



PB153

Operační systémy a jejich rozhraní

I/O systém



Hardware

- HW pro I/O je značně rozmanitý
- Existují však určité běžně používané prvky
 - port
 - sběrnice (bus)
 - řadič (host adapter, controller)
- I/O zařízení jsou řízeny I/O instrukcemi (IN, OUT)
- Adresy I/O zařízení
 - uváděné přímo v I/O instrukcích (např. IN AL, DX : DX port, AL získaný bajt)
 - I/O se mapuje na přístup k paměti (např. grafická karta, videopaměť)
- Základní způsoby ovládání I/O
 - polling, programované I/O operace
 - aktivní čekání na konec operace
 - přerušení
 - DMA

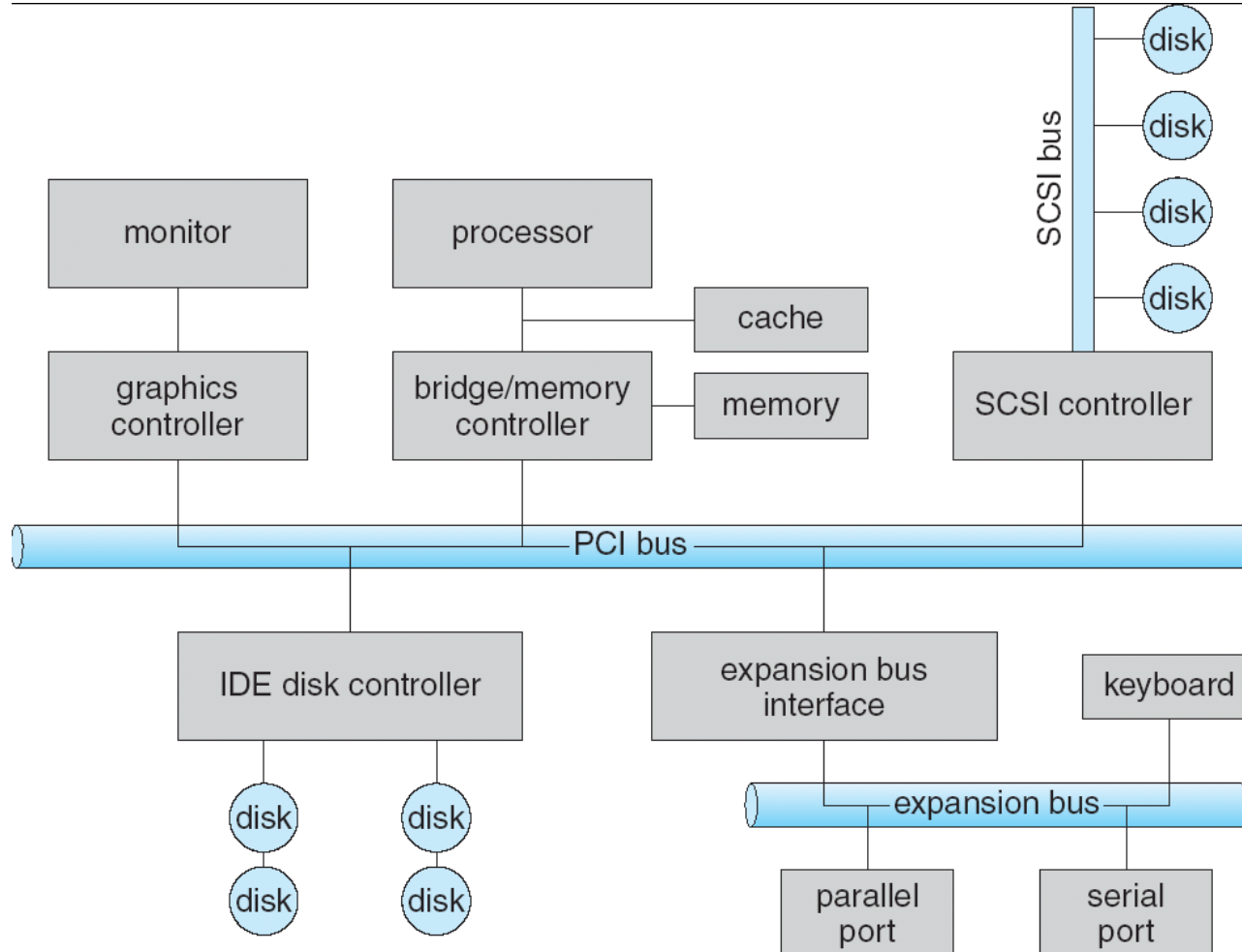
Příklad: instrukce IN

IN—Input from Port

Opcode	Instruction	Op/ En	64-Bit Mode	Compat/ Leg Mode	Description
E4 <i>ib</i>	IN AL, <i>imm8</i>	A	Valid	Valid	Input byte from <i>imm8</i> I/O port address into AL.
E5 <i>ib</i>	IN AX, <i>imm8</i>	A	Valid	Valid	Input word from <i>imm8</i> I/O port address into AX.
E5 <i>ib</i>	IN EAX, <i>imm8</i>	A	Valid	Valid	Input dword from <i>imm8</i> I/O port address into EAX.
EC	IN AL,DX	B	Valid	Valid	Input byte from I/O port in DX into AL.
