Počítačové systémy – přehled

PB 152 ◇ Operační systémy

Jan Staudek
http://www.fi.muni.cz/usr/staudek/vyuka/

Verze: jaro 2017

Profil předmětu

- 1. etapa (1/4 obsahu, přednášek), úvod do OS
  - ✓ Připomenutí základních principů činností počítačů
  - ✓ Seznámení s hlavními komponentami OS
  - ✓ Seznámení s hlavními typy výpočetních prostředí
- 2.etapa, rozbor bázových principů činností OS
  - ✓ Procesy a vlákna
  - ✓ Komunikace a synchronizace procesů/vláken
  - ✓ Uváznutí procesů/vláken
  - ✓ Plánování činností procesoru
  - ✓ Správa hlavní (vnitřní) paměti
  - ✓ Virtualizace hlavní (vnitřní) paměti
  - ✓ Ovládání vstupů a výstupů
  - ✓ Vnější paměti počítačů, práce se soubory dat na nich ukládaných a rozhraní služeb souborových systémů jsou obsahem předmětu PV062

Operační sytém, OS

OS je programové vybavení (software) počítačového systému, které

- umožňuje uživatelům využívat zdroje počítače – procesor, paměť, data na vnější pamětech, IO zařízení, ... efektivně a na komunikační úrovni blízké jim a jejich aplikačním systémům
- poskytuje sestavu služeb pro využívání počítače
- ovládá různorodé periférie a vnější paměti počítače
- ...
- Pokud se má porozumět operačním systémům, je nutné rozumět základním principům operací, tj, konceptům činností hardware počítačových systémů

Proč studujeme OS?

- Pravděpodobně nikdo z Vás (už) nebude skutečně psát OS

Tak proč se OS studují?

- ✓ Mnohé aplikace požadují „vyladit" výkon – je nutné porozumět jak služby poskytované OS ovlivňují návrh aplikací
- ✓ OS je nutné administrovat a efektivně využívat – je nutné rozumět strukturám OS (od interface na HW po aplikační úroveň)
- ✓ OS patří mezi nejrozsáhlejší a nejsložitější IT systémy, techniky používané v OS lze uplatnit i jinde – složité struktury dat, souběžnost, řešení konfliktů, správa zdrojů
- ✓ čas od času je potřeba (část) OS upravit (psaní ovladačů, ...), pak je ovšem potřeba operačním systémům rozumět

Studijní literatura

- Přednášky
- Dobré (doporučené) učebnice:

  William Stallings, .
  Operating systems :
  Internals and Design Principles, 8th ed.
  Prentice Hall, 2014
  ISBN: 978-0-13-380591-8

  Avi Silberschatz, . . .
  Operating Systems Concepts, Essentials 2nd ed.
  John Willey, 2013
  ISBN: 9781118804926

Osnova první přednášky

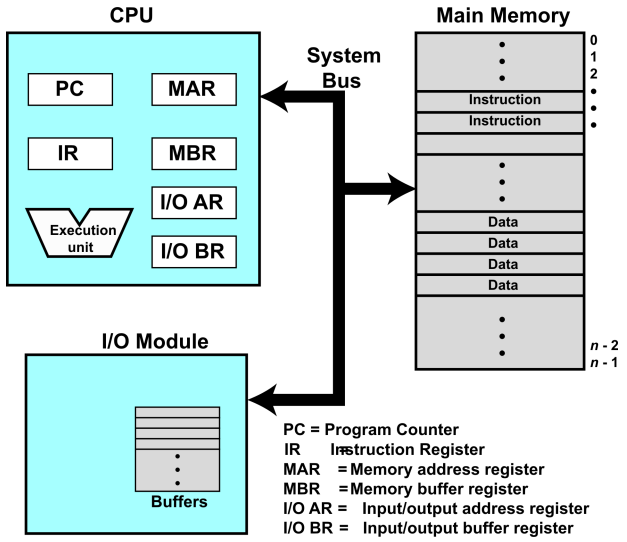
Připomeneme si nejprve co to počítač je

- Principy operací procesoru
- Struktura I/O
- Struktura pamětí
- Hierarchie pamětí
- Hardwarová ochrana
- Architektura univerzálního systému

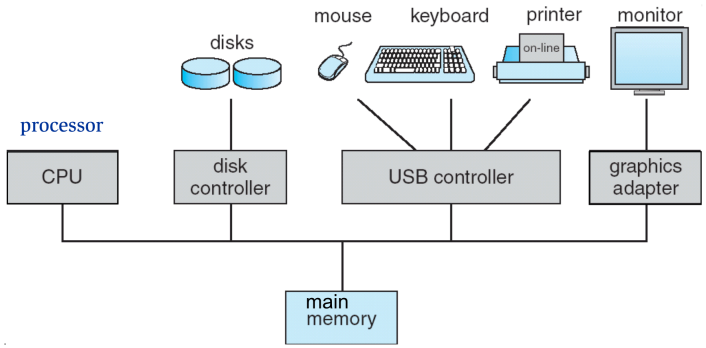
Základní stavební prvky počítače

- procesor, *processor*
  - ✓ řídí činnost počítače
  - ✓ provádí funkce zpracovávající data
  - ✓ pokud počítač obsahuje 1 procesor, nazývá se – *central processing unit* (CPU)
- hlavní paměť, *main memory*
  - ✓ uchovává programy a data
  - ✓ energeticky závislá
  - ✓ další názvy *real memory*, *primary memory*, . . .
- IO moduly
  - ✓ přesun dat mezi počítačem a vnějším prostředím
  - ✓ terminály, vnější paměti, komunikace, . . .
- Systémová sběrnice, *System Bus*
  - ✓ komunikační cesta mezi procesorem, hlavní pamětí a IO moduly

