

Struktur Sistem Operasi



Komponen sistem

- ❑ Manajemen proses
- ❑ Manajemen memori utama
- ❑ Manajemen file
- ❑ Manajemen I/O
- ❑ Manajemen memori sekunder

Manajemen proses (1)

- ❑ **Proses** dinyatakan sebagai program yang dieksekusi
 - Sebuah batch job
 - Sebuah time-shared user program
 - Sebuah system task seperti spooling ke printer
- ❑ Proses memerlukan **resource** untuk menyelesaikan task
 - CPU time, memory, file dan I/O device
 - Semua resource dapat diberikan saat proses dibuat atau dialokasikan saat proses berjalan
- ❑ Program **BUKAN** suatu proses. Program adalah entiti pasif seperti file yang disimpan di disk, proses adalah **entiti aktif**, dg program counter untuk menunjuk ke instruksi berikutnya yang akan dieksekusi
- ❑ Pada sistem, proses dapat dinyatakan sebagai unit kerja yang berisi kumpulan proses :
 - Beberapa merupakan proses OS (mengeksekusi kode sistem)
 - Sisanya adalah user proses (mengeksekusi kode user)

Manajemen proses (2)

- Aktivitas yang harus ditangani oleh OS yang berhubungan dg manajemen proses :
 - Membuat (*creation*) dan menghapus (*deletion*) baik proses user maupun proses sistem
 - Menghentikan sementara (*suspension*) dan melanjutkan (*resumption*) proses
 - Menyediakan mekanisme untuk sinkronisasi proses
 - Menyediakan mekanisme untuk komunikasi proses
 - Menyediakan mekanisme untuk menangani deadlock

Manajemen memori utama (1)

- ❑ Memori adalah array besar ukuran word atau byte, dengan alamat tertentu
- ❑ Memori adalah gudang pengaksesan data shared yang cepat oleh CPU dan I/O device
- ❑ CPU membaca instruksi dari main memory selama siklus instruction-fetch, dan baik membaca dan menulis data dari dan ke memori selama siklus data-fetch
- ❑ I/O diimplementasi melalui DMA juga membaca dan menulis data di main memory
- ❑ Program awalnya harus dipetakan ke alamat absolut dan disimpan di memory untuk dieksekusi
- ❑ Beberapa program harus diletakkan di memori untuk meningkatkan utilitas CPU dan kecepatan dari respon komputer

Manajemen memori utama (2)

- ❑ Terdapat beberapa skema manajemen memory yang berbeda, tergantung beberapa faktor terutama desain hardware
- ❑ Aktivitas yang ditangani oleh OS yang berhubungan dengan manajemen memory :
 - Menyimpan bagian memori yang digunakan dan siapa yang menggunakan
 - Memutuskan proses yang disimpan ke memory bila tersedia ruang memory
 - Mengalokasikan dan men-dealokasikan ruang memory sesuai kebutuhan

Manajemen penyimpanan sekunder

- ❑ *Secondary storage* (disk) digunakan sebagai penyimpanan on-line medium baik program maupun data
- ❑ Kebanyakan program, seperti compiler, assembler, editor dll disimpan di disk sampai disimpan di memory
- ❑ OS bertanggung jawab untuk aktifitas yang berhubungan dengan manajemen disk sbb :
 - Manajemen ruang bebas (free space)
 - Alokasi penyimpanan
 - Penjadwalan disk
- ❑ Penyimpanan sekunder harus digunakan secara efisien.

Manajemen sistem I/O

- ❑ Salah satu tujuan OS adalah menyembunyikan kerumitan device H/W dari user
 - UNIX menyediakan sistem I/O untuk menyembunyikan device I/O dari bulk
- ❑ Sistem I/O terdiri dari :
 - Sistem buffer-catching
 - General device-driver interface
 - Drivers for specific H/W device

Manajemen File (1)

- ❑ Manajemen file adalah satu komponen yang visible dari O/S
- ❑ File adalah unit penyimpan logika, yang merupakan abstraksi dari properti fisik dari perangkat penyimpan yang digunakan OS
 - O/S memetakan file ke media fisik dan mengakses file melalui storage device
- ❑ File adalah kumpulan informasi yang berhubungan dengan pembuatnya
- ❑ File berisi urutan bit, byte, baris atau record yang berarti bagi creatornya
- ❑ OS mengimplementasikan konsep abstrak dari file dengan mengatur mass storage media seperti tape dan disk dan device yang mengontrolnya

Manajemen File (2)

- File secara normal diorganisasikan ke dalam direktori. Bila banyak user mengakses file, perlu ada kontrol oleh siapa dan dengan cara apa file diakses
- Aktivitas yang menjadi tanggung jawab OS yang berhubungan dengan manajemen file :
 - Pembuatan dan penghapusan file
 - Pembuatan dan penghapusan direktori
 - Primitif-primitif yang mendukung untuk manipulasi file dan direktori
 - Pemetaan file ke memori sekunder
 - Backup file ke media penyimpanan yang stabil (nonvolatile)

