

IF3191- Pengenalan OS

Henny Y. Zubir
STEI - ITB

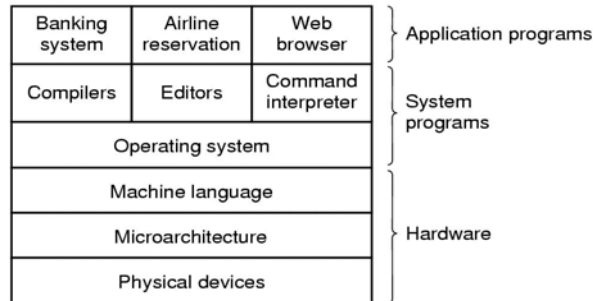


Ikhtisar

- Overview Sistem Komputer
- Mengapa Belajar OS?
- Review Hardware
 - Arsitektur, komponen, memori, control unit, I/O, perangkat penyimpanan, interrupt
- Konsep Dasar OS
 - Sejarah OS, overview fungsi OS, struktur OS



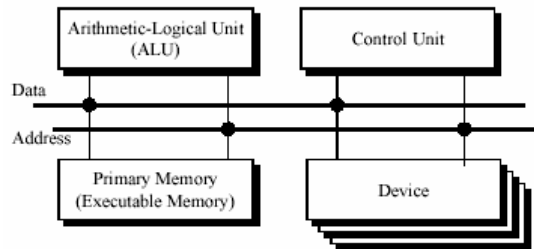
Overview Sistem Komputer



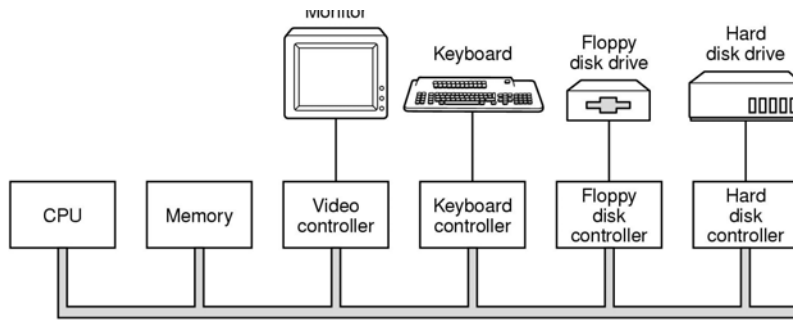
Komponen Sistem Komputer

1. **Hardware** – menyediakan sumber daya komputasi dasar (CPU, memori, perangkat I/O)
2. **Sistem operasi** – mengontrol dan mengkoordinasikan penggunaan H/W antara berbagai program aplikasi untuk beragam user
3. **Program aplikasi** – mendefinisikan cara penggunaan sumber daya sistem untuk memecahkan persoalan komputasi user (kompilator, sistem basisdata, video games, aplikasi bisnis)
4. **Users** (orang, mesin, komputer lainnya)

Arsitektur von Neumann



Tipikal Hardware PC Sederhana



Translasi Program

Source

```
int a, b, c, d;  
...  
a = b + c;  
d = a - 100;
```

Assembly Language

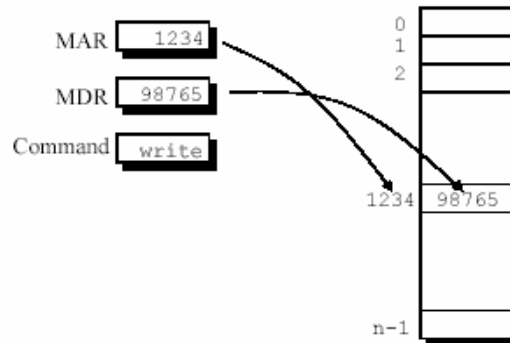
```
; Code for a = b + c  
load    R3,b  
load    R4,c  
add     R3,R4  
store   R3,a  
  
; Code for d = a - 100  
load    R4,=100  
subtract R3,R4  
store   R3,d
```

