatné funkci.

8.10 Zvuková karta

Počítač řady PC je ve své standardní konfiguraci vybaven malým reproduktorem označovaným jako **PC speaker**. Tento reproduktor je součástí skříně počítače a je připojen přímo na základní desku počítače. Jeho zvukové schopnosti jsou však velmi omezené a slouží většinou pouze k vydávání jednoduchých zvuků, jako jsou např. varovné pípnutí při vzniku chyby apod., popř. k přehrávání nějakých jednoduchých zvukových záznamů pro vysloveně amatérské účely. Pokud je požadován kvalitnější zvukový výstup z počítače, je nezbytné tento počítač vybavit zvukovou kartou.

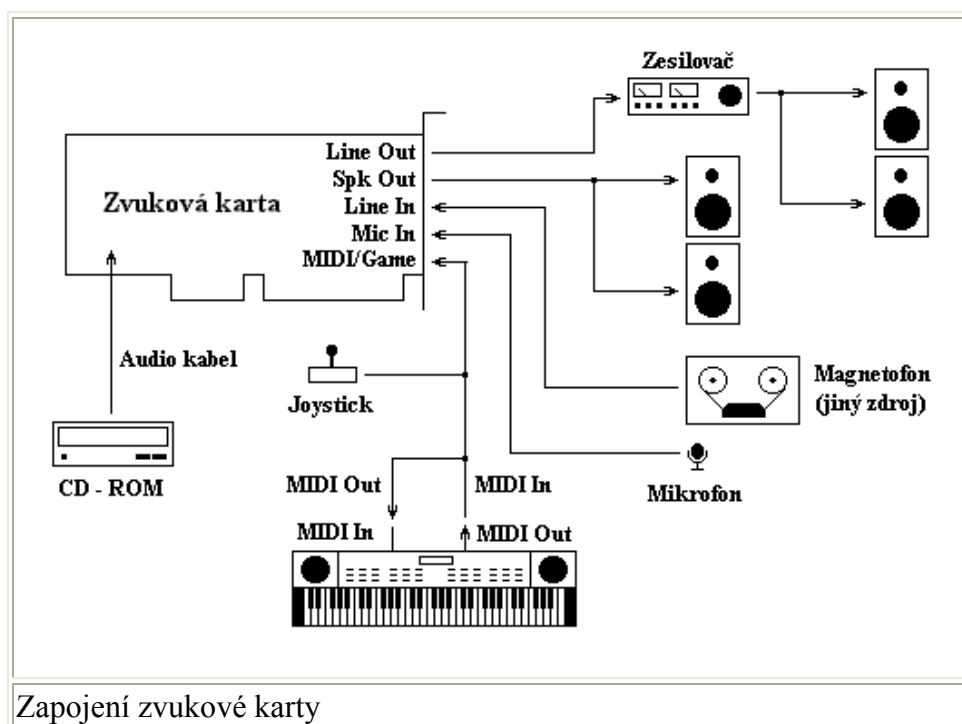


Zvuková karta

Počítačové systémy, periférie, operační systémy

Zvuková karta (sound card) je zařízení, které slouží k počítačovému zpracování zvuku. V závislosti na své kvalitě (a tím i ceně) zajišťuje kvalitní zvukový výstup z počítače vhodný i pro profesionální účely. Ke zvukové kartě lze dále připojit následující zařízení:

- sluchátka
- reproduktory
- zesilovač
- mikrofon
- externí zdroje (rádio, magnetofon, ...)
- je-li karta vybavena rozhraním **MIDI** (Musical Instrument Digital Interface), je možné k ní připojit i elektronické hudební nástroje vybavené také tímto rozhraním (např. elektronické varhany, syntetizátory apod.)



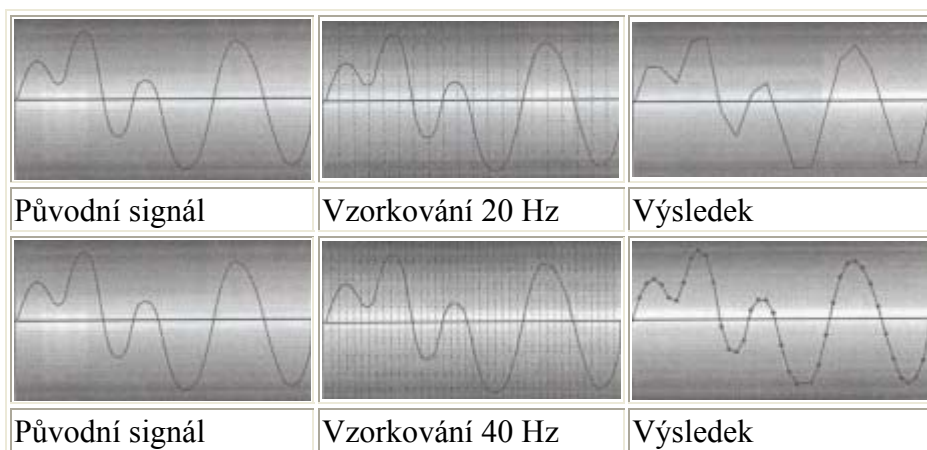
Při záznamu zvuku pomocí zvukové karty je nezbytné rozlišit dva základní případy:

- záznam je prováděn z nějakého zdroje poskytujícího analogový signál (mikrofon, rádio, magnetofon, audio CD). Takovýto signál se skládá z vln (kmitů) o nestejném tlaku, který je vytvářen ve vzduchu hlasivkami, hudebními nástroji nebo přírodními silami.

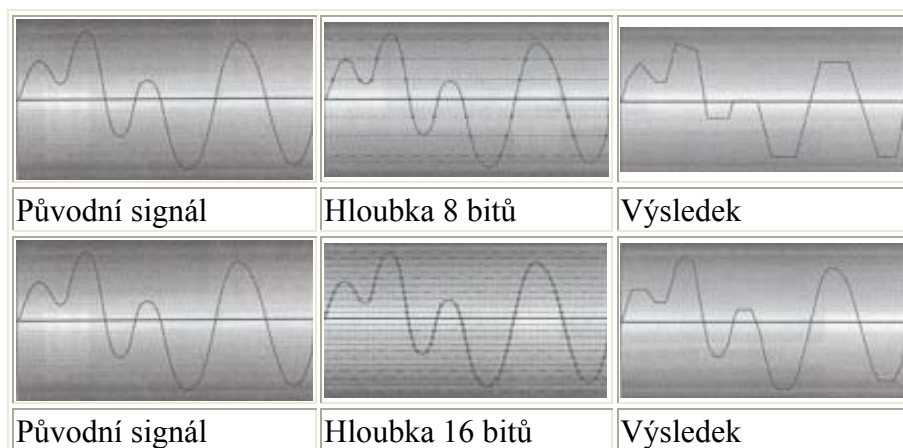


Příklad zvukového signálu

- V takovémto případě je nutné tento analogový signál převést na signál digitální. Převod se uskutečňuje pomocí **vzorkování (sampling)**. To znamená, že v každém časovém intervalu je zjištěn a zaznamenán aktuální stav signálu (vzorek). Je zřejmé, že čím kratší je tento interval, tím vyšší je **vzorkovací frekvence**, tím více vzorků bude pořízeno a tím bude výsledný záznam kvalitnější. Kvalitu je možné dále ovlivnit počtem rozlišitelných úrovní v každém vzorku.
- Ovlivnění kvality záznamu vzorkovací frekvencí.



- Ovlivnění kvality záznamu počtem rozlišitelných úrovní na každý vzorek



Počítačové systémy, periférie, operační systémy

- Při takovémto záznamu se běžně rozlišují následující úrovně kvality záznamu:

Kvalita	Vzorkovací frekvence	Počet bitů na vzorek	Počet vzorků	Délka dig. záznamu (B/s)
Telephone Quality	11025 Hz	8	1 - Mono	11 kB/s
Radio Quality	22050 Hz	8	1 - Mono	22 kB/s
CD Quality	44100 Hz	16	2 - Stereo	172 kB/s

Záznam je prováděn z nějakého zdroje poskytujícího již digitální signál (např. elektronické varhany připojené prostřednictvím MIDI rozhraní). V takovém případě se již neprovádí vzorkování, ale zaznamenávají se přímo jednotlivé byty zasílané tímto rozhraním. Tyto byty obsahují informace, jako jsou:

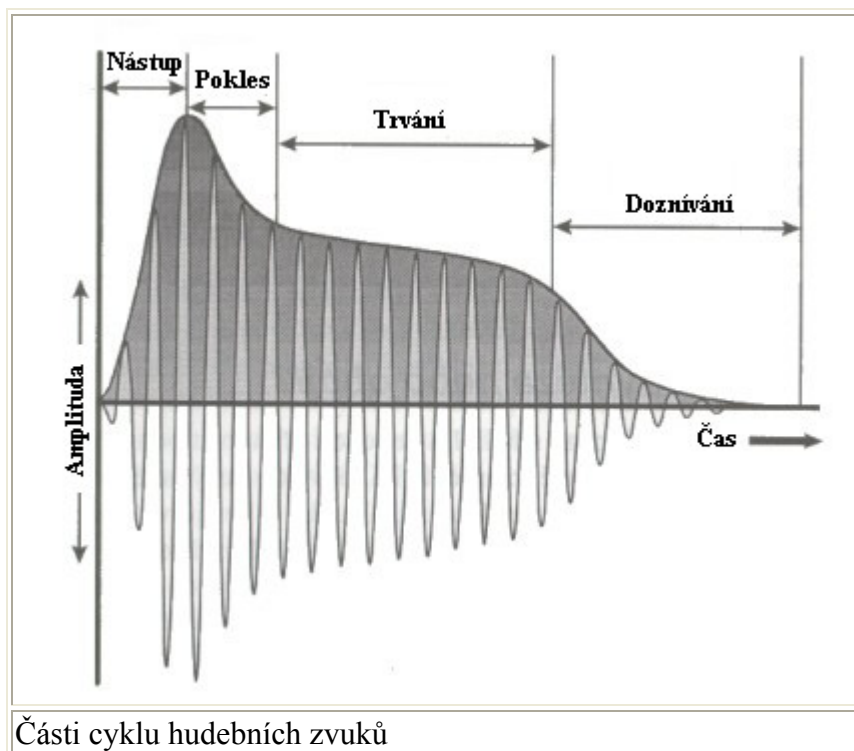
- nástroj, který tón hraje (piano, housle, varhany, ...)
- výška tónu
- délka tónu
- dynamika úhozu na klávesu
- další

Pro přehrání takového záznamu je nezbytné, aby zvuková karta (nebo jiné zřízení) byla schopna podle těchto informací sama vytvářet jednotlivé tóny.

Při vytváření zvuků pomocí zvukových karet je nutné vyjít ze skutečnosti, že u každého hudebního nástroje mají jeho zvuky podobu cyklu, který se skládá ze čtyř částí:

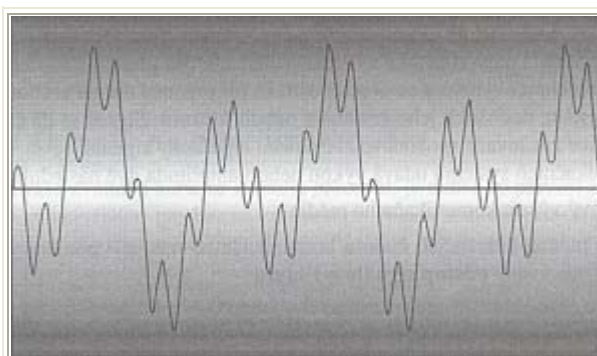
- Nástup (Attack)
- Pokles (Decay)
- Trvání (Sustain)
- Doznívání (Release)

Konkrétní hodnoty jednotlivých fází cyklu jsou charakteristické pro každý hudební nástroj a je potřeba, aby zvuková karta tyto hodnoty co možná nejpřesněji dodržovala. V opačném případě by zvuky ztrácely na věrnosti.

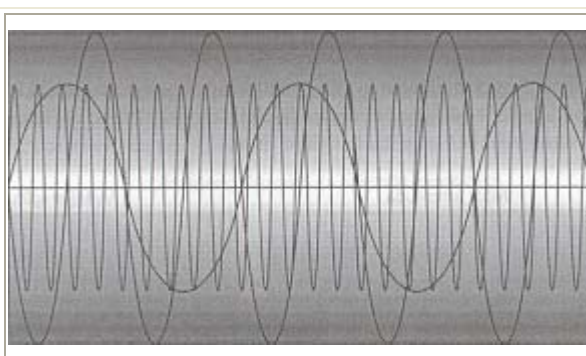


Pro vlastní vytvoření zvuku se využívá dvou rozličných mechanismů:

- **FM syntéza:** realizovaná tzv. FM syntetizátorem (obvod OPL 2, OPL 3 nebo OPL 4). Tato metoda vychází z faktu, že každé vlnění lze sestavit složením vybrané série sinusových kmitů o patřičné frekvenci a amplitudě.



Původní zvukový signál



Sinusové kmity původního signálu

- FM syntéza tedy vychází z popisu příslušného hudebního nástroje na základě Fourierova rozvoje, s jehož pomocí se potom zvuk těchto nástrojů emuluje jako superpozice několika sinusových signálů. Takto získaný signál se může ještě dále upravit různými efekty. Jedná se o levnější realizaci, která se svými výsledky zvukům reálných nástrojů pouze blíží a nikdy jich nemůže dosáhnout. Zvukové karty, které používají pouze tento způsob pro vytváření zvuků, jsou vhodné jen pro amatérské použití (ozvučení her apod.).
- **Wave Table syntéza:** používaná u dražších zvukových karet. Tato metoda používá přímo navzorkovaný signál skutečného nástroje uložený ve své vlastní paměti (ROM nebo RAM). Protože je nemožné,

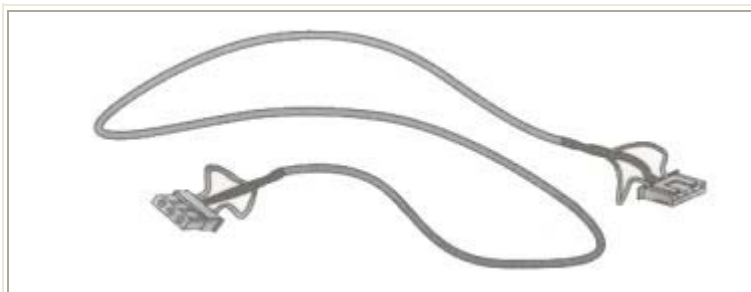
Počítačové systémy, periférie, operační systémy

aby v paměti byly uchovány vzorky všech výšek tónů od všech nástrojů, je v paměti vždy uložen jeden tón od každého nástroje. Různých výšek tohoto tónu se pak dosahuje různou rychlostí přehrání tohoto vzorku.