ED	IN AX,DX	B	Valid	Valid	Input word from I/O port in DX into AX.
ED	IN EAX,DX	B	Valid	Valid	Input doubleword from I/O port in DX into EAX.

Copies the value from the I/O port specified with the second operand (source operand) to the destination operand (first operand). The source operand can be a byte-immediate or the DX register; the destination operand can be register AL, AX, or EAX, depending on the size of the port being accessed (8, 16, or 32 bits, respectively). Using the DX register as a source operand allows I/O port addresses from 0 to 65,535 to be accessed; using a byte immediate allows I/O port addresses 0 to 255 to be accessed.

Sběrnice PC





Rozmístění I/O portů v PC

I/O address range (hexadecimal)	device
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)



I/O porty

- I/O port se obvykle skládá ze 4 registrů
 - Data-in
 - Čtení vstupu od zařízení
 - Data-out
 - Pro zápis výstupu do zařízení
 - Status
 - Aktuální stav (data připravena, chyba, ...)
 - Control
 - Ovládání zařízení, konfigurace, příkazy,...

● ● ● | Příklad: sériový port

- COM1 – porty od 3F8, COM2 – porty od 2F8

8250 SCC Registers

I/O Address (hex)	Description
3F8/2F8	Receive/Transmit data register. Also the L.O. byte of the Baud Rate Divisor Latch register.
3F9/2F9	Interrupt Enable Register. Also the H.O. byte of the Baud Rate Divisor Register.
3FA/2FA	Interrupt Identification Register (read only).
3FB/2FB	Line Control Register.
3FC/2FC	Modem Control Register.
3FD/2FD	Line Status Register (read only).
3FE/2FE	Modem Status Register (read only).
3FF/2FF	Shadow Receive Register (read only not available on original PCs).

Techniky provádění I/O

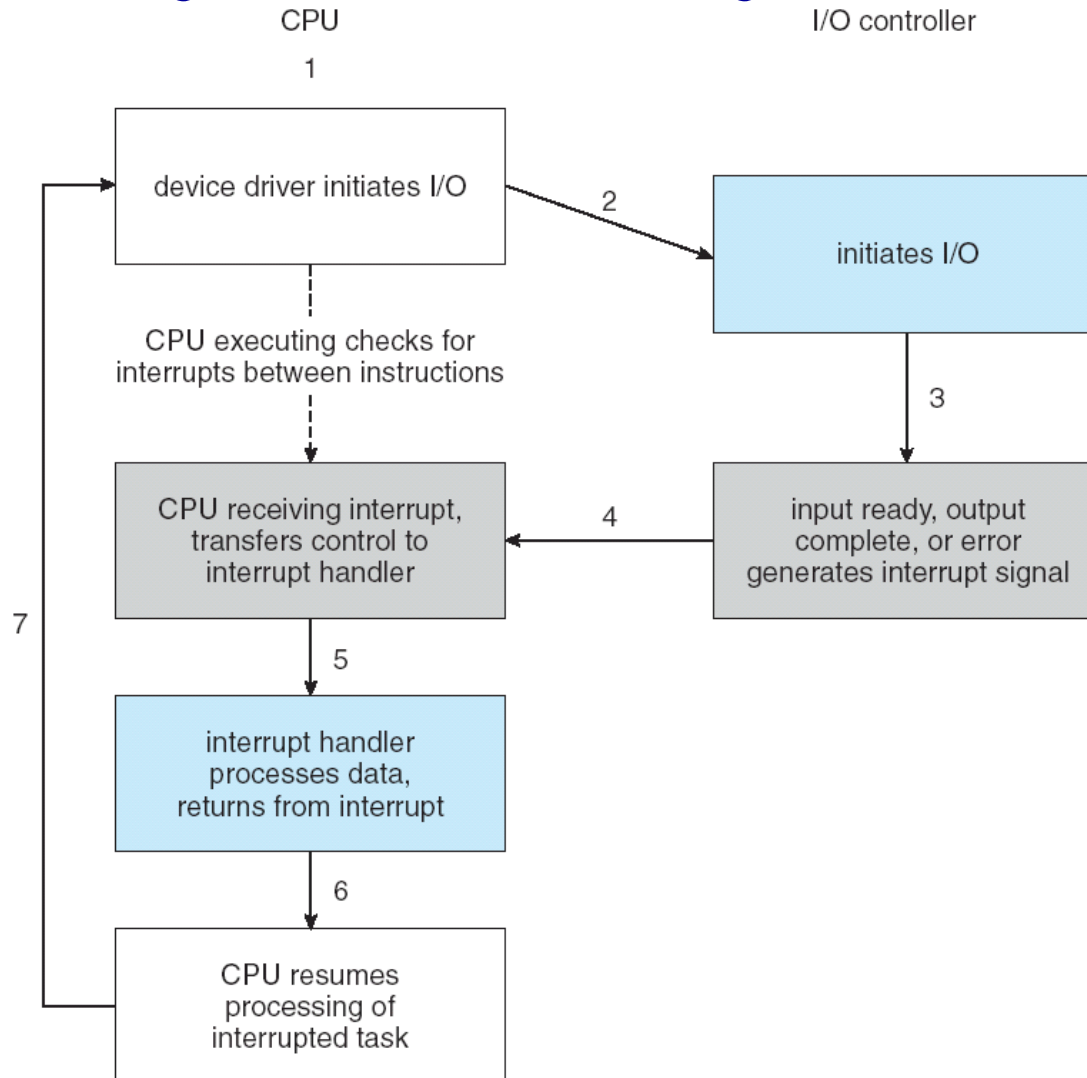
- Programovaný I/O (busy-waiting)
 - opakovaně se ptám na stav zařízení
 - připraven
 - pracuje
 - chyba
- I/O řízený přerušením
 - zahájení I/O pomocí I/O příkazu
 - paralelní běh I/O s během procesoru
 - I/O modul oznamuje přerušením konec přenosu
- Direct Memory Access (DMA)
 - kopírování bloků mezi pamětí a I/O zařízením na principu kradení cyklů paměti
 - přerušlení po přenosu bloku (indikace konce)

