

## Podstředm vstupu a vřtupu

PB 152 ◊ Operační systémy  
PV 062 ◊ Organizace souborů

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



Verze : jaro 2016

## Osnova přednášky

Vstup/vřtup – Input/output – V/V – I/O – periférie – . . .  
Nejrozmanitřší, nejměně systematizovaná stránka OS,  
neexistuje její obecně, plně konzistentní řešení.

- Hardware I/O
- Aplikační rozhraní I/O
- Podstředm I/O v jádru
- Transformace požadavku na I/O požadavky na hardware
- Proudý
- Výkon

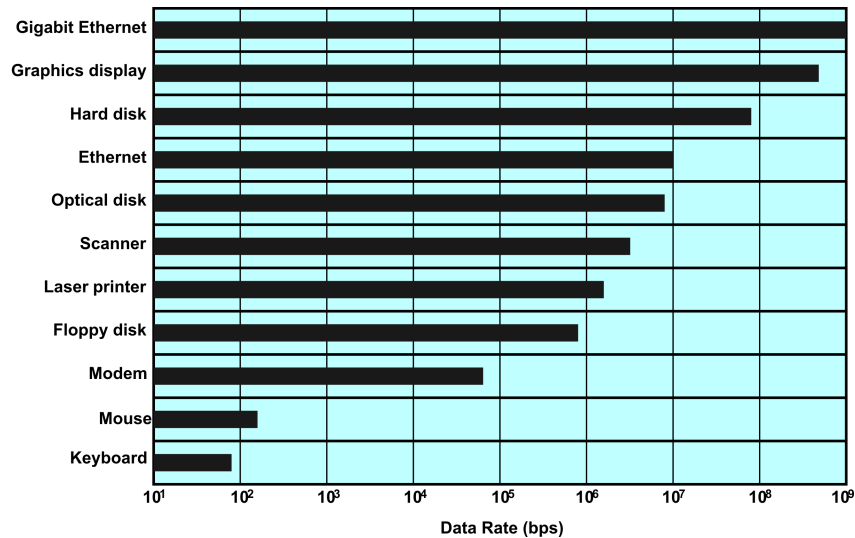
## Klasifikace

- I/O s rozhraním na lidského činitele
  - ✓ tiskárny, terminály, displeje, klávesnice, myš, . . .
- I/O s rozhraním jiné technické zařízení
  - ✓ disky, USB klíče, detektory, řadiče, . . .
- Komunikační IO
  - ✓ komunikační zařízení pro komunikaci počítače s jinými zařízeními

## IO svět – čím se liší I/O zařízení ?

- Rychlostí přenosu dat – ilustrace viz následující obrázky
- Aplikací
  - ✓ **disk** použitý pro uchovávání souborů dat požaduje od OS funkčnost **správa souborů**  
disk použitý pro skladování obrazů LAP vyžaduje funkčnost OS **virtualizace paměti**
  - ✓ **terminál** může používat běžný **uživatel** nebo **administrátor**
- Složitostí obsluhy rozhraní počítač–periférie
- Jednotkou přenosu počítač–periférie
  - ✓ proud jednotlivých bytů (terminál), blok bytů (disk), . . .
- Zobrazením (kódováním) dat
- Chybovostí a potřebným chybovým řízením

## IO svět – svět velmi odlišných rychlostí



## IO svět – svět velmi odlišných rychlostí

Device	Data rate	Device	Data rate
Keyboard	10 bytes/sec	FireWire (IEEE 1394)	50 MB/sec
Mouse	100 bytes/sec	USB 2.0	60 MB/sec
56K modem	7 KB/sec	SONET OC-12 network	78 MB/sec
Scanner	400 KB/sec	SCSI Ultra 2 disk	80 MB/sec
Digital camcorder	3.5 MB/sec	Gigabit Ethernet	125 MB/sec
802.11g Wireless	6.75 MB/sec	SATA disk drive	300 MB/sec
52x CD-ROM	7.8 MB/sec	Ultrium tape	320 MB/sec
Fast Ethernet	12.5 MB/sec	PCI bus	528 MB/sec
Compact flash card	40 MB/sec		