Reproduktory pro připojení ke zvukovým kartám

Zvukové karty bývaly obzvláště dříve vybavovány ještě IDE rozhraním, které sloužilo k připojení CD-ROM disku nebo speciálním rozhraním pro první CD-ROM mechaniky. Dnes toto řešení nemá velké opodstatnění, protože počítače jsou standardně vybaveny EIDE rozhraním, které dovoluje pohodlnější a rychlejší zapojení mechaniky CD-ROM. V případě, že v počítači je osazena CD-ROM mechanika a zároveň i zvuková karta, je velmi vhodné, aby obě tato zařízení byla propojena pomocí tzv. audio kabelu. Díky tomuto propojení je pak možné na CD-ROM přehrávat zvukové CD a poslouchat je z reproduktorů připojených ke zvukové kartě.



Audio kabel

Kromě uvedených vlastností mohou být ještě zvukové karty vybaveny pozicemi pro paměťové moduly RAM, do kterých si uživatel může ukládat vlastní vzorky různých nástrojů vytvořené buď elektronickým syntetizátorem nebo vzniklé nějakou úpravou již existujících vzorků. Dále je možné na zvukových kartách vidět i různé specializované obvody pro vytváření různých efektů v reálném čase (např. prostorový zvuk apod.)

8.11 Síťová karta

Síťová karta je zařízení, které umožňuje připojení počítače do počítačové sítě. Mezi základní parametry každé síťové karty patří:

Parametr	Vysvětlení	Rozsah
Typ sítě	Typ sítě, pro který je daná karta	Ethernet, Fast Ethernet.

Počítačové systémy, periférie, operační systémy

	určena	Arcnet, Tokenring
Rychlost	Množství dat, které je karta do sítě schopna vyslat (ze sítě přijmout) za jednotku času	100 kb/s - 100 Mb/s
Typ média	Typ síťového média (kabelu), které je možné k síťové kartě připojit.	Tenký koaxiální kabel, silný koaxiální kabel, kroucená dvojlinka

Každé síťové médium se k síťové kartě připojuje pomocí specifického konektoru, který karta musí obsahovat. Síťová média:

- **tenký koaxiální kabel:** určený zejména pro vnitřní rozvody uvnitř budovy. Pro jeho připojení se používá konektor **BNC**. V dnešní době bývá častěji nahrazován kroucenou dvojlinkou.
- **silný koaxiální kabel:** používaný dříve k venkovním rozvodům, k jeho připojení se používá konektor **Canon**, který zde bývá označován jako **AUI**. Tento AUI konektor může sloužit také k připojení tzv. **transcieveru**, pomocí něhož je potom možné připojit jiný typ média (transciever AUI - BNC, transciever AUI - RJ45). Dnes je silný koaxiální kabel používaný jen zřídka, protože je nahrazován kvalitnějším optickým kabelem.
- **kroucená dvojlinka:** používaná pro vnitřní rozvody. Kroucená dvojlinka se připojuje pomocí konektoru **RJ-45**

Při realizaci sítě v rámci budovy se dnes poměrně často používá tzv. **strukturovaná kabeláž**, u které se pro horizontální rozvody (v rámci patra) používá kroucená dvojlinka a pro vertikální rozvody je použito optického vlákna.

Některé síťové karty jsou vybaveny patičkou pro obvod zvaný **Boot ROM**. Boot ROM je paměť typu EPROM (EEPROM), která obsahuje programové vybavení nezbytné pro zavádění operačního systému z počítačové sítě místo jeho zavádění z lokálního disku.



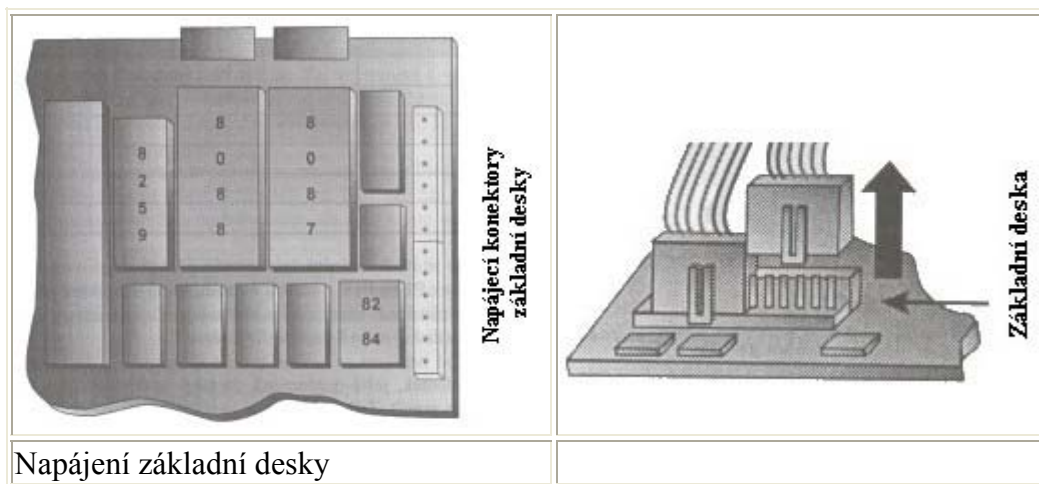
Síťová karta

8.12 Skříň počítače

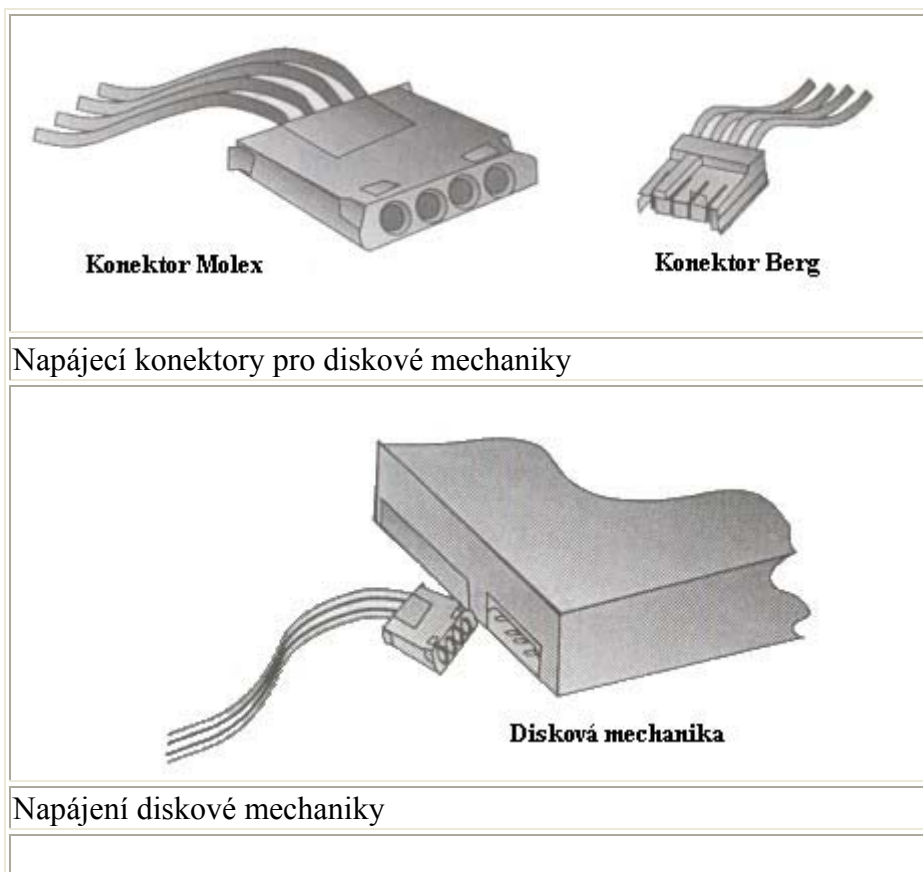
Skříň počítače obsahuje:

Počítačové systémy, periférie, operační systémy

- zdroj (150W - 250W) poskytující napětí (+5 V, -5 V, +12 V, -12 V) s napájecími kabely, které slouží k přivedení napájecího napětí pro:
 - základní desku



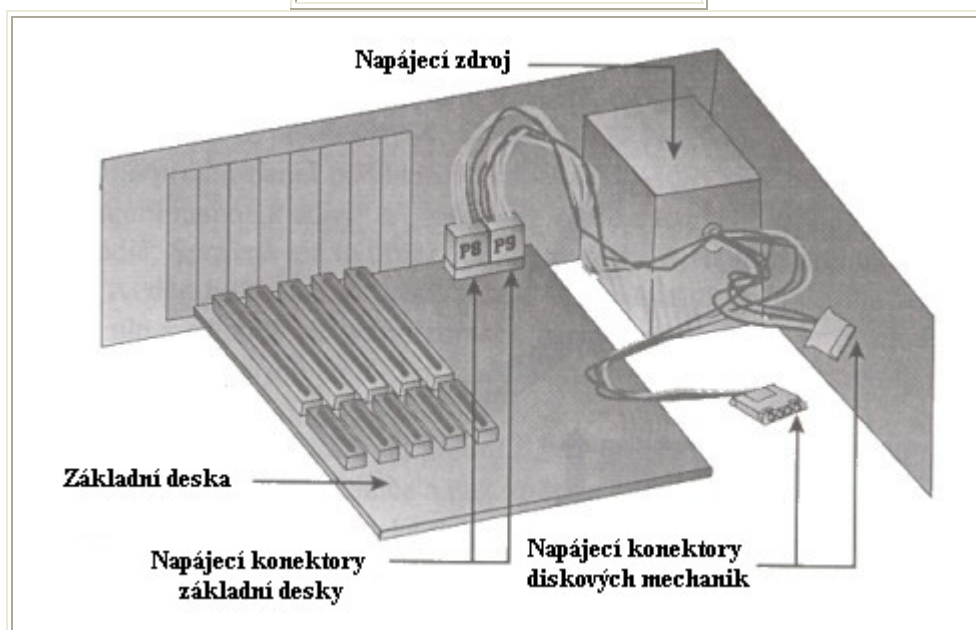
- diskové mechaniky (pružné disky, pevné disky, CD-ROM, apod.)



- aktivní chladič (s ventilátorem) procesoru



Aktivní chladič procesoru



Zapojení napájecího zdroje

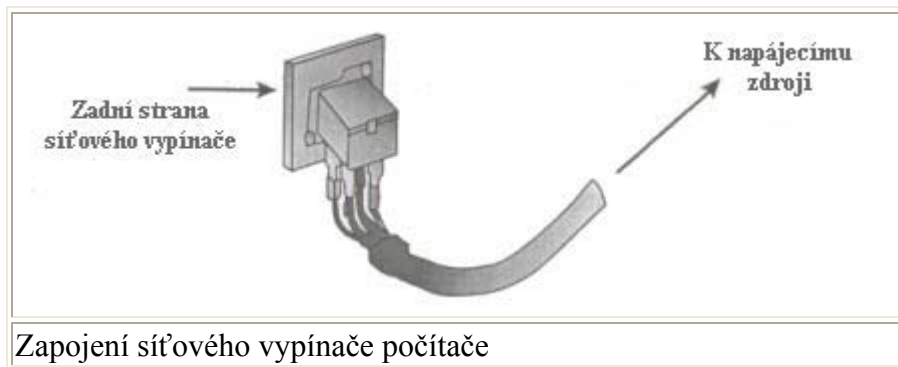


Zadní strana zdroje počítače

- Jednotlivé zapojené karty jsou napájeny ze sběrnice.
- různé LED diody
 - **Power**: signalizuje, že počítač je zapnutý nebo vypnutý
 - **Turbo**: signalizuje, zda počítač pracuje v Turbo režimu (tj. režimu s plným výkonem procesoru) nebo v režimu s nižším výkonem
 - **HDD**: signalizuje aktivitu pevných disků popř. jiných zařízení (např. CD-ROM) připojených k EIDE (SCSI) rozhraní
- kabely pro připojení LED diod a vypínačů

Počítačové systémy, periférie, operační systémy

- síťový vypínač: pro zapnutí a vypnutí počítače



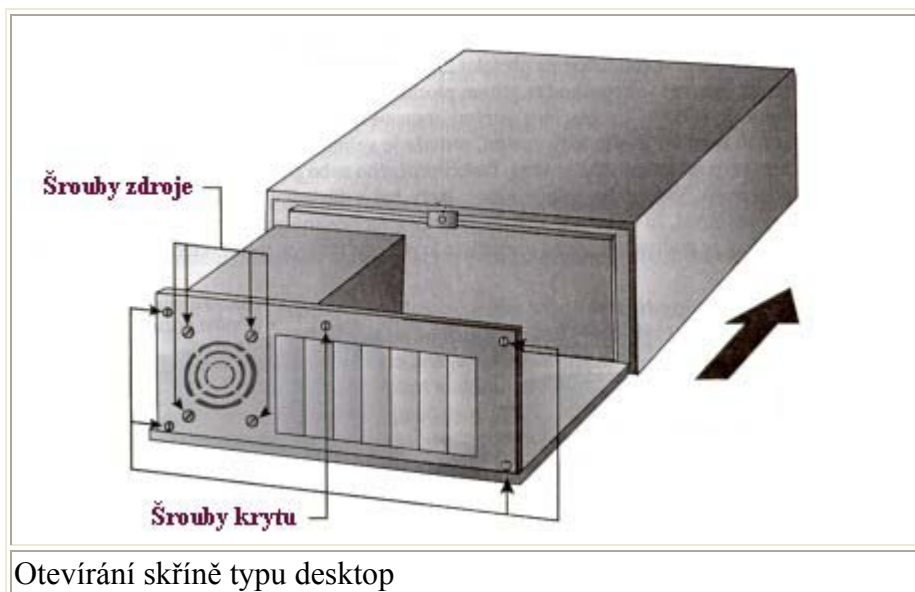
- šachty pro upevnění diskových mechanik
- otvory pro výstupy ze zapojených karet
- pozice pro upevnění základní desky