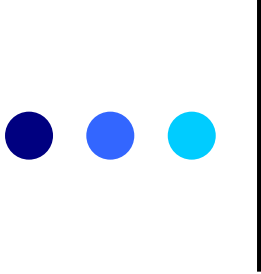


Přerušeni

- Přerušeni obsluhuje ovladač přerušeni (kód OS)
- Maskováním lze některá přerušeni ignorovat nebo oddálit jejich obsluhu
- Patřičný ovladač přerušeni se vybírá přerušovacím vektorem
 - některá přerušeni nelze maskovat
 - přerušeni mohou být uspořádána podle priorit
- Přerušeni se používá i pro řešení výjimek (nejsou asynchronní)

I/O cyklus řízený přerušením





Vektor přerušení procesoru Intel Pentium

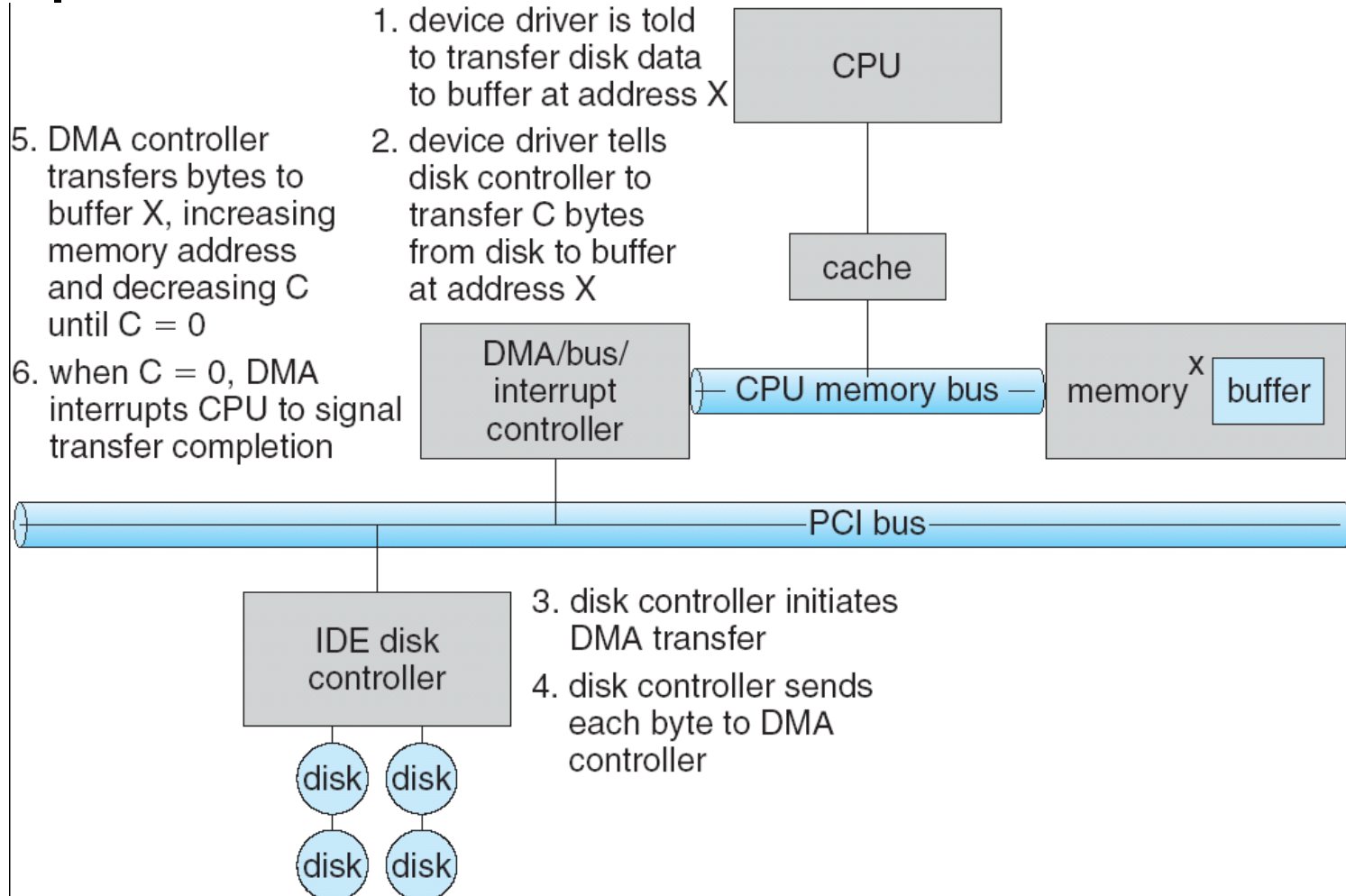
vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
13	general protection
14	page fault
15	(Intel reserved, do not use)
16	floating-point error
17	alignment check
18	machine check
19–31	(Intel reserved, do not use)
32–255	maskable interrupts



DMA

- Přímý přístup do paměti (Direct Memory Access - DMA)
 - nahrazuje programovaný I/O při velkých přesunech dat
 - vyžaduje speciální DMA řadič
 - při přenosu dat se obchází procesor, přístup do paměti zajišťuje přímo DMA řadič
 - procesor a DMA soutěží o přístup k paměti