Komponenty počítače – abstrakce vysoké úrovně



Architektura počítačového systému detailněji



- 1 nebo více procesorů a řadiče IO zařízení propojené společnou systémovou sběrnicí zprostředkovávající přístup do sdílené hlavní paměti
- souběžná činnost CPU a zařízení – soupeření o ,,cykly paměti"

Architektura počítačového systému, součinnost I/O a CPU

- I/O zařízení (periferie) a CPU mohou operovat souběžně
- Každý řadič zařízení je odpovědný za činnost zařízení jistého typu
- Každý řadič zařízení má lokální vyrovnávací paměť, *buffer*
- Data z/do operační paměti do/z lokální vyrovnávací paměti periférie přesouvá mikroprogram I/O instrukce řešené v CPU (nebo zvláštní procesor – DMA, viz později)
- I/O = to co se děje mezi lokální vyrovnávací pamětí řadiče a periferií
- Řadič periférie informuje CPU o ukončení své činnosti přerušením nebo indikacemi ve svých registrech

Procesor (CPU) a hlavní (operační, vnitřní) paměť

- Procesor
  - ✓ získává instrukce z paměti, dekóduje je a provádí je
  - ✓ množina instrukcí je specifická pro jistý typ procesoru
  - ✓ typy instrukcí:
    přesun hodnot mezi hlavní pamětí a registry procesu,
    aritmetické/logické operace nad hodnotami v registrech / paměti,
    větvení (skoky), řízení (start IO, ...)
  - ✓ procesor je vybaven svými rychlými pamětmi – registry:
    obsahují – klíčové proměnné, dočasné výsledky,
    data nutná pro řízení běhů výpočtů, ...
- Hlavní paměť
  - ✓ také operační paměť, primární pamět, RAM,
    fyzický adresový prostor (FAP), ..., někdy jen paměť
  - ✓ energeticky závislá paměť
  - ✓ vedle registrů jediná paměť dostupná z procesoru přímo (sběrnicí)

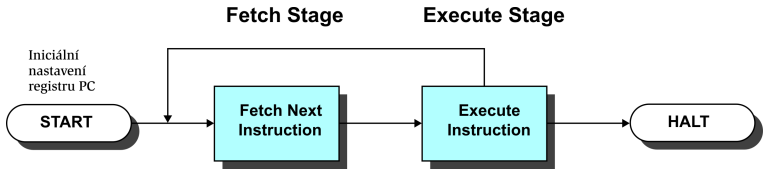
Registry procesoru

- lze je odkazovat (adresovat) ve strojovém jazyku
- Registry viditelné uživateli
  - ✓ jsou dostupné jak OS, tak i uživatelským procesům
  - ✓ obsahují data, adresy (indexy, ukazatele segmentů,
    ukazatele zásobníku, ...), podmínkové kódy – indikace, ...
- Řídicí a stavové registry
  - ✓ obecně nedostupné uživatelským procesům
    (jsou dostupné pouze privilegovanými instrukcemi)
  - ✓ některé používá CPU pro řízení svých vlastních operací
    - *Instruction Register* (IR) – obraz interpretované instrukce
    - *Program Counter* (PC) – adresa následně získávané instrukce
    - *Program Status Word* (PSW)
      bity podmínkových kódů / stavů ($<0,>0,$ přetok, ...)
      bity stavů Interrupt enable/disable, privileg/user mode,
      IO adresa přerušujícího IO zařízení, ...
  - ✓ některé používá OS – řízení bezpečnosti, správy paměti, ...

Jak pracuje procesor, provádění instrukcí

- ✓ procesor interpretuje instrukce uvedené v programu
- ✓ instrukce se získávají z hlavní paměti (FAP) sekvenčně
- ✓ ukazatelem na příště získávanou instrukci z FAP je PC
- ✓ základní cyklus procesoru –
  cyklické provádění dvou fází, FETCH a EXECUTE:

```
loop
    FETCH;      /* FETCH: ((PC)) ⟶ IR */
    PC:= PC+1;
    EXECUTE;    /* EXECUTE: proved' (IR) */
end loop;
```



Jak pracuje procesor, provádění instrukcí

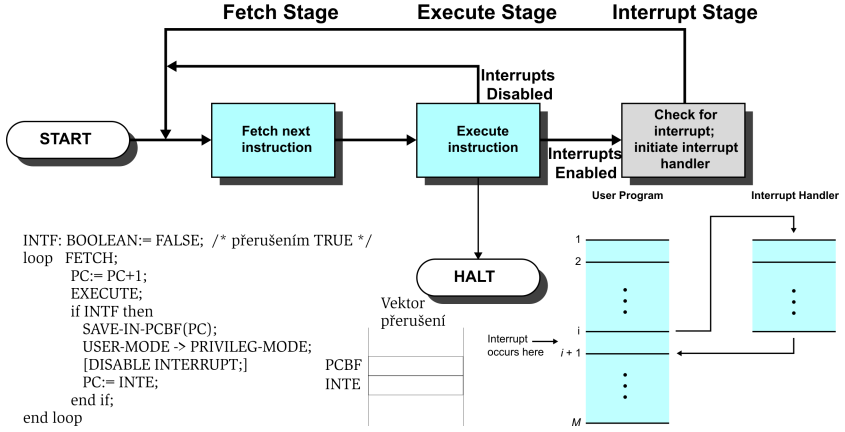
- □ Provádění instrukcí – akce spadající do 4 kategorií
  - ✓ přenosy mezi procesorem a pamětí
  - ✓ přenosy mezi procesorem a IO
  - ✓ zpracování dat (aritmetika, logika)
  - ✓ řízení – změna posloupnosti prováděných instrukcí