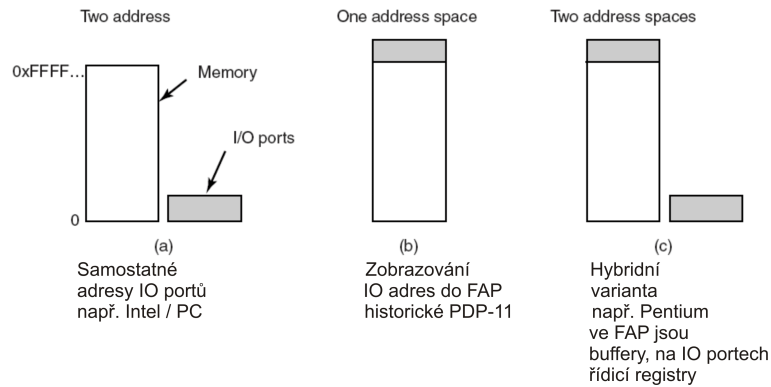
## I/O Hardware

- I když existuje spousta různých V/V zařízení, **I/O devices**, periférií lišících se vlastnostmi a způsoby řízení, ideální OS umožňuje
  - ✓ jak jejich individuální detailní ovládní
  - ✓ tak realizaci univerzálních IO operací přes jednotné rozhraní
- Společné rysy z hlediska architektury
  - ✓ **port**, přípojné adresovatelné místo periferie
    - typicky několika bajtové registry
    - např. stavový, řídicí, vstupní a výstupní, ...
  - ✓ **sběrnice**, pro přenos čtených/zpísovaných dat z/do periferie
  - ✓ **řadič (adapter)** periferie,
    - interface periferie na CPU (přes sběrnici)
    - přijímá příkazy/dat z CPU a dodává data/indikace do CPU
    - dává podnět pro generování přerušení

## I/O instrukce

- **I/O instrukce** strojového jazyka
  - ✓ čtení/zápis dat z/do portu, via datové registry
  - ✓ zadávání příkazů pro IO zařízení, via řídicí registry
  - ✓ odebrání stavových informací od IO zařízení, via stavové/indikační registry
- dvě generické formy I/O instrukcí, dvě formy adresování I/O zařízení
  - ✓ **I/O adresy jsou mapované do adres FAP**, použití v instrukcích přesunu (LOAD, STORE, MOVE, ...), zápisem a čtením z jistých adres se mění konfigurace zařízení, zadávají se výstupní data, získávají se vstupní data, stav, ... např. počítač s 32bit adresovou sběrnici, s 4GB FAP: např. posledních 128MB adres je vyhrazeno pro porty per. zař.
  - ✓ **samostatný IO adresový prostor**, použití v I/O instrukcích (IN, OUT)

