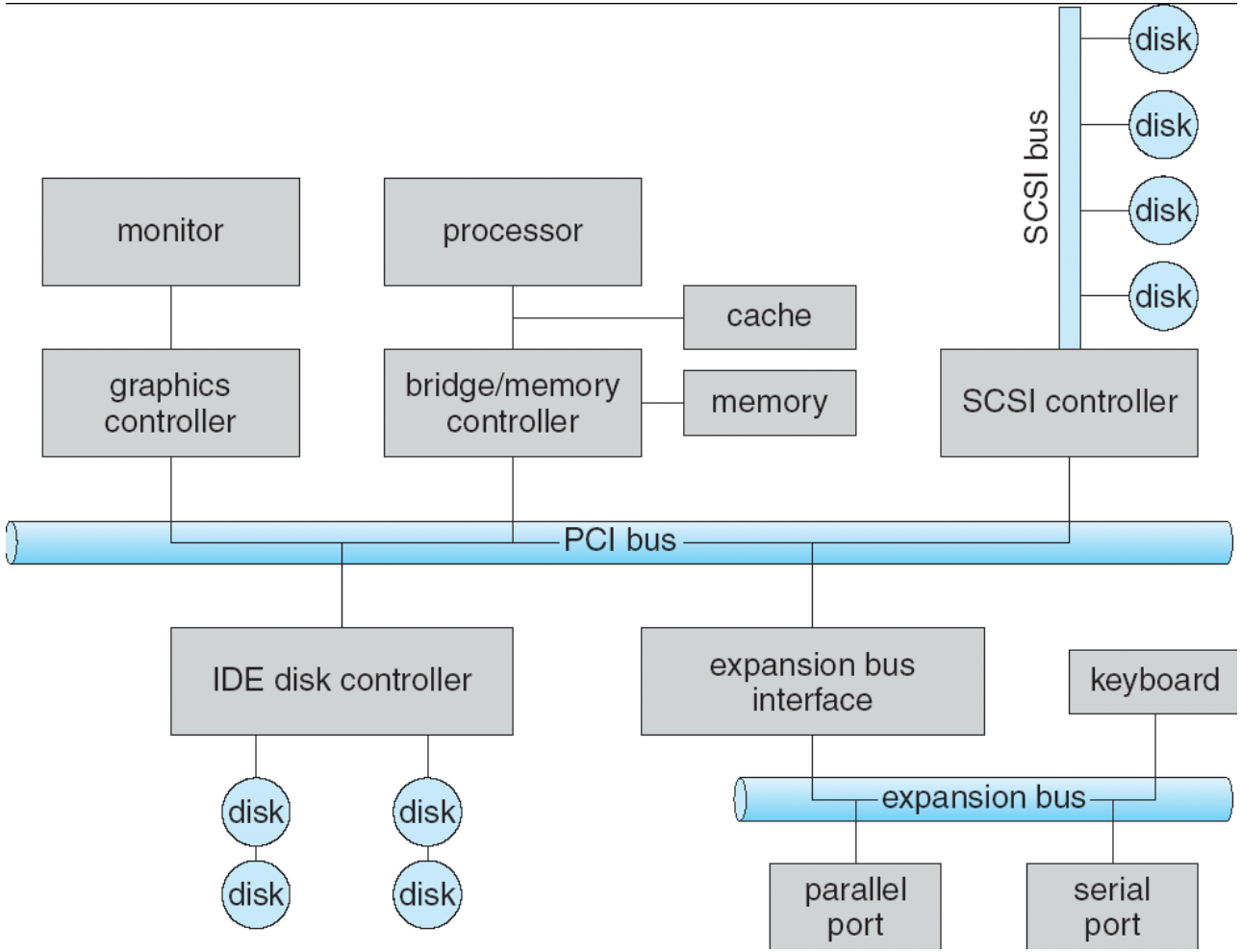
PB153
Operační systémy a jejich rozhraní

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

Sběrnice PC


disk
disk
disk
disk
SCSI bus
monitor
processor
cache
graphics controller
bridge/memory controller
memory
SCSI controller
PCI bus
IDE disk controller
expansion bus interface
keyboard
disk
disk
disk
disk
expansion bus
parallel port
serial port