Machine Language

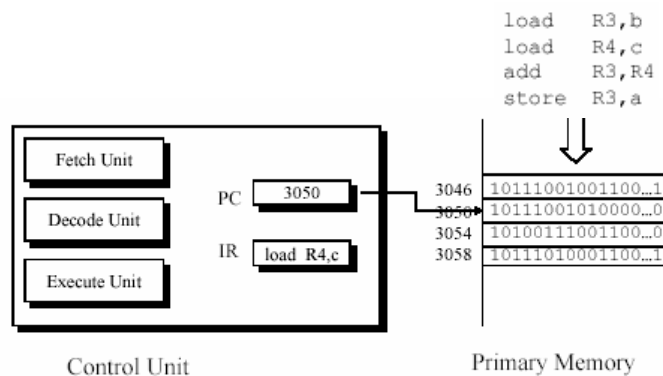
```
10111001001100...1  
10111001010000...0  
10100111001100...0  
10111010001100...1  
10111001010000...0  
10100110001100...0  
10111001101100...1
```



Unit Memori



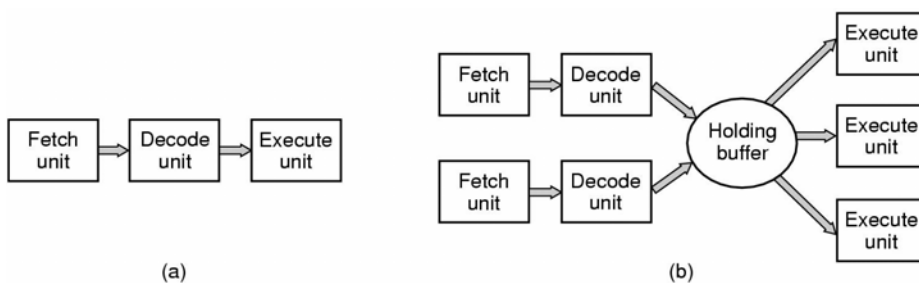
Control Unit



Operasi Control Unit:

- Fase **Fetch**: instruksi diperoleh dari memori
- Fase **Execute**: operasi ALU, referensi data memori, I/O, dll

CPU



- (a) A three-stage pipeline
(b) A superscalar CPU

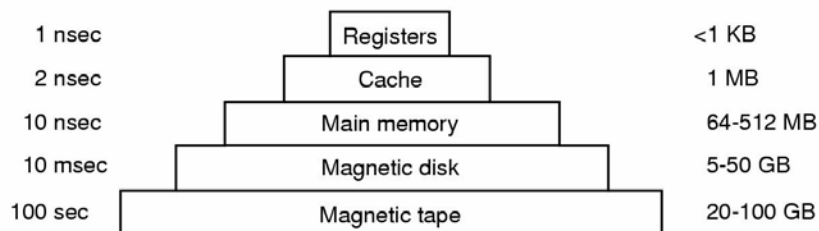
Media Penyimpanan: Struktur

- **Memori utama** – media penyimpanan volatile yg dpt diakses langsung oleh CPU
- **Penyimpanan sekunder** – ekstensi memori utama yg menyediakan kapasitas penyimpanan non-volatile yg sangat besar
- **Disk magnetik** – piringan logam/kaca yg kuat yg ditutupi dgn bahan perekam magnetis
 - Permukaan disk secara logik dibagi menjadi **track**, yg terdiri dari **sektor**
 - **disk controller** menentukan interaksi logik antara perangkat dan komputer