Sistem Proteksi

- Dalam sistem yang memungkinkan multiple user dan concurrent process, proses-2 yang berbeda harus diproteksi dari aktifitas proses yang lain
 - Mekanisme harus ditentukan untuk menjamin file, memory, CPU dan resource lain dapat dioperasikan oleh proses-proses yang menambah kegunaan OS
- Proteksi merupakan mekanisme mengontrol akses program, proses atau user untuk resource yang ditentukan sistem komputer

Networking

- ❑ Hal yang penting dalam sistem terdistribusi dimana kumpulan prosessor tidak membagi memory dan clock; tetapi setiap memory mempunyai memory lokal sendiri dan prosessor berkomunikasi dengan yang lain melalui saluran komunikasi yang berbeda
- ❑ Prosessor dalam sistem dihubungkan melalui jaringan komunikasi yang dikonfigurasi dengan beberapa cara yang berbeda
- ❑ Sistem terdistribusi mengumpulkan sistem yang heterogen secara fisik ke dalam sistem yang koheren, menentukan akses user ke resource berbeda yang ditangan sistem

Sistem Command Interpreter

- *Command interpreter* adalah system program terpenting dari OS dan bertindak sebagai antar muka antara user dengan OS
- Beberapa OS memasukkan command interpreter dalam kernel; beberapa lagi seperti MS-DOS dan UNIX memperlakukan command interpreter sebagai program khusus yang dijalankan saat job diinisialisasi
- Beberapa command yang diberikan OS sebagai pernyataan kontrol :
 - Bila job baru dimulai dalam sistem batch atau bila user masuk ke sistem time-shared, program yg membaca dan meng-interpret pernyataan kontrol dieksekusi otomatis
 - Program diatas biasanya disebut *control-card interpreter*, *command-line interpreter* dan *shell* (UNIX)
- OS secara terus menerus mengakses area command interpreter, interpreter yang user-friendly membuat sistem lebih disukai user

Pelayanan Sistem Operasi (1)

- ❑ OS menyediakan pelayanan untuk program dan untuk user dari program tsb
- ❑ Pelayanan berbeda untuk OS satu dengan lainnya
- ❑ Pelayanan OS disediakan untuk kenyamanan programmer seperti membuat programming task lebih mudah

Pelayanan Sistem Operasi (2)

- ❑ **Eksekusi program** → sistem memanggil program ke memori dan menjalankannya, program dapat mengakhiri eksekusinya dalam bentuk normal atau abnormal
- ❑ **Operasi-operasi I/O** → user tidak boleh mengontrol I/O secara langsung, pengontrolan dilakukan sistem operasi
- ❑ **Manipulasi sistem file** → pembuatan, penghapusan, read dan write
- ❑ **Komunikasi**
 - OS harus menyediakan program yang memungkinkan proses bertukar informasi dengan proses lain
 - Komunikasi antar proses dapat terjadi pada komputer yang sama atau komputer yang berbeda
 - Komunikasi diimplementasikan dengan “shared memory” atau dengan teknik “message passing” dalam paket informasi yang berpindah antar proses dengan OS
- ❑ **Mendeteksi kesalahan** → OS harus sanggup mendeteksi beberapa error yang terjadi pada CPU, memory, device I/O dll

Pelayanan Sistem Operasi (3)

- Beberapa fungsi untuk meyakinkan operasi yang efisien dari sistem :

- **Resource allocation**

- Bila terdapat multiple user atau multiple job berjalan pada waktu yang bersamaan, resource harus dialokasikan masing-masing

- **Accounting**

- Menyediakan keep track user yang menggunakan dan jenis dari resource
 - Menyimpan data yang dibutuhkan untuk accounting atau secara sederhana statistik akumulasi penggunaan

- **Protection**

- Kemungkinan owner mempunyai kontrol ke informasi yang disimpan di sistem komputer multi-user
 - Sekuriti (keamanan) dari sistem dari orang asing harus dipertimbangkan

System Calls (1)

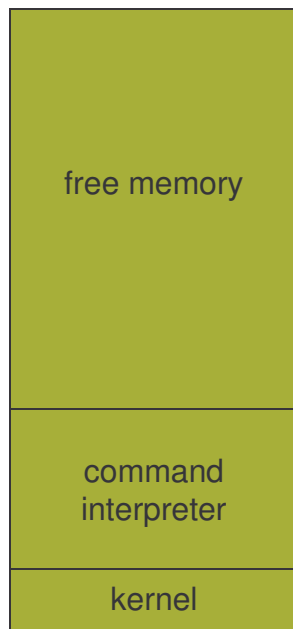
- ❑ Sistem call merupakan interface antara proses dengan OS yang tersedia sebagai instruksi bahasa assembly
- ❑ Beberapa sistem mengizinkan system calls dibuat secara langsung dari program high level language
- ❑ Beberapa bahasa, seperti C ditentukan sebagai pengganti bahasa assembly untuk pemrograman sistem
- ❑ System call terjadi dg cara berbeda tergantung penggunaan komputer :
 - Menambah informasi yang dibutuhkan, tidak hanya identitas dari sistem call tertentu
 - Tipe dan jumlah informasi bervariasi tergantung OS dan system call