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)

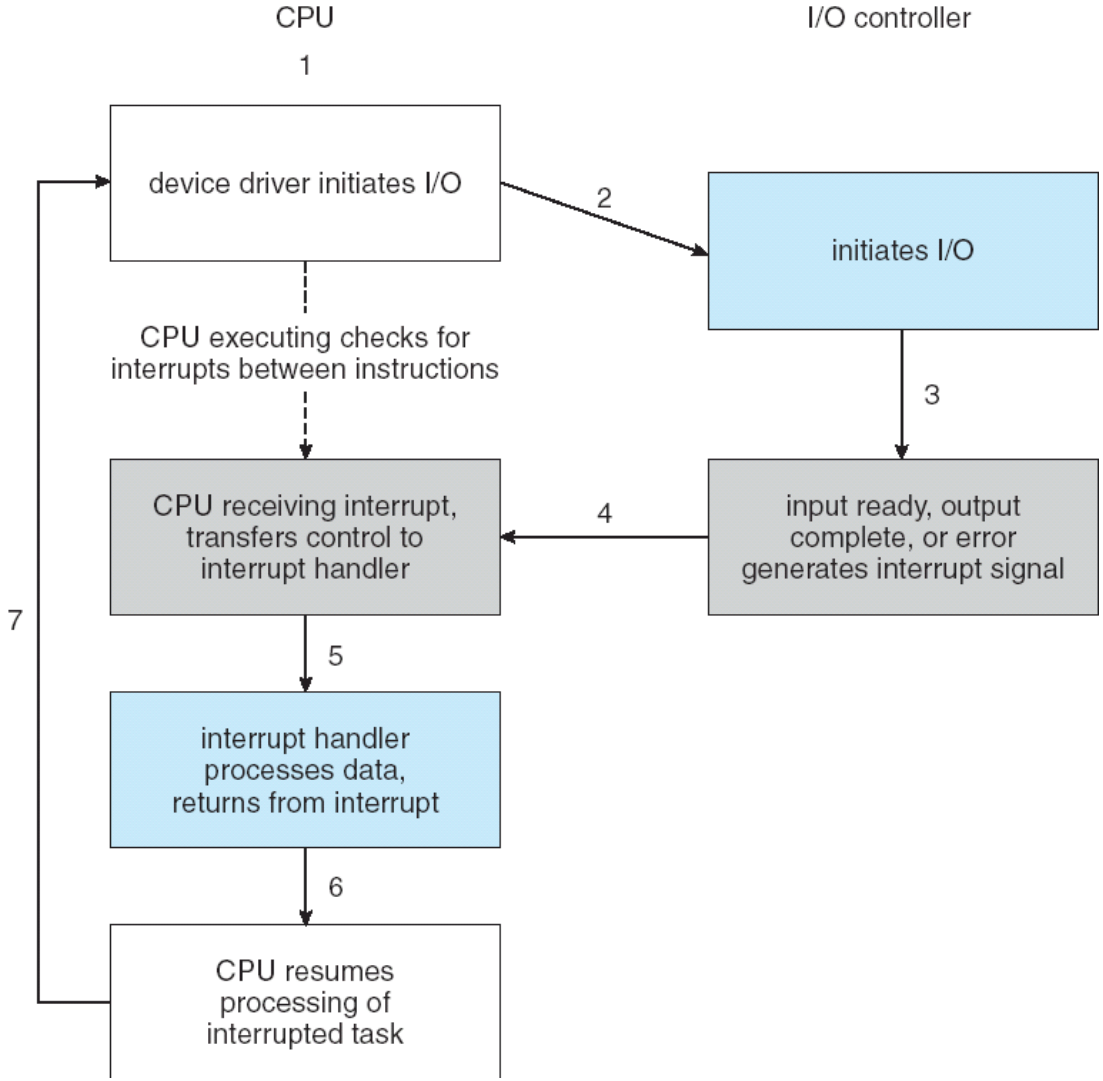
Techniky provádění I/O

- 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
  - přerušení po přenosu bloku (indikace konce)