Media Penyimpanan: Hirarki

Typical access time

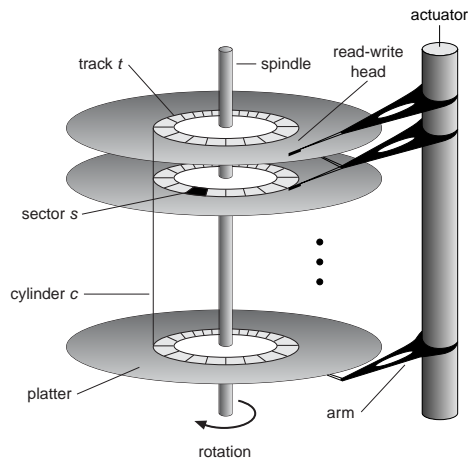
Typical capacity



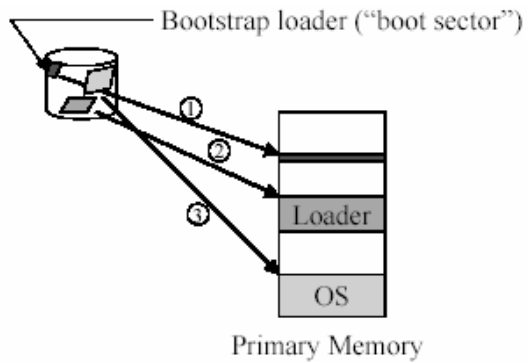
Hirarki media:

- Kecepatan akses
- Biaya
- volatilitas

Media Penyimpanan: Disk

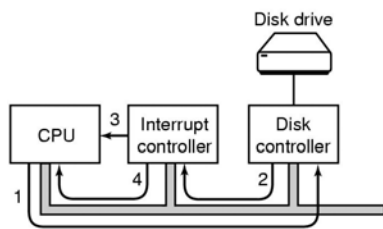


Bootstrapping

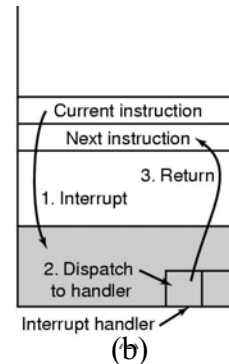


4. Initialize hardware
5. Create user environment
6. ...

Operasi I/O (1)



(a)



(b)

(a) Contoh tahap memulai perangkat I/O dan membangkitkan interrupt

(b) Bagaimana CPU diinterupsi

Operasi I/O

- Perangkat I/O dan CPU dpt melakukan eksekusi secara konkuren
- Tiap **device controller** bertanggung jawab utk tipe perangkat tertentu
- Tiap device controller memiliki **buffer lokal**
- CPU memindahkan data dari/ke memori utama ke/dari buffer lokal
- I/O berlangsung dari perangkat ke buffer lokal di controller
- Device controller membangkitkan **interrupt** untuk memberi tahu CPU bahwa operasinya telah selesai

Interrupt: Fungsi Umum

- **Interrupt** mengalihkan kontrol eksekusi dari instruksi yg sdg berlangsung ke **ISR**
- Alamat instruksi yg diinterupsi hrs disimpan
- Sementara eksekusi suatu ISR berlangsung, interrupt lain yg masuk hrs di-disable
- **Trap** merupakan interrupt yg dibangkitkan software karena adanya error atau permintaan user
- Sistem operasi merupakan **interrupt-driven**

Interrupt: Penanganan Interrupt

- OS memelihara state CPU dgn menyimpan isi register dan PC
- Menentukan tipe interrupt yg terjadi:
 - *polling*
 - vectored interrupt system
- Kode segmen yg terpisah menentukan tindakan apa yg harus dilakukan utk tiap tipe interrupt

Peran OS

- OS sebagai *extended machine*
 - Menyembunyikan detail kerumitan yg harus dilakukan
 - Menyediakan antarmuka bagi user → lebih mudah digunakan
- OS sebagai *resource manager*
 - Tiap program memperoleh alokasi waktu, ruang, dll

Beberapa Definisi OS

- **Resource allocator** – mengelola dan mengalokasikan sumber daya
- **Control program** – mengontrol eksekusi program user dan operasi perangkat I/O
- **Kernel** – suatu program yg berjalan terus-menerus selama komputer dinyalakan (selain dari itu merupakan program aplikasi)

