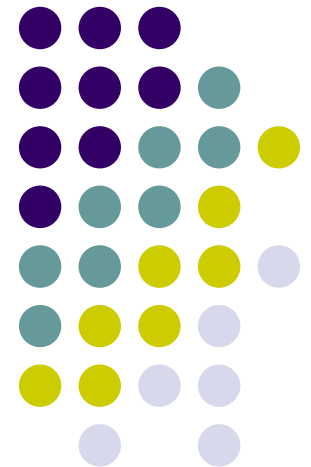


12

Input / Output



Sistem I/O



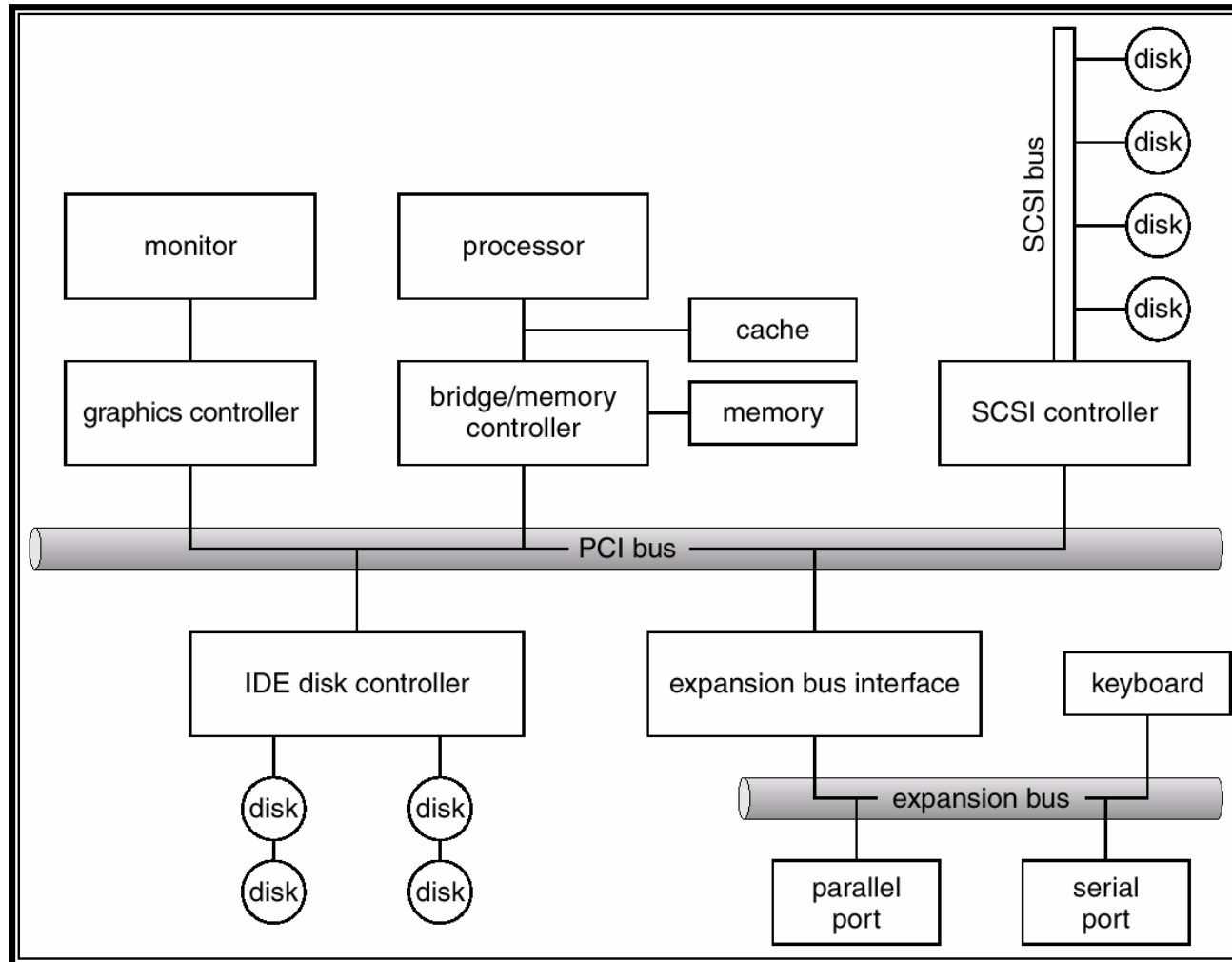
- Hardware I/O
- Interface Aplikasi I/O
- Subsystem Kernel I/O
- Transformasi Permintaan (Request) I/O ke Operasi Hardware
- Stream
- Unjuk Kerja