System Calls (2)

- ❑ Terdapat 3 metode passing parameter pada OS :
 - Pass parameter pada register
 - Jika jumlah parameter lebih banyak dari jumlah register, parameter disimpan dalam "block" (table) dlm memory, dan alamat blok disimpan sebagai parameter pada register
 - Parameter dapat ditempatkan (push) ke "stack" oleh program dan "pop" dari stack oleh OS
- ❑ System calls dikelompokkan dalam 5 kategori :
 - **Process control** : end, abort, load, execute, create process, terminate process, get process attributes, set process attributes, wait for time, wait event, signal event, allocate and free memory
 - **File manipulation** : create file, delete file, open, close, read, write, reposition, get file attributes, set file attributes
 - **Device manipulation** : request device, release device, read, write, reposition, get device attributes, set device attributes, logically attach or detach devices
 - **Information maintenance** : get time or date, set time or date, get system data, set system data, get process, file, or device attributes, set process, file or device attributes
 - **Communication** : create, delete communication connection, send, receive message, transfer status information, attach or detach remote devices

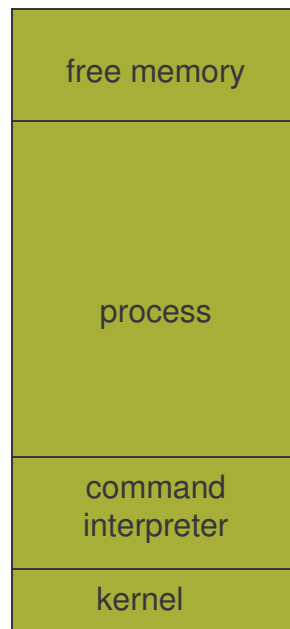
System Calls (3)

MS DOS execution



(a)

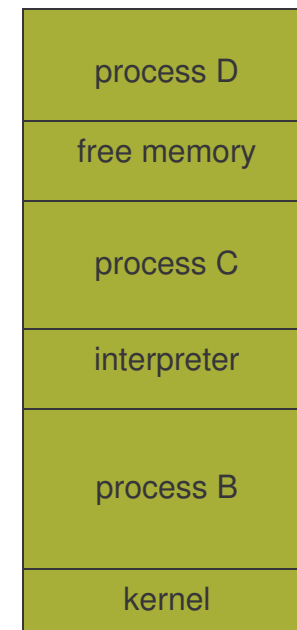
At system startup



(b)