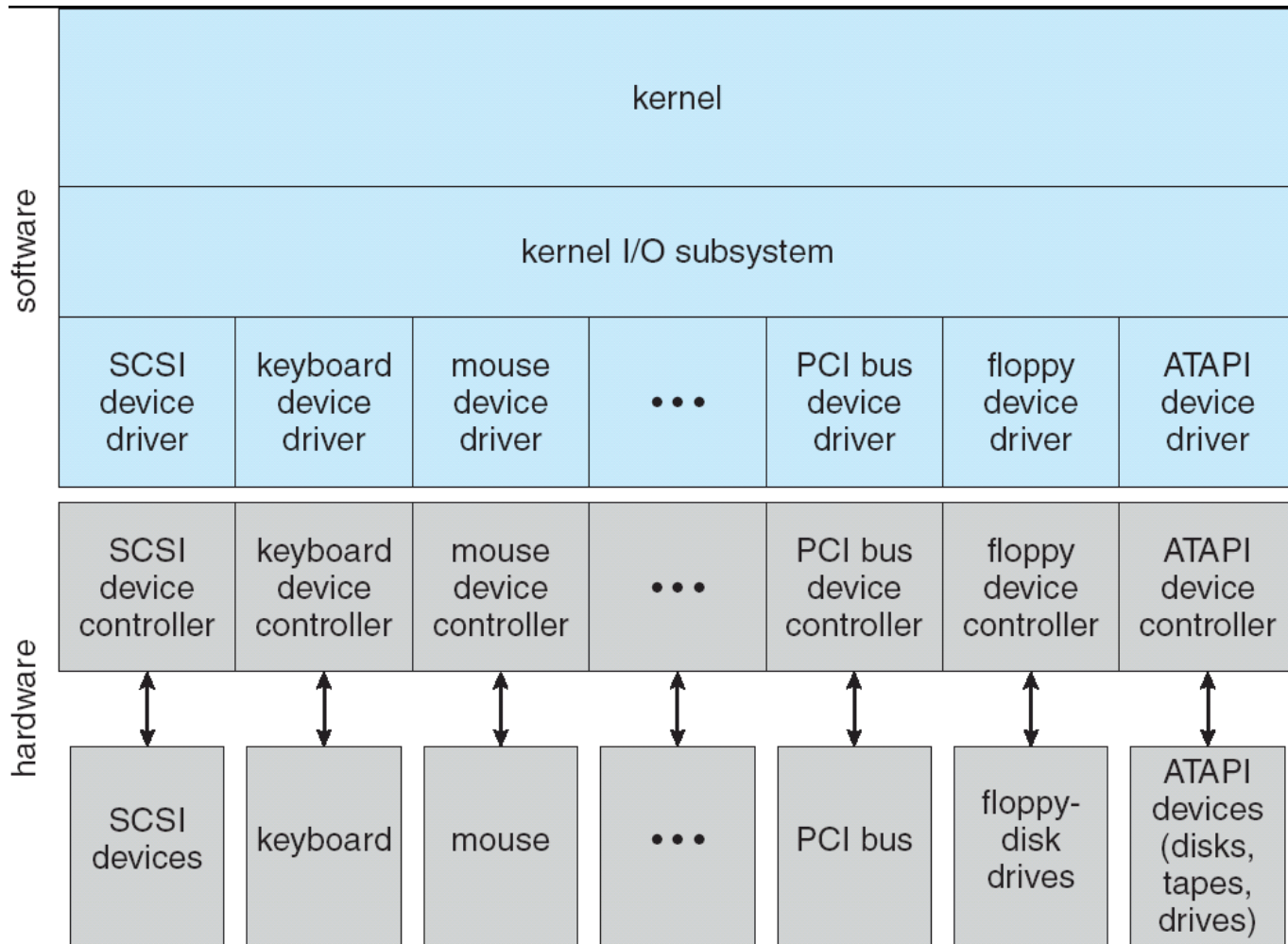
DMA: příklad



Aplikační rozhraní I/O

- Jádro OS se snaží skrýt rozdíly mezi I/O zařízeními a programátorům poskytuje jednotné rozhraní
- Dále vrstva ovladačů ukrývá rozdílnost chování I/O řadičů i před některými částmi jádra
- Některé vlastnosti I/O zařízení
 - mód přenosu dat: znakové (terminál) / blokové (disk)
 - způsob přístupu: sekvenční (modem) / přímý (disk)
 - sdílené/dedikované: klávesnice / páska
 - rychlost přenosu: vystavení, přenos, ...
 - read-write, read only, write only

I/O v jádře a HW





Bloková a znaková zařízení

- Bloková zařízení – typicky disk
 - příkazy: read, write, seek
 - logický způsob přístupu: obecný I/O nebo souborový systém
 - možný přístup formou souboru mapovaného do paměti
- Znaková – klávesnice, myš, sériový port
 - příkazy: get, put
 - nad nimi knihovny podprogramy pro další možnosti (např. řádková editace)



Příklad: Linux

- Mapování souboru do paměti

```
#include <sys/mman.h>
```

```
void *mmap(void *start, size_t length, int prot, int flags, int fd, off_t offset);
```

```
int munmap(void *start, size_t length);
```

The `mmap()` function asks to map `length` bytes starting at offset `offset` from the file (or other object) specified by the file descriptor `fd` into memory, preferably at address `start`. The `prot` argument describes the desired memory protection (and must not conflict with the open mode of the file). It is either `PROT_NONE` or is the bitwise OR of one or more of the other `PROT_*` flags...

Síťová zařízení

- Přístup k nim se značně liší jak od znakových, tak od blokových zařízení
 - proto mívají samostatné rozhraní OS
- Unix i Windows obsahující rozhraní nazývané „sockets“
 - separují síťové protokoly od síťových operací
 - přístup jako k souborům (včetně funkce *select*)
- Existuje celá řada přístupů k síťovým službám
 - Pipes (roury), FIFOs, streams, queues, mailboxes



Blokující a neblokující I/O

- Blokující
 - z hlediska procesu synchronní
 - proces čeká na ukončení I/O
 - snadné použití (programování), snadné porozumění (po provedení operace je hotovo to co jsem požadoval)
 - někdy však není dostačující (z důvodu efektivity)
- Neblokující
 - řízení se procesu vrací co nejdříve po zadání požadavku
 - vhodné pro uživatelské rozhraní, bufferovaný I/O
 - bývá implementováno pomocí vláken
 - okamžitě vrací počet načtených či zapsaných znaků
- Asynchronní
 - proces běží souběžně s I/O
 - konec I/O je procesu hlášen signály
 - obtížné na programování, složité používání, ale v případě vhodně promyšleného programu velice efektivní