Hardware I/O

- Beragam variasi perangkat I/O
- Konsep Umum
 - Port
 - Bus (*daisy chain* atau *shared direct access*)
 - Controller (host adapter)
- Device mengendali instruksi I/O
- Device memiliki alamat, yang digunakan oleh :
 - Direct I/O instructions
 - Memory-mapped I/O

Struktur PC Bus



Lokasi Port Device I/O pada PC (partial)



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)

Polling



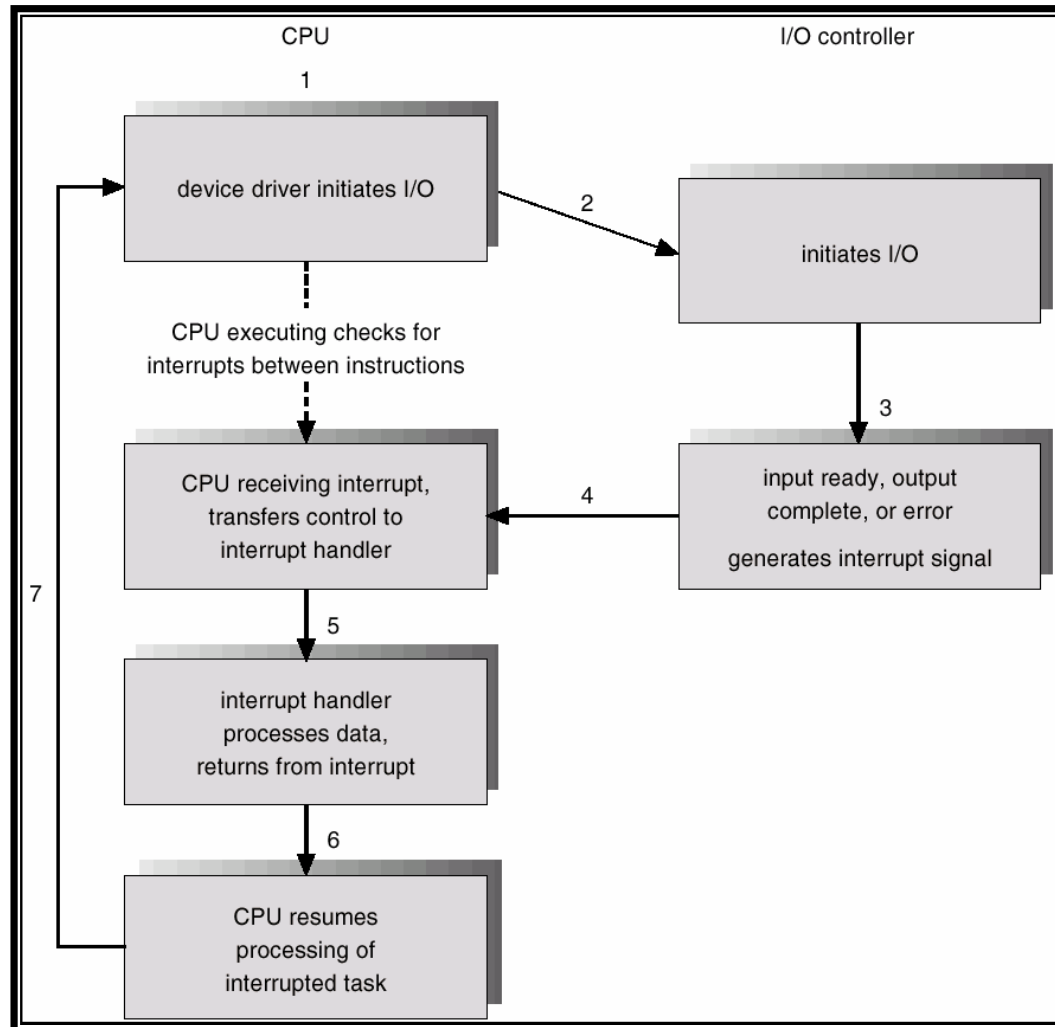
- Menentukan status device
 - `command-ready`
 - `busy`
 - `Error`
- Siklus busy wait ke wait untuk I/O dari device

Interrupt



- Permintaan interrupt CPU dipicu oleh I/O device
- Interrupt handler menerima interrupt
- Maskable interrupt untuk membatalkan atau menunda beberapa interrupt
- Interrupt vector untuk mengirimkan interrupt ke handler yang benar.
 - Berdasarkan prioritas
 - Beberapa unmaskable
- Mekanisme interrupt juga digunakan untuk exception