Dále skříň počítače může obsahovat:

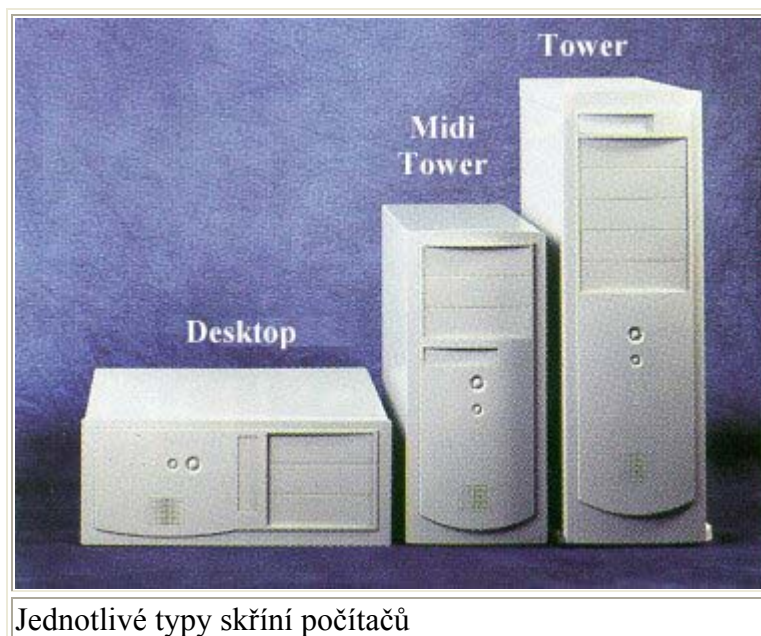
- tlačítko **Reset**: pro uvedení počítače do stavu, který následuje po jeho zapnutí
- tlačítko **Turbo**: pro přepínání mezi režimem s plným výkonem (Turbo) a sníženým výkonem
- zámek klávesnice: pro uzamknutí klávesnice, které způsobí, že veškeré stisky kláves jsou ignorovány.
- ukazatel frekvence: sada sedmissegmentových jednotek vyjadřující frekvenci procesoru. Prakticky žádný počítač neobsahuje měřič frekvence, který by skutečně testoval frekvenci procesoru. Údaj, který je zobrazen na ukazateli frekvence, je nastaven sadou propojek (jumperů), které dovolují nastavit libovolný údaj.

Podle provedení a tvaru skříně je možné rozlišit následující typy:

- **desktop**: skříň, která bývá umístěna vodorovně na stole. Základní deska je ve vodorovné poloze a jednotlivé rozšiřující karty se osazují svisle.
- **slim**: skříň umístěná opět vodorovně na stole, je nižší než desktop. Základní deska je ve vodorovné poloze a rozšiřující karty se osazují vodorovně do slotů umístěných na tzv. stromečku, který je zasunut kolmo na základní desku. Jedná se o provedení, které poskytuje jen velmi málo prostoru pro další rozšiřování počítače (o další pevné disky, CD-ROM mechaniky apod.)



- **Minitower:** skříň postavená na svislo na stole (popř. pod stolem). Základní deska je umístěna ve svislé poloze a rozšiřující karty se zasouvají vodorovně
- **Tower:** velká skříň umístěna většinou ve svislé poloze pod stolem. Základní deska je opět osazena do svislé polohy a rozšiřující karty se zasouvají vodorovně. Tower je skříň, která poskytuje poměrně velké možnosti dalšího rozšiřování počítače a je vhodná zejména pro počítačové servery.

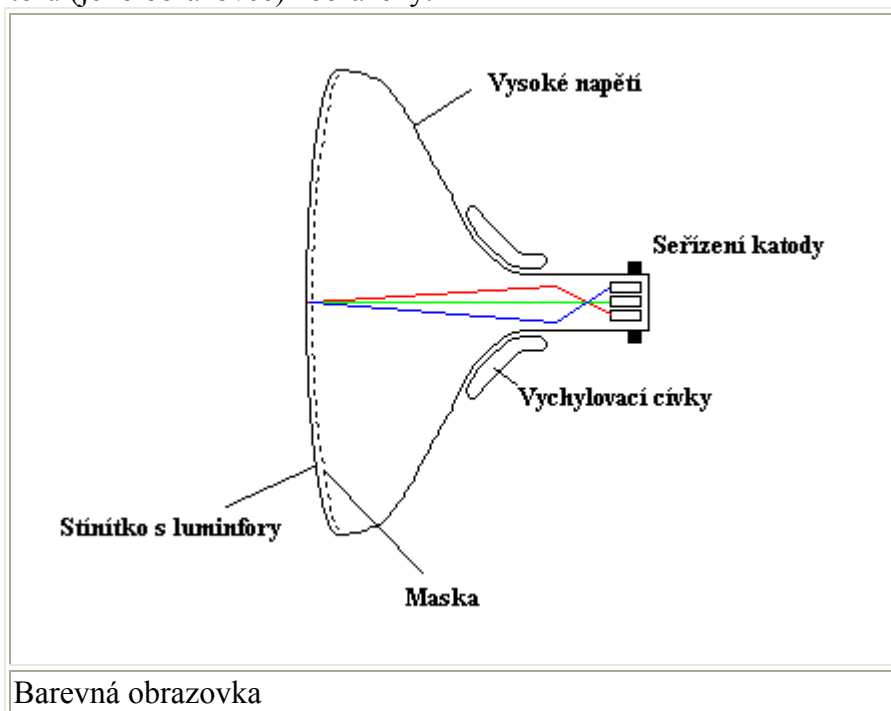


Kromě těchto provedení je možné setkat se i různými dalšími komerčními názvy, jako jsou např.

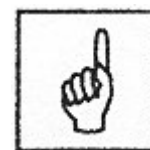
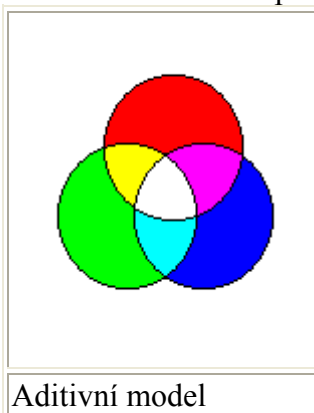
- **miditower:** o něco větší verze minitoweru
- **big tower:** větší verze toweru

8.13 Monitor

Monitory jsou základní výstupní zařízení počítače. Slouží k zobrazování textových i grafických informací. Monitory pracují na principu katodové trubice (CRT - Cathode Ray Tube). Hlavní částí každého monitoru je obrazovka, na jejímž stínítku se zobrazují jednotlivé pixely. Monitor je připojen přímo k videokartě zasílající patřičné informace, které budou na monitoru (jeho obrazovce) zobrazeny.



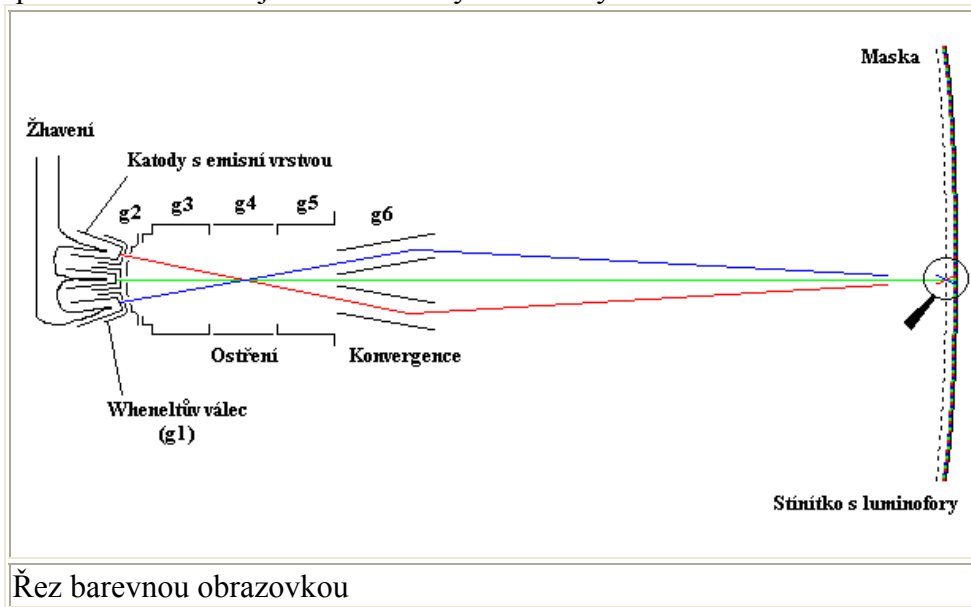
Při práci barevné obrazovky jsou ze tří katod emitovány elektronové svazky, které jsou pomocí jednotlivých mřížek (viz obrázek řez barevnou obrazovkou) taženy až na stínítka obrazovky. Na zadní stěně stínítka obrazovky jsou naneseny vrstvy tzv. **luminoforů** (luminofor = látka přeměňující kinetickou energii na energii světelnou). Tyto luminofory jsou ve třech základních barvách - Red (červená), Green (zelená), Blue (modrá) - pro aditivní model skládání barev. Vlastní elektronové svazky jsou bezbarvé, ale po dopadu na příslušné luminofory dojde k rozsvícení bodu odpovídající barvy.



Protože elektronový svazek je vlastně svazek částic stejného náboje (záporného), mají tyto částice tendenci se odpuzovat a vlivem toho dochází k rozostřování svazku. Proto těsně před stínítkem obrazovky se nachází maska obrazovky. Je to v podstatě mříž, která má za úkol propustit jen úzký svazek

Počítačové systémy, periférie, operační systémy

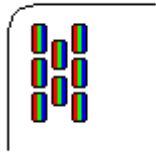
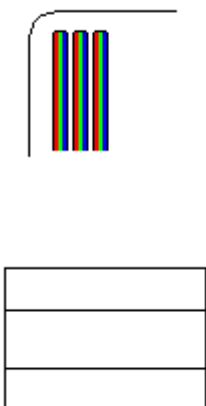
elektronů. Masky obrazovky musí být vyrobeny z materiálu, který co nejméně podléhá tepelné roztažnosti a působení magnetického pole. Oba dva tyto jevy by totiž způsobily, že elektronové svazky nedopadnou přesně na svůj luminofor, což by se projevilo nečistotou barev. Elektronové svazky jsou vychylovány pomocí vychylovacích cívek tak, aby postupně opisovaly zleva doprava a shora dolů jednotlivé řádky obrazovky.



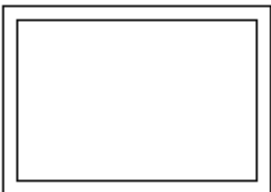
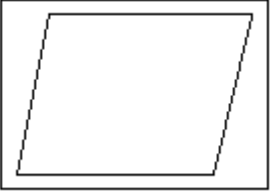
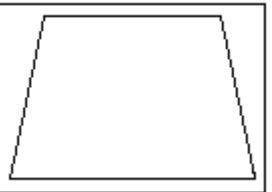
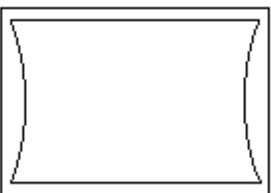
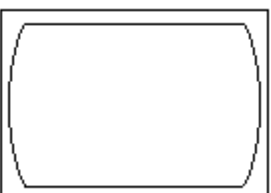

Jednotlivé elektronové svazky jsou emitovány z nepřímo žhavené katody, která má na svém povrchu nanесenu emisní vrstvu. Elektronové svazky pak procházejí tzv. **Whelntovým válcem** (mřížka g1), který má vzhledem ke katodě záporný potenciál. To způsobuje, že elektrony jsou jím odpuzovány a projde jich přes něj jen požadované kvantum. Řízením napětí na Whelntově válci se tedy řídí intenzita jednotlivých elektronových svazků. Po průchodu Whelntovým válcem procházejí elektronové svazky přes jednotlivé mřížky (g2 - g6), které mají naopak vzhledem ke katodě kladný potenciál, díky kterému jsou elektrony přitahovány. Tento kladný potenciál je na mřížce g2 nejnižší, na g3 vyšší a až na g6 nejvyšší. Toto má za úkol elektronové svazky táhnout až na stínítko obrazovky. Speciální funkci zde má mřížka g3 (ostření), která má za úkol zaostřovat elektronové svazky, a mřížka g6 (konvergence), od které se elektronové svazky postupně sbíhají. K jejich setkání dojde u masky obrazovky, kde se prokříží a dopadnou na své luminofory. Podle umístění a tvaru otvorů masky a tím i odpovídajícímu nanесení luminoforů je možné rozlišit tři základní typy barevných obrazovek.

