



**Computer Science, University of Brawijaya**

---

**Putra Pandu Adikara, S.Kom**

**Sistem Operasi**

**Pengantar**



# Pengantar

- ❖ Apa yang dilakukan Sistem Operasi?
- ❖ Organisasi Sistem Komputer
- ❖ Arsitektur Sistem Komputer
- ❖ Struktur Sistem Operasi-
- ❖ Operasi Sistem Operasi
- ❖ Manajemen Proses
- ❖ Manajemen Memori
- ❖ Manajemen Penyimpanan
- ❖ Proteksi dan Keamanan



# Apa itu Sistem Operasi

- ❖ Suatu program yang bertindak sebagai perantara antara pengguna komputer dan perangkat keras komputer.
- ❖ Tujuan Sistem operasi:
  - Menjalankan program pengguna dan membuat pengguna lebih mudah memecahkan masalah.
  - Membuat sistem komputer nyaman digunakan.
- ❖ Menggunakan perangkat keras komputer secara efisien.

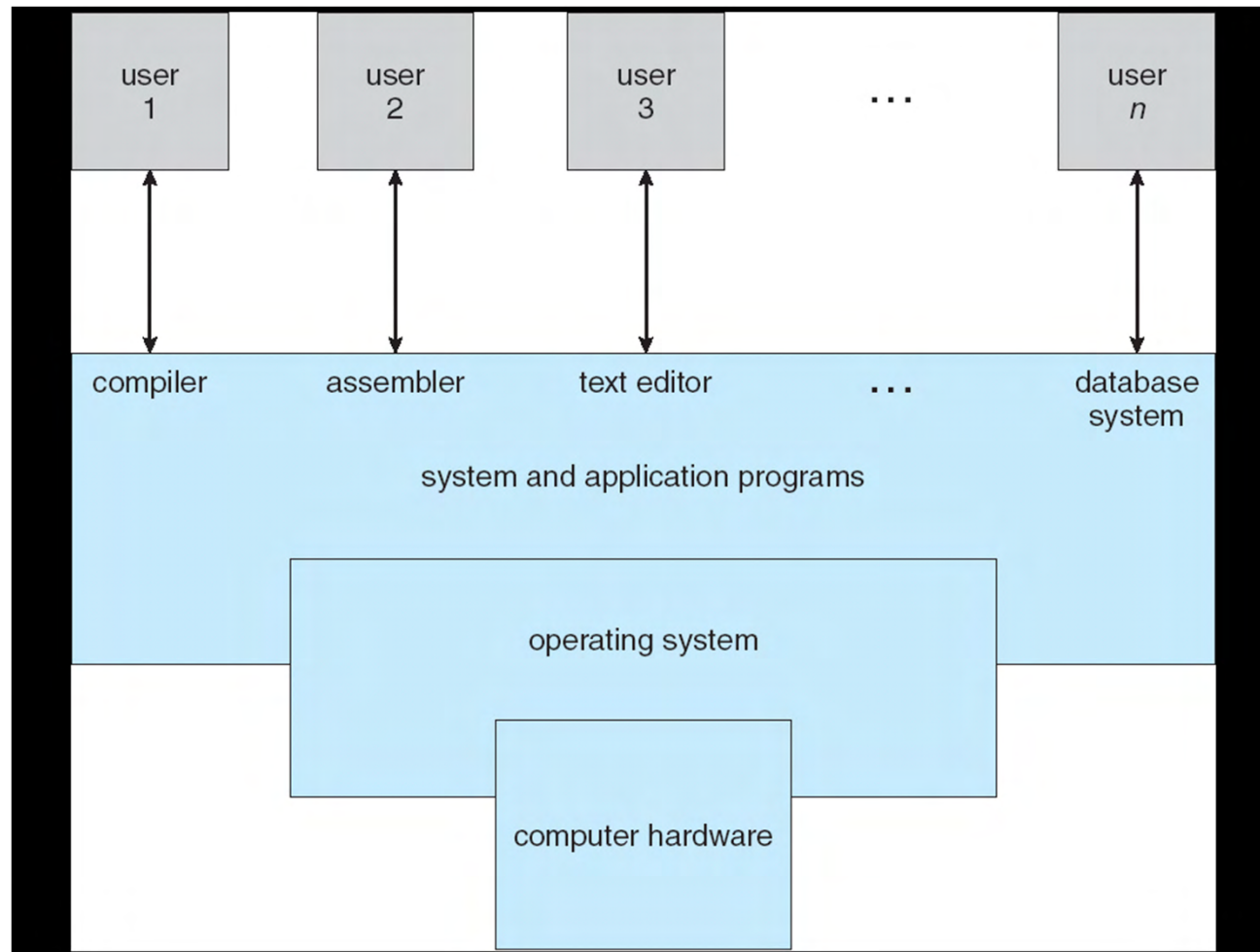


# Struktur Sistem Komputer

- ❖ Sistem komputer dapat dibagi menjadi empat komponen
  - **Hardware** - menyediakan sumber daya komputasi dasar
    - CPU, memori, I / O device
  - **Sistem Operasi**
    - Mengontrol dan mengkoordinasikan penggunaan hardware diantara berbagai aplikasi dan pengguna
  - **Software/Program Aplikasi** - menentukan cara bagaimana sumber daya sistem digunakan untuk memecahkan masalah komputasi dari pengguna
    - Word prosesor, compiler, browser web, sistem database, video game
  - **Pengguna**
    - Orang, mesin, komputer lain



# Empat Komponen Sistem Komputer





# Definisi Sistem Operasi

- ❖ OS adalah **pengalokasi sumber daya**
  - Mengatur semua sumber daya
  - Memutuskan antara permintaan yang mengalami konflik untuk penggunaan sumber daya yang adil dan efisien
  
- ❖ OS adalah **program pengkontrol (control program)**
  - Mengontrol pelaksanaan program untuk mencegah kesalahan dan penyalahgunaan komputer