Siklus Interrupt-Driven I/O



Intel Pentium Processor Event-Vector Table



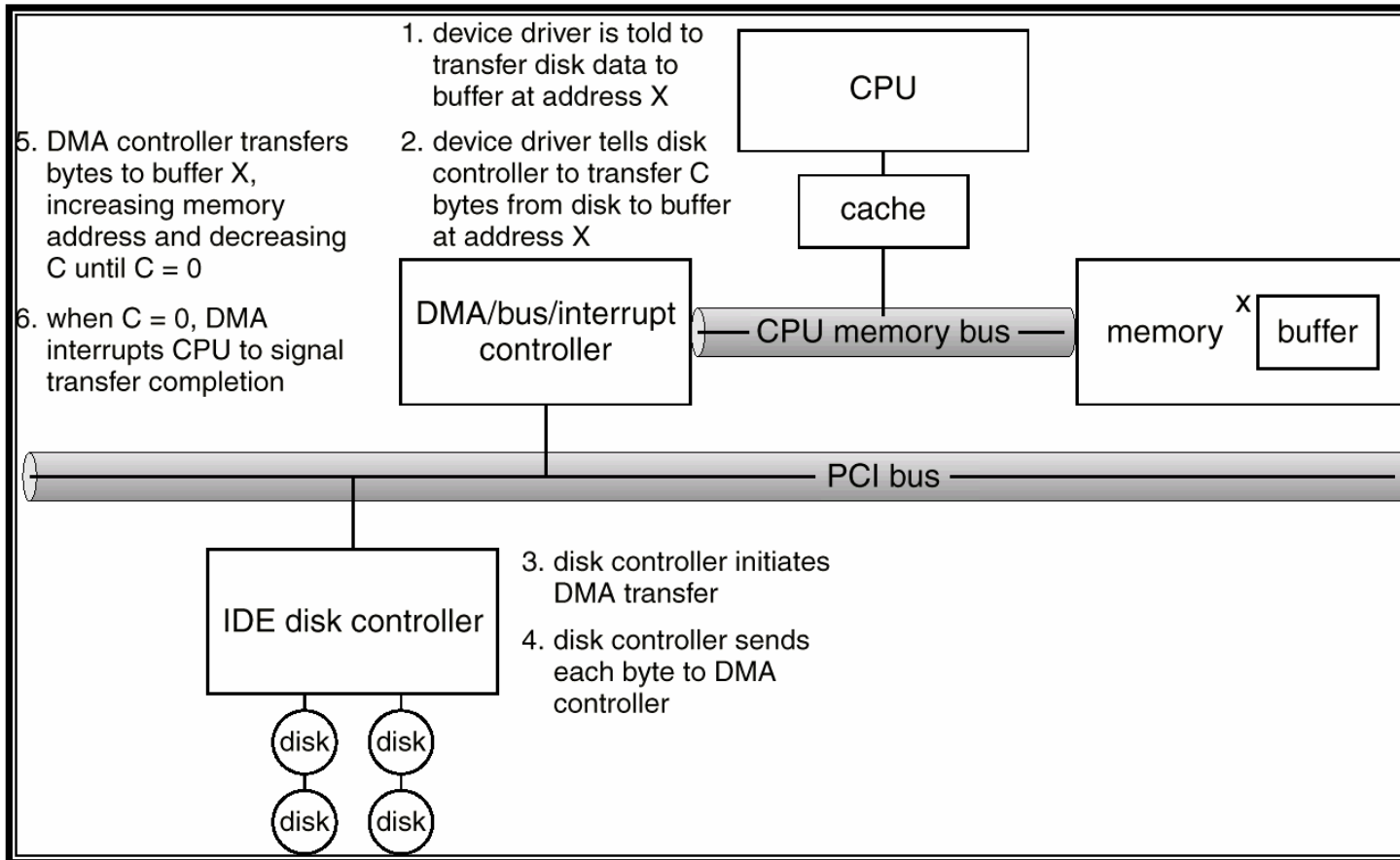
vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
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

Direct Memory Access (DMA)



- Digunakan untuk menghindari diprogramnya I/O untuk pergerakan data yang besar
- Membutuhkan DMA controller
- Dilakukan bypass CPU untuk transfer data secara langsung antara I/O device dan memori