Přerušení

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

I/O cyklus řízený přerušením



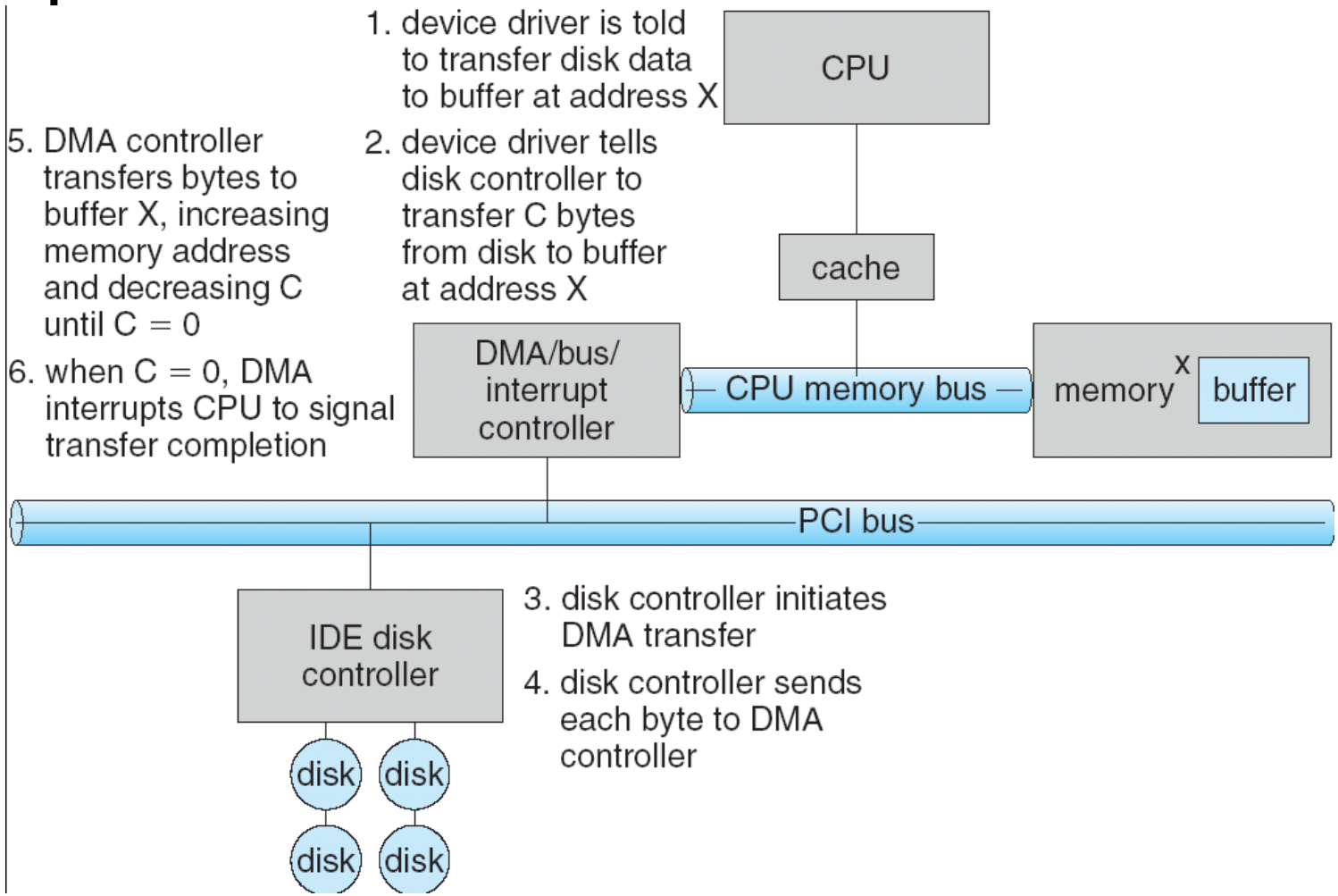
Vektor přerušení procesoru Intel Pentium

