

PERTEMUAN 3:

STRUKTUR SISTEM OPERASI

A. TUJUAN PEMBELAJARAN

Pada bab ini akan dijelaskan mengenai struktur sistem operasi, Anda harus mampu:

- 1.1 Komponen Sistem
- 1.2 Layanan Sistem Operasi
- 1.3 System Calls

B. URAIAN MATERI

Tujuan Pembelajaran 1.1:
Struktur Sistem Operasi

Struktur Sistem Operasi

Komponen-komponen Sistem

Pada kenyataannya tidak semua sistem operasi mempunyai struktur yang sama. Namun menurut Avi Silberschatz, Peter Galvin, dan Greg Gagne, umumnya sebuah sistem operasi modern mempunyai komponen sebagai berikut:

1. Manajemen Proses.
2. Manajemen Memori Utama.
3. Manajemen Secondary-Storage.
4. Manajemen Sistem I/O.
5. Manajemen Berkas.
6. Sistem Proteksi.
7. Jaringan.
8. Command-Interpreter system.

Manajemen Proses

Proses adalah keadaan ketika sebuah program sedang di eksekusi. Sebuah proses membutuhkan beberapa sumber daya untuk menyelesaikan tugasnya. sumber daya tersebut dapat berupa CPU time, memori, berkas-berkas, dan perangkat-perangkat I/O. Sistem operasi bertanggung jawab atas aktivitas-aktivitas yang berkaitan dengan manajemen proses seperti:

- ✓ Pembuatan dan penghapusan proses pengguna dan sistem proses.

- ✓ Menunda atau melanjutkan proses.
- ✓ Menyediakan mekanisme untuk proses sinkronisasi.
- ✓ Menyediakan mekanisme untuk proses komunikasi.
- ✓ Menyediakan mekanisme untuk penanganan deadlock.

Managemen Memori Utama

Memori utama atau lebih dikenal sebagai memori adalah sebuah array yang besar dari word atau byte, yang ukurannya mencapai ratusan, ribuan, atau bahkan jutaan. Setiap word atau byte mempunyai alamat tersendiri. Memori Utama berfungsi sebagai tempat penyimpanan yang akses datanya digunakan oleh CPU atau perangkat I/O. Memori utama termasuk tempat penyimpanan data yang sementara (volatile), artinya data dapat hilang begitu sistem dimatikan.

Managemen Secondary-Storage

Data yang disimpan dalam memori utama bersifat sementara dan jumlahnya sangat kecil. Oleh karena itu, untuk menyimpan keseluruhan data dan program komputer dibutuhkan secondary-storage yang bersifat permanen dan mampu menampung banyak data. Contoh dari secondary-storage adalah harddisk, disket, dll.

Managemen Sistem I/O

Sering disebut device manager. Menyediakan "device driver" yang umum sehingga operasi I/O dapat seragam (membuka, membaca, menulis, menutup). Contoh: pengguna menggunakan operasi yang sama untuk membaca berkas pada hard-disk, CD-ROM dan floppy disk.

Komponen Sistem Operasi untuk sistem I/O:

- ✓ Buffer: menampung sementara data dari/ ke perangkat I/O.
- ✓ Spooling: melakukan penjadualan pemakaian I/O sistem supaya lebih efisien (antrian dsb.).
- ✓ Menyediakan driver untuk dapat melakukan operasi "rinci" untuk perangkat keras I/O tertentu.

Managemen Berkas

Berkas adalah kumpulan informasi yang berhubungan sesuai dengan tujuan pembuat berkas tersebut. Berkas dapat mempunyai struktur yang bersifat hirarkis (direktori, volume, dll.).

Sistem operasi bertanggung-jawab:

- ✓ Pembuatan dan penghapusan berkas.
- ✓ Pembuatan dan penghapusan direktori.
- ✓ Mendukung manipulasi berkas dan direktori.
- ✓ Memetakan berkas ke secondary storage.
- ✓ Mem-backup berkas ke media penyimpanan yang permanen (non-volatile).

Sistem Proteksi

Proteksi mengacu pada mekanisme untuk mengontrol akses yang dilakukan oleh program, prosesor, atau

pengguna ke sistem sumber daya. Mekanisme proteksi harus:

- membedakan antara penggunaan yang sudah diberi izin dan yang belum.
- specify the controls to be imposed.
- provide a means of enforcement.