Enam Tahapan Proses dalam Transfer DMA

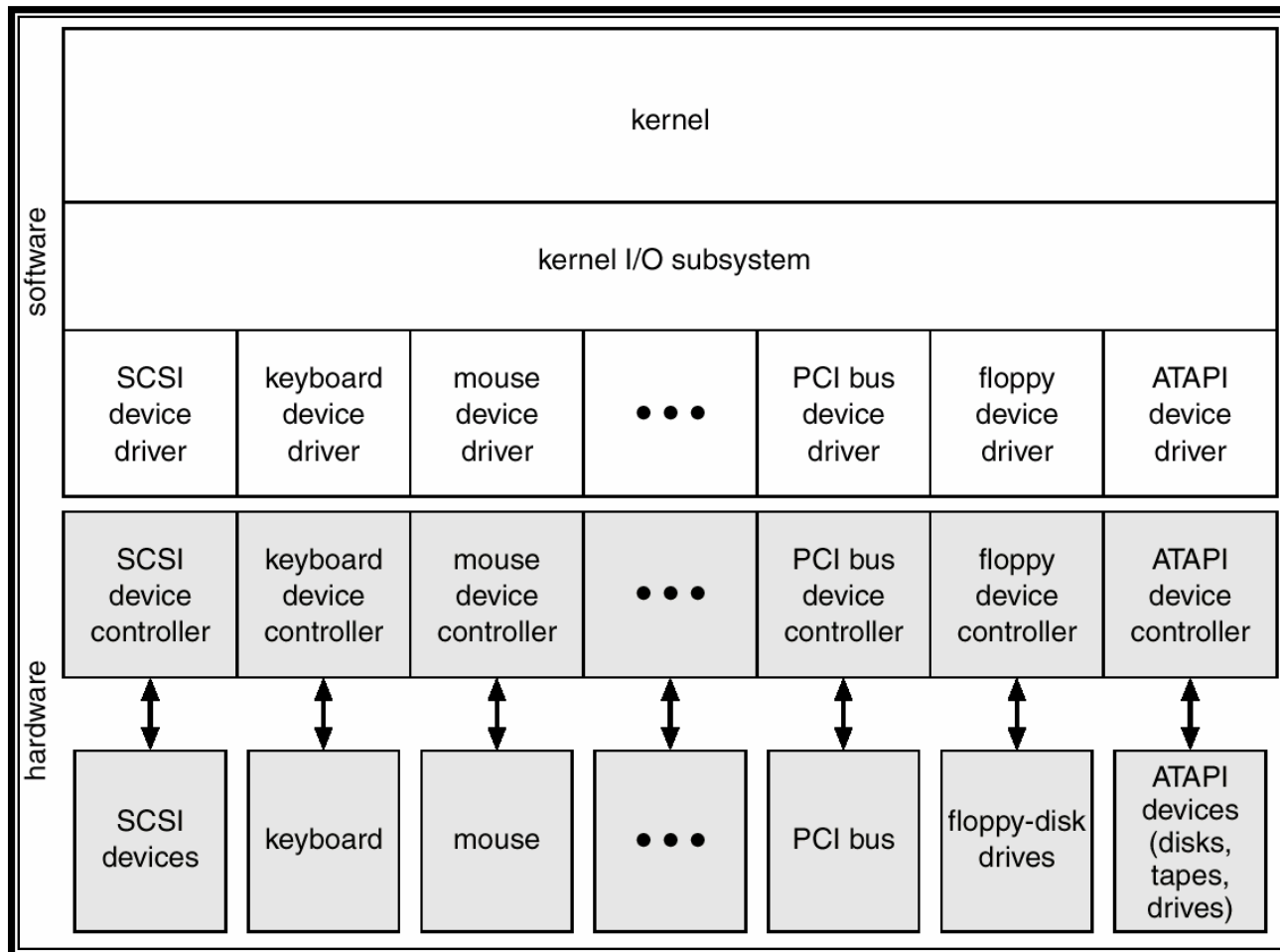




Interface Aplikasi I/O

- System call I/O meng-enkapsulasi device dalam kelas-kelas yang lebih umum
- Device-driver layer menyembunyikan perbedaan diantara I/O controllers dari kernel
- Devices memiliki beragam dimensi :
 - *Character-stream* atau *block*
 - *Sequential* atau *random-access*
 - *Sharable* atau *dedicated*
 - *Speed* atau *operation*
 - *read-write, read only, atau hanya write*

Struktur Kernel I/O



Karakteristik Device I/O

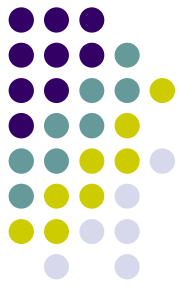


aspect	variation	example
data-transfer mode	character block	terminal disk
access method	sequential random	modem CD-ROM
transfer schedule	synchronous asynchronous	tape keyboard
sharing	dedicated sharable	tape keyboard
device speed	latency seek time transfer rate delay between operations	
I/O direction	read only write only read&write	CD-ROM graphics controller disk