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

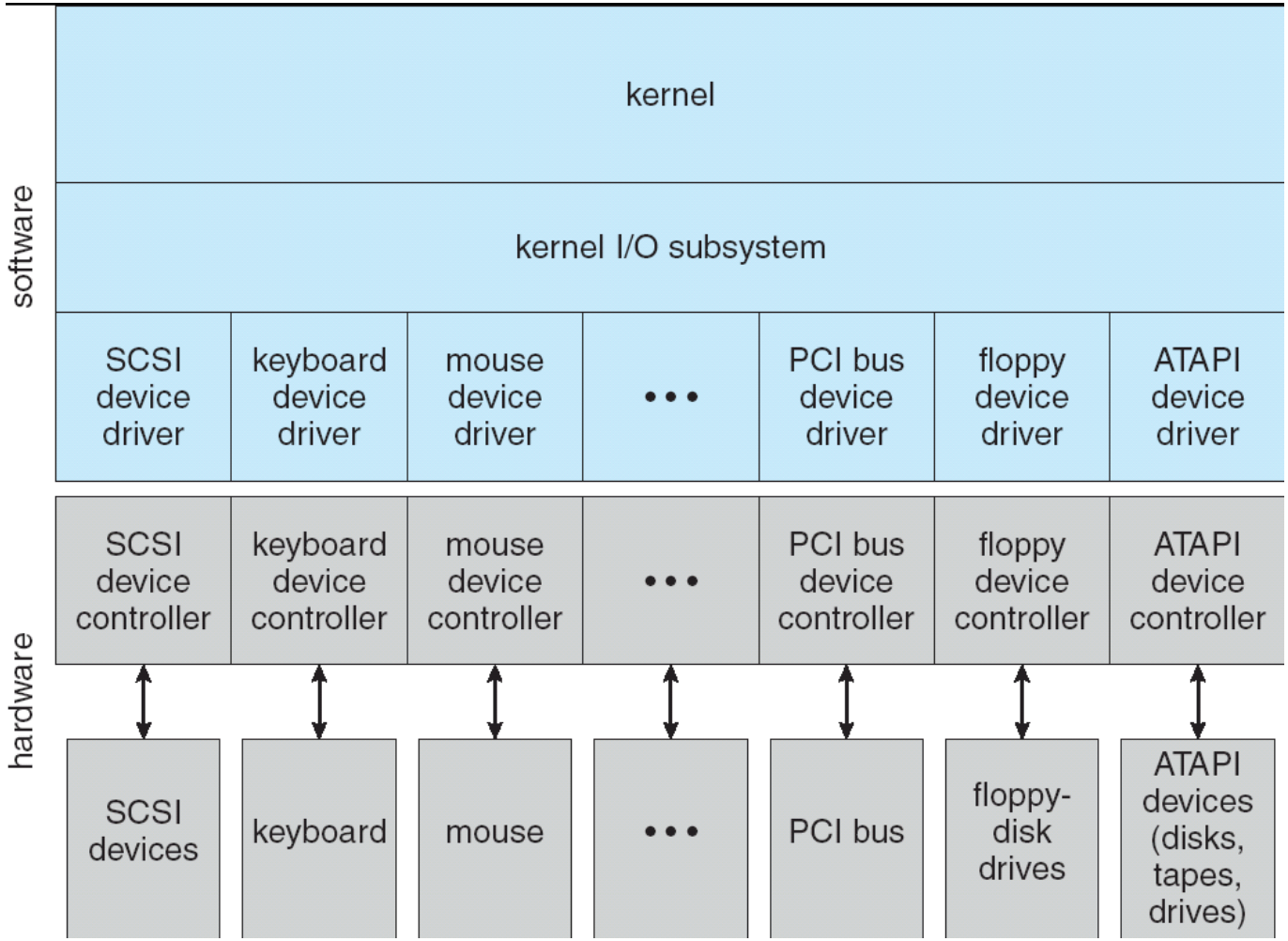
DMA: příklad


1. device driver is told to transfer disk data to buffer at address X
2. device driver tells disk controller to transfer C bytes from disk to buffer at address X
3. disk controller initiates DMA transfer
4. disk controller sends each byte to DMA controller
5. DMA controller transfers bytes to buffer X, increasing memory address and decreasing C until C = 0
6. when C = 0, DMA interrupts CPU to signal transfer completion
CPU
cache
DMA/bus/ interrupt controller
CPU memory bus
memory
x
buffer
PCI bus
IDE disk controller
disk
disk
disk
disk