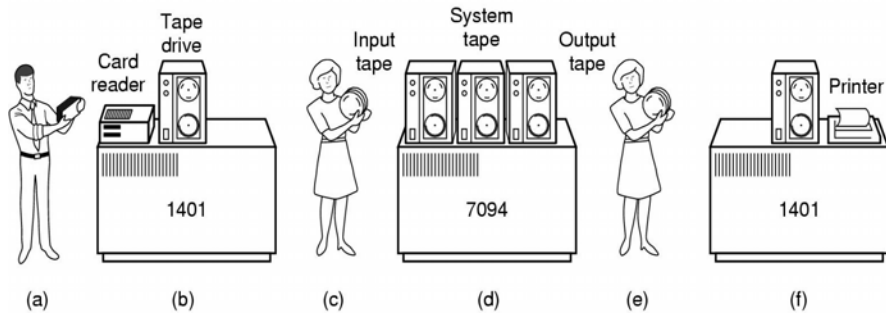
Mengapa Belajar OS?

- Memahami model operasi
 - Mempermudah penggunaan sistem
 - Memungkinkan penulisan kode program secara efisien
- Memahami perancangan sistem
- Bagaimana caranya?
 - Teori & konsep
 - Analogi
 - Terjun langsung → *get your hands dirty*

Sejarah Sistem Operasi

- Generasi I (1945 - 1955)
 - Tabung hampa, plugboards
- Generasi II (1955 - 1965)
 - Transistor, sistem batch
- Generasi III (1965 - 1980)
 - IC & Multiprogramming
- Generasi IV (1980 - sekarang)
 - PC

Sistem *Batch* (1)

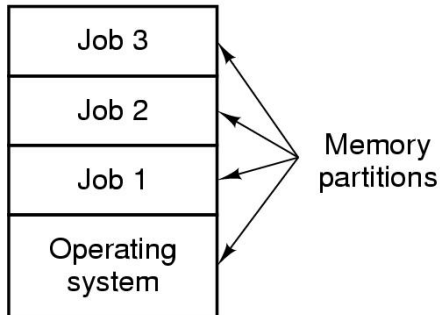


Sistem *Batch* (2)

- Job (file perintah OS) dipersiapkan secara offline
- Sekumpulan (batch) job diberikan pd OS sekaligus
- OS memproses job satu per satu
- Tidak ada interaksi manusia-komputer
- OS mengoptimasi penggunaan sumberdaya

Sistem Batch Multiprogramming (1)

- Beberapa job berada di memori utama pd saat yg sama
- Penggunaan CPU digilirkan antar job



Sistem Batch Multiprogramming (2)

- Fitur OS yg diperlukan utk multiprogramming:
 - I/O routine yg disediakan oleh sistem
 - Manajemen memori – sistem harus mengalokasikan memori utk beberapa job
 - Penjadwalan CPU – sistem harus memilih diantara beberapa job yg siap utk dijalankan
 - Alokasi perangkat

Sistem Time-sharing

- CPU digilirkan diantara beberapa job yg berada di memori dan disk (CPU dialokasikan utk suatu job hanya jika job tsb berada di memori)
- Job dipindahkan dari/ke memori ke/dari disk
- Menyediakan komunikasi on-line antara user dan sistem; perintah kendali diberikan bukan dari card reader, melainkan dari keyboard user
- Sistem on-line harus tersedia bagi user utk mengakses data dan kode

Sistem Personal Computer

- **Personal computers** – sistem komputer yg didedikasikan utk satu user
- Perangkat I/O – keyboards, mouse, layar peraga, printer sederhana
- Dpt mengadopsi teknologi yg dikembangkan utk OS yg besar
- Penggunaan personal seringkali tidak memerlukan CPU atau fitur proteksi yg canggih

Sistem Paralel (1)

- Sistem multiprosesor dgn > 1 CPU yg saling berkomunikasi
- **Tightly coupled system** – prosesor menggunakan memori dan clock yg sama; komunikasi biasanya melalui memori yg digunakan
- Keuntungan sistem paralel:
 - Peningkatan **throughput**
 - Ekonomis
 - Peningkatan keandalan

Sistem Paralel (2)