● ● ● | Příklad: Linux

○ Blokující čtení

- `ssize_t read(int fd, void *buf, size_t count);`

○ Neblokující volání

- `int select(int nfd, fd_set *readfds, fd_set *writefds, fd_set *exceptfds, struct timeval *timeout);`

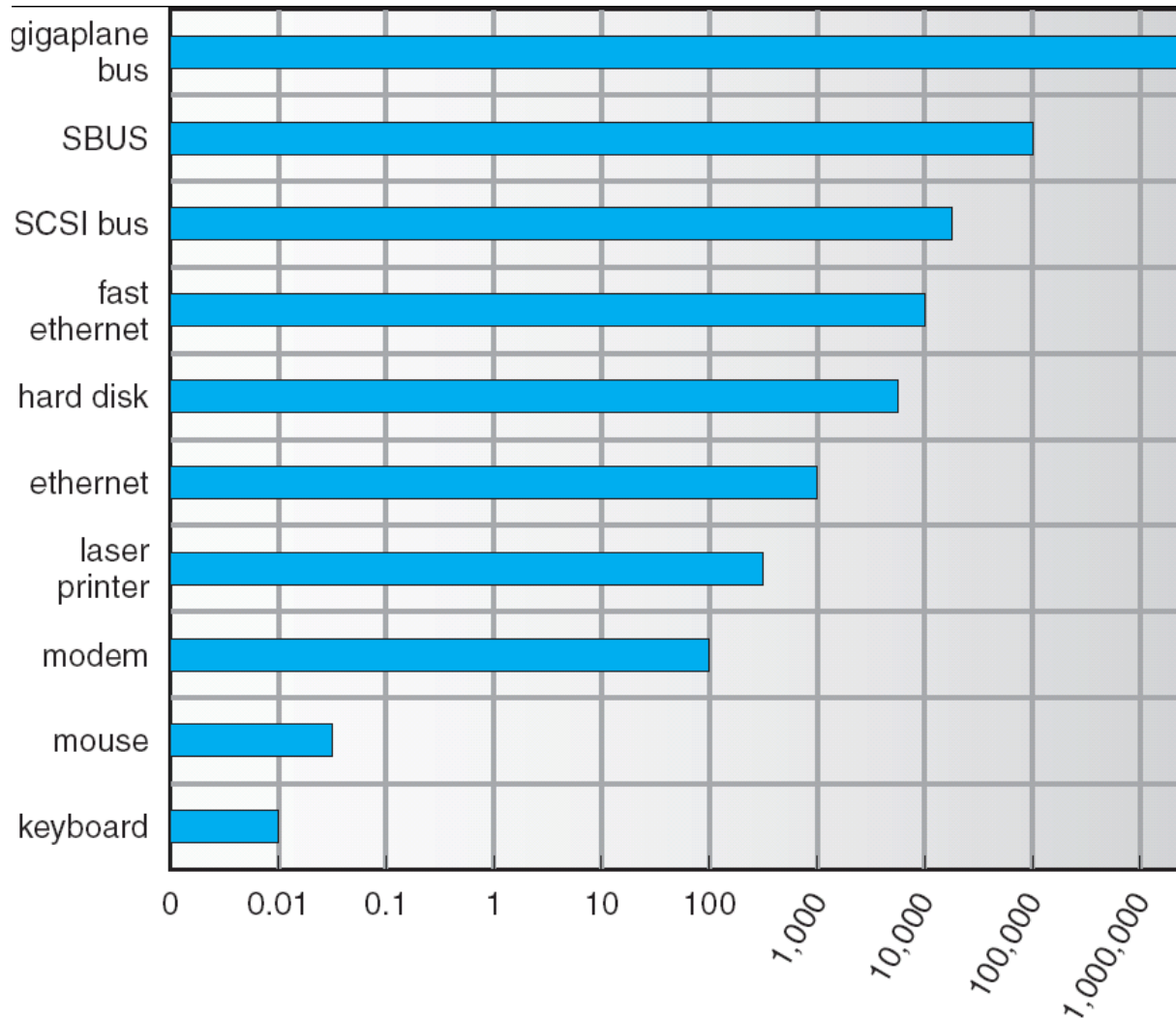
○ Asynchronní čtení

- `int aio_read(struct aiocb *aiocbp);`
- `int aio_error(const struct aiocb *aiocbp);`
- `ssize_t aio_return(struct aiocb *aiocbp);`