Přerušení

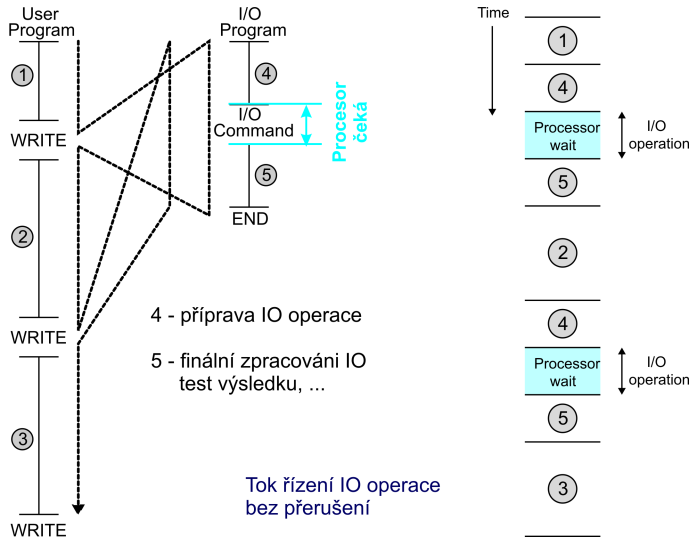
- □ Přerušení normálního běhu procesoru
  - ✓ cílem je umožnit překrýváním více činností v čase
  - ✓ dynamicky vzniká potřeba provést jistou posloupnost příkazů (OS) jako reakci na nějakou přerušující událost
  - ✓ přerušující událost způsobí, že se potlačí provádění běžícího procesu tak, aby ho bylo možné později obnovit
- □ v době řešení I/O operace se umožní,
  aby CPU prováděla jiné instrukce než
  instrukce programu čekajícího na konec I/O operace
  - ✓ činnost CPU se později přeruší iniciativou ,,I/O modulu"
  - ✓ CPU předá řízení na ,,Interrupt Handler Routine"
    (standardní součást jádra OS)
- □ CPU (na úrovni mikroprogramu) testuje nutnost přerušení
  alespoň po každém provedení instrukce

Jak pracuje procesor s přerušovacím systémem

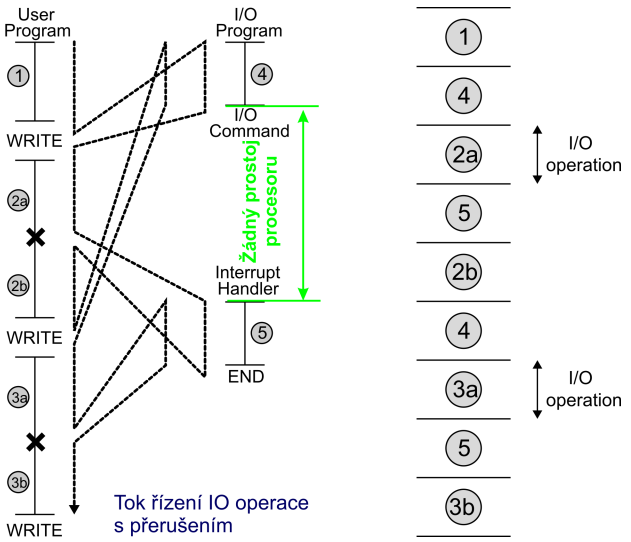
- ✓ Existuje-li nevyřízená žádost o přerušení, a je povoleno přerušování, provede se *interrupt handler*, správce přerušení, součást jádra OS



Tok řízení IO operací bez přerušení



Tok řízení IO operací s přerušením



Obecná funkce přerušení

- Přerušení předává řízení správci přerušení (*interrupt handler*) prostřednictvím vektoru přerušení
- Vektor přerušení obsahuje adresy vstupních bodů všech správců přerušení, pokud jsou replikovaní podle příčin přerušení
- Mechanismus přerušení musí uchovat adresu instrukce prováděné jako příští po obnově výpočtu po obsluze přerušení (na definovaném místě v paměti)
- Obsluha přerušení vesměs nebývá násobně přístupná – aby se zajistila validní obsluha přerušení, jsou během obsluhy přerušení další indikovaná přerušení maskována (*disabled*)

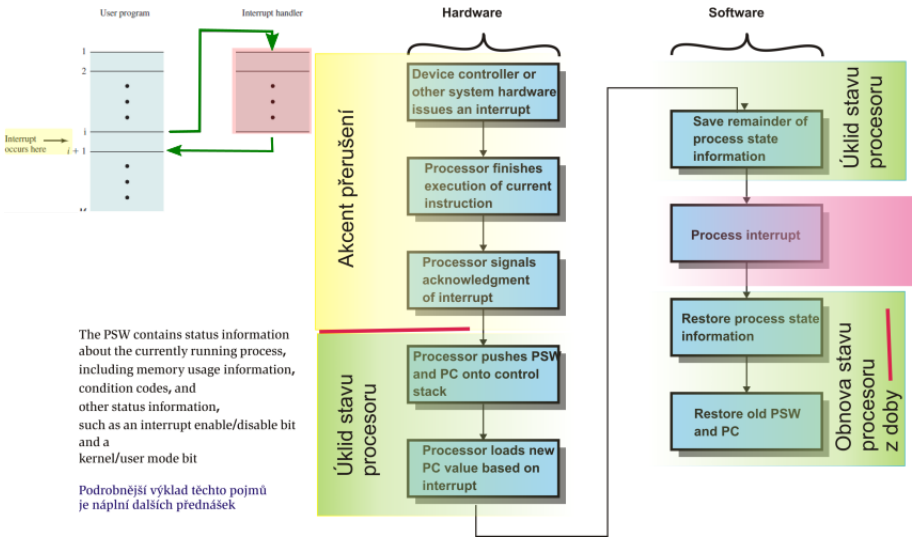
Obecná funkce přerušení

- „Trap" – softwarově generované přerušení
  - žádostí o službu řešenou operačním systémem (*System call*) vyvolanou provedením speciální instrukce
- trap – také synchronní přerušení
- interrupt – také asynchronní přerušení
- Operační systém je systém řízený přerušeními (*interrupt driven*)