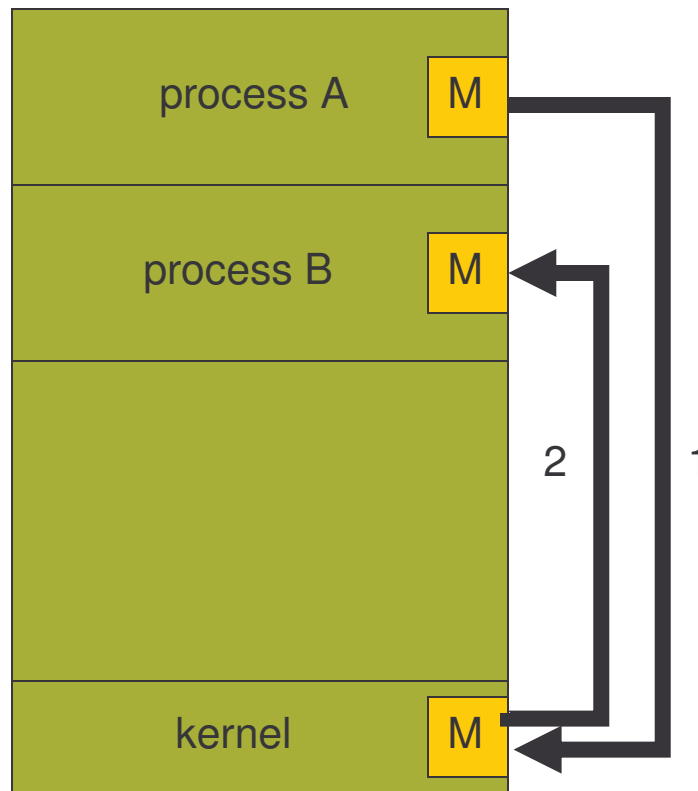
Running a program

UNIX running multiple program

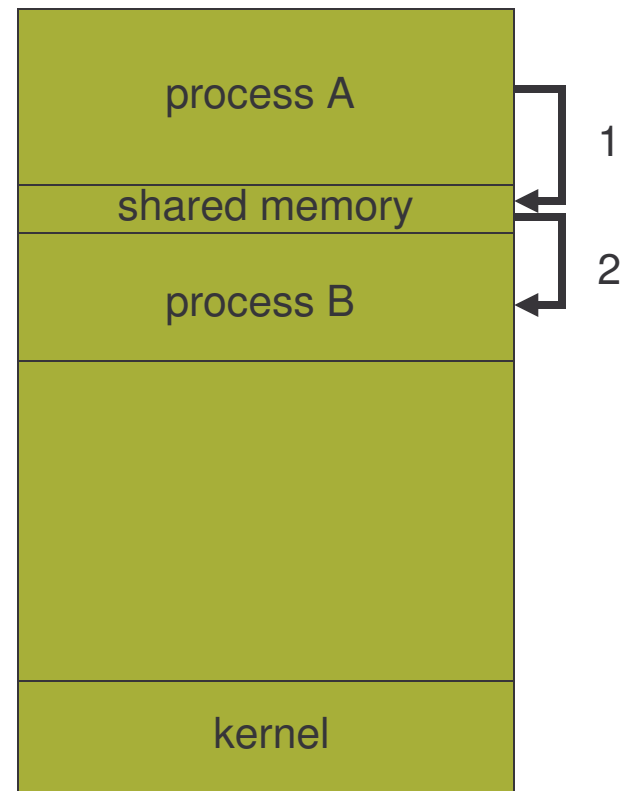


System Calls (4)

Model Komunikasi



(a) Message Passing



(b) Shared Memory

System program

- ❑ **OS terdiri dari kumpulan system program**
- ❑ **System program berada antara OS dengan program aplikasi**
- ❑ **System program dibagi menjadi beberapa kategory**
 - **Manipulasi file**
 - ❑ Program ini umumnya memanipulasi file dan direktory : create, delete, copy, rename dll
 - **Informasi status**
 - ❑ Beberapa program menanyakan sistem untuk informasi status seperti tanggal, waktu, jumlah user, jumlah penggunaan memori atau disk space dll
 - **Modifikasi file**
 - ❑ Menyediakan beberapa text editor
 - **Bahasa pemrograman**
 - ❑ Menyediakan ompiler, assembler dan interpreter
 - **Loading dan eksekusi program**
 - ❑ Menyediakan loader untuk menyimpan program yang di ompile ke main memory, seperti absolute loader dan relocatable loader
 - **Komunikasi**
 - ❑ Menyediakan mekanisme untuk membuat hubungan virtual antar proses, user dan sistem komunikasi yang berbeda
 - **Program-program aplikasi**
 - ❑ Program program aplikasi yang digunakan bersama dengan OS, contoh : compiler to compiler, pemformat teks dll

Struktur Dasar Sistem Operasi

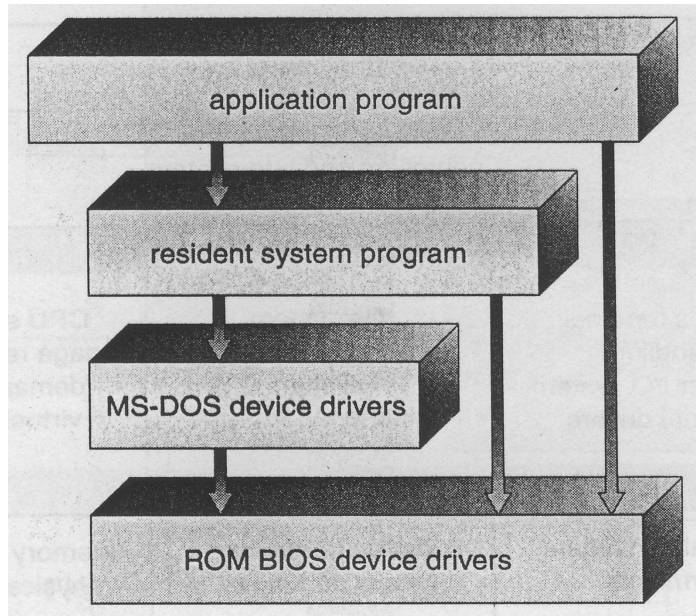
- ❑ Struktur Sederhana
- ❑ Sistem berlapis (layered)
- ❑ Mesin maya (virtual machine)

Sistem Sederhana (1)