# Definisi Sistem Operasi

- ❖ Tidak ada definisi yang diterima secara universal
- ❖ "Program yang berjalan setiap saat di komputer" adalah **kernel**. Segala sesuatu yang lain adalah salah satu program sistem (bersamaan dengan sistem operasi) atau program aplikasi



# Startup Komputer

- ❖ **Program bootstrap** dimuat di power-up atau reboot
  - Biasanya disimpan dalam ROM atau EPROM, umumnya dikenal sebagai firmware
  - menginisialisasi semua aspek dari sistem
  - Memuat kernel sistem operasi dan memulai eksekusi
  
- ❖ **BIOS** salah satu program sistem bootstrap yang disimpan dalam ROM/EPROM dan menginisialisasi aspek dari sistem komputer
  
- ❖ Setelah itu diteruskan ke bootstrap loader/boot loader dari sistem operasi, seperti GRUB, LILO, atau NTLDR

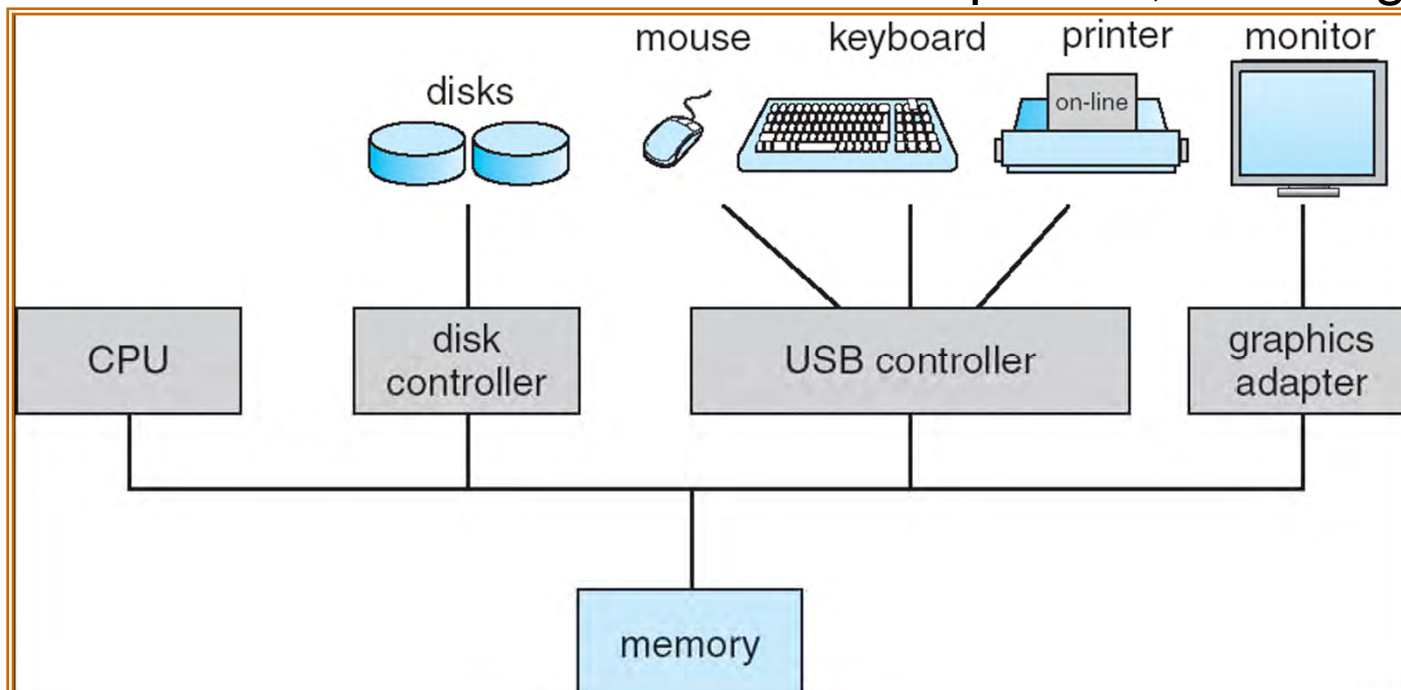




# Organisasi Sistem Komputer

## ❖ Operasi sistem komputer:

- Satu atau lebih CPU dan *device controller* (pengendali peranti) terhubung melalui jalur (bus) umum menyediakan akses ke memori berbagi-pakai
- Eksekusi bersamaan dari CPU dan peranti, bersaing untuk





# Operasi Sistem Komputer

- ❖ Peranti I/O dan CPU dapat dieksekusi secara bersamaan.
- ❖ Setiap *device controller* bertanggung jawab atas suatu jenis peranti tertentu.
- ❖ Setiap device controller memiliki buffer lokal.
- ❖ CPU memindahkan data dari / ke memori utama ke / dari buffer lokal
- ❖ I/O adalah yang berpindah dari peranti ke buffer lokal *controller*.
- ❖ *Device controller* memberitahu CPU bahwa ia telah selesai operasi dengan menyebabkan *interrupt*.



## Fungsi umum interrupt

- ❖ Interrupt mentransfer kontrol ke *interrupt service routine*, melalui *interrupt vector*, yang berisi alamat dari semua rutinitas pelayanan.
- ❖ Arsitektur interrupt harus menyimpan alamat dari instruksi yang diinterupsi.
- ❖ Interupsi masuk dinonaktifkan (*disabled*) sementara interrupt yang lain sedang diproses untuk mencegah *lost interrupt*.
- ❖ *Trap* (perangkap) adalah interrupt yang dihasilkan oleh perangkat lunak yang disebabkan oleh kesalahan atau permintaan pengguna.
- ❖ Sistem operasi adalah *interrupt driven* (dikendalikan oleh interrupt).