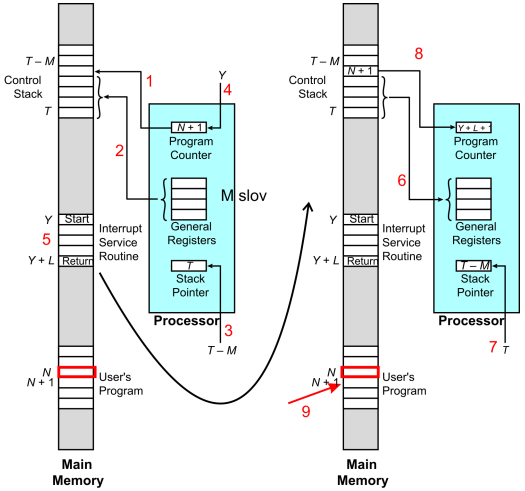
Správa přerušení

- třídu/typ přerušení určí mikroprogram CPU
  – specifikuje položky PCBF a INTE ve vektoru přerušení
- mikroprogram CPU zapamatuje stav CPU uchováním čítače instrukcí
- (INTE) určuje vstupní bod relevantního správce přerušení
- správce přerušení zapamatuje stav CPU uchováním ostatních registrů např. v určeném zásobníku
- správce přerušení detailně specifikuje příčinu přerušení dotazy na stavové registry a určí konkrétní způsob obsluhy příčiny přerušení

Idea zpracování přerušení



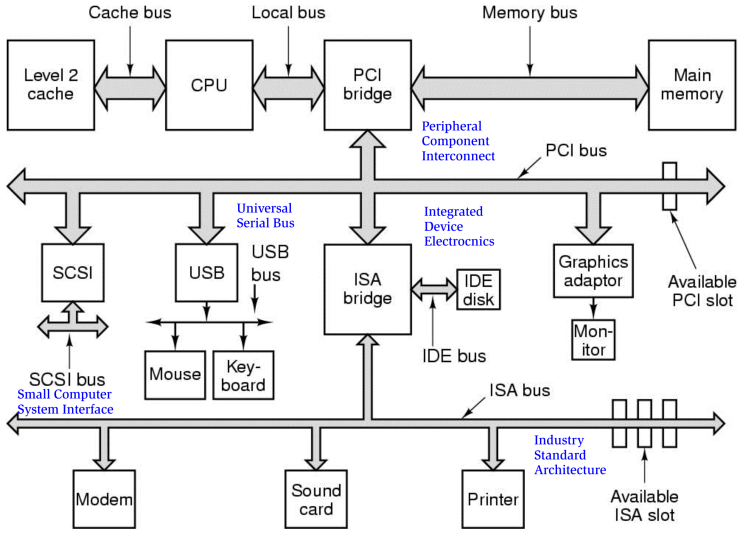
Změny v paměti a v registrech při obsluze přerušení



Systémová sběrnice

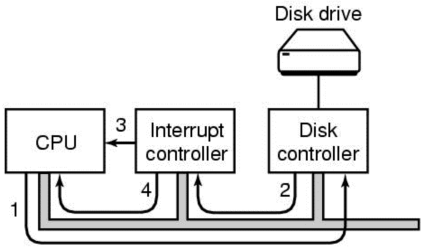
- komunikační prostředek mezi CPU, pamětí a IO
- komunikační cesty pro přenos adres
- komunikační cesty pro přenos dat
- $n$bitově paralelní (8, 16, 32, 64, . . . paralelních bitových cest)