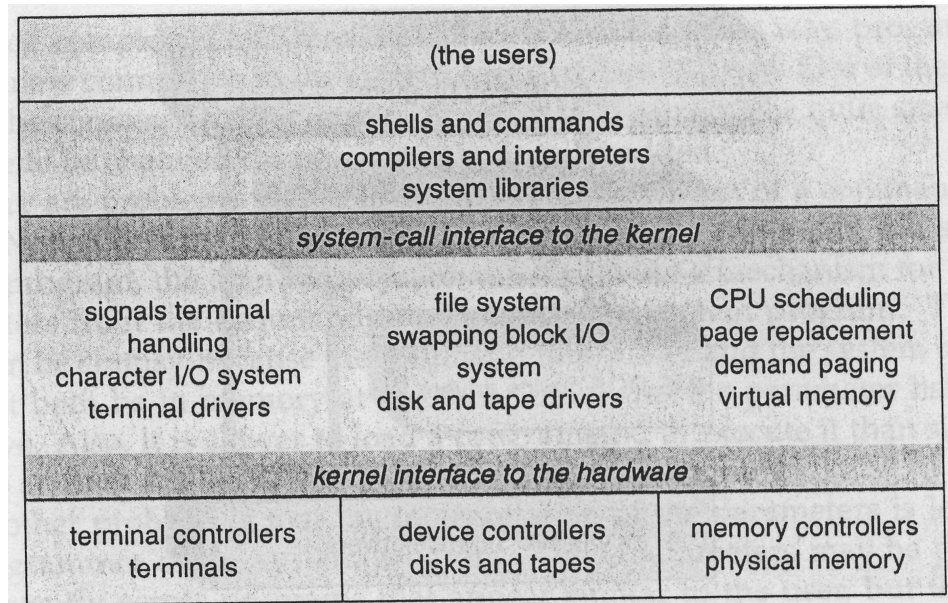
- ❑ Karakteristik :
 - Struktur sistem tidak didefinisikan dengan baik
 - Biasanya merupakan sistem kecil, sederhana dan terbatas, dan kemudian berkembang
 - Contoh : MS-DOS, Original UNIX
- ❑ MS-DOS :
 - Meskipun mempunyai beberapa struktur, fungsi interface dan level tidak dipisahkan dg baik
 - Program aplikasi dapat mengakses routine dasar I/O untuk menulis langsung ke display dan disk drive
 - Terbatas oleh H/W tempat menjalankan
- ❑ ORIGINAL UNIX :
 - Struktur terbatas dan fungsi H/W terbatas
 - Terdiri dari 2 bagian terpisah : "Kernel" dan "System Program"
 - Kernel berada di bawah antar muka system call dan diatas H/W fisik
 - Kernel lebih jauh dibagi dalam deretan interface dan device driver
 - System program menggunakan sistem call yang didukung kernel untuk menyediakan fungsi yang berguna seperti kompilasi dan manipulasi file
 - Sistem call mendefinisikan "program interface" ke UNIX

Sistem Sederhana (2)