Jaringan

Sistem terdistribusi adalah sekumpulan prosesor yang tidak berbagi memori atau clock. Tiap prosesor mempunyai memori sendiri. Prosesor-prosesor tersebut terhubung melalui jaringan komunikasi. Sistem terdistribusi menyediakan akses pengguna ke bermacam sumber-daya sistem. Akses tersebut menyebabkan:

- Computation speed-up.
- Increased data availability.
- Enhanced reliability.

Command-Interpreter System

Sistem Operasi menunggu instruksi dari pengguna (command driven). Program yang membaca instruksi dan mengartikan control statements umumnya disebut: control-card interpreter, command-line interpreter, dan UNIX shell. Command-Interpreter System sangat bervariasi dari satu sistem operasi ke sistem operasi yang lain dan disesuaikan dengan tujuan

dan teknologi I/O devices yang ada. Contohnya: CLI, Windows, Pen-based (touch), dan lain-lain.

Layanan Sistem Operasi

Eksekusi program adalah kemampuan sistem untuk "load" program ke memori dan menjalankan program. Operasi I/O: pengguna tidak dapat secara langsung mengakses sumber daya perangkat keras, sistem operasi harus menyediakan mekanisme untuk melakukan operasi I/O atas nama pengguna.

System Calls

System call menyediakan interface antara program (program pengguna yang berjalan) dan bagian OS. System call menjadi jembatan antara proses dan sistem operasi. System call ditulis dalam bahasa assembly atau bahasa tingkat tinggi yang dapat mengendalikan mesin (C). Contoh: UNIX menyediakan system call: read, write => operasi I/O untuk berkas.

Mesin Virtual

Sebuah mesin virtual (Virtual Machine) menggunakan misalkan terdapat sistem program => control program yang mengatur pemakaian sumber daya perangkat keras. Control program = trap System call + akses ke perangkat keras.

CPU dan memori. Menyediakan interface "identik" dengan apa yang disediakan oleh perangkat keras =>

sharing devices untuk berbagai proses.

Mesin Virtual (MV) => control program yang minimal MV memberikan ilusi multitasking : seolah-olah terdapat prosesor dan memori eksklusif digunakan MV. MV memilah fungsi multitasking dan implementasi extended machine (tergantung proses pengguna) => flexible dan lebih mudah untuk pengaturan.

System Generation (SYSGEN)

Sistem operasi dirancang untuk dapat dijalankan di berbagai jenis mesin; sistemnya harus di konfigurasi untuk tiap komputer. Program SYSGEN mendapatkan informasi mengenai konfigurasi khusus dari sistem perangkat keras.

- Booting: memulai komputer dengan me-load kernel.

- Bootstrap program: kode yang disimpan di code ROM yang dapat menempatkan kernel, memasukkannya kedalam memori, dan memulai eksekusinya.

Struktur Sistem Operasi

- Komponen Sistem
- Layanan Sistem Operasi
- System Calls
- Program System
- StrukturSystem
- Virtual Machines
- System Design dan Implementation
- System Generation

Komponen Sistem Umum

- Manajemen Proses
- Manajemen Main Memory
- Manajemen File
- Manajaemen I/O System
- Manajemen Secondary Storage
- Command-Interpreter System

Manajemen Proses

- Proses adalah sebuah program yang sedang dijalankan (eksekusi).
 - Suatu proses memerlukan resources pada saat eksekusi: CPU time, memory, files dan I/O devices
- Sistim operasi bertanggungjawab terhadap aktifitas yang berhubungan dengan manajemen proses:
 - Process creation & deletion.
 - Process suspension (block) & resumption.
 - Mekanisme:
 - § Sinkronisasi antar proses
 - § Komunikasiantarproses

Manajemen Main - Memory

- Memori sebagai tempat penyimpanan instruksi / data dari program

- Storage yang cepa tsehingga dapat mengimbangi kecepatan eksekusi instruksi CPU
 - Terdiri dari “array of words / bytes” yang besar
 - Address digunakan untuk mengakses data (shared oleh CPU dan I/O devices)
- Umumnya main memory bersifat “volatile” – tidak permanent. Isinya akan hilang jika komputer dimatikan.
- Manajemen memori:
 - Melacak pemakaian memori (siapa dan berapa besar?).
 - Memilih program mana yang akan diload ke memori.
 - Alokasi dan De-alokasi memorifisik untuk program.