Block dan Character Device

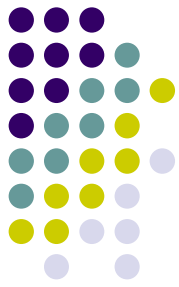


- Block devices termasuk ke dalamnya adalah disk drives
 - Perintahnya meliputi read, write, seek
 - Raw I/O atau file-system access
 - Memungkinkan dilakukannya Memory-mapped file access
- Character devices termasuk ke dalamnya keyboards, mice, serial ports
 - Perintahnya meliputi `get`, `put`
 - Libraries layered terletak pada bagian atas baris editing



Network Device

- Bervariasi dari block dan character yang digunakan untuk interface
- Unix and Windows NT/9//2000 memasukkan socket interface
 - Memisahkan network protocol dari network operation
 - Memasukkan fungsi `select`
- Pendekatannya cukup bervariasi (pipes, FIFOs, streams, queues, mailboxes)



Clock dan Timer

- Menyediakan current time, elapsed time, timer
- Jika programmable interval time digunakan untuk timing, interrupt dilakukan secara periodik
- `ioctl` (pada UNIX) menutupi aspek yang tidak wajar dari I/O seperti clock dan timer

Blocking dan Nonblocking I/O



- Blocking – proses tertahan hingga I/O selesai
 - Mudah digunakan dan dimengerti
 - Kadang tidak cocok untuk semua kebutuhan
- Nonblocking - I/O dikembalikan
 - User interface, data copy (buffered I/O)
 - Diimplementasi melalui multi-threading
 - Pengembalian dilakukan cepat dengan menghitung byte yang dibaca atau ditulis
- Asynchronous – proses dijalankan saat eksekusi I/O
 - Sulit digunakan
 - Signal subsystem I/O melakukan proses ketika I/O selesai

Subsystem Kernel I/O



- Scheduling
 - Beberapa permintaan I/O dipesan melalui urutan device
 - Untuk beberapa Sistem Operasi cukup wajar
- Buffering – menyimpan data di memori ketika transfer antar device
 - Untuk mengatasi kecepatan yang tidak sepadan
 - Untuk mengatasi ukuran transfer yang tidak sepadan
 - Untuk pengelolaan “copy semantics”

Subsystem Kernel I/O (cont.)



- Caching – memori yang cepat menangani copy data
 - Hanya untuk copy
 - Kunci dalam unjuk kerja
- Spooling - menangani output pada device
 - device dapat melayani sendiri satu permintaan pada satu waktu
 - contoh, mencetak (printing)
- Device reservation – menyediakan akses exclusive untuk device
 - System calls untuk alokasi dan dealokasi
 - Berhati-hati dalam deadlock