Struktur layer MS-DOS



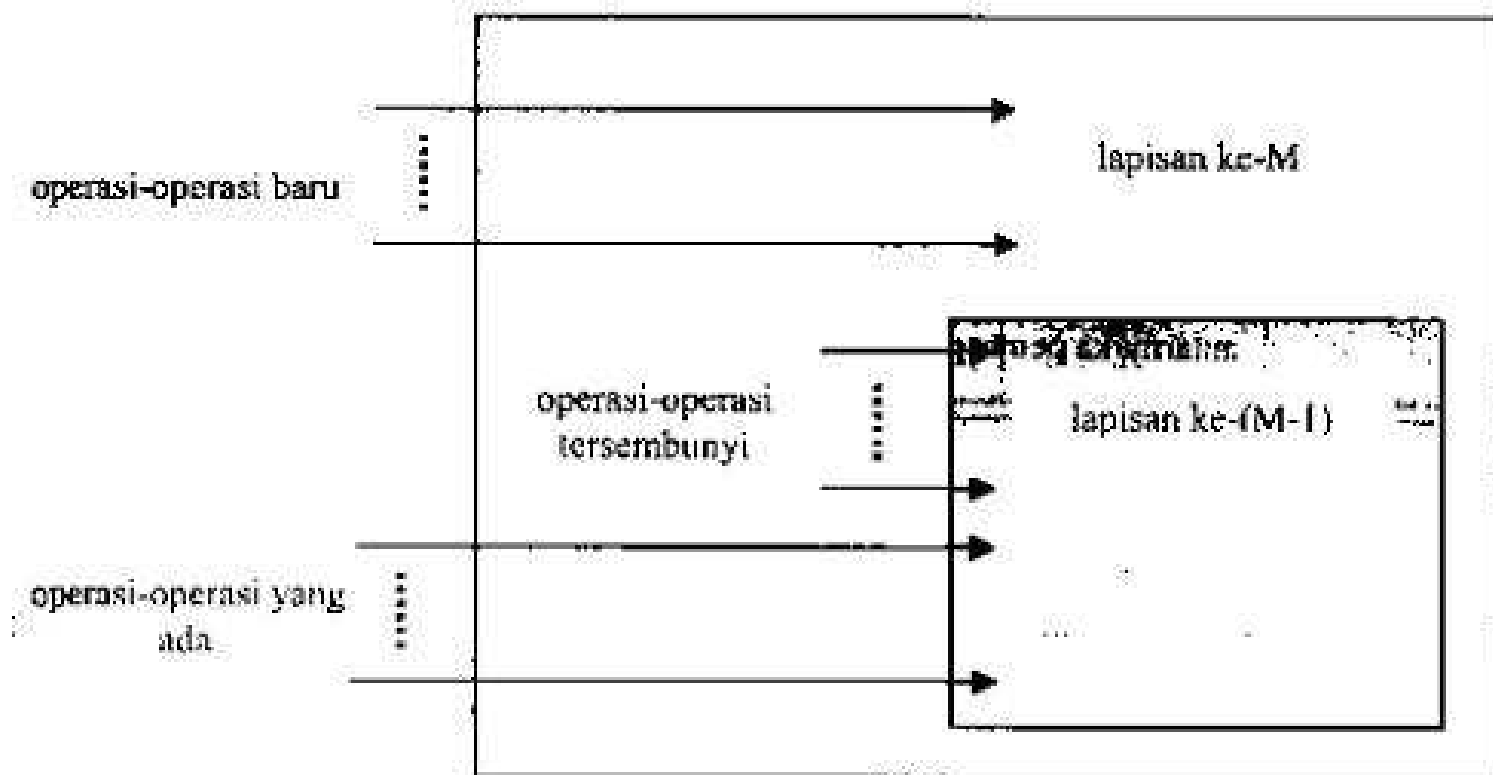
Struktur sistem UNIX



Sistem Layered (1)

- IDE :
 - menggunakan pendekatan top-down untuk menentukan fungsi secara keseluruhan dan gambaran OS dan membaginya ke dalam sejumlah komponen yang lebih kecil
 - Memecah OS ke sejumlah layer (level), level terbawah (layer 0) adalah H/W dan level tertinggi (layer N) adalah user interface
- Keuntungan :
 - Modularity : layer dipilih sesuai fungsi kegunaan (operasi) dan melayani layer yang lebih rendah
 - Debugging dan verifikasi sistem yang lebih sederhana : layer yang lebih rendah tertentu dapat di-debug tanpa menyangkut layer lebih tinggi
 - Setiap layer dapat menyembunyikan keberadaan struktur data, operasi dan H/W dari layer yang lebih tinggi. Suatu layer tidak perlu mengetahui bagaimana operasi diimplementasikan, yang perlu diketahui hanya apa yang dikerjakan suatu operasi
- Contoh : Technische Hogeschool Eindhoven (THE), sistem Venus

Sistem Layered (2)



Sistem Layered (3)