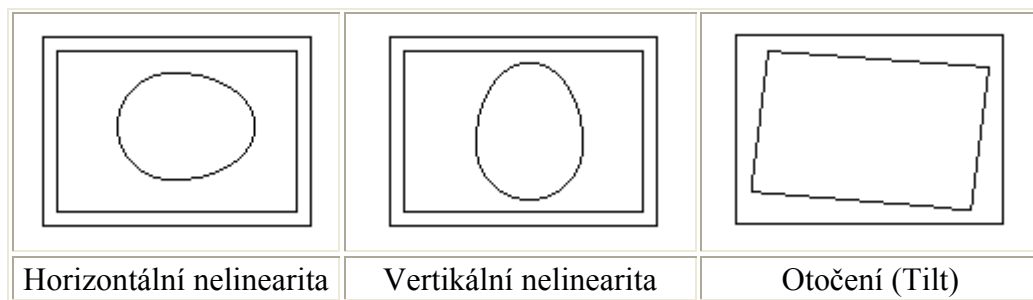
Typ	Maska	Poznámky
Delta		Jednotlivé otvory v masce jsou kruhové a jsou uspořádány do trojúhelníků (velké písmeno delta). Stejným způsobem jsou uspořádány i luminofory na stínítku. Nevýhodou tohoto typu masky (obrazovky) je velká plocha, která je tvořena kovem masky a která způsobuje větší náchylnost k tepelné roztažnosti. Vzhledem k tomuto poskytovaly obrazovky typu Delta poměrně nekvalitní obraz a dnes se již

Počítačové systémy, periférie, operační systémy

		nepoužívají
Inline		Otvory v masce jsou obdélníkového tvaru a jednotlivé luminofory jsou naneseny v řadě vedle sebe. Obrazovka Inline je dnes nejrozšířenějším typem obrazovky
Trinitron		<p>Obrazovky Trinitron jsou propagovány zejména firmou Sony. Jejich luminofory jsou naneseny v řadě vedle sebe podobně jako u obrazovky typu Inline. Vlastní maska je tvořena svislými pásy, které ve vodorovném směru nejsou nikde přerušeny. Toto řešení s sebou nese problém - pásy masky jsou tenké a na celé výšce obrazovky se neudrží. Tento se řeší dvěma způsoby:</p> <ul style="list-style-type: none"> • u monitorů: natažením dvou vodorovných drátů (cca v jedné třetině a dvou třetinách výšky obrazovky) přes obrazovku. Tyto dráty jsou potom bohužel na obrazovce vidět (hlavně na světlém pozadí) • u televizorů: silnějšími pásy masky. Maska pak působí o něco hrubším dojmem.

U konkrétních obrazovek se mohou projevit následující základní poruchy geometrie obrazu.

		
Ideální obraz	Rovnoběžníkovitost (Parallelogram)	Lichoběžníkovitost (Trapezoid)
		
Poduškovitost (Pincushion)	Soudkovitost	Posunutí (Shift)



Některé z těchto poruch bývá možné napravit pomocí korekcí vyvedených na předním panelu monitoru. Pokud tyto korekce monitor nemá nebo jejich rozsah pro nápravu nedostačuje, je nutné provést servisní zásah.

8.13.4 Parametry monitorů

Každý monitor musí být přizpůsoben videokartě (např.: VGA, SVGA), ke které má být připojen. Monitory je možné rozdělit do dvou základních skupin:

- **monochromatické (černobílé):** informace zobrazují pouze v odstínech jedné barvy (obvykle bílá, oranžová, zelená)
- **barevné (color):** umožňují zobrazovat více různých barev současně

Dalším parametrem každého monitoru je velikost jeho obrazovky. Stínítko obrazovky monitoru je tvaru přibližného obdélníku s poměrem stran 4/3. Velikost každé obrazovky je udávána její úhlopříčkou. Úhlopříčka udává její celou velikost a nikoliv velikost její aktivní plochy (plocha, na které je možné zobrazit obraz), která je vždy o něco menší (např. u 17" monitoru je 15,4" až 16,1"). Běžně používané velikosti obrazovek u počítačů jsou:

- **14", 15" :** monitory určené hlavně pro zpracování informací v textovém režimu. V grafickém režimu jsou vhodné pro rozlišení 800 x 600 bodů. Vyšší rozlišení na těchto monitorech bývá hůře čitelné. Ve vyšších rozlišovacích režimech také tyto monitory neposkytují příliš dobré obnovovací frekvence.
- **17" :** monitory určené pro práci s graficky orientovanými programy (tabulkové procesory, textové a grafické editory, prezentační programy). Je možné je použít i pro amatérskou práci s programy CAD/CAM a DTP. 17" monitory jsou vhodné pro rozlišení 1024 x 768 bodů až 1280 x 1024 bodů.
- **19" - 21" :** monitory určené zejména pro profesionální práci s náročnými aplikacemi CAD/CAM a DTP. Jedná se o monitory vhodné pro práci s rozlišením 1280 x 1028 bodů až 1600 x 1200 bodů.

S velikostí obrazu souvisí také parametr označovaný jako **FS (Full Screen)**, který říká, že monitor je schopen využívat celou viditelnou plochu obrazovky. Díky tomu nevznikají na obrazovce nevyužitá černá okraje, do kterých není možné obraz roztáhnout a které byly pozorovatelné zejména u starších 14" monitorů.

Jak bylo uvedeno v předcházející kapitole, jsou při práci monitoru elektronové svazky vychylovány vychylovacími cívkami tak, aby proběhly celou aktivní plochu stínítka obrazovky. Pro kvalitu obrazu je velmi podstatné, jak rychle

Počítačové systémy, periférie, operační systémy

jsou tyto svazky schopné jednotlivé řádky probíhat. U každého monitoru se proto udává:

- **horizontální frekvence (řádkový kmitočet):** měří se v kHz a udává, kolik řádků vykreslí elektronové svazky monitoru za jednu sekundu.
- **vertikální frekvence (obnovovací kmitočet obrazu):** úzce souvisí s horizontální frekvencí, měří se v Hz a udává počet obrazů zobrazených za jednu sekundu.



Obecně platí, že čím vyšší jsou tyto frekvence pro dané rozlišení, tím kvalitnější a stabilnější obraz monitor poskytuje. Při nízkých frekvencích je obraz nestabilní (poblikává) a při delší práci působí únavu zraku. Konkrétní parametry, které jsou ještě vyhovující a které již ne, jsou silně subjektivní a závisí na člověku, který s monitorem pracuje a jak dlouho s ním denně pracuje. Uvádí se, že při rozlišení 1024 x 768 by vertikální frekvence měla být okolo 72 Hz.

V případě požadavků na režimy s vysokým rozlišením je možné se setkat také s tzv. **prokládanými režimy** (interlaced mode). Tento režim použije monitor v okamžiku, kdy není schopen zvládnout vysoké řádkovací frekvence pro režimy s vysokým rozlišením. Aby tento režim mohl monitor zobrazit, obraz se rozloží do dvou dílů. Při prvním průchodu elektronových svazků se vykreslí všechny liché řádky a po návratu paprsku se vykreslí všechny sudé řádky. Tento systém poskytuje lepší obraz, než kdyby monitor zobrazoval s nízkou frekvencí všechny řádky postupně jako u neprokládaného (non-interlaced) režimu, avšak podstatně horší obraz než monitor, který dokáže použít vyšší frekvenci a pomocí ní potom neprokládaně zobrazit celý obraz. Prokládaný režim je charakteristický tím, že obraz se chová mírně neklidně - "mrká" a jsou pozorovatelné slabé tmavé vodorovné pruhy. Při dlouhé práci s takovým monitorem dochází k únavě zraku.

Pokud má monitor zobrazovat různé grafické režimy (s různým rozlišením), je nutné, aby pracoval s různými frekvencemi. Výsledkem je, že při přepnutí grafického režimu může dojít ke změně umístění obrazu (obraz již není přesně vycentrován na střed obrazovky), popř. i ke změnám geometrie obrazu (špatná horizontální a vertikální velikost, poduškovitost apod.). Tyto poruchy lze odstranit pomocí korekcí monitoru, avšak je velmi nepraktické při každém přepnutí režimu měnit nastavení monitoru. Tento problém vyřešily moderní monitory, které používají **digitální ovládání** společně s tzv.

mikroprocesorovým řízením. Tyto monitory jsou vybaveny pamětí, do níž je možné uložit nastavení obrazu pro různé režimy. U starších monitorů, které tuto možnost nemají, je nutné použít program, který bývá dodáván většinou k videokartě a který dovoluje uložení informací o nastavení obrazu pro jednotlivá rozlišení.

Někteří výrobci monitorů používají při výrobě obrazovek tzv. **odzrcadlení**, které omezuje odrazy okolního světa v obrazovce. Tohoto efektu se dosáhne leptáním, mechanickým zdrsněním nebo nanesením speciální vrstvy na stínítko obrazovky. Dalším trendem při výrobě obrazovek jsou obrazovky **flat screen**. Vyznačují se jen velmi malým zakřivením a tím i realističtější zobrazením informací.

Vzhledem k tomu, že monitor má při své práci poměrně vysoký příkon (u 17" monitoru asi 125 W), bývají monitory vybaveny funkcí **green**, která dovoluje

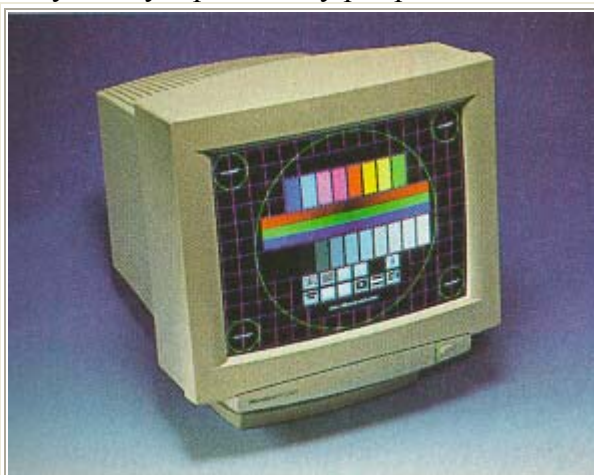
Počítačové systémy, periférie, operační systémy

přepnutí monitoru po určité době od posledního ovládní počítače uživatelem (poslední stisk klávesy, poslední pohyb myši apod.) do pohotovostního režimu. V tomto režimu monitor nic nezobrazuje, jeho příkon je podstatně nižší (8 W - 15 W) a po započítí práce s počítačem se opět automaticky přepne do pracovního režimu.

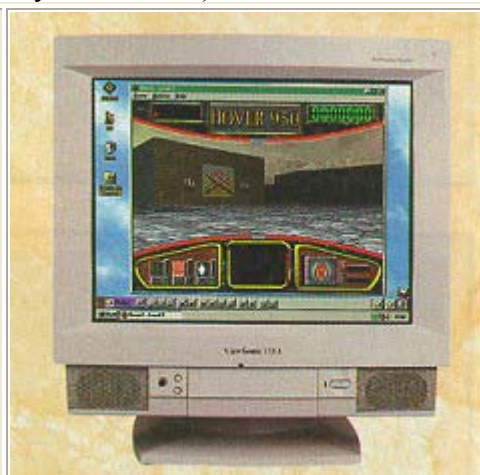
Při práci monitoru může vlivem magnetického pole Země, popř. působením magnetického pole některých předmětů (permanentní magnet, reproduktory apod.) dojít ke zmagnetování masky obrazovky, které se projeví nečistotou barev. Každý monitor provádí proto po svém zapnutí automaticky demagnetizaci masky obrazovky. Novější monitory bývají vybaveny speciálním tlačítkem označovaným degauss (degaussing), které provádí manuální demagnetizaci za chodu monitoru.

U moderních monitorů je také kladen požadavek, aby nedocházelo k nežádoucímu vyzařování škodlivého záření. Jako první vznikla norma **LR** (Low Radiation), která označuje monitory se sníženým vyzařováním. Jako další a přísnější byla později přijata norma **TCO**.

Spolu se stále větším rozmachem nasazování počítačů i v oblastech vzdělávání či zábavy je možné se setkat i s **multimediálními** monitory, které bývají vybaveny reproduktory pro přehrávání zvukových záznamů).



17" Monitor firmy ADI



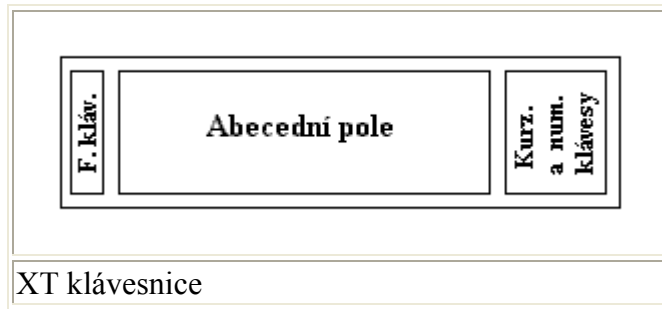
17" Multimediální monitor ViewSonic

8.14 Klávesnice a myš

Klávesnice (keyboard) slouží jako základní vstupní zařízení pro zadávání údajů. Dnes existují dva nejčastěji používané typy klávesnice:

- **PC/XT**: obvykle nazývaná jako "XT klávesnice", má 83 kláves a byla určena k prvním počítačům řady PC a PC/XT. Klávesy této klávesnice lze rozdělit do 3 základních skupin:
 - Abecední pole: obsahuje litery abecedy, číslice, speciální znaky (!,@,#,...) a některé speciální klávesy (SHIFT, CTRL, ALT, ENTER, ...)
 - Funkční klávesy: klávesy označené F1 až F10, jejichž význam závisí na konkrétním programu, se kterým uživatel pracuje
 - Kurzorové a numerické klávesy: obsahují klávesy pro číslice a ovládní kurzoru

Počítačové systémy, periférie, operační systémy



- **PC/AT:** obvykle nazývaná jako "AT klávesnice", obsahuje 101 (US standard) nebo 102 (European standard) kláves. Tyto klávesy lze rozdělit do 4 bloků:
 - Abecední pole: podobně jako u XT klávesnice
 - Funkční klávesy: obsahuje klávesy F1 - F12, jejichž význam je opět podobný jako u XT klávesnice.
 - Kurzorové klávesy: klávesy pro ovládání kurzoru
 - Kurzorové a numerické klávesy: podobně jako u XT klávesnice



- S příchodem operačního systému MS Windows 95 byla AT klávesnice doplněna o speciální klávesy pro ovládání tohoto systému (klávesa pro vyvolání Start menu a pro vyvolání kontextového menu) a nese označení Win95 Natural. Je možné se setkat i se speciálními ergonomickými klávesnicemi majícími speciální tvar, který má zaručit, že ruce uživatele budou při práci v co možná nejpřirozenější poloze.