- **Symmetric multiprocessing (SMP)**
 - Tiap proses menjalankan OS yg sama
 - Banyak proses dpt dijalankan sekaligus tanpa menurunkan performansi
 - Kebanyakan OS modern mendukung SMP
- **Asymmetric multiprocessing**
 - Tiap prosesor diberi tugas khusus; prosesor master menjadwalkan dan mengalokasikan pekerjaan ke prosesor slave
 - Lebih banyak digunakan pd sistem yg sangat besar

Sistem Real-time

- Sering digunakan pd perangkat kendali dgn aplikasi khusus spt pengendalian eksperimen, sistem kendali industri, dan beberapa sistem peraga
- Constraint waktu yg tepat dan terdefinisi dgn baik
- **Hard real-time system**
 - Data disimpan di memori jangka pendek atau ROM, media penyimpanan sekunder terbatas atau tidak ada
 - Konflik dgn sistem time-sharing, tidak didukung oleh OS biasa
- **Soft real-time system**
 - Penggunaan terbatas pd kendali industri atau robotika
 - Bermanfaat utk aplikasi yg memerlukan fitur OS canggih (multimedia, virtual reality)

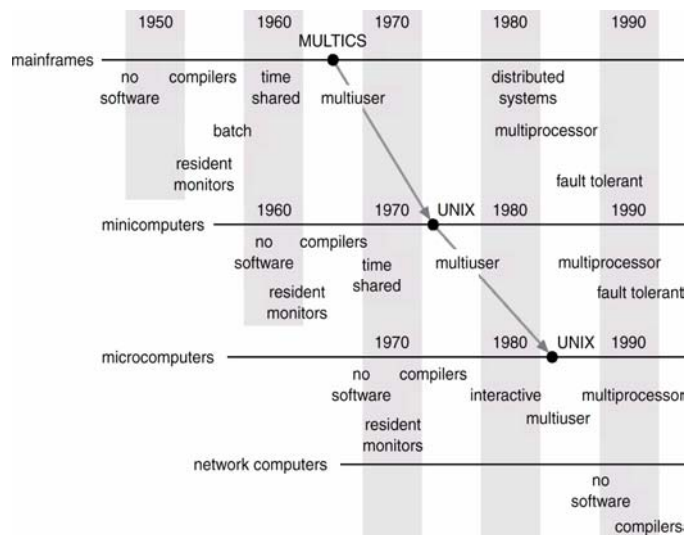
Sistem Terdistribusi (1)

- Mendistribusikan komputasi pd beberapa prosesor
- **Loosely coupled system** – tiap prosesor memiliki memori lokal tersendiri; prosesor saling berkomunikasi melalui berbagai jalur komunikasi (bus kecepatan tinggi, saluran telepon)
- Keunggulan sistem terdistribusi
 - Penggunaan bersama sumber daya
 - Peningkatan komputasi – pemerataan beban
 - Keandalan
 - Komunikasi

Sistem Terdistribusi (2)

- Network OS
 - Menyediakan penggunaan file bersama
 - Menyediakan skema komunikasi
 - Independen thd komputer lainnya di jaringan
- Distributed OS
 - Tingkat otonomi antar komputer lebih rendah
 - Seolah-olah ada satu OS yg mengontrol jaringan

Migrasi Konsep dan Fitur OS



Ragam OS yg Ada Saat Ini

- Mainframe OS
- Multiprosesor OS
- PC OS
- Realtime OS
- Network OS
- Embedded OS
- Smart Card OS

Konsep Dasar OS

- Komponen dasar OS:
 - Manajemen proses
 - Manajemen file
 - Manajemen I/O dan media penyimpanan
 - Sistem proteksi
 - Jaringan

Layanan OS

- **Eksekusi program** – me-load program ke memori dan menjalankannya
- **Operasi I/O** – utk membantu program user melakukan I/O krn program user tdk dpt melakukan operasi I/O secara langsung
- **Manipulasi sistem file** – membaca, menulis, membuat, dan menghapus file
- **Komunikasi** – pertukaran informasi antar proses yg berjalan pd komputer yg sama atau berbeda
- **Pendeteksian kesalahan** – menjamin kebenaran komputasi utk mendeteksi kesalahan pd CPU, memori, perangkat I/O, atau program user