Struktur THE

Lapis-5 : user program

Lapis-4 : buffering untuk I/O device

Lapis-3 : operator-console device driver

Lapis-2 : manajemen memori

Lapis-1 : penjadwalan CPU

Lapis-0 : hardware

Struktur Venus

Lapis-6 : user program

Lapis-5 : device driver & scheduler

Lapis-4 : virtual memory

Lapis-3 : I/O channel

Lapis-2 : penjadwalan CPU

Lapis-1 : instruksi interpreter

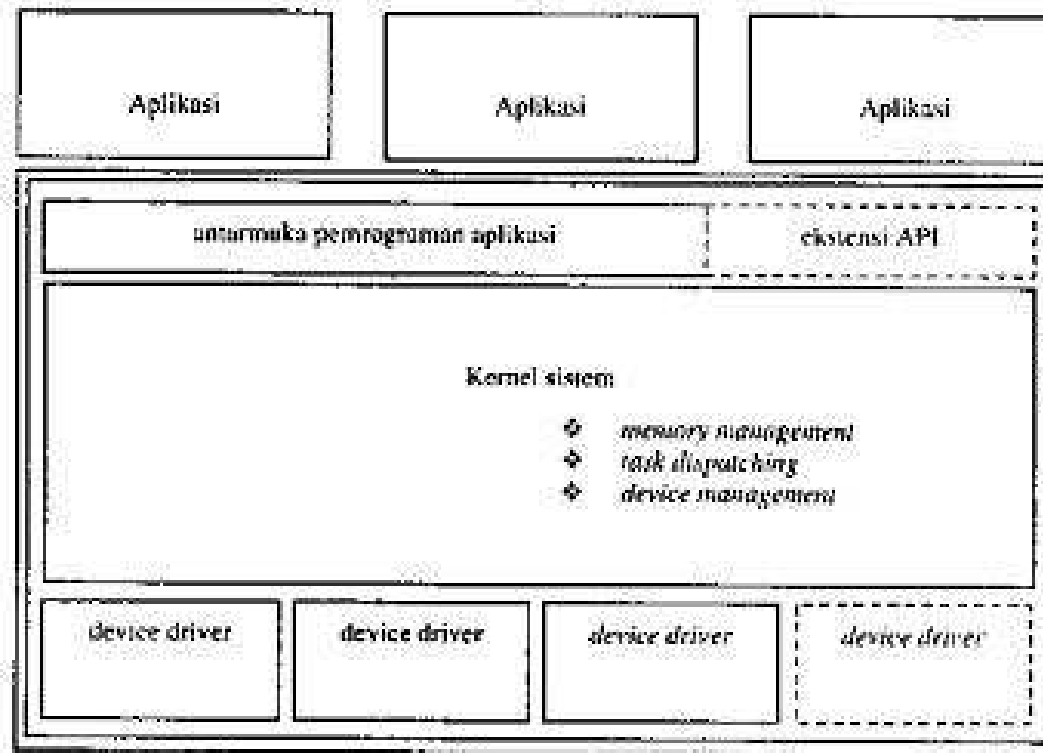
Lapis-0 : hardware

Sistem Layered (4)

- ❑ Kelemahan :
 - Kesulitan mendefinisikan layer yang tepat, karena sebuah layer hanya menggunakan layer yang berada di level yang lebih rendah. Diperlukan perencanaan secara hati-hati
 - Implementasi layer cenderung lebih tidak efisien dibandingkan sistem lain
- ❑ Sistem Operasi OS/2
 - Turunan langsung dari MS-DOS, dibuat untuk mengatasi keterbatasan MS-DOS dan menghindari kesulitan definisi dan interaksi layer
 - Operasi multi-tasking dan dual-mode dan feature lain ditambahkan ke OS/2
 - Sistem diimplementasikan lebih fashionable
 - Akses user langsung ke fasilitas pada level lebih rendah tidak diperbolehkan

Sistem Layered (5)

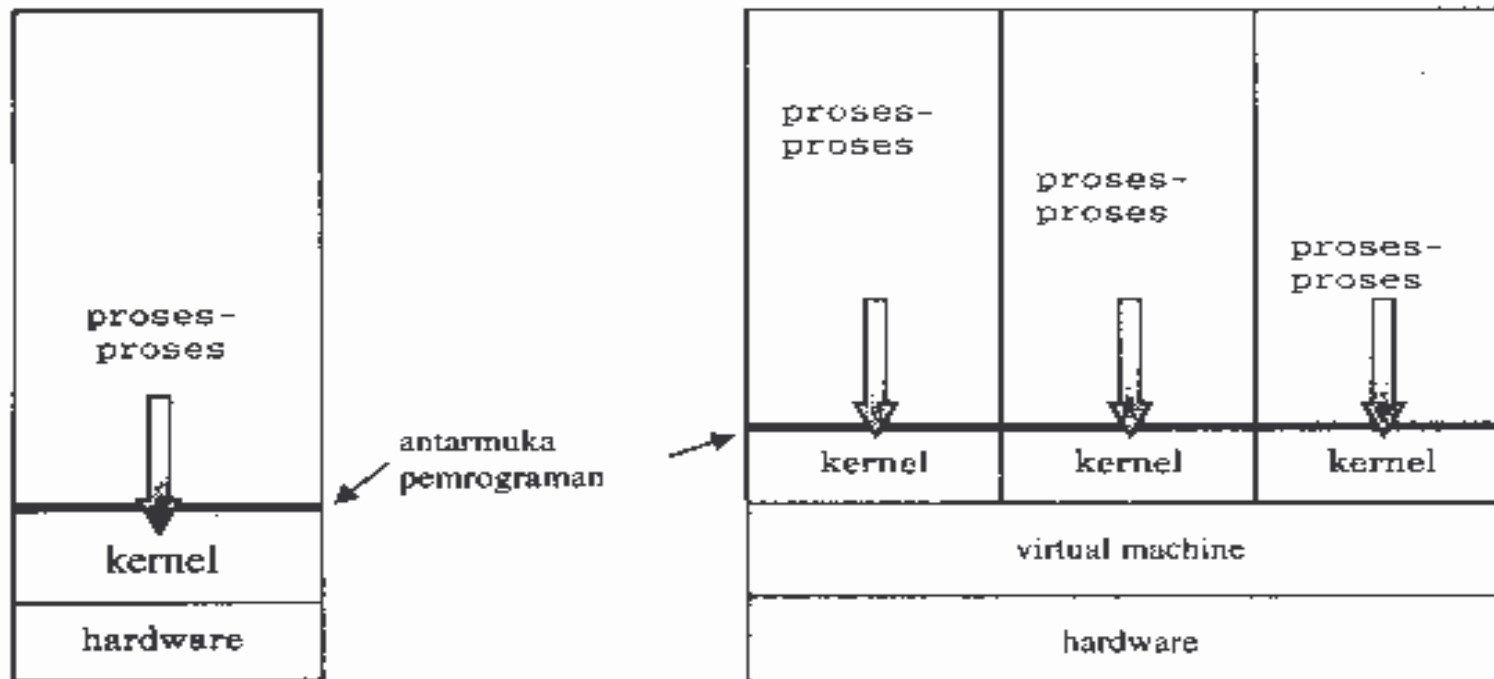
Sistem OS/2



Virtual Machine (1)