Win95 Natural klávesnice



Ergonomická klávesnice

Rozdíl mezi klávesnicí XT a AT je hlavně v tom, že XT klávesnice má mikroprocesor klávesnice zabudovaný přímo v sobě, zatímco AT předpokládá

Počítačové systémy, periférie, operační systémy

procesor pro klávesnici na základní desce počítače. Tyto klávesnice jsou tedy mezi sebou nekompatibilní, takže není možné XT klávesnici použít u počítače AT. Naopak je pravdou, že většina AT klávesnic je vybavena přepínačem, který dovoluje AT klávesnici přepnout do režimu XT a používat ji tak u počítače XT.

Dále podle realizace funkce jednotlivých kláves je možné rozdělit klávesnice na:

- **pracující na principu spínačů:** používá pro každou klávesu mikrosvítač
- **kapacitní:** stisknutí klávesy vyvolá úhoz na kapacitní modul, jenž vysílá patřičné signály, které jsou potom interpretovány procesorem 8048 umístěným přímo v klávesnici, a jejich kódy jsou pak vysílány do počítače.

Klávesnice bývá k počítači připojena většinou 5 kolíkovým konektorem DIN, popř. pomocí PS/2 konektoru.



Konektor DIN

8.15 Myš

Myš (mouse) je zařízení, které umožňuje přenášet pohyb ruky po vodorovné podložce na obrazovku počítače. Slouží většinou jako ukazovátka při práci s mnoha dnešními programy. Dnes rozlišujeme dva základní typy myši:

- Microsoft Mouse (má dvě tlačítka)
- PC Mouse (má tři tlačítka)

Microsoft Mouse i PC mouse se připojují k počítači většinou přes sériový port, popř. přes PS/2 port a každá z nich komunikuje s počítačem pomocí jiného protokolu. Z toho vyplývá, že dvoutlačítková a třítlačítková myš jsou vzájemně nekompatibilní. Podobně jako u klávesnic i u třítlačítkových myši často existuje přepínač, pomocí kterého je možné třítlačítkovou myš přepnout do dvoutlačítkového režimu.

Většina myši pracuje tak, že ve své spodní části obsahuje kuličku, která se při pohybu po podložce otáčí a toto otáčení je přenášeno na dva otočné válečky (jeden pro horizontální a jeden pro vertikální směr). Podle jejich otáčení jsou

Počítačové systémy, periférie, operační systémy

vysílány informace o pohybu myši do počítače, které způsobují patřičný pohyb kurzoru myši po obrazovce. Kromě toho je možné vidět i myši, které využívají speciální podložku (obsahující jemnou mřížku) a které ve spodní části mají místo kuličky LED diodu a speciální čidlo, dovolující rozeznat pohyb myši po této mřížce.

8.16 PCMCIA

Sdružení PCMCIA (Personal Computer Memory Cards International Association) bylo ustaveno v roce 1989. Tento standard brzy přijalo asi 200 firem a dnes k této nevýdělečné organizaci patří asi 600 společností. Původně se jednalo o standard, který byl určen pro rozšiřující paměťové karty a jejich sloty pro přenosné počítače. Dnes se jedná o rozhraní s univerzálním použitím, ke kterému je možné připojit celé spektrum různých zařízení.

Standard PCMCIA není závislý na hardwarové platformě a operačním systému. To znamená, že je možné se s ním setkat nejen na počítačích PC, Apple Macintosh, ale často i v "nepočítačových" zařízeních. Jeho hlavní těžiště použití je u přenosných počítačů (notebooků, laptopů, palmtopů atd.), které jsou PCMCIA osazeny až z 90%

Rozhraní PCMCIA vytváří sběrnici, na kterou je možné připojovat PCMCIA karty. Tato sběrnice je kompatibilní se sběrnici ISA, EISA, MCA, VL-bus i PCI, takže není žádný problém, aby počítač byl vybaven např. PCI a PCMCIA sběrnici zároveň.

Hlavní výhodou PCMCIA je, že se jedná o rychlý a efektivní systém pro připojování různých periférií bez otevírání počítače, má jednoduchou instalaci dovolující automatickou konfiguraci. Není tedy nutné provádět manuální nastavování pomocí propojek (jumperů). PCMCIA je navrženo tak, aby umožňovalo tzv. "**hot swap**", tj. kartu je možné vyměnit za chodu počítače (není nutné počítač vypínat a po jeho zapnutí znovu zavádět operační systém). PCMCIA karty mají všechny stejnou velikost 85,6 x 54 mm (šířka x délka) a liší se pouze svou tloušťkou:

- **typ I:** jeho tloušťka je 3,3 mm a jedná se nejstarší typ používaný zejména pro paměťové karty Flash, SRAM
- **typ II:** o tloušťce 5 mm, který je dnes nejpoužívanější. K dispozici je řada různých zařízení:
 - faxmodemy
 - síťové karty
 - SCSI karty
 - zvukové karty
 - disky





PCMCIA síťová karta

PCMCIA SCSI rozhraní

- **typ III:** tloušťka 10,5 mm. Jedná se zatím o poslední mezinárodně přijatou specifikaci používanou hlavně pro pevné disky.
- **typ IV:** standard, o jehož zavedení se pokouší firma Toshiba. Jeho tloušťka je 16 mm.
- **Extended Cards:** rozšířené (prodloužené) karty: asi o 50 mm delší, určené pro speciální aplikace.

Jednotlivé typy jsou vzájemně kompatibilní: kartu typu I je možné použít ve slotu typu II i III a podobně. Obráceně to z mechanických důvodů není možné.

8.17 Externí paměťová média

Data a programy, se kterými uživatel na počítači pracuje, je nezbytné uchovávat na nějakých paměťových médiích. Standardně je proto každý počítač vybaven mechanikou pružných disků a pevným diskem (kromě prvních PC a PC/XT). Disketová mechanika slouží k záznamu dat na pružný (floppy) disk. Jedná se o přenosné médium, které má však pro dnešní účely poměrně malou kapacitu (disketa 3¹/₂ má kapacitu 1,44 MB) a poskytuje poměrně nízkou přenosovou rychlost.

Naopak pevný disk má vysokou kapacitu, poskytuje vysokou přenosovou rychlost, ale jde o nepřenositelné médium, které je pevně osazeno v počítači. Pevný disk tedy slouží k ukládání dat a programů, se kterými na počítači momentálně pracujeme, a jen výjimečně slouží k přenosu dat mezi dvěma počítači.

Protože, jak bylo uvedeno výše, disketová mechanika pro přenos většího objemu dat není vhodná, vzniká poměrně velké množství jiných paměťových médií určených zejména jako přenosná média s větší kapacitou a vyšší přenosovou rychlostí, než má floppy disk.

Mezi základní parametry každého takového média patří:

Parametr	Vysvětlení	Rozsah
Kapacita	Maximální množství dat, které je možné na dané médium zaznamenat	1 MB - 10 GB
Přenosová rychlost	Množství dat, které je možné z média přenést do počítače za jednotku času	10 MB/min - 1 MB/s
Přístup k	Způsob, kterým je možné přistupovat k	Sekvenční, přímý

datům	datům	
Připojení k počítači	Rozhraní, řadič, pomocí kterého je možné čtecí (zapisovací) mechaniku pro dané médium připojit k počítači	EIDE, SCSI, řadič pružných disků, paralelní port
Princip záznamu	Způsob, kterým se jednotlivé bity na médium zaznamenávají	magnetický, optický, magnetický s optickým naváděním hlav
Provedení čtecí jednotky	Čtecí jednotka může být umístěna uvnitř skříně počítače (interní), nebo naopak je umístěna ve vlastní skříně vně počítače (externí).	Externí, interní
Pořizovací cena	Cena čtecí (zapisovací) mechaniky pro dané médium	1000 Kč - 10000 Kč
Cena za bit	Poměr ceny za jedno médium ku kapacitě média	
Spolehlivost	Střední doba mezi poruchami	roky

8.18 CD mechaniky

8.18.4 CD-ROM

Médium CD-ROM vznikalo původně jako audio nosič a jeho autory byly firmy Philips a Sony. Jedná se o médium, které je určeno pouze ke čtení informací. Dovoluje uložení až 650 MB programů a dat.

Na rozdíl od dříve uvedených diskových zařízení (pružné disky, pevné disky, ZIP disky, Magnetooptické disky apod.) nejsou data ukládána do soustředných kružnic, ale do jedné dlouhé spirály podobně jako na gramofonové desce.

Spirála začíná u středu média a rozvíjí se postupně až k jeho okraji. Záznam (spirála dat) je pouze na spodní straně disku, tj. záznam na CD-ROM disku je jednostranný. Délka celé spirály je zhruba 6 km a hustota dat v ní uložených je konstantní. Podle rychlosti, kterou je CD-ROM mechanika schopna číst tato data, se mechaniky rozlišují na:

- **single speed:** rychlost čtení dat je 150 kB/s, dostačuje pouze pro přenos souborů
- **double speed:** data je schopna číst rychlostí 300 kB/s, což poskytuje plynulou rychlost pro práci s datovými soubory. Nedostačuje pro přehrávání videa
- **triple speed:** dovoluje číst data rychlostí až 450 kB/s
- **quadruple speed:** mechanika dovolující čtení dat rychlostí 600 kB/s
- **6x:** rychlost čtení: 900 kB/s
- **8x:** rychlost čtení: 1200 kB/s
- **12x:** rychlost čtení: 1800 kB/s
- **16x:** rychlost čtení: 2400 kB/s
- **24x:** rychlost čtení: 3600 kB/s

Rychlost čtení spirály je v single speed mechanice asi 1,3 m/s. Rychlost otáčení CD-ROM disku není konstantní, ale je kontinuálně přizpůsobována podle toho, zda se čtení provádí blíže kraji nebo středu disku. U středu disku je rychlost otáčení vyšší (asi 500 otáček za minutu) a u kraje naopak nižší (asi 200 otáček za minutu). Toto přizpůsobování otáček disku zaručuje, že data jsou čtena ze spirály konstantní rychlostí.

Přístupová doba u datových CD-ROM disků je potom závislá na čase nutném k regulaci otáček. Je tedy velmi nevhodné číst data uložená v různých částech disku, protože je neustále nutné přizpůsobovat rychlost otáčení. Tento problém plně neodstraňují ani mechaniky s vyšší přístupovou rychlostí, i když samozřejmě mechaniky s vyšší rychlostí čtení mají i nižší přístupovou dobu. Přístupová doba se u CD-ROM mechanik pohybuje od 100 ms do 300 ms. Protože šířka stopy spirály je velmi malá, data jsou uložena s poměrně velkou hustotou a vlastní CD-ROM nosič není ničím chráněn, je velká pravděpodobnost, že i při běžné manipulaci s CD-ROM diskem může dojít ke špatnému přečtení některých uložených bitů. Proto informace uložené na médiu CD-ROM jsou silně redundantní (nadbytečné) a mechanika má obvody realizující na základě těchto nadbytečných informací poměrně složité algoritmy pro korekturu chyb vzniklých při čtení.

CD-ROM mechaniky se k počítači připojují pomocí:

- EIDE rozhraní (**ATAPI**: rozšíření normy ATA o příkazy pro práci s CD-ROM, streamery a dalšími perifériemi)
- SCSI rozhraní
- zvukové karty obsahující rozhraní pro CD-ROM
- vlastního řadiče



Čtecí mechanika disku CD-ROM

8.18.5 Mechaniky CD-R

Mechaniky CD-R (Compact Disk - Recordable) jsou zařízení, jež dovolují provedení záznamu na disk CD-R, který je potom čitelný v běžné CD-ROM mechanice.

Proces zaznamenávání dat na CD-R disk je velmi náročný na kontinuální přísun dat ze zařízení, z něhož záznam (většinou pevný disk) do mechaniky CD-R provádíme. V případě, že CD-R mechanika nedostane požadovaná data

Počítačové systémy, periférie, operační systémy

včas, dojde k přerušení záznamu na CD-R a tím ke zničení celého média. Proto ve většině případů bývá mechanika CD-R připojována pomocí SCSI rozhraní a doporučuje se, aby i pevný disk, ze kterého se provádí záznam na CD-R, byl připojen na SCSI, které dovoluje vyšší a plynulejší přenos dat. Rovněž se doporučuje, aby po dobu, kdy je prováděn záznam, nebyl na počítači spuštěn žádný jiný program, který by mohl vyvolat jeho zátěž a tím i přerušení přísunu dat do CD-R mechaniky. Rychlost záznamu na CD-R je závislá na mnoha faktorech (rychlost počítače, rychlost pevného disku, ze kterého se záznam provádí, kapacita operační paměti apod.). V současnosti se pohybuje rychlost záznamu od 4x do 16x. Při použití rychlosti záznamu 1x trvá záznam plného CD-R média zhruba 75 min.



Mechanika pro zápis disku CD-R

8.18.6 Mechaniky CD-RW

Mechaniky CD-RW (Compact Disk - Rewritable) jsou určeny k záznamu nejen na disky CD-R, ale také pro záznam na speciální disky CD-RW. CD-RW disky dovolují na rozdíl od CD-R disků, aby záznam byl přemazán a proveden znovu. Přemazání však nemůže být prováděno libovolně, jako např. na pevném disku, ale pouze na celém disku.