Manajemen Secondary - Storage

- Secondary Storage: penyimpanan permanen
 - Kapasitas harus besar untuk menyimpan semua program dan data.
 - Secondary storage dapat dijadikan “backup” storage main memory supaya dapaten jalankan banyak program.
 - Umumnya menggunakan “magnetic disks” (hard disk).
- OS bertanggungjawab untuk manajemen disk:
 - Manajemen ruang kosong
 - Alokasi storage
 - Penjadualan disk

Manajemen I/O System

- Sering disebut device manager
 - Menyediakan “device driver” yang umum sehingga operasi I/O dapat seragam (open, read, write, close)
 - § Contoh: user menggunakan operasi yang sama untuk read file pada hard disk, CD-ROM dan floppy disk sama.
- Komponen OS untuk sistim I/O:
 - Buffer: menampung sementara i data dari / keI / O devices
 - Spooling: melakukan scheduling pemakaian I/O sistim supaya lebih efisien (antrian dsb)
 - Menyediakan “driver” untuk dapat melakukan operasi “rinci” (detail) untuk hardware I/O tertentu.

Manajemen File

- File: kumpulan informasi yang berhubungan (sesuai dengan tujuan pembuat file tsb).
 - File dapat mempunyai struktur yang bersifat hirarkis (direktori, volume dll).
- OS bertanggungjawab:
 - Membuat dan menghapus file.
 - Membuat dan menghapus directory.
 - Dukungan primitif untuk manipulasi file dan directory.
 - Pemetaan file kedalam secondary storage.
 - Backup file kedia storage yang stabil (nonvolatile).

Command-Interpreter System

- OS: menunggu instruksi dari user (command driven)
- Program yang membaca instruksi dan mengartikan keinginan user (lebih dari sejenis).
 - Contoh:
 - § control-card interpreter
 - § command-line interpreter
 - § shell (in UNIX)
 - Sangat bervariasi dari satu OS ke OS yang lain dan disesuaikan dengan tujuan, teknologi I/O devices yang ada.
 - § CLI, Windows, Pen-based (touch) etc.

Layanan Sistem Operasi

- Eksekusi Program
 - Kemampuan sistem untuk “load” program ke memori dan menjalankan program.
- Operasi I/O
 - User tidak dapat secara langsung mengakses H/W resources, OS harus menyediakan mekanisme untuk melakukan operasi I/O atas nama user
- Manipulasi File-system
 - Kemampuan program untuk operasi pada file (to read, write, create, and delete files).

Layanan Sistem Operasi (Cont.)

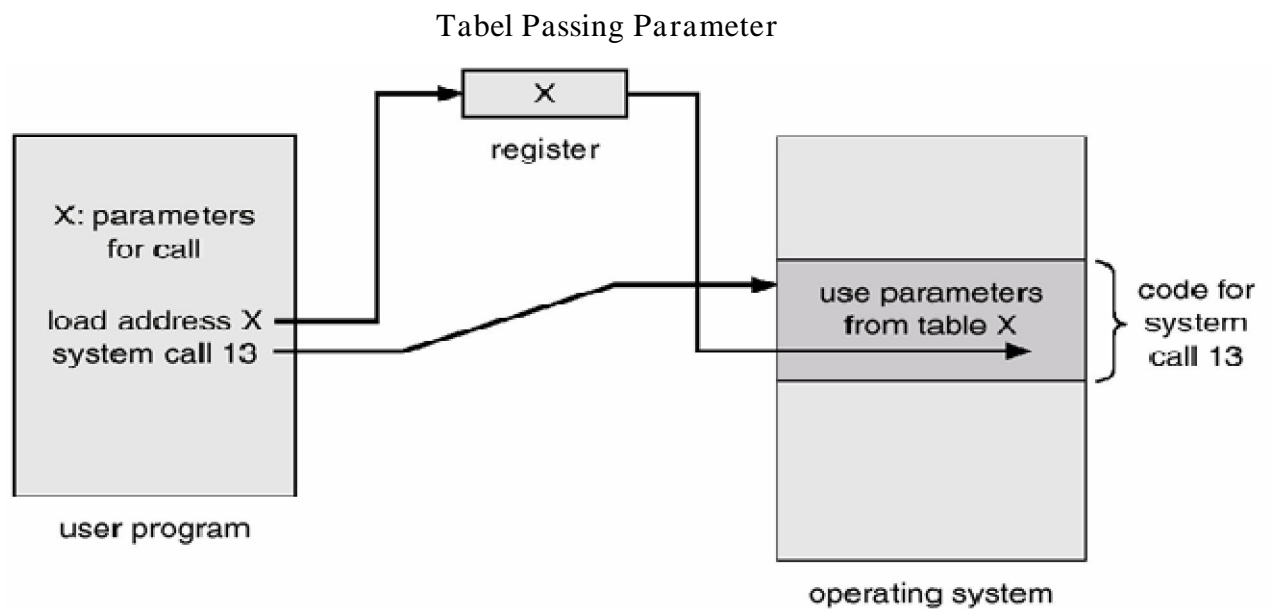
- Komunikasi
 - Pertukaran data / informasi antar dua atau lebih proses yang berada pada satu komputer (atau lebih).
- Deteksi Error
 - Menjaga kestabilan sistem dengan mendeteksi "error": hardware maupun operasi.
- Penggunaan System yang Efisien
 - Proteksi: menjamin akses ke system resources dikendalikan (user dikontrol akses ke sistem).
 - Accounting: merekam kegiatan users, jatah pemakaian resources (fairness atau policy).