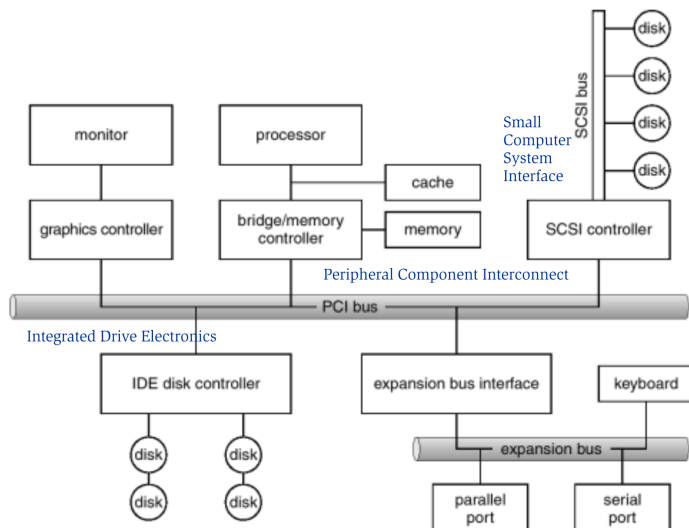
## IO adresování



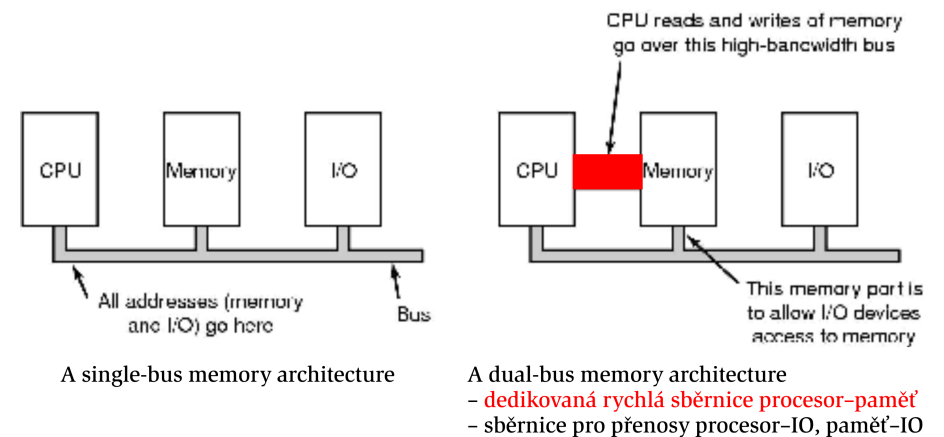
## I/O instrukce, samostatný IO adresový prostor

- *IN* registr-kam, registr-zdroj  
registr-kam: cíl načtení vstupující hodnoty  
registr-zdroj: obsahuje IO adresu / číslo portu vstupu
- *OUT* registr-kam, registr-zdroj  
registr-kam: obsahuje IO adresu / číslo portu výstupu  
registr-zdroj: registr s vypisovanou hodnotou
- ilustrace části *IO address range* (hexadecimal) pro PC:
  - 000-00F: DMA Controller
  - 020-021: Interrupt Controller
  - 040-043: Timer
  - 200-20F: Game Controller
  - 2F8-2FF: Serial port (secondary)
  - 320-32F: Hard disk Controller
  - 378-37F: Parallel port
  - 3D0-3DF: Graphics Controller
  - 3F0-3F7: Diskette drive Controller
  - 3F8-3FF: Serial port (primary)

## Klasická struktura počítače se jednou sběrnici



## Struktura počítače se více sběrnici

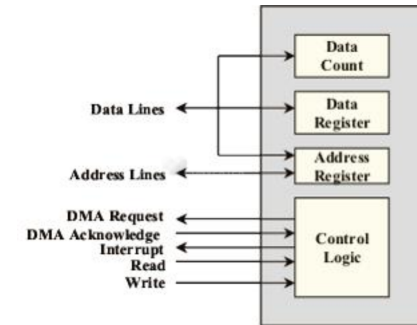


## Techniky provádění I/O

- **programovaný I/O, polling, busy-waiting**
  - ✓ vydání instrukce – **I/O command**
  - ✓ cyklické dotazování na stav přenosu až do zjištění jeho konce, tj. činné čekání, **synchronní operace**
- **programovaný I/O, řízený přerušením**
  - ✓ vydání instrukce – **I/O command**
  - ✓ paralelní běh s procesorem, typicky několikaslabikový přenos
  - ✓ I/O modul (řadič) oznamuje konec přenosu přerušením, **asynchronní operace**