Penanganan Kesalahan

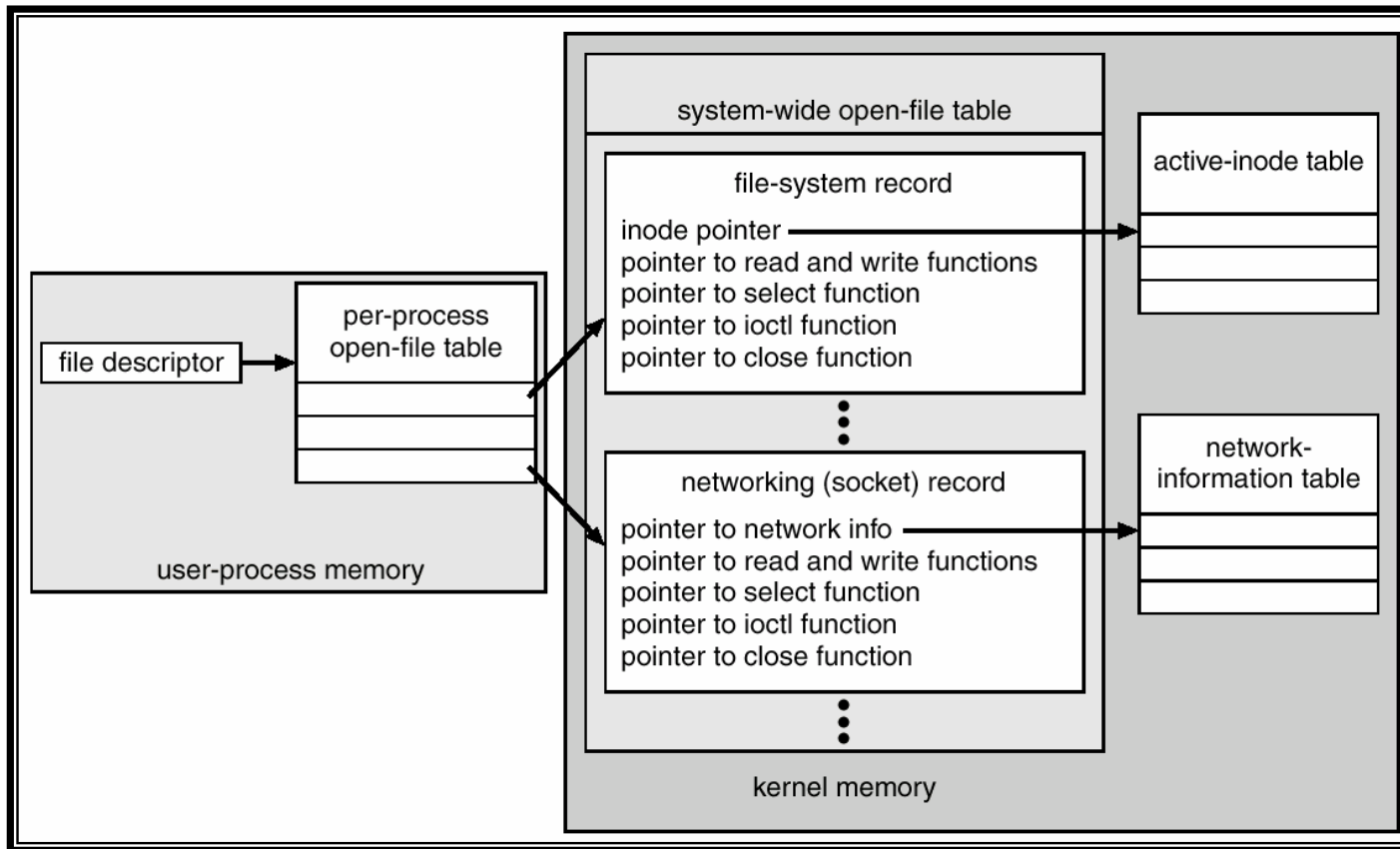
- Sistem operasi dapat mengembalikan pembacaan disk, device unavailable dan kegagalan penulisan karena transient
- Pengembalian error number atau kode terjadi ketika permintaan terhadap I/O gagal.
- System error log memberikan laporan kesalahan



Struktur Kernel Data

- Kernel menjaga status informasi dari komponen I/O termasuk open file tables, koneksi network, status character device
- Banyak struktur data yang kompleks untuk track buffer, alokasi memori, dan blok kotor.
- Penggunaan metode object-oriented methods dan message passing untuk implementasi I/O