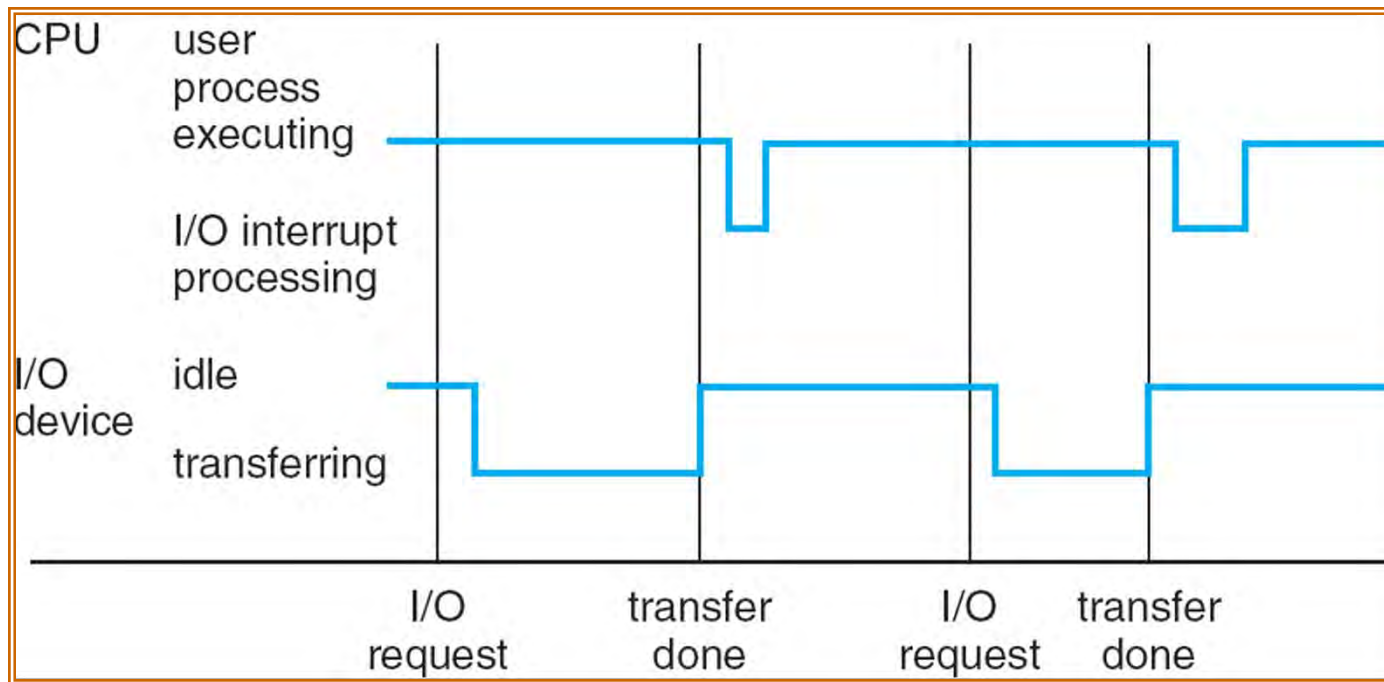


# Penanganan Interrupt

- ❖ Sistem operasi menjaga keadaan CPU dengan menyimpan register dan program counter.
- ❖ Menentukan jenis interrupt telah terjadi saat:
  - *polling*
  - vektor interrupt sistem
- ❖ Segmen terpisah dari kode menentukan tindakan apa yang harus diambil untuk setiap jenis interrupt