8.19 Typy externích paměťových médií

8.19.4 Páskové paměti

Páskové paměti jsou typickým sekvenčním zařízením, to znamená, že pokud je potřeba zpřístupnit libovolnou informaci na pásce, je nutné, aby nejdříve byly přečteny všechny informace předcházející. Mezi první páskové paměti patří devítistopá páska o šířce $\frac{1}{2}$ ". Hustota záznamu na těchto páskách dosahovala až 6250 bpi (bits per inch = bitů na palec). Tyto páskové paměti se používaly zejména u velkých sálových počítačů a vyžadovaly poměrně náročnou obsluhu, protože páska byla navinuta pouze na cívce (nikoliv umístěna v kazetě) a tudíž se musela pracně zavádět do čtecího zařízení.

Páskové paměti jsou vhodné zejména pro zálohování velkého objemu dat a jeho případné obnovy. Jsou naprosto nevhodné pro časté zpřístupňování určitých částí dat. Toto je dáno jejich sekvenčním přístupem k datům, který



Počítačové systémy, periférie, operační systémy

může způsobit, že přístupová doba k datům uloženým na konci pásky může dosáhnout až několika hodin.

Připojování pásky se provádělo přes rozhraní SCSI, záznam byl prováděn magneticky a životnost pásky byla odhadována na 25 let.

Kazety 3480, 3490E

Jedná se o kazety s magnetickou páskou, které byly vyvinuty firmou IBM pro velká výpočetní střediska. Na rozdíl od svého předchůdce se již jedná o pásku umístěnou v kazetě, což usnadňuje manipulaci.

Další výhodou těchto pásek je vyšší přenosová rychlost a kratší doba převíjení. Kazety 3480 a 3490E jsou opět vhodné zejména pro zálohování velkých objemů dat, protože podobně jako $\frac{1}{2}$ " pásky jsou i tyto kazety sekvenční média. Později vznikla k těmto kazetám ještě automatická zařízení na manipulaci s archivem těchto kazet (jukebox).

Streamer

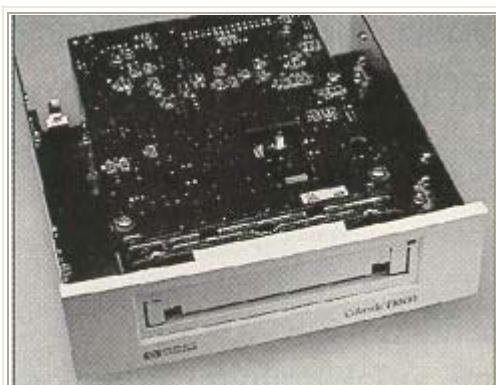
Streamer je páskovým médiem s podélným proudovým záznamem, které není již určeno výhradně pro velké sálkové počítače, jako tomu bylo u předešlých médií, ale i pro malé osobní počítače.

Čtecí (zpisovací) mechaniky pro streamery se vyrábějí jak v interním, tak externím provedení. Páska streameru je uložena v kazetě, se kterou potom mechanika pracuje. Zaznamenávaná data se nezapisují po blocích, ale jsou zapisována jako celistvý proud dat. Vlastní záznam je prováděn podélně (podobně jako u audio kazety).

Oproti předešlým páskovým pamětem a kazetám poskytují streamery vyšší přenosovou rychlost (cca 10-15 MB/min) a mají kapacitu 60 MB, 120 MB, 250 MB, 500 MB, 1,2 GB, 2,5 GB a více. Jejich hlavní využití spočívá podobně jako u předchozích médií v zálohování velkých objemů dat.

Připojení k počítači se provádí prostřednictvím:

- řadiče pružných disků
- paralelního portu
- SCSI rozhraní



Streamer

Kazety 8 mm "Helical" (Exabyte)

Počítačové systémy, periférie, operační systémy

Tyto kazety, vyrobené firmou Exabyte, obsahují opět magnetický pásek, tentokrát o šířce 8 mm, na který se provádí šikmý záznam dat (podobně jako na video kazetě). Kapacita této kazety se pohybuje v závislosti na délce pásku od 1 GB až do 20 GB. Rychlost přenosu dat je zhruba 20 MB/min.

Vzhledem k uvedeným parametrům a skutečnosti, že se jedná opět o sekvenční médium, jsou kazety Exabyte vhodné pro zálohování velkých objemů dat.

Připojení k počítači se provádí většinou přes rozhraní SCSI

Kazety 4 mm DAT

Kazety DAT (Digital Audio Type) byly vyrobeny Hewlett Packard. Jedná se o podobné zařízení jako jsou kazety Exabyte, které pracuje na podobném principu (šikmý záznam dat na magnetickou 4 mm pásku).

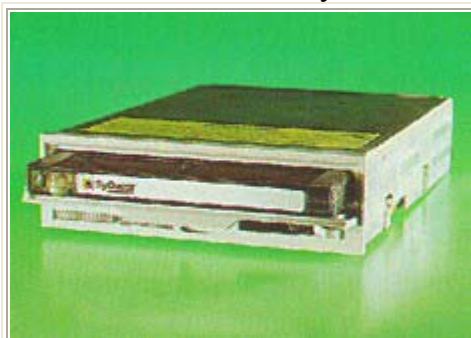
Kapacita těchto kazet se pohybuje v rozmezí 1 GB - 8 GB a přenosová rychlost je 22 MB/min. Připojení je podobně jako u Exabyte prováděno většinou přes rozhraní SCSI.

8.19.5 SyQuest disk

SyQuest disk byl vyroben firmou SyQuest a jedná se o výměnný kotouč pevného disku o průměru 3¹/₂" umístěný v plastové kazetě. Tvoří přechod mezi pružnými a pevnými disky. Jeho kapacita je 105 MB, 130 MB a 270 MB. Připojení k počítači se provádí prostřednictvím:

- SCSI rozhraní
- IDE rozhraní
- paralelního portu

Nevýhodou SyQuest disků je jejich častá vzájemná nekompatibilita. Stává se, že disk zapsaný v jedné mechanice není čitelný v mechanice jiné.



SyQuest Disk

8.19.6 Bernoulliho disk

Je vyroben firmou Iomega a jde o pružný kotouč otáčející se v proudu vzduchu, který přitlačuje (podle Bernoulliho jevu) povrch média k hlavičce. Bernoulliho disky se vyrábějí ve formátu 5¹/₄" a jejich kapacita se pohybuje od 20 MB do 200 MB. Připojení k počítači je provedeno pomocí paralelního portu nebo SCSI rozhraní. Mezi výhody Bernoulliho disků patří poměrně vysoká odolnost média proti přetížení (pád, náraz apod.).

Počítačové systémy, periférie, operační systémy

Jedná se o přenosné zařízení s přímým přístupem k datům, které poskytuje vyšší přenosovou rychlost a vyšší kapacitu než disketa.

8.19.7 Floptical disk

Floptical disk (Floppy Optical) je pružný disk o formátu $3\frac{1}{2}$ " , na který se provádí magnetický záznam. Při tomto záznamu se používá optické navádění čtecích (zapisovacích) hlav na značky vytvořené pevně při výrobě diskety. Floptical disk má vylepšený povrch a dovoluje uložení až 21 MB dat. Výhodou těchto disků je také to, že v mechanice pro floptical disk je možné používat i běžné $3\frac{1}{2}$ " diskety. Na tyto diskety sice není možné uložit výše zmíněných 21 MB (pouze 1.44 MB), ale uživatel tak nemusí mít ke své floptical mechanice ještě standardní mechaniku pro $3\frac{1}{2}$ " diskety.

8.19.8 Flash disk

Flash disk jsou zařízení vyráběná většinou jako PCMCIA karty a simulují disk pomocí Flash paměti. Jejich kapacita se pohybuje v rozmezí 5 MB - 20 MB. Jejich hlavní nevýhodou je jejich poměrně vysoká cena.

8.19.9 Magnetooptické disky

Magnetooptické disky jsou média, u nichž se záznam provádí zaměřením laserového paprsku za současného působení magnetického pole. Záznam na tento disk probíhá ve dvou fázích:

- nejprve se smaže místo, na které se bude záznam provádět (zapsání samých nul)
- v další otáčce (po smazání) se na příslušná místa zapíše jedničky

Fyzikální princip záznamu je následující:

- laserový paprsek zahřeje bod na citlivé vrstvě nad Curiovu teplotu (teplota stanovená pro každý materiál, při níž stačí pouze malá intenzita magnetického pole ke změně jeho magnetických vlastností).
- magnetickým polem příslušné orientace se změní zmagnetování bodu
- po ochlazení magnetizace zůstane

Fyzikální princip čtení:

- čtení se provádí laserovým paprskem nižší intenzity
- sleduje se stáčení paprsku Kerrovým efektem (elektrooptický dvojlom), který je závislý na magnetické orientaci bodu

Protože záznam na médium je podmíněn zahřátím příslušného bodu disku, není nebezpečí, že by mohlo dojít k nechtěnému smazání dat vlivem magnetického pole, např. silného permanentního magnetu, reproduktoru apod. Díky těmto vlastnostem je pro magnetooptické disky charakteristická vysoká životnost (desítky let). Vyrábějí se ve formátu $3\frac{1}{2}$ " i $5\frac{1}{4}$ " a poskytují kapacitu od stovek MB až po jednotky GB.

Připojení magnetooptických disků se provádí přes rozhraní SCSI.

Počítačové systémy, periférie, operační systémy

8.19.10 ZIP disky

ZIP disky jsou média vyrobená firmou Iomega a jedná se disk o průměru 3¹/₂", na který je možné uložit 100 MB dat. Princip práce ZIP disku je podobný jako u disketové mechaniky. Provádí se na magnetickou vrstvu pomocí čtecích (zapisovacích) hlav, které při práci přímo dosedají na povrch média. Mechaniky pro ZIP disky se vyrábějí v interním i externím provedení.

- Interní mechaniky se připojují přes
 - EIDE rozhraní
 - SCSI rozhraní
- Externí disky se připojují přes
 - SCSI rozhraní
 - Paralelní port



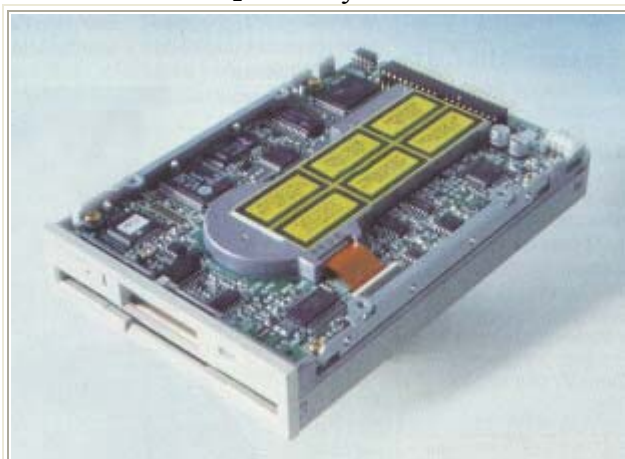
Interní ZIP disk



Externí ZIP disk

8.19.11 Disky LS120

Disky LS120 jsou svými vlastnostmi velmi podobné diskům ZIP. Dovolují uložení až 120 MB dat a na rozdíl od ZIP disků je možné v mechanice pro disky LS120 používat i běžné 3¹/₂" diskety.



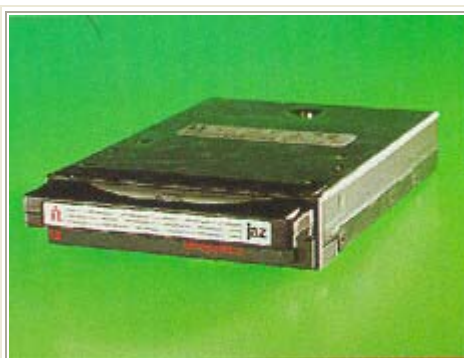
Disk LS 120

8.19.12 JAZZ disky

Počítačové systémy, periférie, operační systémy

Disky JAZZ jsou média, která pracují na podobném principu jako pevný disk. Jsou vyráběny firmou Iomega a dovolují uložení 1 GB - 2 GB dat. Záznam je prováděn do magnetické vrstvy pomocí hlav, které plovou na tenké vzduchové vrstvě nad vlastním médiem. JAZZ disky se vyrábějí ve formátu 3 1/2", a to jak v interním, tak i v externím provedení. Připojení se provádí:

- u interního provedení přes SCSI rozhraní
- u externího provedení také přes SCSI rozhraní nebo přes paralelní port



Jazz disk

8.19.13 SyJet disky

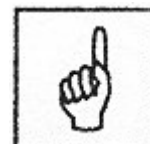
SyJet disky jsou v podstatě reakcí firmy SyQuest na disky JAZZ. Jde o zařízení pracující na stejném principu jako JAZZ disk, jehož kapacita je 1,5 GB. Připojování, formát a provedení je prakticky totožné jako JAZZ disků.



SyJet Disk

8.20 Tiskárna

Tiskárny jsou výstupní zařízení sloužící pro výstup údajů z počítače. Prostřednictvím tiskárny je možné data uchovaná doposud v elektronické formě vytisknout (nejčastěji na papír).
Základní parametry tiskárny jsou:



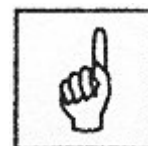
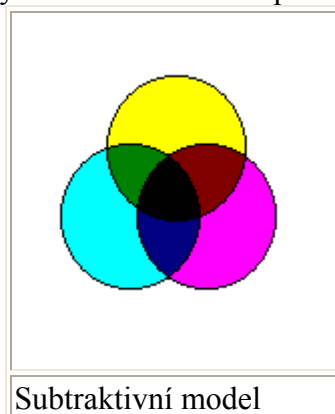
Počítačové systémy, periférie, operační systémy

Parametr	Vysvětlení	Rozsah
Typ tisku	Způsob použitý k tisku jednotlivých znaků	Jehličková, tepelná, inkoustová, laserová tiskárna
Rychlost tisku	Počet znaků vytištěných za jednotku času	řádově 100 zn/s - 10 stránek/min
Kvalita tisku	Počet bodů, které je tiskárna schopna vytisknout na jeden palec (bpi - bits per inch)	120 - 1200 bpi
Barevnost	Schopnost tisknout pouze černobíle nebo i barevně.	Černobílé, barevné
Pořizovací náklady	Cena, za kterou je možné tiskárnu pořídit	řádově 1000 Kč - 100000 Kč
Cena za vytištěnou stránkou	Cena, kterou uživatel zaplatí za vytištěnou stránku. Je dána cenou listu požadovaného papíru, cenou a životností tiskové náplně (páska, inkoust, toner)	

V případě barevného tisku je nutné pracovat se **subtraktivním modelem** mísení barev (na rozdíl od obrazovky, kde pracujeme s aditivním mísením). Tento model označovaný také jako CMYK používá pro tisk tři až čtyř základních barev, jejichž mísením se dostávají barvy ostatní:

- Cyan - indigově modrá
- Magenta - fialová
- Yellow - žlutá
- Black - černá

V případě levnějších tiskáren bývá vynechána černá barva, která se nahrazuje smísením tří zbývajících barev. Tyto barvy však neposkytují čistě černou barvu, a proto jejich tisk bývá co do barevného podání méně kvalitní.