Systémová sběrnice, příklad Pentium

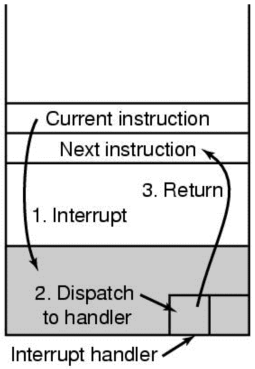


Obsluha I/O zařízení pomocí přerušení

Start IO operace a přerušení indikující konec IO operac



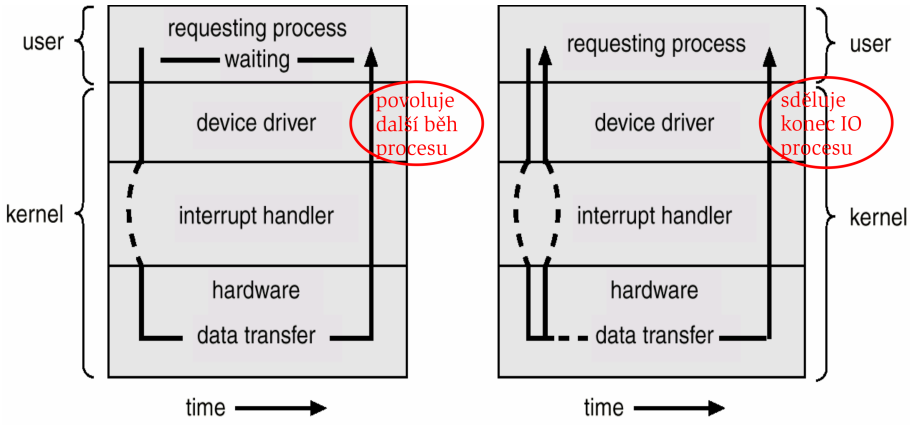
Reakce na přerušení



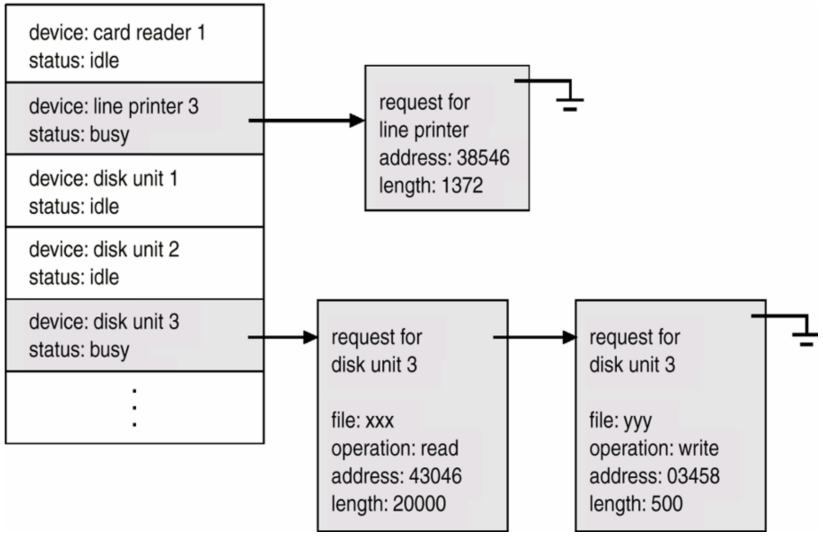
Struktura I/O – dvě metody obsluhy

- činné čekání, *busy waiting*
  - ✓ v systémech bez řízení IO pomocí OS
  - ✓ žádné souběžné zpracovávání I/O, nedořešený zůstává nejvýše jeden I/O požadavek
  - ✓ program testuje konec IO operace opakovanými dotazy na příslušný stavový registr IO zařízení
- přerušením (a OS) řízená souběžná realizace IO
  - ✓ v systémech s řízením IO pomocí OS
  - ✓ souběžné zpracovávání I/O s během programu(ů)
  - ✓ IO operaci zahajuje OS na žádost procesu
  - ✓ proces čeká na dokončení IO operace synchronní řešení IO
  - ✓ proces nečeká na dokončení IO operace asynchronní řešení IO, může běžet souběžně s IO operací

Synchronní a asynchronní řešení I/O pomocí OS

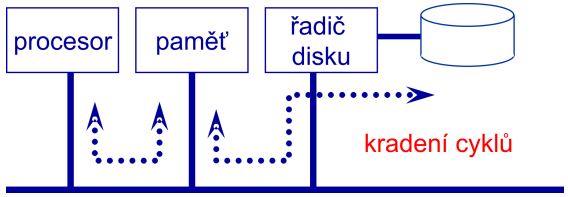


Přehledu o stavu IO operací v OS – Device-Status Table



Direct Memory Access, DMA

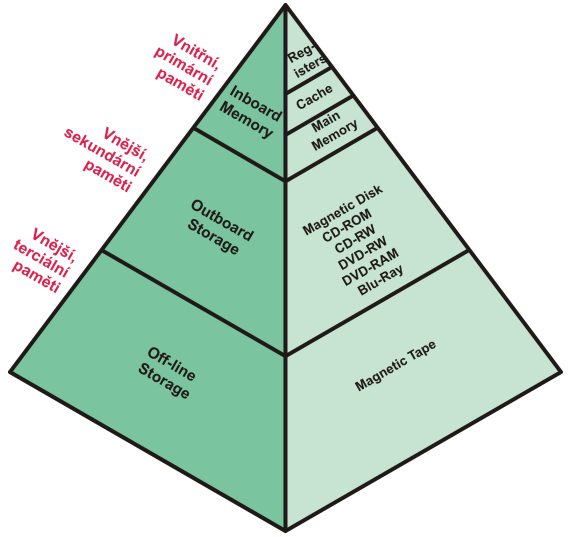
- Používá se pro velmi rychlá I/O zařízení pro přenos dat do/z pamětí rychlostí ,,blízkou" rychlosti vnitřní paměti
- Řadič periférie přenáší bloky dat mezi vyrovnávací pamětí a periférií bez zásahů ze strany CPU – kradení cyklů (*cycle stealing*)
- Přerušení se generuje po přenesené celého bloku, ne po každé dílčí jednotce (byte)



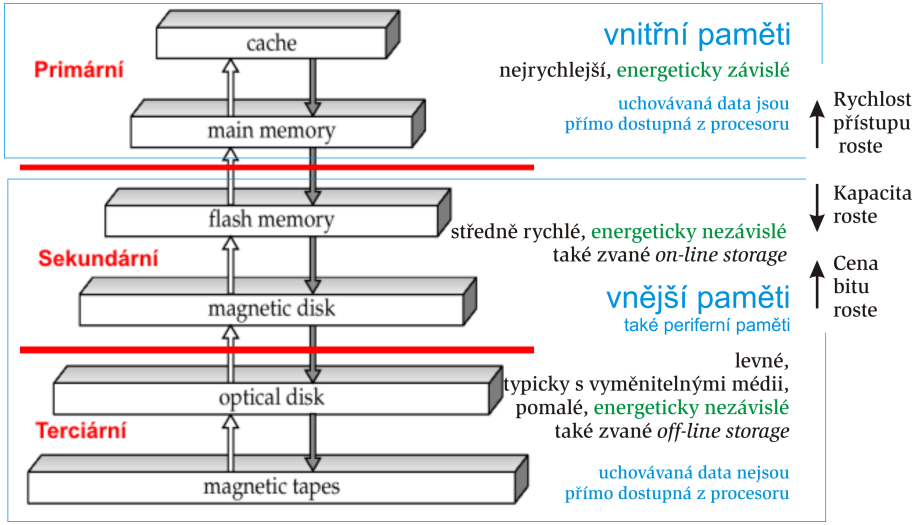
Struktura (hierarchie) pamětí

- Hlavní paměť, operační paměť, RAM, primární paměť, ...
  - ✓ Jediná velká paměť, do které CPU může přistupovat přímo
  - ✓ Vesměs energeticky závislá z hlediska uchovávání obsahu
  - ✓ Kapacita, řádově: desítky MB až jednotky GB,
  - ✓ rychlost přístupu $< 10^{-6}$ s (typicky desítky až stovky ns)
- Sekundární paměť
  - ✓ Rozšíření paměti poskytující energeticky nezávislou (*nonvolatile*) paměťovou kapacitu (jednotky GB až TB)
  - ✓ Rychlost přístupu typicky jednotky ms
  - ✓ Typický reprezentant – magnetický disk
- terciární paměť
  - ✓ archivní média
  - ✓ Typický reprezentant – roboticky řízené sklady pásek