# Interrupt Timeline



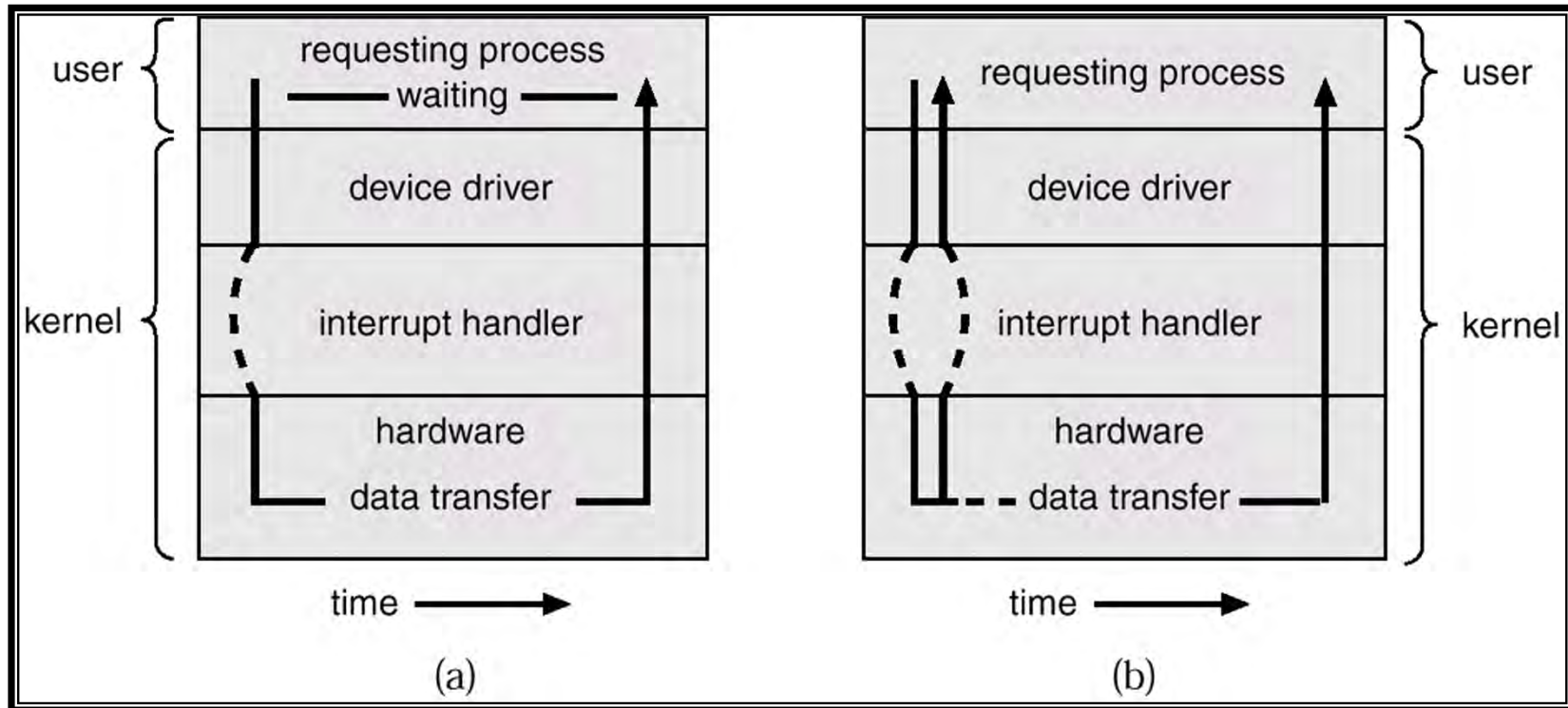


# Struktur I / O

- ❖ Setelah I/O dimulai, kendali kembali ke user program hanya pada I / O selesai.
  - Menunggu instruksi ruang CPU hingga interrupt berikutnya
  - Menunggu *loop* (pembuatan untuk akses memori).
  - Paling banyak satu permintaan terhadap I/O yang beredar pada suatu waktu, tidak ada pemrosesan I/O secara bersamaan.
- ❖ Setelah I/O dimulai, kendali kembali ke user program tanpa menunggu I/O selesai.
  - **System call** - request ke sistem operasi untuk mengizinkan pengguna untuk menunggu I/O selesai.
  - **Device-status table** berisi entri untuk setiap peranti I/O, menentukan tipe, alamat, dan keadaannya.
  - Sistem operasi mengindeks I/O device table untuk menentukan status peranti dan untuk memodifikasi tabel entry untuk memasukkan interrupt.

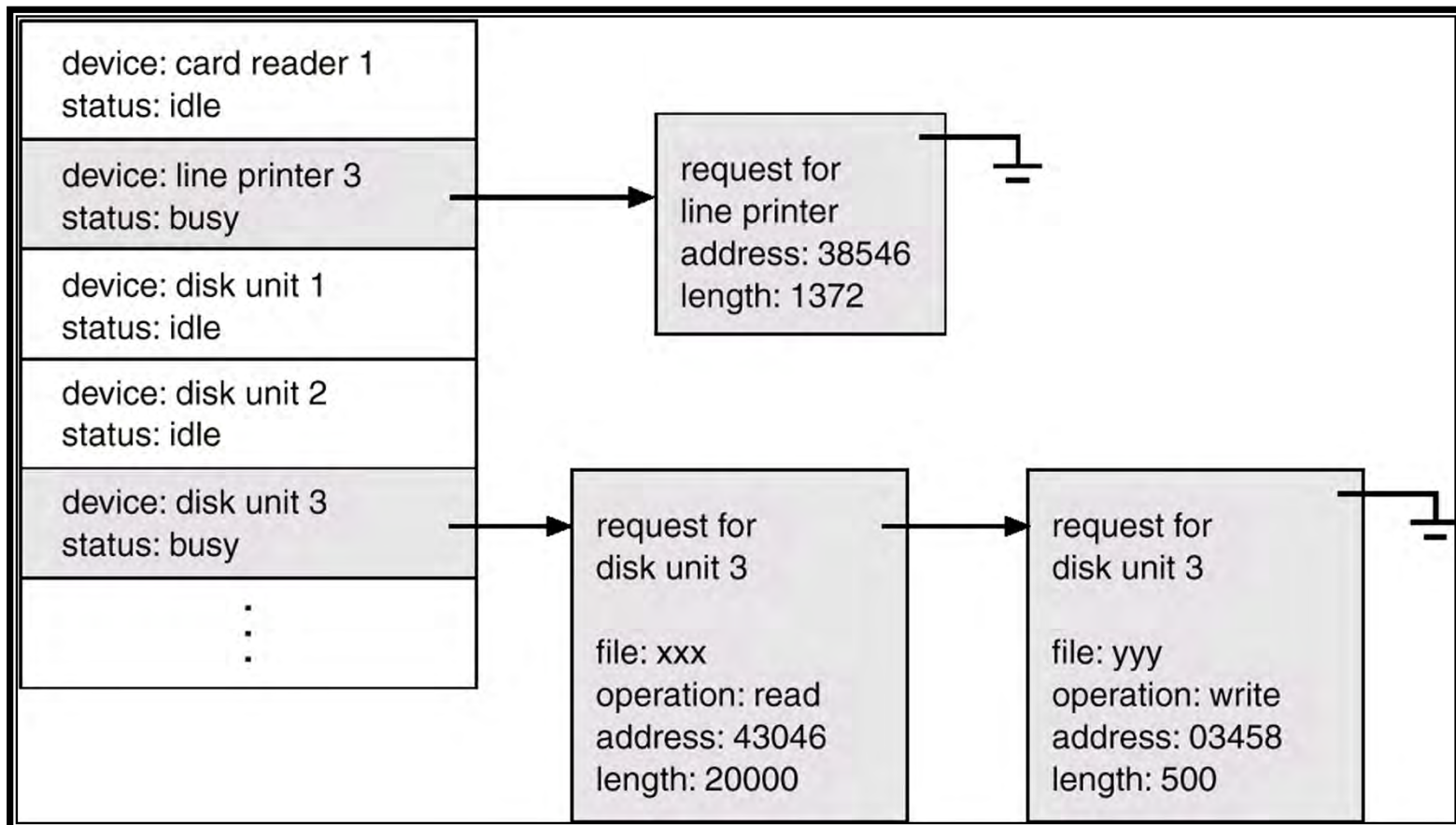


# Dua Metode I/O





# Device-Status Table







## Struktur Direct Memory Access (DMA)

- ❖ Digunakan untuk peranti I/O berkecepatan tinggi untuk dapat mengirimkan informasi mendekati kecepatan memori.
- ❖ *Device controller* memindahkan blok data dari buffer langsung ke memori utama tanpa intervensi CPU.
- ❖ Hanya satu interrupt dibangkitkan per blok, daripada satu interrupt per byte.