Struktur Kernel I/O UNIX

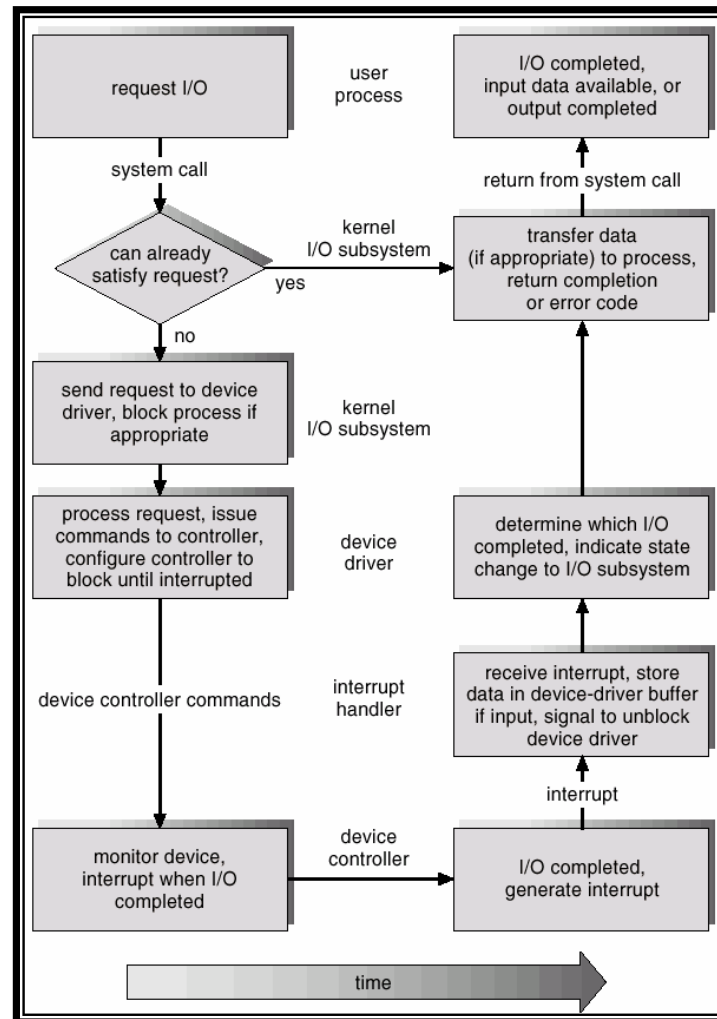


Permintaan I/O untuk Operasi Hardware



- Proses pembacaan file dari disk :
 - Menentukan device yang menangani file
 - Translasi nama ke device yang representatif
 - Pembacaan fisik data dari disk melalui buffer
 - Membuat data menjadi tersedia sehingga dapat diminta (request) untuk proses
 - Mengembalikan kontrol ke proses.

Siklus Hidup Permintaan I/O

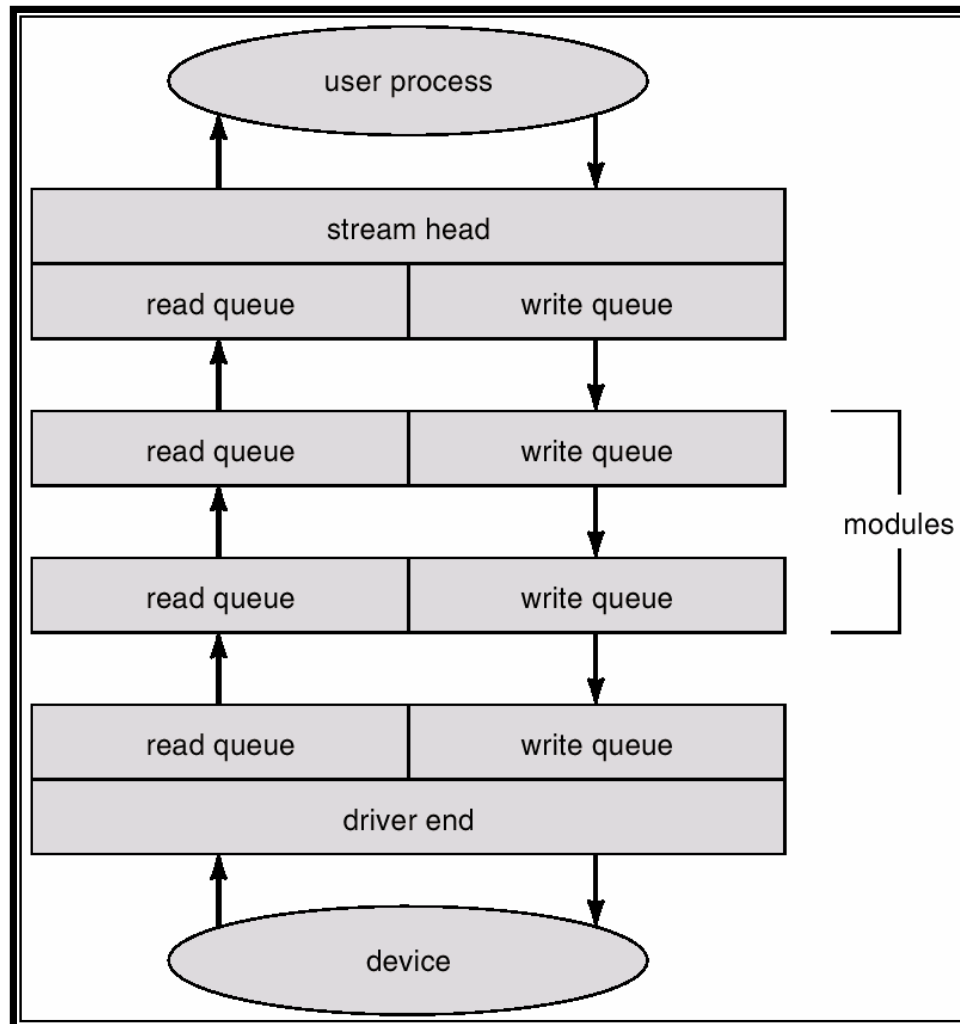


STREAM



- **STREAM** – suatu saluran komunikasi full-duplex antara proses dan device
- STREAM terdiri dari :
 - **STREAM head** , interface dengan user proses
 - **driver end**, interfaces dengan device
 - zero atau lebih dari satu modul STREAM yang ada antar proses dan device
- Setiap modul berisi **read queue** dan **write queue**
- Message passing digunakan untuk komunikasi antar queue.