Hierarchie pamětí



Hierarchie pamětí



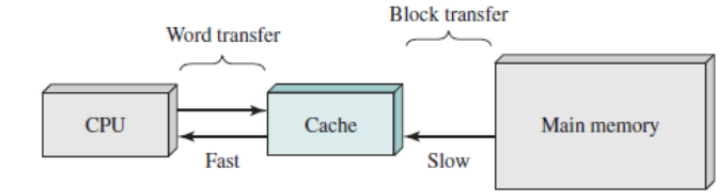
Caching, cache paměť

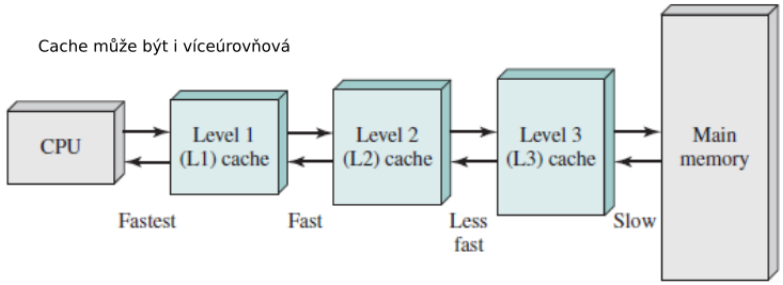
- Caching *CPU-main memory*
  - ✓ Mikroprogramem řízené kopírování dat nedávné historii zpřístupňovaných v operační (hlavní) paměti a jejich okolí do rychlejší paměti (s menší kapacitou) – *cach memory*
  - ✓ Používání rychlejší paměti pro zpřístupňování aktuálních dat/instrukcí
  - ✓ Uplatňuje se princip časové a prostorové lokálnosti běžných programů
  - ✓ Jimi řízené procesy se s vysokou pravděpodobností po jistou dobu pohybují v omezeném adresovém prostoru paměti
  - ✓ Procesor se obrací na operační paměť až v případě, když se zpřístupňovaná data nenacházejí v cache paměti
- Caching zavádí jinou úroveň hierarchie pamětí
  - ✓ tatáž data se souběžně uchovávají ve více než jedné úrovni
  - ✓ Je nutné proto řešit problém udržení konzistence více kopií těchže dat

Caching, cache paměť

- ✓ Cache –"skrýš"
- ✓ Drahá ale velmi rychlá paměť
- ✓ Stýká se s pomalejší ale větší pamětí
- ✓ OS a uživatelské programy ji nevidí
- ✓ Cache je udržovaná pomocí hardware správy paměti
- ✓ Procesor hledá odkazované slovo nejprve v cache
- ✓ Jestliže procesor slovo nenalezne v cache, přesune se do cache blok z RAM, který toto slovo obsahuje
- ✓ Princip časo-prostorové lokality programů způsobuje,že příště zpřístupňované slovo bude s velkou pravděpodobností nalezeno v cache
- ✓ Používá se dynamicky podobně jako virtuální paměť
- ✓ Rychlost procesor je bližší rychlosti cache než rychlosti vnitřní paměť(RAM)

Caching, cache paměť





Bezpečnostní mechanismy – 2 stavy procesoru

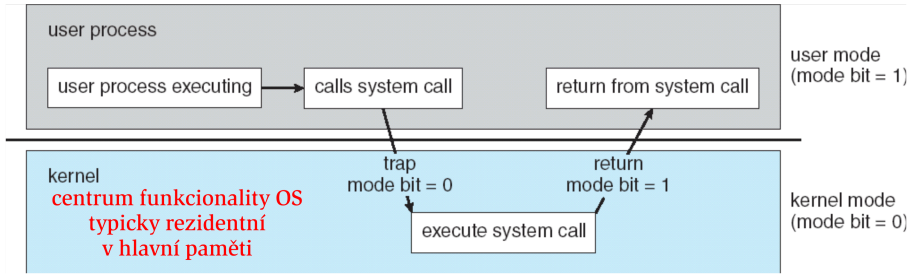
- Sdílení systémových zdrojů požaduje, aby OS měl záruku, že nesprávný program negativně neovlivní běh ostatních procesů nebo OS
- Z pravomoci (a odpovědnosti) uživatelských programů se vyjímají I/O operace, operace ovlivňující stav systémových zdrojů (registry ochrany,. . .) apod.
- mnohé funkční vlastnosti smí spravovat pouze OS, ne aplikační programy – je nutný duální režim činnosti CPU
  - ✓ user mode – CPU může interpretovat omezený instrukční repertoár a nemůže zpřístupňovat zdroje systému dostupné výhradně OS
  - ✓ kernel mode – CPU není nijak omezovaný, může provádět i tzv. privilegované instrukce

2 stavy procesoru

- Privilegované instrukce se mohou provádět pouze v privilegovaném režimu
- Po přijetí přerušení (vč. trap-přerušení, žádostí o provedení služby) se procesor automaticky přepíná do privilegovaného režimu (spouští se OS)
- do uživatelské režimu procesor přepíná jádro OS při spouštění uživatelského procesu

2 stavy procesoru

- procesor přechází do *kernel mode* přijetím přerušení
- procesor přechází do *user mode* privilegovanou instrukcí provedenou OS při spouštění běhu procesu
- stav procesoru indikuje mode bit ve stavových registrech CPU