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 knihovní podprogramy pro další možnosti (např. řádková editace)

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

Hodiny a časovače

- Poskytují skutečný čas, měří uplynulý čas, fungují jako časovače
- Nízkoúrovňové programování pomocí *ioctl* (pod Unixem) umožňuje přístup k časovačům a hodinám

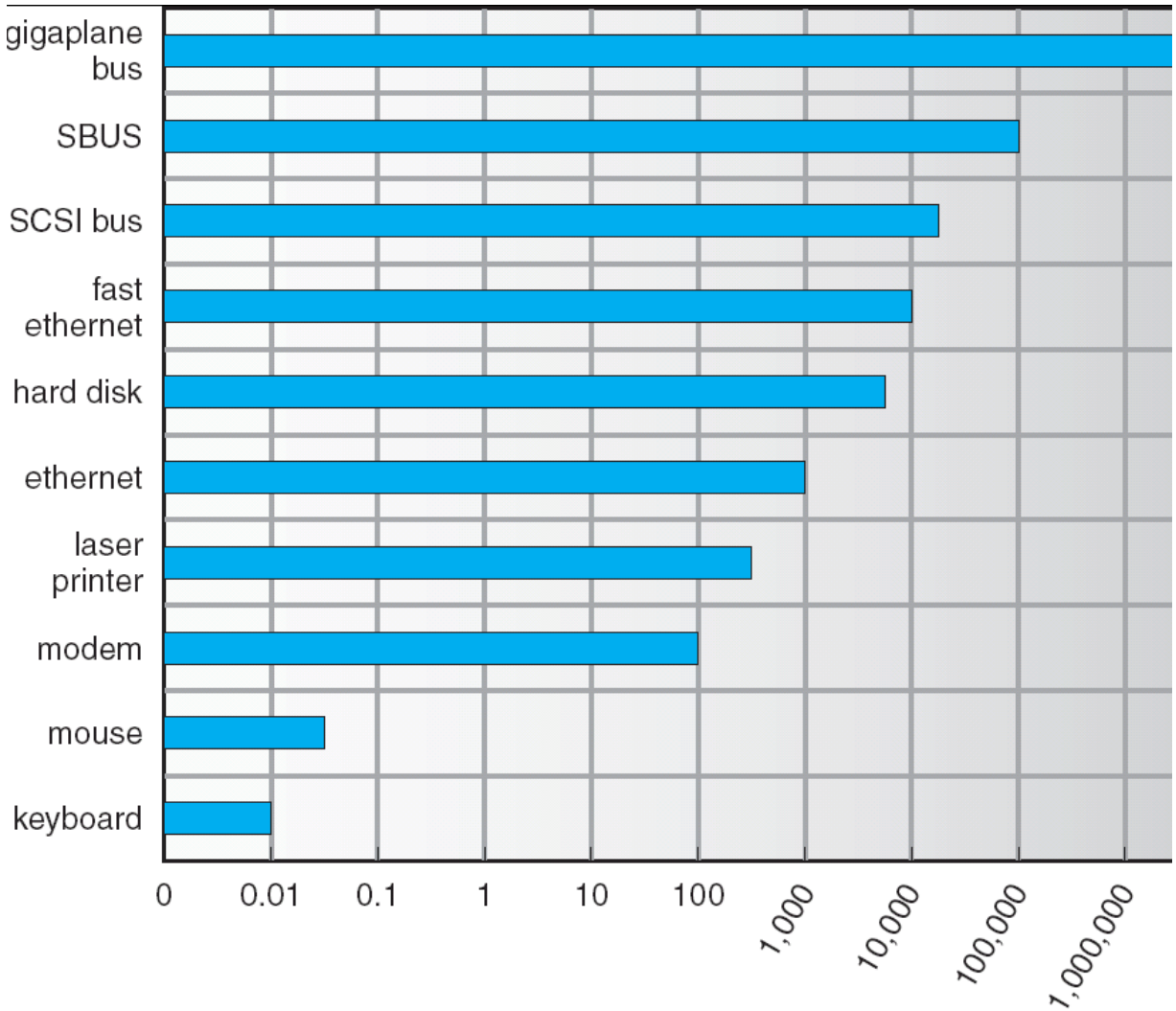
Blokující a neblokující I/O