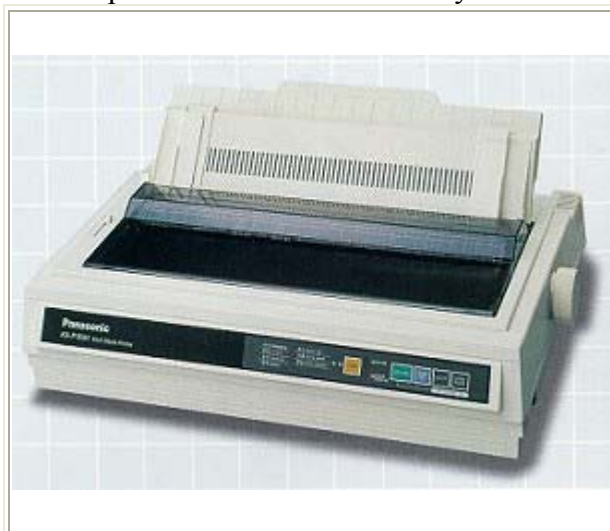
Tiskárna se k počítači připojuje většinou přes paralelní port pomocí rozhraní **Centronics**, popř. **Bitronics**. Některé tiskárny dovolují i připojení přes sériový port, ale tento způsob propojení počítače s tiskárnou bývá méně častý.

8.20.1 Jehličková tiskárna

U jehličkové tiskárny se k tisku využívá tisková hlava, která obsahuje sadu pod sebou umístěných jehliček. V závislosti na počtu těchto jehliček se dále jehličkové tiskárny rozlišují na:

- **1 jehličkové a 2 jehličkové:** technické rarity vyráběné svého času v ČSSR
- **7 jehličkové:** poskytují tisk s velmi nízkou kvalitou a jsou používány pouze ve speciálních případech, jako jsou např. pokladny v prodejně, kde na kvalitu tisku nejsou kladeny téměř žádné nároky.
- **9 jehličkové:** dovolují tisk v tzv. **NLQ** (**N**ear **L**etter **Q**uality - blízký dopisní kvalitě) režimu. Tento režim svou kvalitou tisku odpovídá přibližně kvalitě elektrického psacího stroje. Tyto tiskárny jsou vhodné pro tisky výpisů programů a pro tisk dokumentů, na jejichž kvalitě příliš nezáleží.
- **24 jehličkové:** umožňují kvalitnější tzv. **LQ** (**L**etter **Q**uality - dopisní kvalita) režim tisku. Oproti 9 jehličkovým tiskárnám poskytují také větší rychlost tisku. Jsou používány opět zejména pro dokumenty, na jejichž kvalitu jsou kladeny nižší nároky.

Jednotlivé jehličky jsou připojeny k elektromagnetům, které je při práci (tisku) vystřelují proti barvicí pásce. Tato barvicí páska dopadne v daném bodě pak na papír, kde způsobí malý barevný bod. Obecně platí, že jehličkové tiskárny jsou poměrně hlučná zařízení, která nejsou vhodná pro tisk grafických dokumentů a neposkytují příliš velkou rychlost tisku (řádově 100 zn/s). Barevný tisk je u jehličkových tiskáren možný použitím vícebarevné pásky. Vzhledem k výše popsanému principu tisku nevyžadují tyto tiskárny žádný speciální papír. Jejich pořizovací cena i cena za vytištěnou stránku jsou poměrně nízké.



Jehličková tiskárna Panasonic



Barevná jehličková tiskárna Panasonic

8.20.2 Tepelná tiskárna

Tepelné tiskárny tisknou na podobném principu jako tiskárny jehličkové. Jsou opět vybaveny tiskovou hlavou, která obsahuje sadu jehliček připevněných k

Počítačové systémy, periférie, operační systémy

elektromagnetům. Jednotlivé jehličky jsou však na rozdíl od jehličkové tiskárny zahřáty na vyšší teplotu, která poté, co se jehlička přiblíží ke speciálnímu papíru citlivému na teplo, způsobí jeho zbarvení.

Tepelné tiskárny poskytují podobnou kvalitu a rychlost tisku jako tiskárny jehličkové. Jejich velkou nevýhodou je nutnost použít speciální papír a tím i vyšší cena za vytištěnou stránku. V dnešní době se tyto tiskárny používají jen výjimečně.

8.20.3 Inkoustová tiskárna

Inkoustová tiskárna tiskne pomocí inkoustu, který je stříkán na papír. Inkoust bývá umístěn v malé nádržce, jež se pohybuje společně s tiskovou hlavou. Kvalita tisku inkoustové tiskárny je silně závislá na použitém papíru. V případě kvalitního papíru je možné dosáhnout velmi kvalitního tisku (za cenu vyšších nákladů na tuto vytištěnou stránku). Barevný tisk bývá prováděn pomocí různobarevných inkoustů.

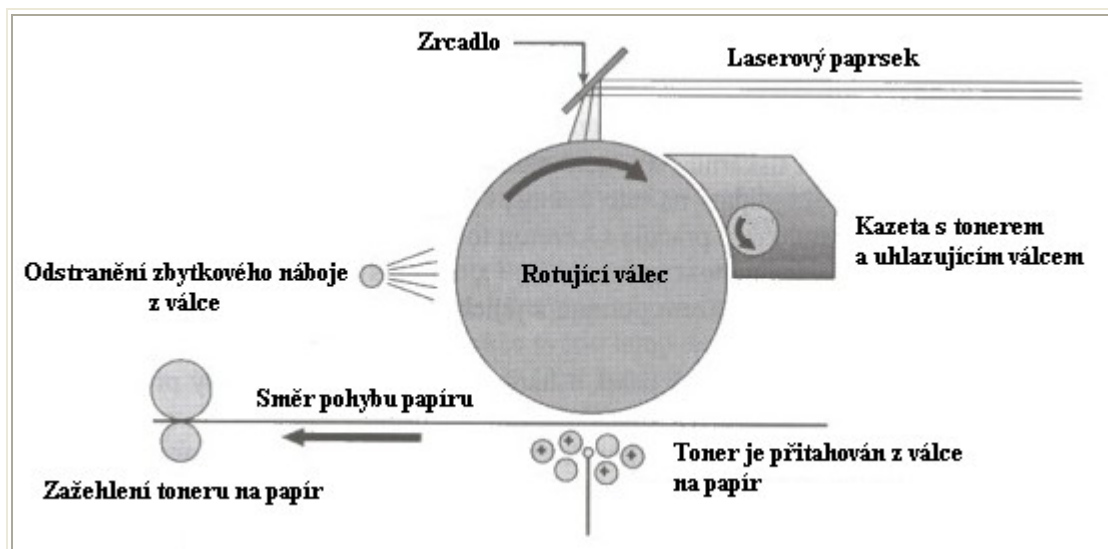
Inkoustové tiskárny poskytují vyšší rychlost tisku než tiskárny jehličkové. Jedná se o zařízení vhodná pro tisk běžných textových i grafických dokumentů. Jejich pořizovací cena dnes již není příliš vysoká. Jejich nevýhodou je však poměrně vysoká cena za vytištěnou stránku, která je dána cenou inkoustu a vyšší cenou kvalitního papíru.



Inkoustová tiskárna firmy Hewlett Packard

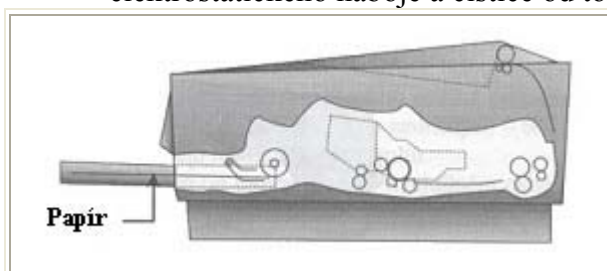
8.20.4 Laserová tiskárna

Laserová tiskárna je zařízení určené zejména pro profesionální použití. Poskytuje velmi vysokou kvalitu (300 dpi - 1200 dpi) i rychlost tisku (desítky stránek za minutu). Jedná se o poměrně drahé zařízení - ale cena za vytištěnou stránku bývá většinou nižší než u inkoustových tiskáren.

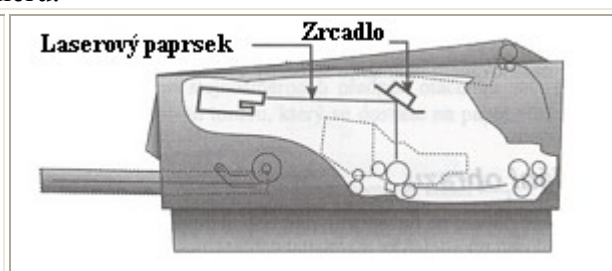


Princip tisku laserové tiskárny

Při tisku laserové tiskárny jsou nejdříve znaková data zasílaná počítačem převáděna řadičem na videodata. Ta jsou zasílána na vstup polovodičového laseru. Polovodičový laser vysílá laserový paprsek, který je vychylován soustavou zrcadel na rotující válec. V místech, kam tento paprsek na válec dopadne, dojde k jeho nabití statickou elektřinou na potenciál řádově 1000 V. Rotující válec dále prochází kolem kazety s barvicím práškem (tonerem), který je vlivem statické elektřiny přitážen k nabitým místům na povrchu válce. Papír, který vstoupí do tiskárny ze vstupního podavače, je nejdříve nabit statickou elektřinou na potenciál vyšší než jsou nabitá místa na válci (cca 2000 V). V okamžiku, kdy tento papír prochází kolem válce, dojde k přitahování toneru z nabitých míst válce na papír. Toner je do papíru dále zažehlen a celý papír je na závěr zbaven elektrostatického náboje a umístěn na výstupní zásobník. Rotující válec po otištění na papír prochází dále kolem sběrače elektrostatického náboje a čističe od toneru.



Řez laserovou tiskárnou 1



Řez laserovou tiskárnou 2

Barevný tisk je u laserových tiskáren možné docílit použitím různobarevných tonerů.



Laserová tiskárna firmy Hewlet Packard

8.21 Scanner

Scanner je zařízení, které slouží ke snímání předlohy do počítače. Pracuje na principu **digitalizace** (převodu na číselnou hodnotu) odstínu barvy na předloze procházející pod snímacím prvkem. Scanery je možné rozdělit podle následujících kritérií:

- **černobílý scanner:** umožňuje snímat pouze v odstínech šedi, barevné odstíny jsou do nich převedeny
- **barevný scanner:** dovoluje snímat nejen v odstínech šedi, ale i v barvách. Většina dnešních scannerů má možnost snímat v TrueColor (tj. 16.7 mil barev).
- **ruční scanner:** vyžaduje, aby uživatel při snímání předlohy sám pohyboval scannerem. Při práci tedy velmi záleží na zkušenosti a zručnosti uživatele, který musí scannerem pohybovat rovnoměrně, správnou rychlostí a rovně. Jedná se o levnější zařízení, které dává méně kvalitní výsledek. Nevýhodou těchto scannerů bývá také to, že většinou nedokáží obsáhnout celou šíři strany formátu A4.



Ruční scanner

Počítačové systémy, periférie, operační systémy

- **stolní scanner:** zařízení, které samo pohybuje snímacím ramenem a díky tomu poskytuje podstatně lepší výsledek oproti ručnímu scanneru. Nevýhodou je jeho vyšší cena.



Stolní scanner firmy UMAX



Stolní scanner firmy Hewlet Packard

- **rozlišení:** počet bodů na palec, které dokáže scanner rozlišit. U dnešních běžných scannerů se pohybuje od 300 x 300 dpi až po 600 x 1200 dpi. Pomocí softwarové interpolace je možné dosáhnout rozlišení až 2400 x 2400 dpi. U profesionálních scannerů je možné vidět i rozlišení 5000 x 5000 dpi.