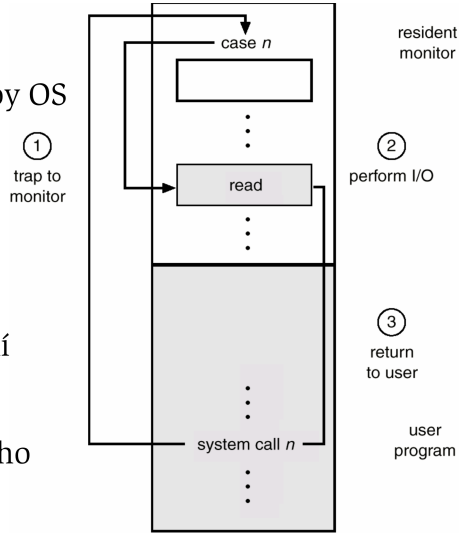
Bezpečnostní mechanismy – hardwarové ochrany

Ochrana I/O

Povinné je použití volání služby OS k zahájení I/O

Všechny I/O instrukce jsou privilegované

Musí platit, že uživatelský program nikdy nezíská řízení v privilegovaném režimu. tj. nesmí mít např. možnost uložit adresu do přerušovacího vektoru



Bezpečnostní mechanismy – hardwarové ochrany

- Ochrana paměti
  - ✓ Souvisí s metodami správy paměti
  - ✓ Detaily později při výkladu správy paměti
- Ochrana (dostupnosti) CPU
  - ✓ Záruku, že vládu nad procesorem si udrží OS a ne aplikační program, poskytuje časovač
  - ✓ Časovač – přednastavený registr privilegovanou instrukcí, –1 (decrement) při každém hodinovém tiku, jakmile hodnota registru dosáhne 0, generuje se přerušení
  - ✓ Souvisí s metodami plánování
  - ✓ Detaily později při výkladu plánování činnosti procesoru

Klasifikace počítačů – osobní/personální počítače

- typicky dedikované pro jednoho uživatele
  - ✓ v současné době ale vesměs s multiprogramováním (multitasking)
- typické I/O vybavení
  - ✓ klávesnice, myš, obrazovka, malá tiskárna, komunikační připojení
- Upřednostňovaným cílem je uživatelovo pohodlí,
- Iniciální trend – vesměs minimum ochran –
  - hlavní roli hraje uživatelova odpovědnost
  - vesměs minimální využívání ochranných rysů CPU
- jejich OS posléze adoptovaly technologie vyvinuté pro OS větších počítačů (střediskových, podnikových serverů, ...)
- Mohou se na nich provozovat různé typy operačních systémů
  - ✓ Windows, MacOS, UNIX, Linux, ...

Multiuživatelské systémy, Time-Sharing Systems, TSS

- multiprogramování
  - orig. technologie pro efektivní dávkové zpracování
  - souběžné řešení více programů
- Multiuživatelské systémy rozšiřují plánovací pravidla o rychlé (spravedlivé, cyklické) přepínání mezi procesy řešícími zakázky interaktivních uživatelů
- podporuje se on-line komunikace mezi uživatelem a OS
  - ✓ původně v konfiguraci počítač–terminál
  - ✓ v současnosti v síťovém prostředí
- systém je uživatelům on-line dostupný
  - jak pro zpřístupňování dat
  - tak i pro řešení programů

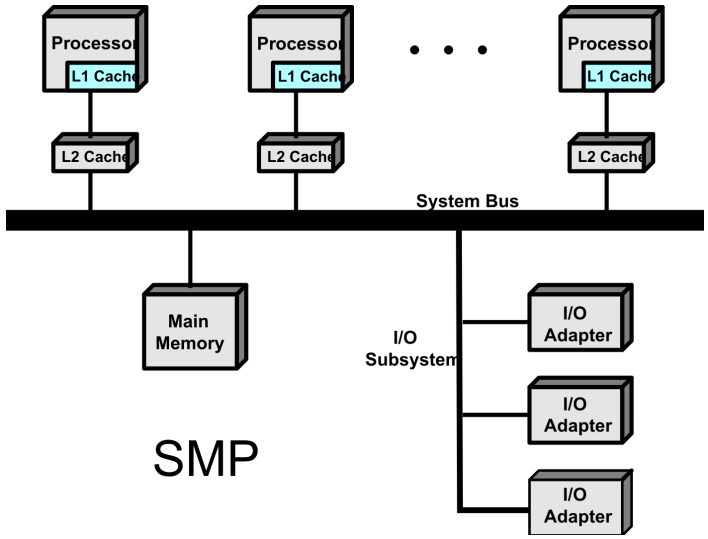
Klasifikace počítačů – paralelní a distribuované systémy

- Paralelní systémy –
  - ✓ více procesorů
  - ✓ sdílí společný FAP
  - ✓ všechny procesory mohou ,,současně'' vidět stav paralelně řešené úlohy (každého z participujících procesorů) udržovaný ve sdíleném FAP
  - ✓ paralelní systémy jsou řízeny paralelními algoritmy
- Distribuované systémy –
  - ✓ více počítačů
  - ✓ nesdílí společný FAP, každý počítač má svůj lokální FAP
  - ✓ komunikují periferními operacemi (spoje, síť) – výměnou zpráv
  - ✓ stav distribuovaně řešené úlohy si musí každý zúčastněný počítač postupně získávat výměnou zpráv
  - ✓ Distribuované systémy jsou řízeny distribuovanými algoritmy

Paralelní systémy, klasická klasifikace

- zvyšují dosažitelnou propustnost, ekonomičnost a spolehlivost
- Multiprocesorové systémy
  - ✓ také těsně vázané systémy
  - ✓ systémy s více než jednou CPU propojených se společným FAP systémovou sběrnicí a sdílejících rovněž IO
- Symetrický multiprocessing (SMP)
  - ✓ Celý systém je řízený integrovaným OS
  - ✓ OS může být interpretovaný souběžně více procesory
  - ✓ Podporuje většina soudobých OS
  - ✓ Najednou může běžet více procesů, aniž dojde ke snížení výkonu
  - ✓ Kterýkoliv proces může kdykoliv běžet na kterémkoliv procesoru