- IDE :
 - Memperluas pendekatan layer dengan memperbolehkan system program dipanggil dengan mudah oleh program aplikasi
 - Programmer sistem diberikan “virtual machine” tersendiri dan pengembangan sistem dilakukan pada virtual machine
- Contoh : VM
- Keuntungan utama :
 - Resource dari komputer fisik di-share untuk membuat virtual machine
 - Penjadwalan CPU dapat digunakan untuk membagi CPU dan menampilkan bahwa user mempunyai prosessor sendiri
 - Spooling dan sistem file disediakan oleh virtual card reader dan virtual line printer
 - Terminal time-sharing user yang normal menyediakan fungsi console operator dari virtual machine
- Kelemahan : implementasi yang efisien merupakan masalah yang sulit karena sistem menjadi besar dan kompleks

Virtual Machine (2)



Tanpa VM

Dengan VM

Studi Kasus – MS DOS

- Pandangan Pemakai :
 - Command language : perintah internal (dir, copy, del, cd, md, rd dll) & eksternal (.com, .exe, .bat)
- Pemrogram :
 - Layanan ROM BIOS berupa instruksi interupsi (utilitas Print-Screen, Video I/O, daftar perangkat, ukuran memori, disk I/O, serial port I/O, keyboard I/O , printer I/O dll)
 - Layanan MS DOS (IO.SYS dan MSDOS.SYS)
 - System Calls : operasi terhadap disk, direktori, pengelolaan file, perangkat masukan/keluaran, memori, kendali program, lingkungan eksekusi

Studi Kasus – MS Windows 95

- Pandangan Pemakai :
 - GUI, plug and play, nama file yang panjang
 - Build-in networking, pengamanan level pemakai, registry
 - Interface : menu, icon, task bar, window perintah
- Pemrogram :
 - System Calls disebut Win32 API :
 - window management
 - window controls
 - shell features : namespace dan shell link (shortcut)
 - graphics device interface(GDI)
 - system services
 - international features
 - network services

Studi Kasus – Windows NT

- Sistem operasi single-user, multitasking (multiprogramming)
- Perangkat lunak berorientasi aplikasi & sistem operasi yang berjalan pada privilege mode / kernel mode (NT executive)
- Sasaran : extensibility, portability, reliability dan robustness, compatibility, performance
- Arsitektur dasar :
 - Hardware abstraction Layer (HAL) : memetakan perintah dan tanggapan perangkat keras menjadi perintah dan tanggapan unik platform tertentu
 - Kernel : berisi komponen-komponen paling mendasar SO
 - Subsystem : modul fungsi-2 spesifik menggunakan layanan dasar kernel
 - System services : menyediakan interface ke perangkat lunak mode pemakai

Studi Kasus – UNIX^{TD}

- Karakteristik :
 - Interactive timesharing system : dirancang oleh pemrogram, untuk pemrogram, mengasumsikan pemakai menyukai pengembangan perangkat lunak
 - Expert friendly OS : menyediakan banyak fasilitas yang dapat bekerja sama dan berbagi informasi secara terkendali yang disukai para pakar
- UNIX Shell : bourne shell (sh), C shell (csh), korn shell (ksh), bourne again shell (bash)
- Pandangan Pemrogram :
 - Program utilitas dasar : canggih dan bagus, ditulis dengan bahasa script
 - Pustaka standard : fungsi-fungsi standar POSIX
 - System Calls (API) : seperti fungsi di bahasa C untuk memasuki layanan kernel

Studi Kasus – LINUX

- Multitasking, multiuser, multiplatform, multiprocessor, standard POSIX, pengaksesan transparan ke partisi MS-DOS, system file UMSDOS, implementasi TCP/IP networking
- Arsitektur dasar : kernel (jantung SO) yang menyediakan tool untuk semua layanan melalui system call, mencegah proses aplikasi mengakses h/w secara langsung, memberi proteksi kepada pemakai dari gangguan pemakai lain
- Pandangan pemakai : melalui perintah-perintah dan aplikasi-aplikasi pada system linux
- Pandangan Pemrogram : pengkombinasian program-program utilitas untuk suatu tujuan, rutin-rutin pustaka, system calls