I/O subsystém v jádru

- Plánování
 - některé I/O operace požadují řazení do front na zařízení
 - některé OS se snaží o „spravedlnost“
- Vyrovnání (vyrovnávací paměti), buffering
 - ukládání dat v paměti v době přenosu k/ze zařízení
 - řeší rozdílnost rychlosti
 - řeší rozdílnost velikosti datových jednotek

Příklad: Sun Enterprise 6000





I/O Subsystém v jádru

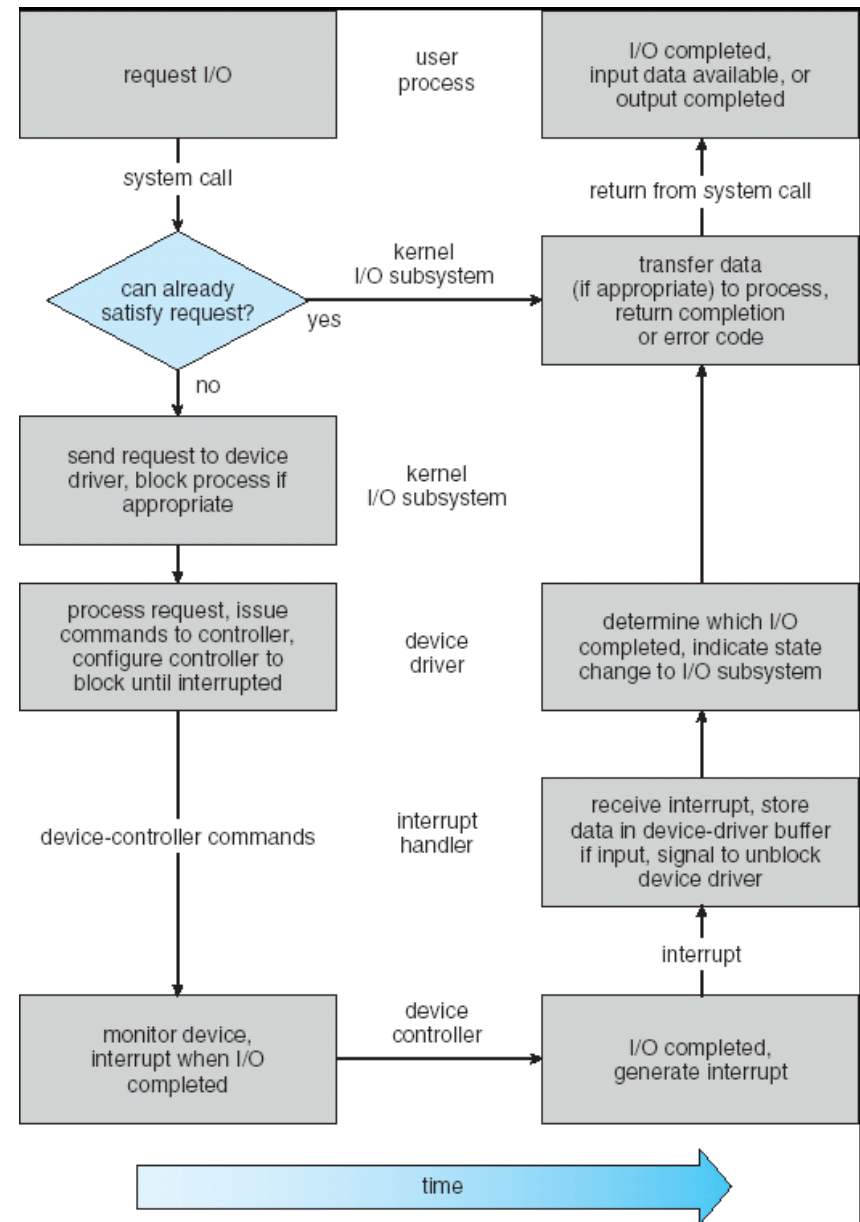
- Caching
 - rychlá paměť udržuje kopii dat
 - vždy pouze *kopii*
 - caching je klíčem k dosažení vysokého výkonu
- Spooling
 - udržování *fronty dat* určených k výpis na zařízení
 - pokud zařízení může vyřizovat požadavky pouze sekvenčně
 - typicky tiskárna
- Rezervace zařízení
 - exkluzivita přístupu k zařízení pro proces
 - rezervace / uvolnění – volání systému
 - pozor na uváznutí (deadlock)

● ● ● | Chybové řízení

- Vzpamatování se po poruše při chybě čtení z disku, zjištění nedostupnosti zařízení, po náhodné chybě zápisu, ...
- Volání požadující I/O operaci získá číslo chyby
 - typicky záporná hodnota
- Udržuje se záznam o chybách v systému
 - pro následné analýzy
 - syslog, events (viewer), ...