Paralelní systémy, klasická klasifikace



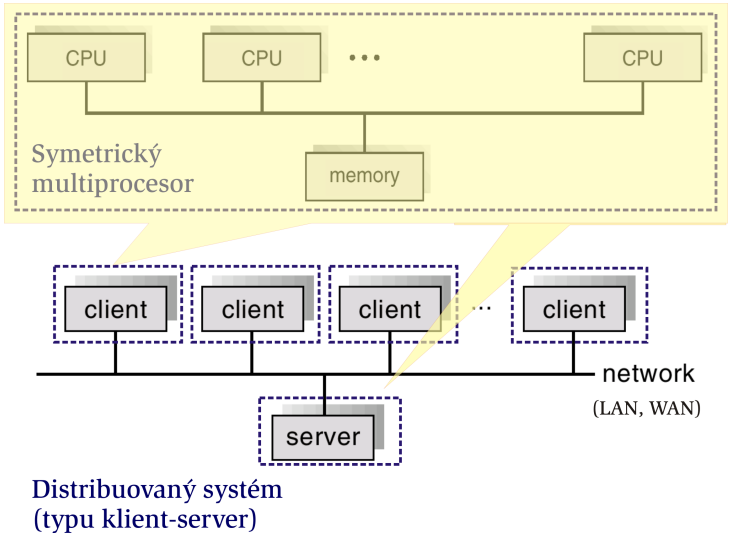
Paralelní systémy, klasická klasifikace

- Asymetrický multiprocessing (AMP)
  - ✓ každý procesor má přidělený specifický úkol
  - ✓ hlavní (*master*) procesor (CPU) vedle výpočtů plánuje a přiděluje práci podřízeným (*slave*) procesorům (komunikační procesor, ...)

Distribuované systémy

- Distribuce výpočtů mezi více počítačů propojených sítí
- lze sdílet zátěž (*load-sharing*), výpočty se tudíž zrychlují
- zvyšuje se spolehlivost, komunikativnost
  - ✓ také volně vázané systémy
  - ✓ každý samostatný procesor má svoji vlastní lokální paměť (FAP)
  - ✓ vzájemně komunikují pomocí komunikačních spojů výměnou zpráv
- vynucují si použití vhodné síťové infrastruktury
  - ✓ LAN, Local Area Networks
  - ✓ WAN, Wide Area Networks
- klasifikace
  - ✓ symetrické distribuované systémy – peer-to-peer
  - ✓ asymetrické distribuované systémy – klient–server
- Distribuovaný operační systém x síťový operační systém

Paralelní a distribuované systémy, ilustrace architektur



Distribuovaný systém
(typu klient-server)

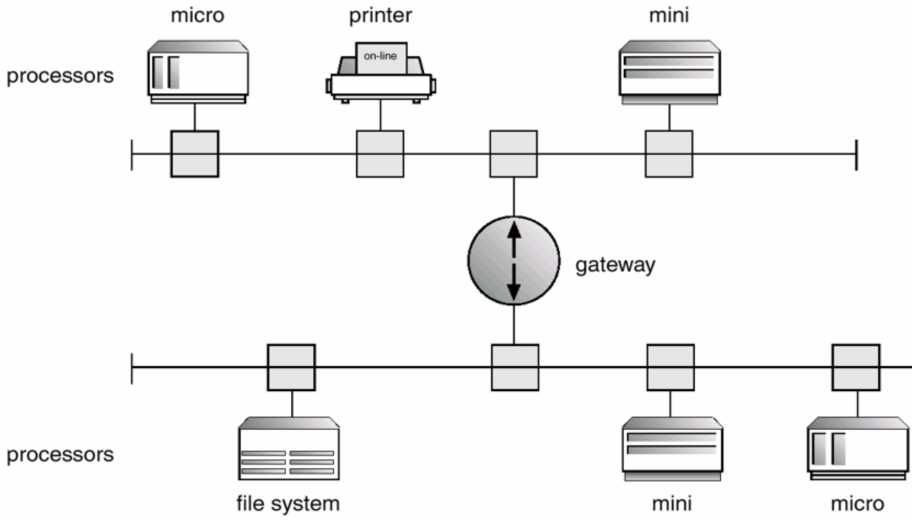
Flynnova kategorizace výpočetních systémů

- *Single Instruction Single Data* (SISD)
  - ✓ jediný procesor provádí jediný instrukční proud
  - ✓ data jsou uložena v jediné, sériově dostupné paměti
  - ✓ klasický 1-procesorový systém, příp. řízený OS s multitaskingem
- *Single Instruction Multiple Data* (SIMD)
  - ✓ jedna (a táž) instrukce se provádí na množině dat více procesory
  - ✓ specializované maticové / vektorové počítače (koprocesory)
- *Multiple Instruction Single Data* (MISD)
  - ✓ jedna posloupnost dat je přenášena k množině procesorů
  - ✓ každý procesor nad daty provádí jinou posloupnost instrukcí
  - ✓ nikdy neimplementováno, formální model
- *Multiple Instruction Multiple Data* (MIMD)
  - ✓ multiprocesor – těsně vázané systémy
  - ✓ distribuovaný systém – volně vázané systémy

Klasifikace počítačů – shluky, Clustered Systems

- Shlukování (clustering)
  - ✓ pod řízením OS lze ,,shlukovat'' více procesorů pro řešení dílčích problémů jednoho zadání
  - ✓ paralelní systémy – na bázi SMP
  - ✓ distribuované systémy – na bázi LAN, sdílení vnější paměti – příklad – ORACLE Parallel Server (jedna verze DBS ORACLE)
- cílem bývá dosažení vysoké dostupnosti řešené služby

LAN, Local Area Network



WAN, Wide Area Network