- Blokující
  - z hlediska procesu synchronní
  - proces čeká na ukončení I/O
  - snadné použití (programovaní), 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 programovaní, složité používání, ale v případě vhodně promyšleného programu velice efektivní

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)

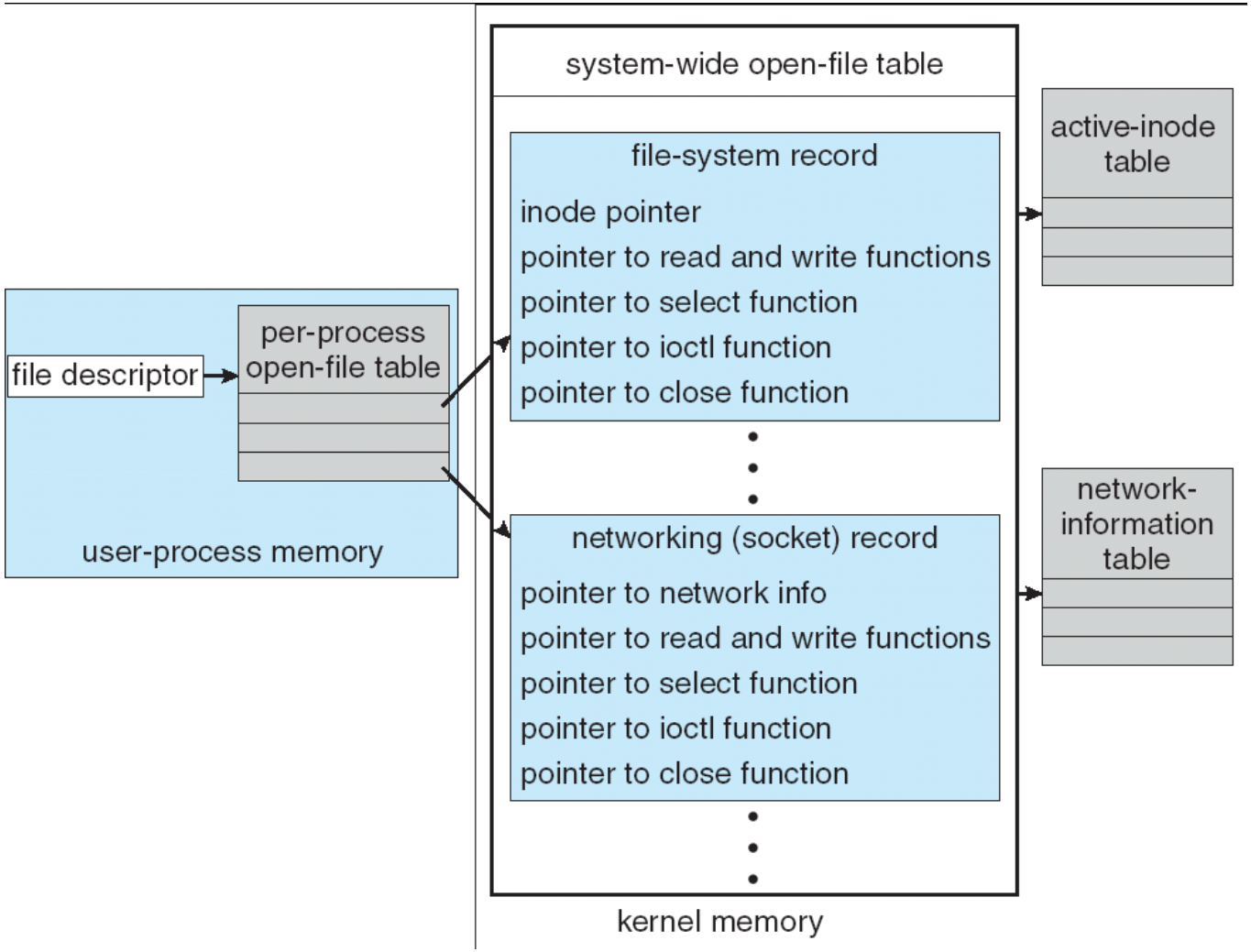
Chybové řízení

- 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), ...