System Call

- System call:
 - Menyediakan interface antara program (user program yang berjalan) dan bagian OS.
- System call menjadi jembatan antara proses dan OS.
 - System call ditulis dalam assembly language (machine specific) atau bahasa tingkat tinggi yang dapat mengendalikan mesin (C).
 - Contoh: UNIX menyediakan system call: read, write => operasi I/O untuk file.

System Call : Passing Parameter

- Sering user program harus memberikan data (parameter) ke rutin OS yang akan dipanggil.
 - UNIX: read (buffer, max_size, file_id);
- Tiga cara memberikan parameter dari program ke sistem operasi:
 - Melalui registers (resources di CPU).
 - Menyimpan parameter pada data struktur (table) di memory, dan alamat table tsb ditunjuk oleh pointer yang disimpan di register.
 - Push (store) melalui "stack" pada memori dan OS mengambilnya melalui pop pada stack tsb.



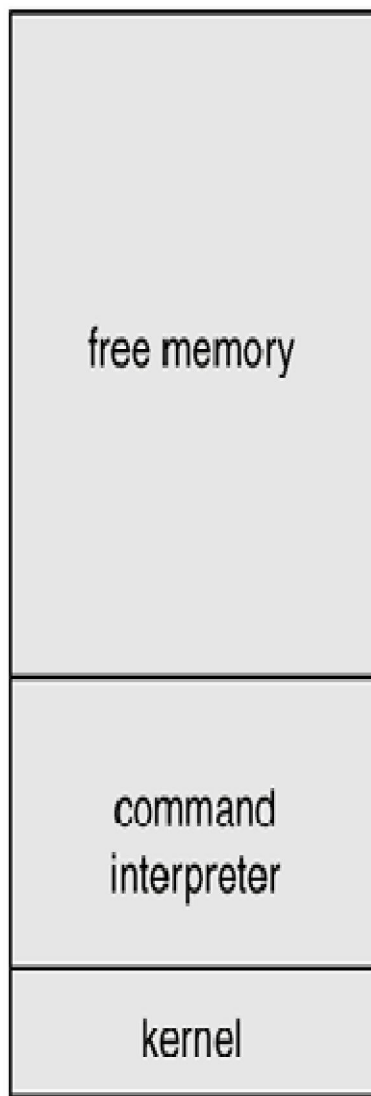
Tipe System Calls

- Tipe System Calls
- Kontrol Proses
- Manipulasi File
- Manajemen Device
- Informasi Lingkungan
- Komunikasi

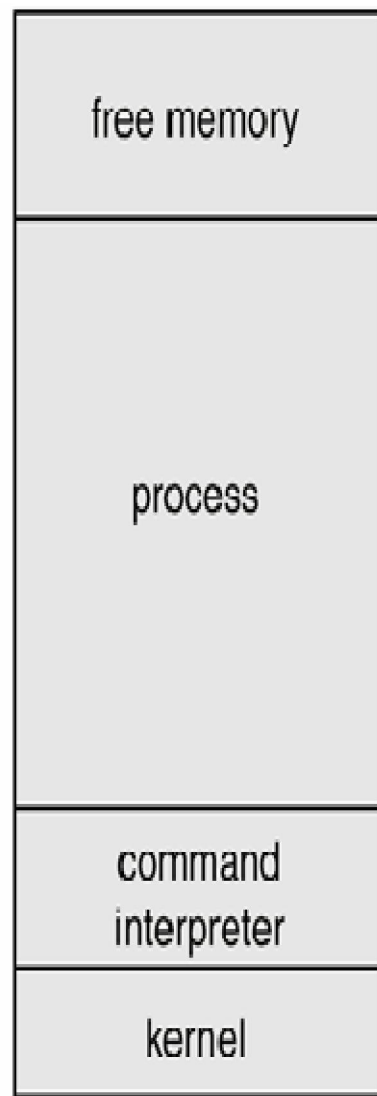
Kontrol Proses

- Mengakhiri (end) dan membatalkan (abort)
- Mengambil (load) dan eksekusi (execute)
- Membuat dan mengakhiri proses
- Menentukan dan mengeset atribut proses
- Wait for time
- Wait event, signal event
- Mengalokasikan dan membebaskan memori