# Struktur Penyimpanan

- ❖ **Main memory** - hanya media penyimpanan besar yang mana CPU dapat mengakses langsung.
- ❖ **Secondary storage** - perpanjangan dari memori utama yang menyediakan kapasitas penyimpanan nonvolatile besar.
- ❖ **Magnetic disk** – logam kaku atau piringan (platter) kaca ditutupi dengan bahan magnetic recording
  - Permukaan disk secara logis dibagi menjadi *track*, yang dibagi lagi menjadi *sector*.
  - *Disk controller* menentukan interaksi logis antara peranti dan komputer.

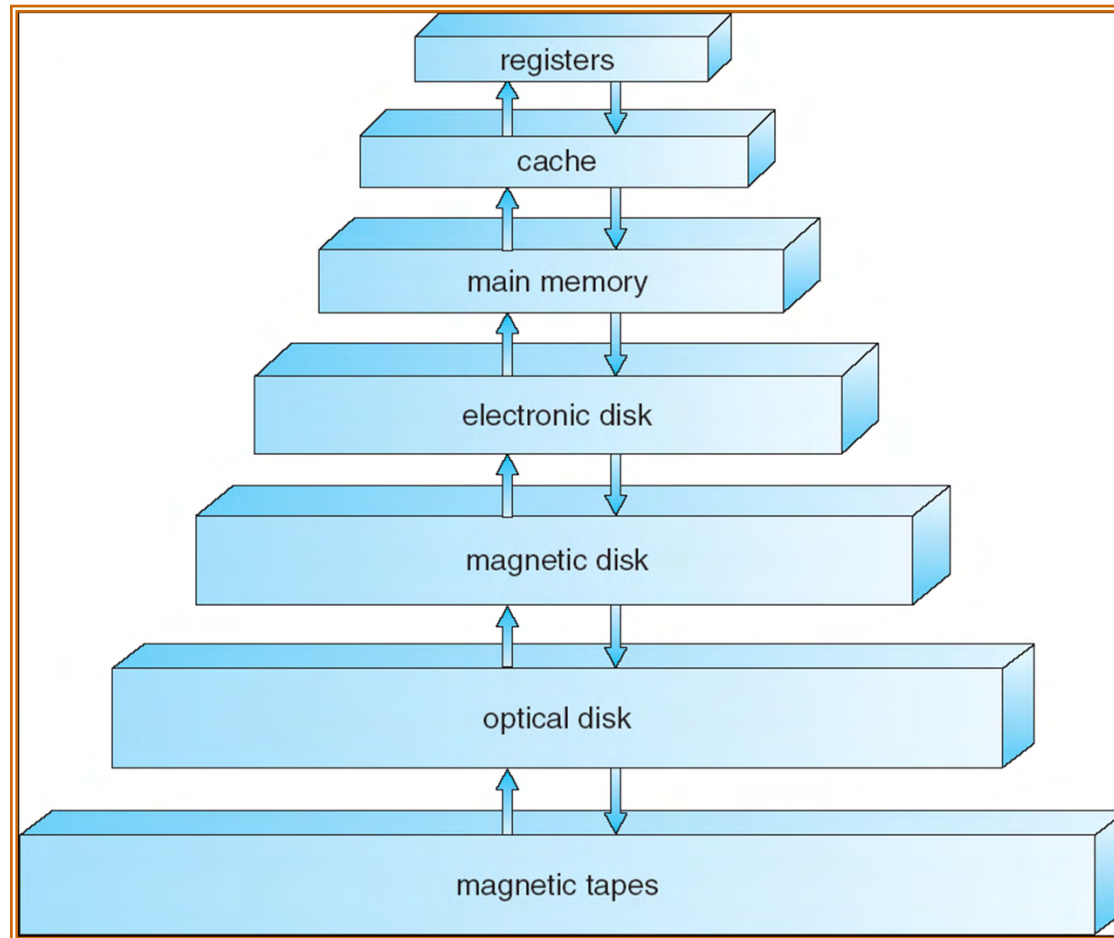


# Hirarki Penyimpanan

- ❖ Sistem penyimpanan, diorganisasikan dalam hirarki.
  - Kecepatan
  - Biaya
  - Volatilitas
- ❖ *Caching* - menyalin informasi ke dalam sistem penyimpanan cepat; memori utama dapat dilihat sebagai *cache* terakhir untuk penyimpanan sekunder.



# Hirarki Peranti Penyimpanan





# Caching

- ❖ **Prinsip penting**, dilakukan di berbagai tingkatan di komputer (di hardware, sistem operasi, perangkat lunak)
- ❖ Informasi yang digunakan disalin sementara dari penyimpanan lebih lambat untuk penyimpanan lebih cepat
- ❖ Penyimpanan cepat (cache) diperiksa terlebih dahulu untuk menentukan apakah informasi itu ada atau tidak
  - Jika ada, informasi yang digunakan langsung dari cache (cepat)
  - Jika tidak, data yang disalin ke cache dan digunakan di sana
- ❖ Cache yang lebih kecil dari penyimpanan sedang di-cache



# Tingkat Kinerja Berbagai Penyimpanan

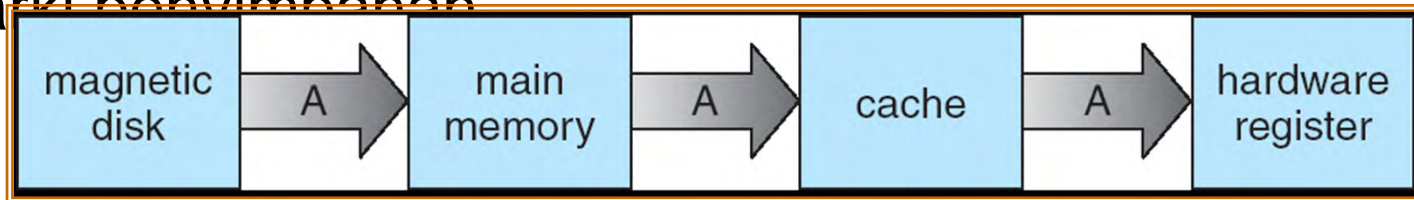
- ❖ Pergerakan antara tingkat hirarki penyimpanan dapat eksplisit atau implisit