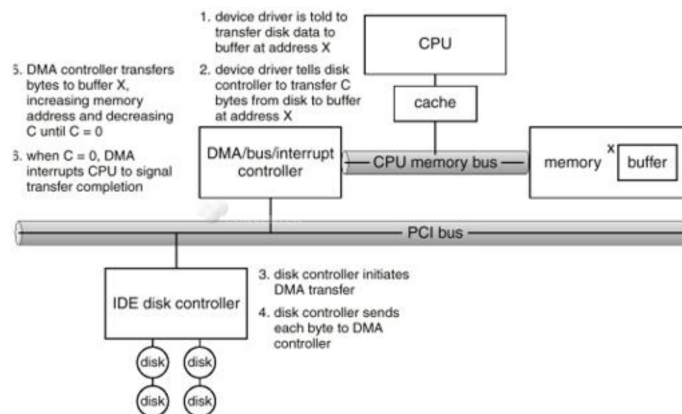
## Základní koncepce ovládání I/O

- **Direct Memory Access (DMA)**
  - ✓ zadání definice IO operace
  - ✓ výměna bloků mezi FAP a I/O zařízením („kradení cyklů paměti“)
  - ✓ přerušování generované DMA po ukončení přenosu bloku

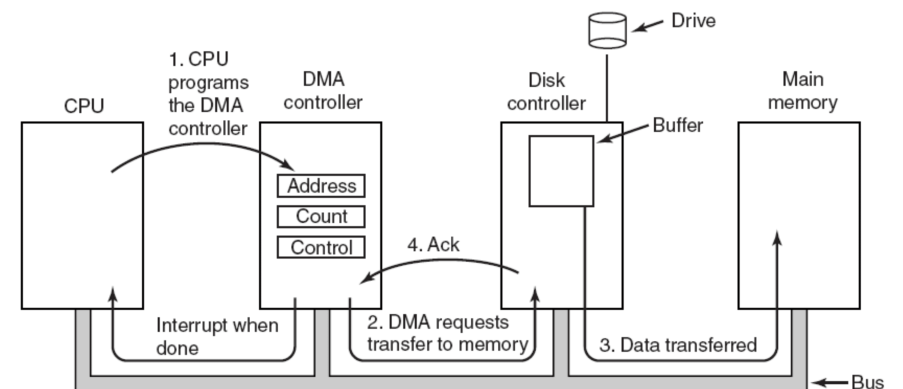


## Direct Memory Access, DMA

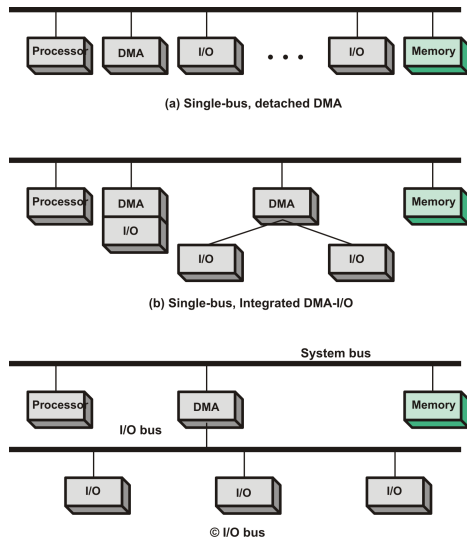
- ✓ Náhrada programovaného I/O při velkých přesunech dat
- ✓ požaduje se speciální DMA řadič
- ✓ při přenosu dat se obchází procesor, přímý tok zařízení – paměť



## Direct Memory Access, DMA



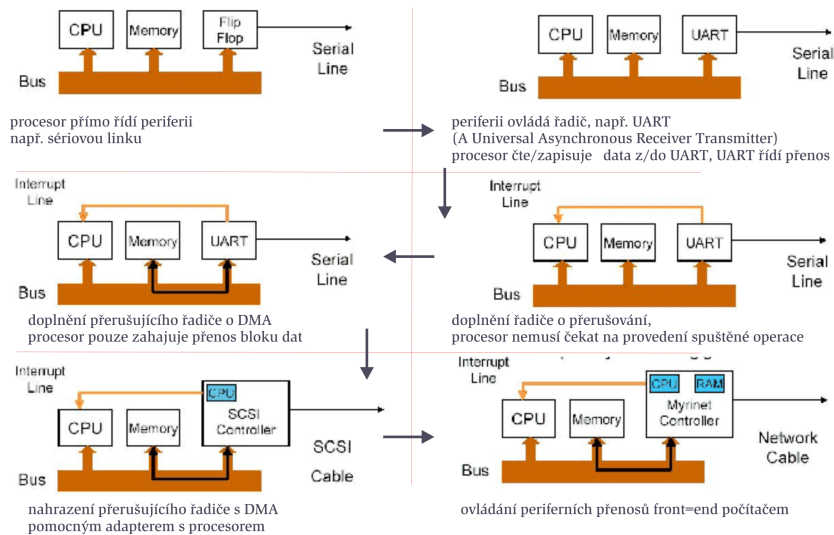
## Direct Memory Access – možná alternativní řešení