Datové struktury jádra

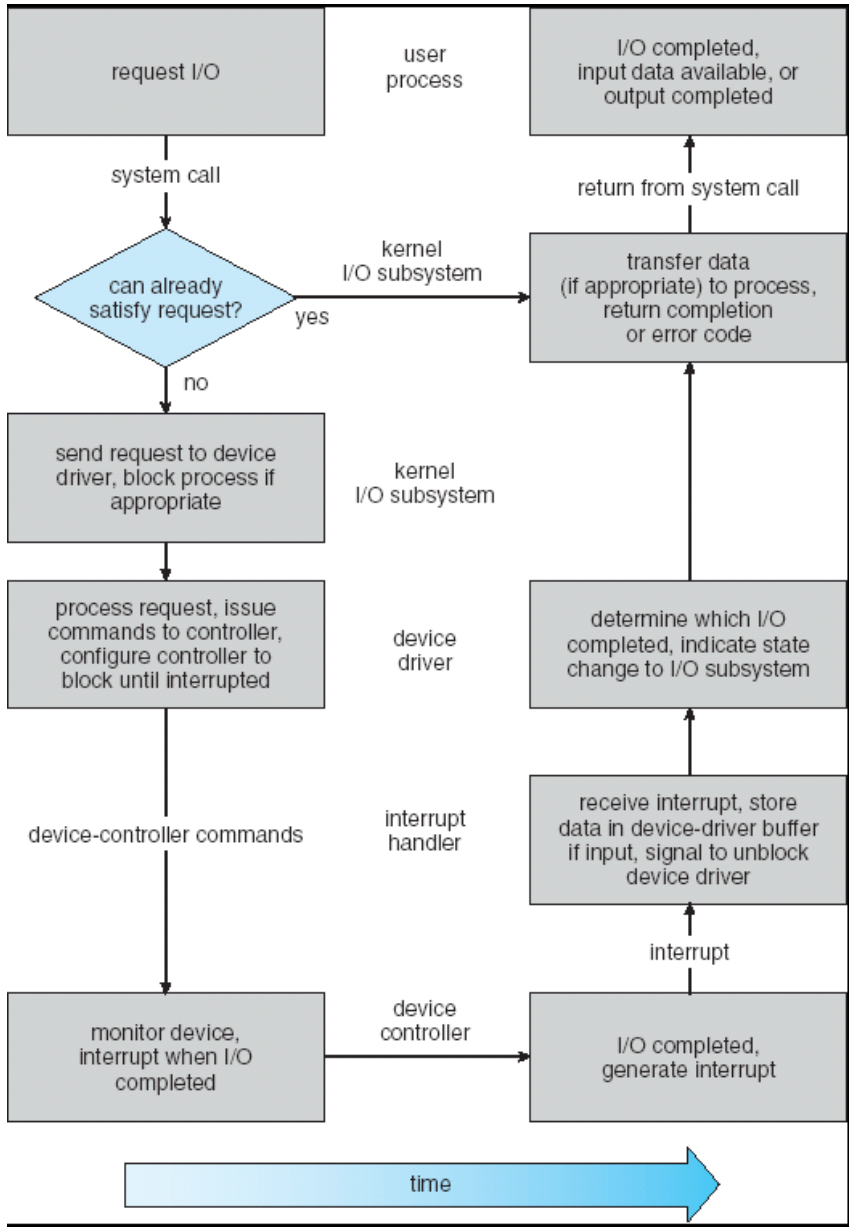
- Jádro udržuje stavovou informaci o komponentách I/O
  - tabulky otevřených souborů, síťových spojení, stavů znakových zařízení, …
- Mnoho složitých datových struktur sleduje využívání vyrovnávacích pamětí, alokaci paměti, změněné a nezapsané bloky, …
- Některá jádra používají pro implementaci I/O objektově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

I/O část jádra UNIXu



Př.: I/O pož.