Level	1	2	3	4
Name	registers	cache	main memory	disk storage
Typical size	< 1 KB	> 16 MB	> 16 GB	> 100 GB
Implementation technology	custom memory with multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk
Access time (ns)	0.25 – 0.5	0.5 – 25	80 – 250	5,000.000
Bandwidth (MB/sec)	20,000 – 100,000	5000 – 10,000	1000 – 5000	20 – 150
Managed by	compiler	hardware	operating system	operating system
Backed by	cache	main memory	disk	CD or tape



## Migrasi Integer A dari Disk ke Register

- ❖ Lingkungan *Multitasking* harus diperhatikan untuk menggunakan nilai terakhir, di mana pun disimpan dalam hirarki penyimpanan



Lingkungan *Multiprocessor* harus memberikan koherensi cache pada perangkat keras sehingga semua CPU memiliki nilai yang paling terbaru dalam cache mereka

- ❖ Lingkungan Terdistribusi (*Distributed*) lebih kompleks
  - Beberapa salinan dari data yang dapat eksis
  - Berbagai solusi yang dibahas dalam Bab 17



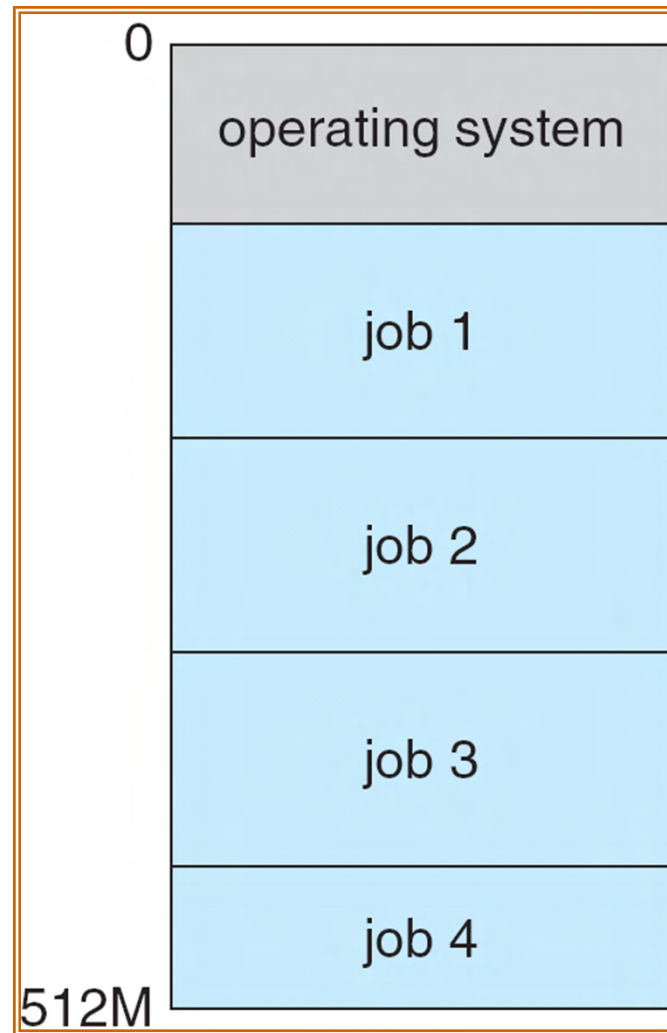
# Struktur Sistem Operasi

- ❖ **Multiprogramming** diperlukan untuk efisiensi
  - Single user tidak dapat menjaga CPU dan perangkat I/O sibuk setiap saat
  - Multiprogramming mengatur job (kode dan data) sehingga selalu memiliki satu CPU untuk mengeksekusi
  - Suatu bagian dari total job dalam sistem disimpan dalam memori
  - Satu job dipilih dan dijalankan melalui **job scheduling**
  - Ketika harus menunggu (untuk I / O misalnya), OS beralih ke job lain
- ❖ **Timesharing (multitasking)** adalah perluasan logis di mana CPU memindah job begitu sering sehingga pengguna dapat berinteraksi dengan setiap pekerjaan walau sedang berjalan, membuat komputasi yang **interaktif**
  - **Waktu respon** harus <1 detik
  - Tiap user punya paling sedikit 1 program dieksekusi dlm memori → **proses**
  - Jika beberapa job siap untuk dijalankan pada saat yang sama → **job scheduling**
  - Jika proses tidak cukup di memori, **swapping** bergerak masuk dan keluar untuk dijalankan
  - **Virtual memory** memungkinkan eksekusi proses yang tidak masuk sepenuhnya dalam memori