Př.: I/O pož.

- Příklad čtení souboru z disku:
 1. Zjistíme, které zařízení obsahuje soubor
 2. Převédeme jméno na reprezentaci na zařízení
 3. Fyzicky přečteme data z disku do bufferu
 4. Získaná data zpřístupníme procesu
 5. Vratíme řízení procesu

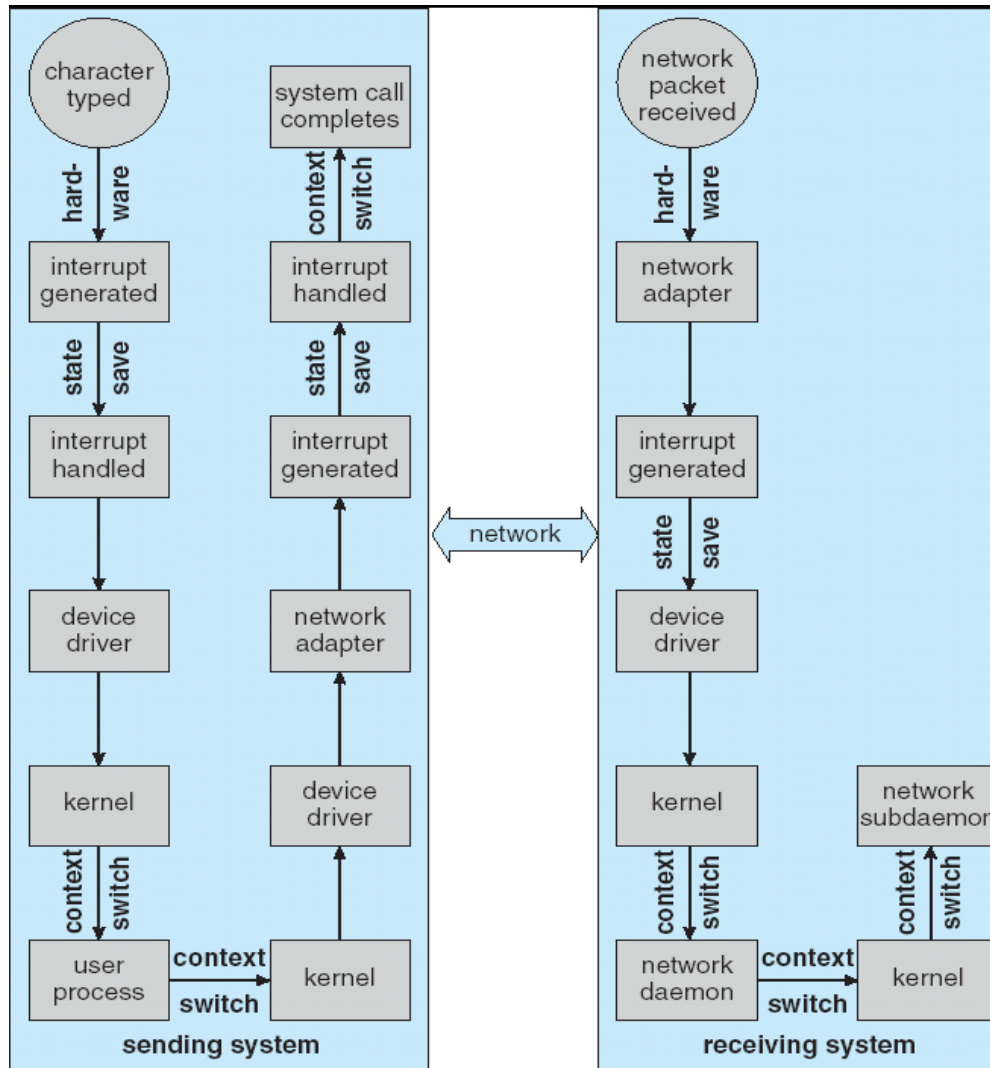




Výkon

- I/O je nejvýznamnějším faktorem výkonu celého systému
 - CPU musí provádět ovladače a programy I/O části jádra
 - při přerušení se přepíná kontext
 - provádí se kopírování dat
 - zvláště významný je síťový provoz

Příklad: Síťová aplikace





Zvyšování výkonu

- Omezujeme počet přepnutí kontextu
- Omezujeme zbytečné kopírování dat
- Omezujeme počet přerušení tím, že přenášíme delší bloky
- Využíváme všech výhod (funkcí) moderních řadičů
- Používáme co nejvíce DMA
- Všechny komponenty kombinujeme s cílem dosažení co nejvyšší propustnosti
 - CPU, paměť, sběrnice, I/O zařízení