## Základní koncepce ovládání I/O

- I/O procesor, kanál –
  - analogie DMA + multiplexing více přenosů,
  - DMA = selektorový kanál, I/O procesor = multiplexorový kanál
- I/O počítač –
  - samostatné zpracování IO
  - (např. ze sdíleného disku apod.)
  - počítače s vlastním FAP

## Techniky provádění I/O



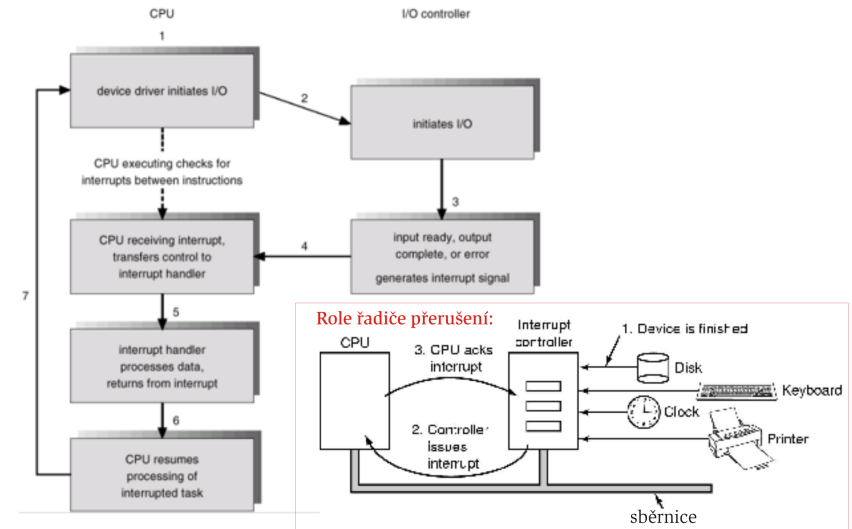
## Polling, vyzývání

- Určení stavu zařízení
  - ✓ připravené
  - ✓ obsazené
  - ✓ chybový stav
- vydání IO instrukce
- činné čekání na konec operace I/O zařízení – monoprogramové systémy
- oznámení konce operace I/O zařízení přerušením – multitaskingové systémy

## Přerušeni

- Přerušeni obsluhuje **správce přerušeni**
- **Maskováním** lze některá přerušeni ignorovat nebo oddálit jejich obsluhu
- Potřebný správce přerušeni se vybírá **přerušovacím vektorem**
  - ✓ dvojice {schránka pro úschovu čítače instrukcí, vstupní bod správce}
  - ✓ na pevném místě FAP (typicky na začátku)
  - ✓ vesměs prioritní uspořádání
  - ✓ některá přerušeni nelze maskovat (výpadek energie, ...)
- Přerušeni se používá i pro řešení výjimek
  - ✓ chybové stavy, ...