# Memory Layout untuk Sistem Multiprogram





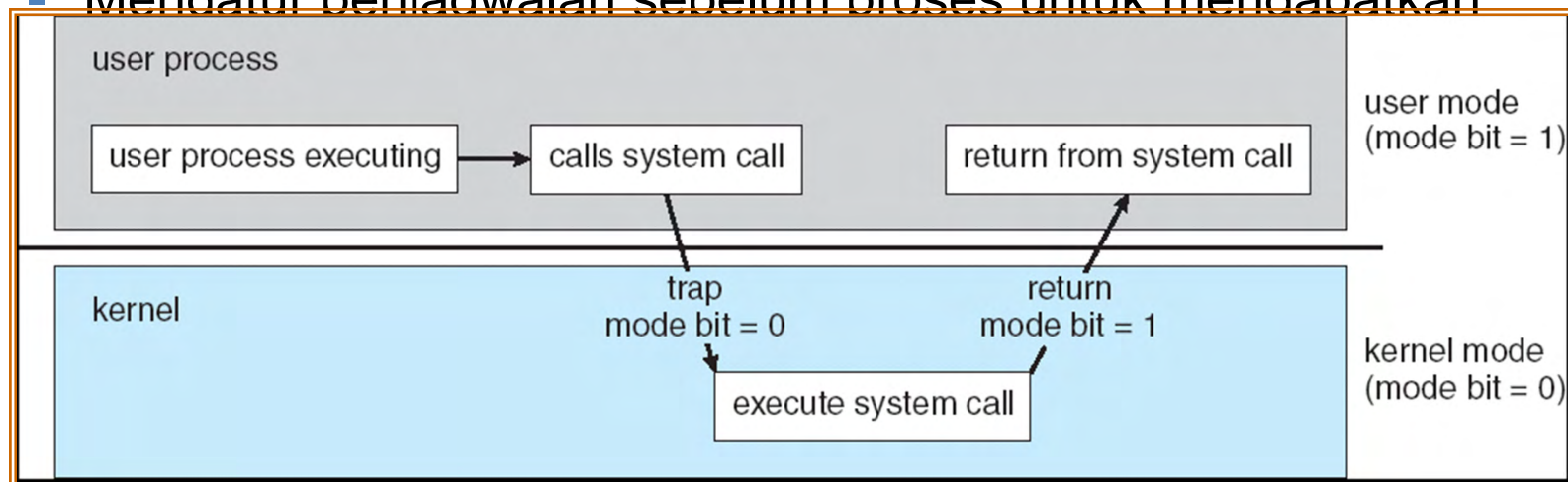
# Operasi Sistem Operasi

- ❖ Interrupt dikendalikan oleh perangkat keras
- ❖ Kesalahan atau permintaan perangkat lunak membuat **exception** atau **trap**
  - Contoh: pembagian dengan nol, permintaan untuk layanan sistem operasi
- ❖ Permasalahan proses lainnya termasuk *infinite loop*, proses memodifikasi satu sama lain atau sistem operasi
- ❖ Operasi **Dual-mode** memungkinkan OS untuk melindungi diri dan komponen sistem lainnya
  - **User mode** dan **kernel mode**
  - **Mode bit** yang disediakan oleh perangkat keras
    - Menyediakan kemampuan untuk membedakan ketika sistem sedang menjalankan kode pengguna atau kode kernel
    - Beberapa instruksi didesain sebagai **privileged (istimewa)**, hanya dieksekusi dalam mode kernel
    - Sistem call memubah mode untuk kernel, kembali dari call mereset



# Transisi dari Mode User ke Kernel

- ❖ Timer untuk mencegah infinite loop/proses memonopoli sumber daya
  - Mengeset interrupt tertentu setelah periode tertentu
  - Sistem operasi mengurangi (*decrement*) counter
  - Ketika nol counter menghasilkan interrupt
  - Mengatur penjadwalan sebelum proses untuk mendapatkan





# Manajemen Proses

- ❖ Proses adalah program sedang dieksekusi. Merupakan unit kerja dalam sistem. Program merupakan sebuah **entitas pasif**, proses adalah suatu **entitas aktif**.
- ❖ Proses membutuhkan sumber daya untuk menyelesaikan tugasnya
  - CPU, memori, I/O, file
  - Inisialisasi data
- ❖ Terminasi proses membutuhkan pengembalian setiap sumber daya yang dapat digunakan ulang
- ❖ Proses Single-threaded memiliki satu **program counter** menetapkan lokasi instruksi berikutnya untuk eksekusi
  - Proses mengeksekusi instruksi secara berurutan, satu per satu, sampai selesai
- ❖ Proses Multi-threaded memiliki satu program counter per thread
- ❖ Biasanya sistem memiliki banyak proses, pengguna, beberapa sistem operasi berjalan secara bersamaan pada satu atau lebih CPU
  - Konkurensi oleh multiplexing CPU diantara proses / thread



# Aktivitas Manajemen Proses

- ❖ Sistem operasi bertanggung jawab atas aktivitas berikut yang berhubungan dengan manajemen proses:
  - Membuat dan menghapus dan proses sistem dan proses pengguna
  - Menangguhkan (suspend) dan melanjutkan (resume) proses
  - Menyediakan mekanisme untuk sinkronisasi proses
  - Menyediakan mekanisme untuk proses komunikasi
  - Menyediakan mekanisme untuk penanganan deadlock



# Manajemen Memori

- ❖ Semua data dalam memori sebelum dan sesudah pengolahan
- ❖ Semua instruksi dalam memori untuk eksekusi
- ❖ Memori manajemen menentukan apa yang ada di memori saat
  - Mengoptimalkan utilisasi CPU dan respon komputer untuk pengguna
- ❖ Memori manajemen kegiatan
  - Mencatat bagian mana dari memori yang sedang digunakan dan oleh siapa
  - Memutuskan proses mana (atau bagiannya) dan data untuk masuk dan keluar dari memori
  - Mengalokasikan dan dealokasi ruang memori yang



# Manajemen Penyimpanan

- ❖ OS menyediakan keseragaman, pandangan logis dari penyimpanan informasi
  - Abstraksi sifat-sifat fisik ke unit penyimpanan logis - **file**
  - Setiap media dikendalikan oleh peranti (misalnya, disk drive, tape drive)
    - Properti bervariasi seperti kecepatan akses, kapasitas, kecepatan transfer data, metode akses (sequential atau random)
- ❖ Manajemen Sistem Berkas (File-Sistem)
  - File biasanya diatur ke dalam direktori
  - Kontrol akses (**Access Control**) pada kebanyakan sistem untuk menentukan siapa yang dapat mengakses apa
  - Aktivitas OS meliputi
    - Membuat dan menghapus file dan direktori
    - Primitif untuk memanipulasi file dan dirs
    - Pemetaan file ke secondary storage
    - Backup file ke media penyimpanan stabil (non-volatile)