Scanner se k počítači připojuje pomocí:

- vlastní karty
- SCSI rozhraní
- sériového portu
- paralelního portu
- USB

8.22 Zdroje využívané zařízeními

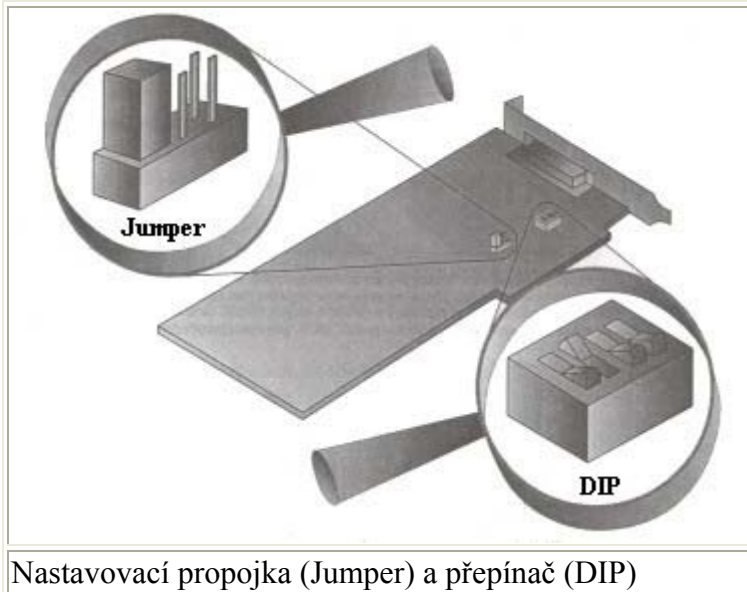
Počítače řady PC jsou navrženy tak, aby bylo možné poměrně snadno vyměňovat jejich jednotlivé části (např. vyměnit vadný modul, vyměnit starší kartu, která již svým výkonem nedostačuje za novou), popř. aby bylo možné snadno stávající počítač rozšířit o nové zařízení. Při instalaci nových karet do počítače je nezbytné postupovat v následujících třech krocích:

- **konfigurace nové desky:** její správné nastavení tak, aby mohla v počítači pracovat
- **vlastní instalace:** její fyzické zasunutí do počítače
- **testování:** ověření, zda nová karta skutečně správně pracuje

Při konfiguraci nové desky je vždy nezbytné si pečlivě přečíst její dokumentaci a zjistit, jaké zdroje (resources) daná karta vyžaduje a jaké jejich nastavení umožňuje. Vlastní nastavení se provádí většinou pomocí nastavovacích propojek (jumperů), popř. pomocí přepínačů (DIP). U novějších karet se konfigurace provádí většinou programově pomocí speciálního programu dodaného společně s kartou. Nastavení zdrojů, které karta ke své činnosti

Počítačové systémy, periférie, operační systémy

vyžaduje, je nutné provést tak, aby nebylo v konfliktu s žádným již existujícím zařízením. Pokud by k takovému konfliktu došlo, znamená to ve většině případů špatnou funkci obou zařízení, popř. nefunkčnost celého počítače.



Nastavovací propojka (Jumper) a přepínač (DIP)

Může se stát, že konkrétní zařízení nepodporuje žádné z možných nastavení, která jsou v počítači ještě volná. V takovém případě bývá nezbytné provést rekonfiguraci některých již dříve zapojených karet a tím uvolnit některý ze zdrojů požadovaných k činnosti nové karty.

Vlastní instalace karty do počítače bývá většinou bezproblémová. Je nutné pouze dbát na opatrnou manipulaci s kartou a otevřeným počítačem a veškeré zapojování a vypořádání provádět, až na výjimky (např. PCMCIA), s vypnutým počítačem, aby nedošlo k poškození karet nebo k poškození celého počítače.

8.23 Technologie Plug & Play (PnP)

Předešlé kapitoly popisovaly postup při instalaci nové karty do počítače. Je pravdou, že v některých případech může být tato instalace poměrně komplikovaná a správné nastavení jednotlivých zdrojů využívaných jednotlivými kartami může působit problémy. Pravděpodobnost výskytu takovýchto problémů je tím vyšší, čím větší je počet karet zapojených do počítače. Takže zapojení další karty do počítače, který již kromě standardních a nezbytných věcí obsahuje ještě např. SCSI rozhraní, síťovou kartu, zvukovou kartu, může způsobit nemalé problémy.

Nové trendy, se snaží problém řešit tím, že počítač by si sám provedl výše zmíněné konfigurace. V ideálním případě by tedy uživatel zasunul kartu do počítače a po jeho zapnutí by počítač sám:

1. Rozpoznal zasunutou desku
2. Zeptal se desky, které IRQ, DMA, I/O Adresy, RAM a ROM adresy potřebuje
3. Zeptal se desky, které IRQ, DMA, I/O Adresy, RAM a ROM adresy může použít
4. Nastavil prostředky tak, aby nedošlo k žádnému konfliktu s již existujícími deskami

Počítačové systémy, periférie, operační systémy

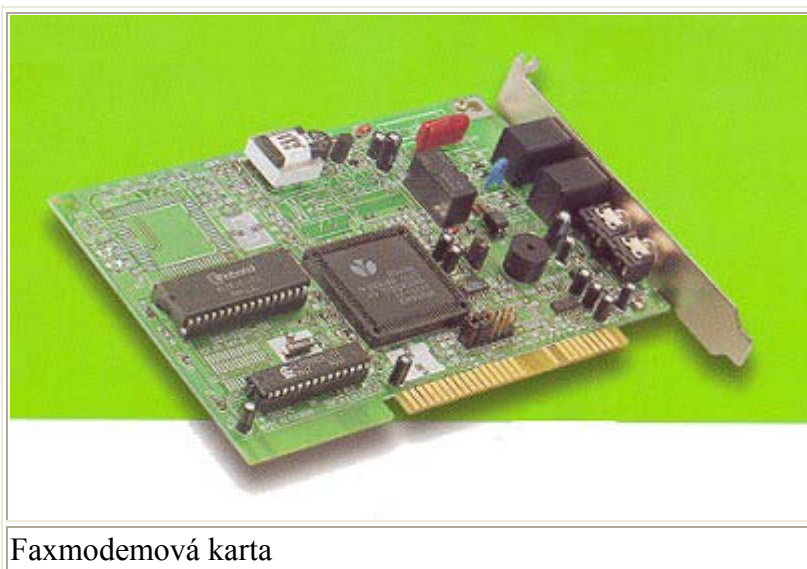
5. Vyhledal a nakonfiguroval potřebné ovladače, které by s nově nainstalovanou deskou spolupracovaly

Pro zvládnutí tohoto problému navrhly firmy Intel, Microsoft a Compaq normu nazvanou **Plug & Play** ("zapoj a hraj"). Myšlenka Plug & Play spočívá v tom, že výrobci přídatných karet přidají ke svým kartám takové obvody, aby bylo možné automaticky desky nastavovat a vznášet dotazy na požadované zdroje. Operační systém pak může sám přímo konfigurovat desky a dotazovat se na ně, čímž by se redukovala nutnost otevírání počítače na minimum.

8.24 Další zařízení

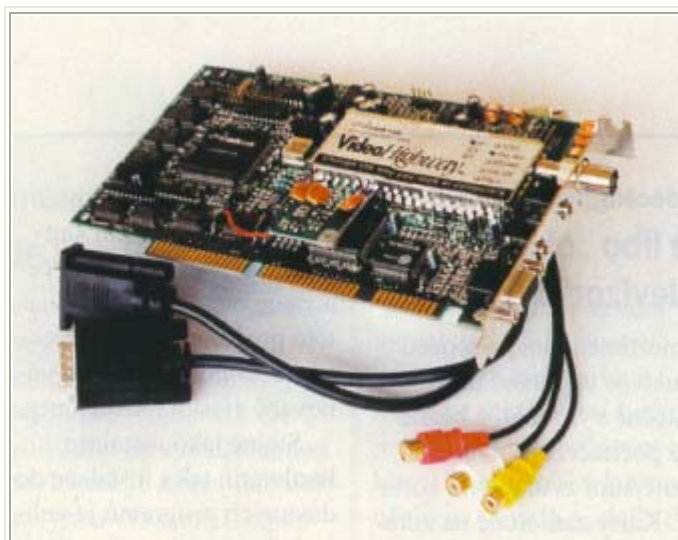
Tato kapitola je věnována výčtu a stručnému popisu některých dalších zařízení, která je možné připojit k počítači, ale jejich rozšíření je již menší, než tomu bylo u zařízení uvedených dříve.

- **Faxmodemové karty:** karty pro příjem a odesílání faxových zpráv pomocí počítače. Tyto karty také dovolují propojení počítačů pomocí telefonní sítě.



Faxmodemová karta

- **Karty pro zpracování videa:** zařízení určená ke zpracování videosekvencí v reálném čase.
- **Karty pro příjem teletextu:** karty umožňující na počítači přijímat teletext.
- **Karty pro příjem rádia a televize:** karty sloužící k příjmu televizního a rádiového signálu na počítači. V případě televizního signálu tyto karty většinou signál pouze zobrazují, ale buď jej nedovedou v reálném čase zaznamenávat, nebo jej zaznamenávají v poměrně nízké (pro profesionální účely nedostačující) kvalitě.



Karta pro příjem rádio a televize VideoHighway

- **Karty pro připojení měřicích přístrojů:** většinou poměrně specializované karty určené k připojení různých měřicích přístrojů (osciloskopy, multimetry, ...), jimiž naměřené hodnoty je možné dále za pomoci počítače zpracovávat.
- **Bezpečnostní karty:** mají za úkol omezit nebo úplně znemožnit přístup cizí osoby k datům v počítači tím, že po uživateli je vyžadováno uživatelské jméno a heslo. Některé tyto karty dovolují i zakódování veškerých dat na pevném disku počítače, čímž je znemožněn přístup neoprávněné osoby k datům i poté, co bezpečnostní kartu z počítače vyjmeme.
- **Tablety:** specializovaná zařízení pro profesionální práci s konkrétním typem programu (většinou programy typu CAD/CAM). Jedná se o vstupní zařízení tvaru tabulky (destičky), po které se rukou pohybuje ukazovacím zařízením, tužkou, zaměřovacím křížem s tlačítky. Tablet slouží k zadávání souřadnic, převodu křivek na číselné údaje nebo k ovládání programů místo myši
- **Digitizéry:** zařízení podobné jako tablet, které poskytuje větší přesnost souřadného systému (řádově 0.001 mm).
- **Trackbally:** vstupní zařízení používané někdy místo myši. Jedná se o kuličku zčásti uzavřenou a z části vyčnívající, která je zpřevodovaná na elektronické snímače otáčení. Při práci pohybuje uživatel touto kuličkou, čímž dosahuje podobného efektu jako uživatel, který pohybuje myší po podložce.
- **Snímače čárového kódu:** vstupní zařízení určená ke čtení čárových kódů. Snímač čárového kódu se většinou zapojují přes rozhraní klávesnice.



Snímač čárového kódu

- **Plottery:** výstupní zařízení pro grafický výstup na papír. Plotter obsahuje pohyblivě upevněnou hlavu nesoucí speciální pisátko, které se pohybuje vzhledem k papíru. Papír bývá buď pevně upevněn, pak se hlava s pisátkem musí pohybovat v obou osách, nebo papír je v jednom směru posouván otáčejícím se válcem, pak se hlava pohybuje pouze v jednom směru (kolmém k pohybu papíru). Plotter se používá zejména ve spojení s programy CAD/CAM ke kreslení technických výkresů, pro něž tiskárny nedostačují.



Literatura:

- 1) Kosek, J.- Janíková, V.: Internet první kroky českého uživatele. Grada, Praha 1996.
- 2) Kusala, J.: Internet ve škole. Fortuna, Praha 2000.
- 3) ABC: Internet od ABC až do Z. Gryf, Praha 2001.
- 4) Čermák, M.: Internet snadno a rychle. Moraviapres.
- 5) Brázda, J.: Internet.cz-jak hledat a najít. Grada, Praha 1999.
- 6) Internet: <http://archiv.czech.net>.
- 7) Hlavenka J.: Microsoft® Windows 98 CZ Second Edition. Praha: Computer Press, 2000.
- 8) Hlavenka J.: První kroky s počítačem. Brno: Computer Press, 1995.
- 9) Born G.: Registr Microsoft® Windows 98 Konfigurujeme a vyladujeme operační systém. Microsoft Press.
- 10) Pisklák, B., Novotná, V.: Základy elementární matematiky a elementární geometrie, skripta OU
- 11) Kolektiv autorů: Číselné soustavy, Kancelářské stroje
- 12) Microsoft: Windows Resource Kit, Computer Press
- 13) Born, G.: Registr Windows 98, Computer Press
- 14) Krol, E.: Vše o Internetu, Science
- 15) Engst, A., Low, C., Orchard, S.: Internet pro uživatele Windows 95, UNIS

Počítačové systémy, periférie, operační systémy

- 16) Kosek, J.- Janíková, V.: Internet první kroky českého uživatele. Grada, Praha 1996
- 17) Kusala, J.: Internet ve škole. Fortuna, Praha 2000
- 18) ABC: Internet od ABC až do Z. Gryf, Praha 2001
- 19) Čermák, M.: Internet snadno a rychle. Moraviapres
- 20) Brázda, J.: Internet.cz-jak hledat a najít. Grada, Praha 1999
- 21) Hlavenka J.: Microsoft® Windows 98 CZ Second Edition. Praha: Computer Press, 2000.
- 22) Hlavenka J.: První kroky s počítačem. Brno: Computer Press, 1995.
- 23) Born G.: Registr Microsoft® Windows 98 Konfigurujeme a vyladíme operační systém. Microsoft Press.
- 24) Morkes David, Komprimační a archivační programy, Computer Press, Brno, 1998;
- 25) Čapek Jan, Fabian Peter, Komprimace dat principy a praxe, Computer Press, Praha, 2000
- 26) Jiří Mrnušík, Viry 98, Grada Publishing
- 27) Jalůvka, J.: Moderní počítačové viry
- 28) Mueller Scott, Osobní počítač, Computer Press Praha 2001
- 29) Pelikán, J.: Architektura počítačů PC. Texty v elektronické podobě. Masarykova univerzita v Brně 1998
- 30) Brandejs, M.: Architektura počítačů. Texty v elektronické podobě. Masarykova univerzita v Brně 1998
- 31) Klimeš, C.: Realizace počítačových systémů. Texty v elektronické podobě. Ostravská univerzita, 2000

Další zdroje informací

- 1) <http://archiv.czech.net/>
- 2) <http://www.microsoft.com>
- 3) <http://home.netscape.com>
- 4) <http://www.internetjam.net>
- 5) <http://www.seznam.cz;>
- 6) <http://www.click.cz;>
- 7) <http://domeny.czechia.com>
- 8) <http://www.viry.cz/>
- 9) <http://www.hoax.cz/>
- 10) <http://biohazard.zde.cz>
- 11) www.mujweb.cz/www/computerweb/viry.htm
- 12) www.mujweb.cz/www/mirap/virus.htm