Struktur STREAM

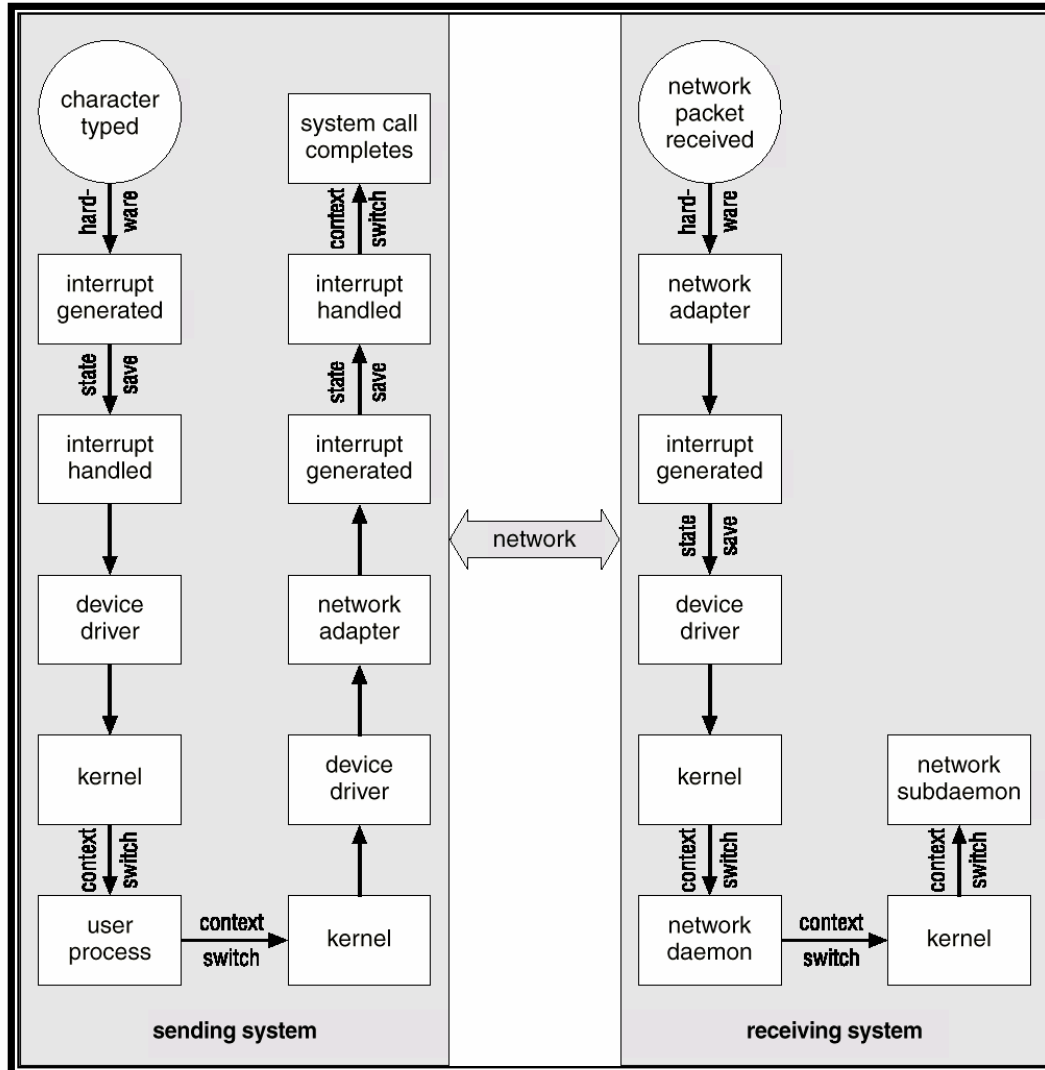




Unjuk Kerja

- Faktor utama dalam unjuk kerja sistem I/O :
 - Permintaan CPU untuk mengeksekusi device driver, kernel I/O code
 - Context switch untuk interrupt
 - Peng-copy-an data
 - Kepadatan Network

Intercomputer Communications





Peningkatan Unjuk Kerja

- Mengurangi jumlah context switches
- Mengurangi peng-copy-an data
- Mengurangi interrupts pada transfer yang besar, pengendalian yang cerdas, polling
- Menggunakan DMA
- Menyeimbangkan CPU, memory, bus, dan unjuk kerja I/O untuk menghasilkan throughput yang tinggi