## IO cyklus řízený přerušeni, role řadiče přerušeni



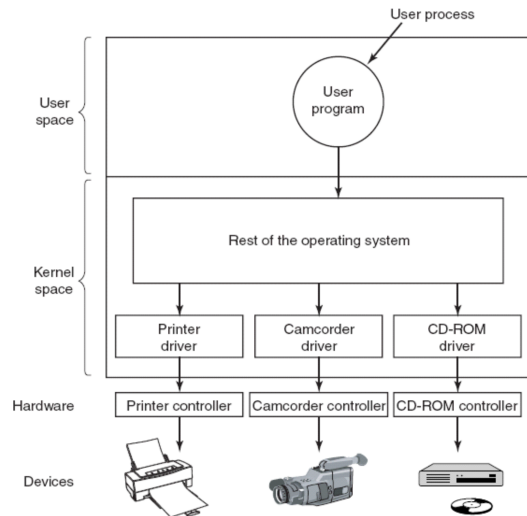
## Vektor přerušeni procesoru Intel Pentium

vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
13	general protection
14	page fault
15	(Intel reserved, do not use)
16	floating-point error
17	alignment check
18	machine check
19D31	(Intel reserved, do not use)
32D255	maskable interrupts

## Aplikační rozhraní I/O

- v generických třídách skrývá chování konkrétních zařízení volání systému (I/O)
- vrstva ovladačů ukrývá rozdílnost chování I/O řadičů pro jádro
- Mnoho odlišných typů zařízení
  - ✓ mód přenosu dat – znakové (terminál) / blokové (disk)
  - ✓ metoda přístupu – sekvenční (modem) / přímý (CD-ROM)
  - ✓ sdílené / dedikované – klávesnice / páska (disk)
  - ✓ rychlost – rotační zpoždění, vystavení, rychlost přenosu, prodlevy
  - ✓ směr přenosu – read-write (disk), read only (CD-ROM), write only (řadič grafiky)
  - ✓ ...

## Jádro a jeho I/O struktura



## Bloková a znaková zařízení

- bloková – typicky disk
  - ✓ Příkazy: read, write, seek (blok)
  - ✓ Na úrovni bazového I/O nebo na úrovni přístupu k souborům
  - ✓ možný přístup formou **Memory-Mapped File**
- znaková – klávesnice, myš, sériový port, ...
  - ✓ příkazy: get, put (character)
  - ✓ nad nimi knihovní podprogramy pro možnost řádkové editace

## Síťová zařízení

- na rozhraní se chovají jinak než bloková či znaková zařízení
- OS typu Unix / Windows obsahují rozhraní **socket**
  - ✓ separace síťových protokolů od síťových operací
  - ✓ funkcionality **klient-server**
- Mnoho různých forem modelů
  - ✓ pipes,
  - ✓ streams,
  - ✓ queues,
  - ✓ mailboxes, ...

## Hodiny a časovače

- poskytují hodnotu času
- měří uplynulý čas
- stopky
- programovatelný časovač je použitelný pro časovou synchronizaci formou periodických přerušení
- ryzí, syrové, bazové programování na úrovni ioctl (UNIX)

## Blokující a neblokující I/O

### □ blokující

- ✓ z hlediska procesu synchronní
- ✓ proces čeká na ukončení I/O
- ✓ snadné použití, snadné porozumění
- ✓ mnohdy nevyhovující

### □ neblokující

- ✓ řízení se procesu vrací bez zbytečného prodlení po vydání volání systému
- ✓ uživatelské rozhraní, vyrovnávaný I/O (buffered I/O)
- ✓ implementováno často pomocí sledů (vláken)
- ✓ promptně vrací počet načtených či vypsanych znaků

### □ asynchronní

- ✓ proces běží souběžně s I/O
- ✓ konec I/O je procesu hlášen signály (přerušeni)
- ✓ obtížnější, složitější používání

## I/O Subsystem v jádru

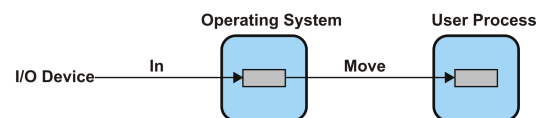
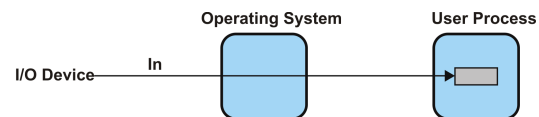
### □ Plánování

- ✓ některá I/O zařízení požadují řazení I/O požadavků do front na zařízení
- ✓ některé OS se snaží o spravedlnost