Manajemen Proses

- **Process:** program yg dieksekusi.
- Proses memerlukan sumberdaya (waktu CPU, memori, file, dan perangkat I/O) utk menyelesaikan tugasnya
- Fungsi OS dalam manajemen proses:
 - Membuat dan menghapus proses
 - Menunda dan memulai kembali proses
 - Mendukung mekanisme utk sinkronisasi dan komunikasi antar proses

Manajemen Memori Utama

- **Memori:** array words atau bytes yg berukuran besar, masing2 dgn alamatnya sendiri
- Datanya dpt diakses secara cepat oleh CPU dan perangkat I/O
- Bersifat volatile, isinya dpt hilang jika terjadi kegagalan sistem
- Peran OS dlm manajemen memori:
 - Mengetahui bagian memori yg sedang diakses dan siapa yg mengakses
 - Menentukan proses mana yg akan di-load jika ada ruang memori yg tersedia
 - mengalokasikan dan mendealokasikan ruang memori ketika diperlukan

Manajemen Media Penyimpanan Sekunder

- Media penyimpanan sekunder merupakan back up thd memori utama
- Kebanyakan sistem komputer modern menggunakan disk sbg media penyimpanan on-line utk program dan data
- Peran OS dalam manajemen disk:
 - Pengelolaan ruang kosong
 - Alokasi penyimpanan
 - Penjadwalan disk

Manajemen Sistem I/O

- Sistem I/O terdiri dari
 - Sistem buffer-caching
 - Antarmuka device-driver umum
 - Drivers utk H/W spesifik

Manajemen File

- **File:** kumpulan informasi terkait yg didefinisikan oleh pembuatnya
- Umumnya, file merepresentasikan program dan data
- Peran OS dalam manajemen file:
 - Membuat dan menghapus file dan direktori
 - Mendukung primitif utk manipulasi file dan direktori
 - Pemetaan file ke media penyimpanan sekunder
 - Back up file pada media penyimpanan yg stabil

Sistem Proteksi

- **Proteksi:** mekanisme utk mengontrol akses oleh program, proses, atau user thd sumberdaya sistem dan pengguna
- Mekanisme proteksi haruslah:
 - Membedakan antara penggunaan sah dan tidak sah
 - Menspesifikasikan kontrol yg diterapkan
 - Menyediakan sarana utk menerapkannya

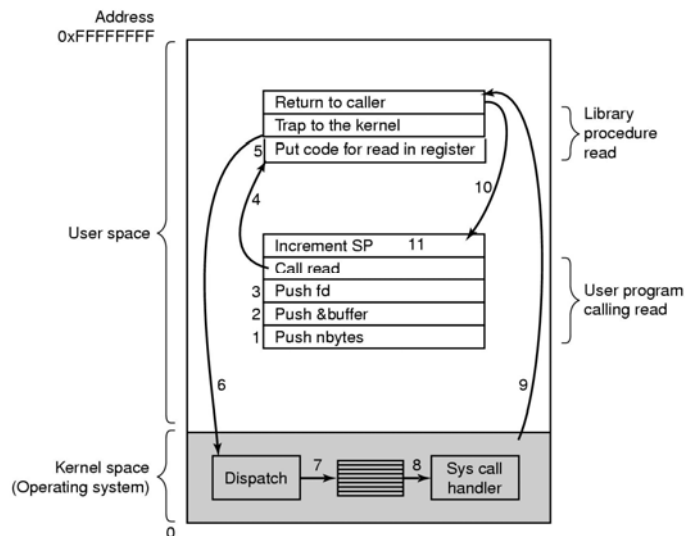
Jaringan (Sistem Terdistribusi)