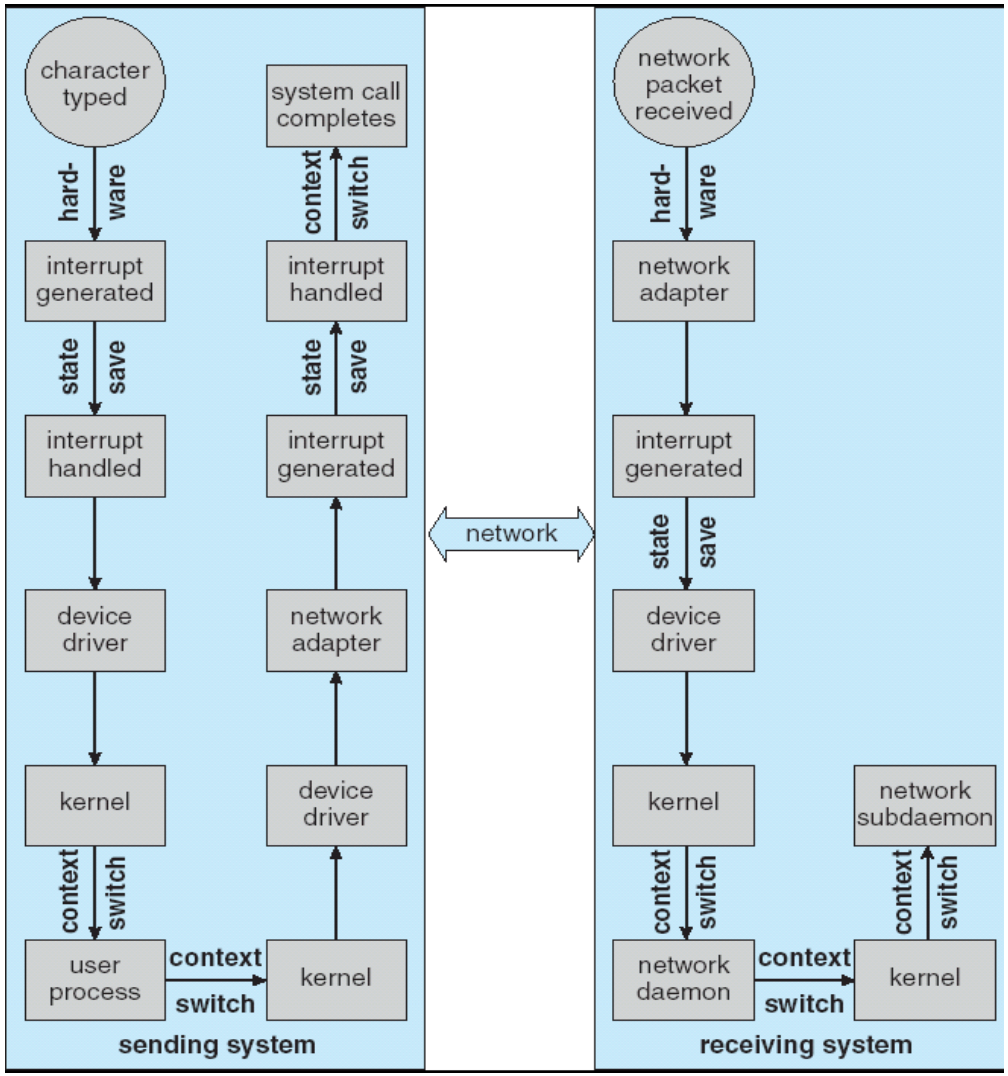
- Příklad čtení souboru z disku:
  1. Zjistíme, které zařízení obsahuje soubor
  2. Převedeme jméno na reprezentaci na zařízení
  3. Fyzicky přečteme data z disku do bufferu
  4. Získaná data zpřístupníme procesu
  5. Vrátíme řízení procesu



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