# Manajemen Mass-Storage

- ❖ Biasanya disk digunakan untuk menyimpan data yang tidak cukup di memori utama atau data yang harus disimpan untuk periode waktu lama
- ❖ Manajemen yang tepat adalah sangat penting
- ❖ Semua kecepatan operasi komputer di subsistem disk dan algoritma
- ❖ Aktivitas OS
  - Manajemen free space
  - Alokasi penyimpanan
  - Penjadwalan disk
- ❖ Beberapa penyimpanan tidak perlu cepat
  - penyimpanan tersier meliputi penyimpanan optik, pita magnetik
  - Tetap harus dikelola
  - Bervariasi antara WORM (write-sekali, read-banyak-kali) dan RW (baca-tulis)





## Subsistem I/O

- ❖ Salah satu tujuan OS adalah untuk menyembunyikan kerumitan perangkat keras dari pengguna
- ❖ subsistem I/O bertanggung jawab atas
  - Manajemen memori I/O termasuk buffering (menyimpan data sementara ketika sedang ditransfer), caching (bagian menyimpan data dalam penyimpanan lebih cepat untuk kinerja), spooling (tumpang tindih output dari satu pekerjaan dengan input dari pekerjaan lain)
  - Antarmuka driver-perangkat yang umum
  - Driver untuk perangkat keras tertentu



# Proteksi dan Keamanan

- ❖ **Proteksi** - mekanisme untuk mengontrol akses dari proses atau pengguna ke sumber daya yang didefinisikan oleh OS
- ❖ **Keamanan** - sistem pertahanan terhadap serangan internal dan eksternal
  - Jangkauan besar, termasuk denial-of-service (DOS), worm, virus, pencurian identitas, pencurian layanan
- ❖ Sistem umumnya pertama kali membedakan pengguna, untuk menentukan siapa yang dapat melakukan apa
  - Identitas pengguna (**user ID**, security ID n) termasuk nama dan nomor tertentu yang berasosiasi, satu per pengguna
  - User ID kemudian dikaitkan dengan semua file, proses pengguna tersebut untuk menentukan kontrol akses
  - Identifier Group (**group ID**) memungkinkan kumpulan pengguna untuk didefinisikan dan kontrol yang dikelola, kemudian juga mengaitkan dengan tiap proses, file
  - **Privilege escalation (perluasan hak istimewa)** memungkinkan pengguna untuk mengubah ID efektif dengan hak-hak lebih



# Lingkungan Komputasi

## ❖ Komputer tradisional

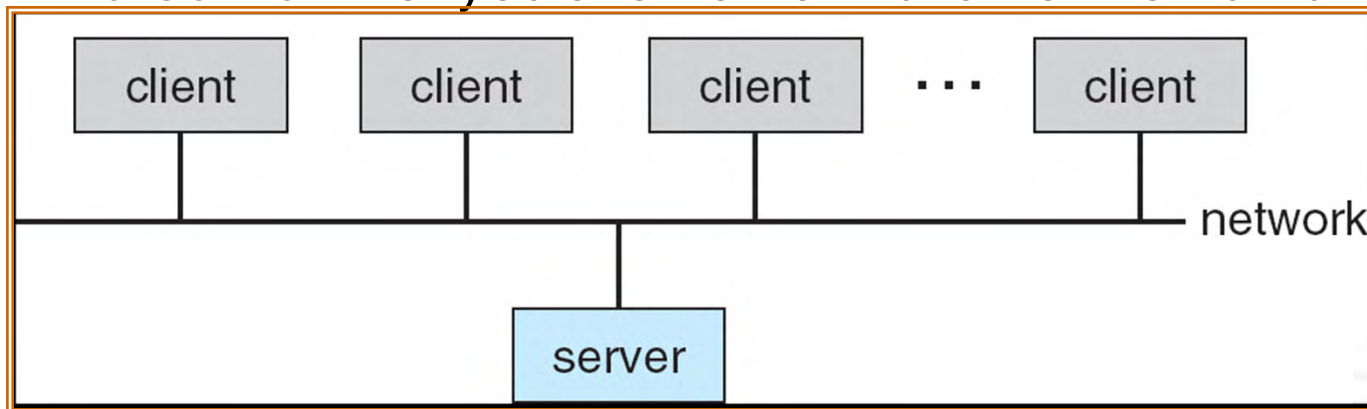
- Menjadi kabur seiring waktu
- Lingkungan Kantor
  - PC terhubung ke jaringan, terminal melekat pada mainframe atau minikomputer menyediakan batch dan timesharing
  - Portal sekarang memungkinkan sistem jarak jauh dan jaringan mengakses sumber daya yang sama
- Jaringan rumah
  - Dulunya sistem tunggal kemudian modem
  - Sekarang firewall, jaringan



## Lingkungan Komputasi (lanj.)

### ❖ Komputasi Client-Server

- Terminal dungu digantikan PC cerdas
- Banyak sistem sekarang berupa **server**, merespon request yang diberikan oleh **client**
  - **Compute-server** menyediakan antarmuka ke klien untuk meminta layanan (misal database, web)
  - **File-server** menyediakan antarmuka ke klien untuk





# Komputasi Peer-to-Peer

- ❖ Model lain dari sistem terdistribusi
- ❖ P2P tidak membedakan klien dan server
  - Sebaliknya semua node dianggap peer (rekan)
  - Masing-masing dapat bertindak sebagai klien, server atau keduanya
  - Node harus bergabung dengan jaringan P2P
    - Meregistrasi layanan dengan pusat layanan pencarian pada jaringan, atau
    - Mem-broadcast permintaan untuk layanan dan menanggapi permintaan untuk layanan melalui protokol penemuan
  - Contohnya termasuk *Napster* dan *Gnutella* sekarang  
Torrent



# Komputasi Web

- ❖ Web telah menjadi ubiquitous (dimana-mana)
- ❖ PCs menjadi perangkat yang lazim
- ❖ Berbagai perangkat tergabung dalam jaringan memungkinkan akses web
- ❖ Kategori baru peranti untuk mengatur lalu lintas web diantara server yang mirip: **load balancer**
- ❖ Penggunaan sistem operasi seperti Windows 95, client-side, telah berevolusi ke Linux dan Windows XP, yang dapat menjadi klien dan server