- Sistem terdistribusi merupakan kumpulan prosesor yg masing2 memiliki memori lokal dan clock sendiri
- Pd sistem ini prosesor terhubung melalui jaringan komunikasi
- Sistem terdistribusi menyediakan akses ke berbagai sumber daya sistem
- Akses ke sumber daya yg digunakan bersama memungkinkan:
 - Peningkatan kecepatan komputasi
 - Peningkatan ketersediaan data
 - Peningkatan keandalan

System Call

- System call menyediakan antarmuka antar program yg sedang berjalan dan OS
- 3 metode yg umum digunakan utk melewati parameter ant program yg sdg berjalan dan OS:
 - Melewatkan parameter pd register
 - Menyimpan parameter pd tabel di memori dan alamat tabel dilewatkan sbg parameter di register
 - Menyimpan (*push*) parameter pd stack oleh program dan mengeluarkan (*pop*) dari stack oleh OS

System Call: Tahapan



System Call: Contoh (1)

Process management

Call	Description
pid = fork()	Create a child process identical to the parent
pid = waitpid(pid, &statloc, options)	Wait for a child to terminate
s = execve(name, argv, environp)	Replace a process' core image
exit(status)	Terminate process execution and return status

File management

Call	Description
fd = open(file, how, ...)	Open a file for reading, writing or both
s = close(fd)	Close an open file
n = read(fd, buffer, nbytes)	Read data from a file into a buffer
n = write(fd, buffer, nbytes)	Write data from a buffer into a file
position = lseek(fd, offset, whence)	Move the file pointer
s = stat(name, &buf)	Get a file's status information

System Call: Contoh (2)

Directory and file system management

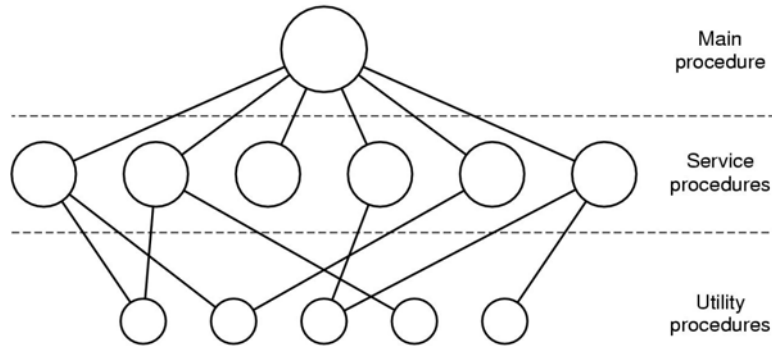
Call	Description
s = mkdir(name, mode)	Create a new directory
s = rmdir(name)	Remove an empty directory
s = link(name1, name2)	Create a new entry, name2, pointing to name1
s = unlink(name)	Remove a directory entry
s = mount(special, name, flag)	Mount a file system
s = umount(special)	Unmount a file system

Miscellaneous

Call	Description
s = chdir(dirname)	Change the working directory
s = chmod(name, mode)	Change a file's protection bits
s = kill(pid, signal)	Send a signal to a process
seconds = time(&seconds)	Get the elapsed time since Jan. 1, 1970

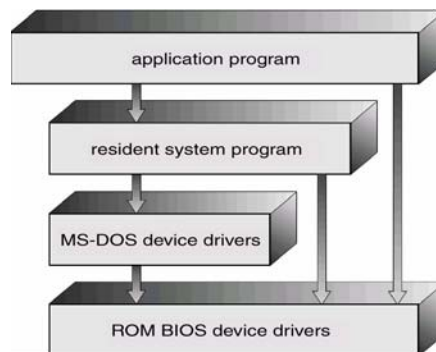
Struktur OS: Struktur Sederhana (1)

- Monolitik → tanpa struktur, tidak modular



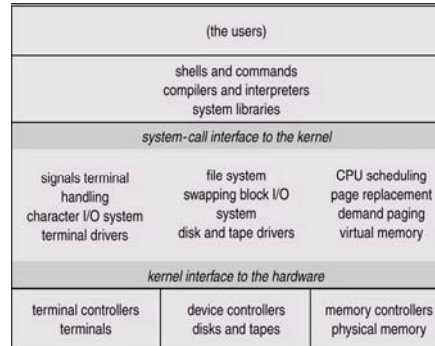
Struktur OS: Struktur Sederhana (2)