## I/O Subsystem v jádru

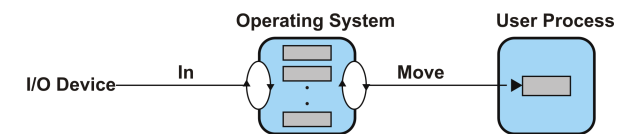
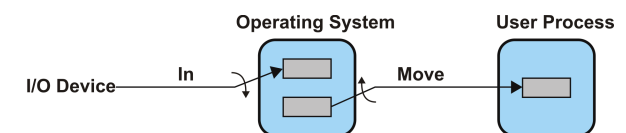
### □ vyrovnávání, buffering

- ✓ ukládání dat v paměti v době přenosu k/e zařízení
- ✓ řeší se rozdílnost rychlostí
- ✓ řeší se rozdílnost datových jednotek



(b) Single buffering

## Vyrovnávání, buffering



(d) Circular buffering



## I/O Subsystém v jádru, 2

- Caching
  - ✓ rychlá paměť udržující kopii dat
  - ✓ vždy pouze kopii
  - ✓ klíč k dosažení vysokého výkonu
- Spooling
  - ✓ udržování fronty dat určených k výpis na zařízení
  - ✓ pokud zařízení může vyřizovat požadavek po požadavku
  - ✓ typicky – tiskárna
- rezervace zařízení
  - ✓ exkluzivita přístupu k zařízení pro proces
  - ✓ rezervace / uvolnění – volání systému
  - ✓ ochrana proti uváznutí !

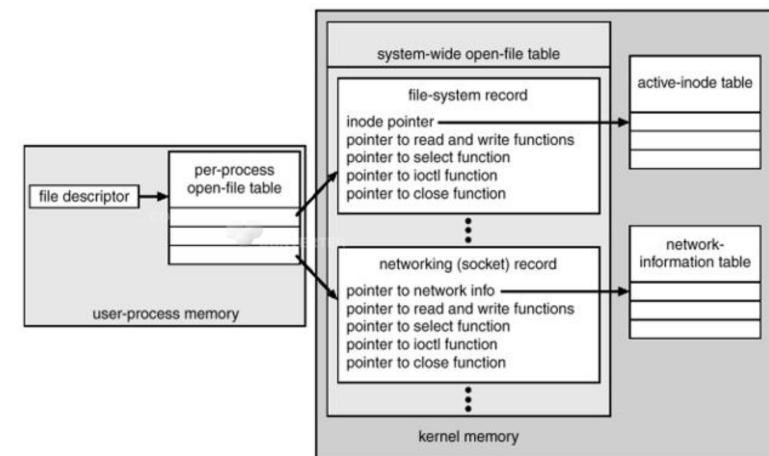
## Chybové řízení

- vzpamatování se po poruše při chybě čtení z disku, zjištění nedostupnosti zařízení, po náhodné zápisové chybě, ...
- typické chybové hlášení
  - ✓ záporná funkční hodnota poskytnutá voláním systému
- udržuje se záznamník o chybách v systému
  - ✓ System error logs

## Datové struktury jádra

- Jádro udržuje stavovou informaci o komponentách I/O
  - ✓ tabulky otevřených souborů, síťových spojení, stavů znakových zařízení, ...
- mnoho složitých datových struktur sleduje využívání vyrovnávacích pamětí, alokaci paměti, vadné bloky, ...
- Některá jádra používají pro implementaci I/O objektově orientované metody a předávání zpráv

## Struktura I/O části jádra UNIX



## Filozofie provedení IO požadavku

- Uvažme čtení souboru z disku procesem
  - ✓ určí zařízení uchovávající soubor
  - ✓ přelož jméno na reprezentaci zařízení
  - ✓ přečti data z disku fyzicky do vyrovnávací paměti
  - ✓ zpřístupni data volajícímu procesu
  - ✓ vrať řízení procesu

## Životní cyklus I/O požadavku