- MS-DOS – menyediakan fungsionalitas dgn ruang seminimal mungkin
 - Tidak dibagi dlm modul
 - Meskipun ada sedikit struktur, namun antarmuka dan tingkat fungsionalitasnya tdk terpisah dgn baik



Struktur OS: Struktur Sederhana (3)

- UNIX – original UNIX memiliki struktur yg terbatas
- Terdiri dari 2 bagian
 - Program sistem
 - Kernel (terdiri dari semua yg dibawah antarmuka system-call dan diatas H/W)



Struktur OS: Struktur Berlapis (1)

- OS dibagi dlm beberapa lapisan:
 - lapisan plg bawah (layer 0) → hardware
 - Lapisan paling atas (layer N) → antarmuka user
- Dgn modularitas, lapisan dipilih sedemikian shg masing2 hanya menggunakan fungsi dan layanan yg disediakan lapisan di bawahnya

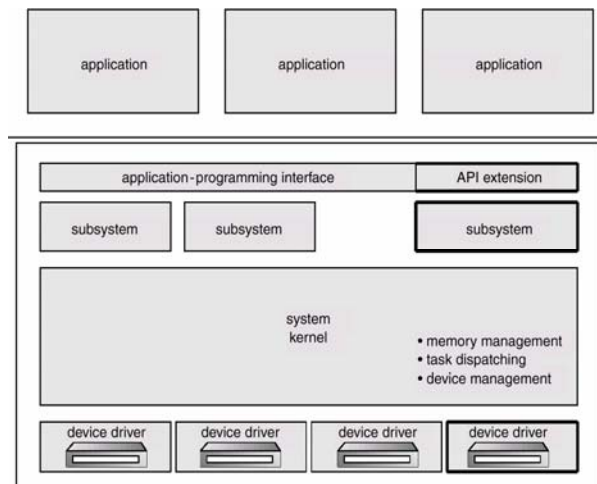
Struktur OS: Struktur Berlapis (2)

- Contoh: THE

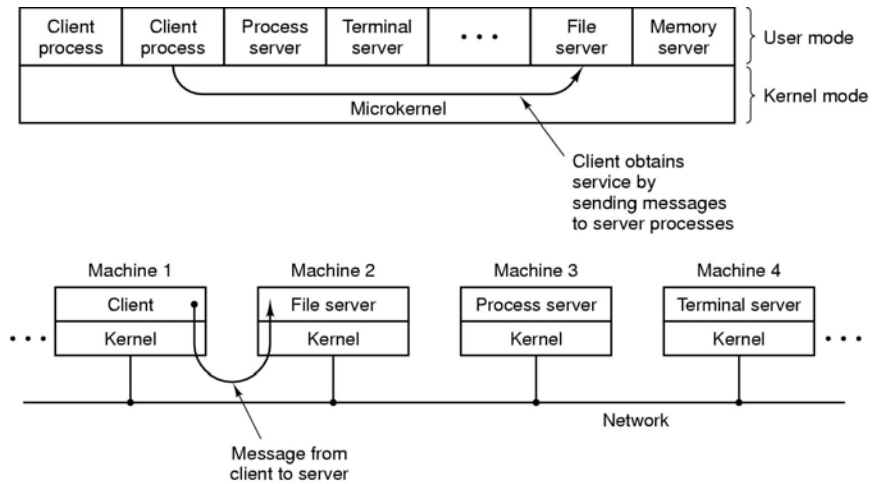
Layer	Function
5	The operator
4	User programs
3	Input/output management
2	Operator-process communication
1	Memory and drum management
0	Processor allocation and multiprogramming

Struktur OS: Struktur Berlapis (3)

- Contoh: OS2



Struktur OS: Client-Server (1)



Struktur OS: Client-Server (2)

- Contoh: Windows NT