Eksekusi MS-DOS

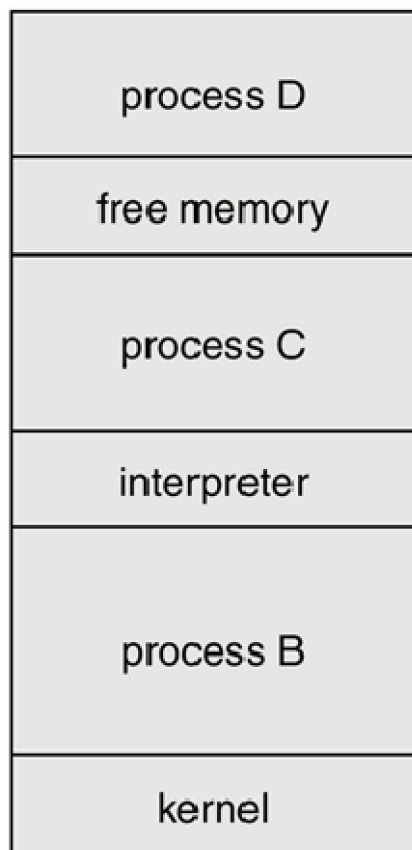


(a)



(b)

UNIX Menjalankan Multiple Program



Manipulasi File

- Membuat dan menghapus file
- Membuka dan menutup file
- Membaca, menulis dan mereposisi file
- Menentukan dan mengeset atribut file

Managemen Device

- Meminta dan membebaskan device
- Membaca, menulis dan mereposisi device
- Menentukan dan mengeset atribut device

Informasi Lingkungan

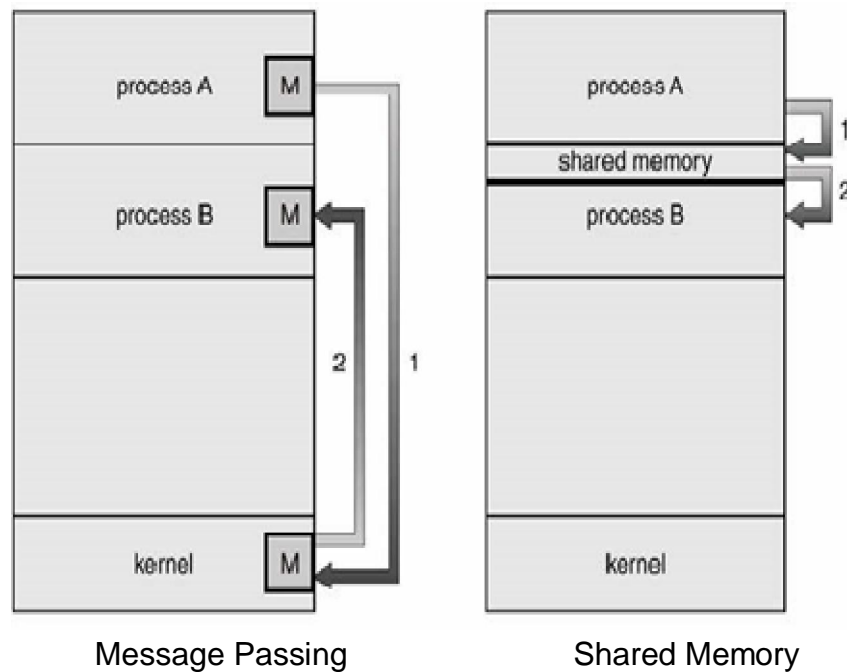
- Mengambil atau mengeset waktu atau tanggal
- Mengambil atau mengeset sistem data
- Mengambil atau mengeset proses, file atau atribut-atribut device

Komunikasi

- Membuat dan menghapus sambungan komunikasi
- Mengirim dan menerima pesan
- Mentransfer status informasi

Komunikasi(2)

- Komunikasi dilakukan dengan melewati pesan atau sharing memori



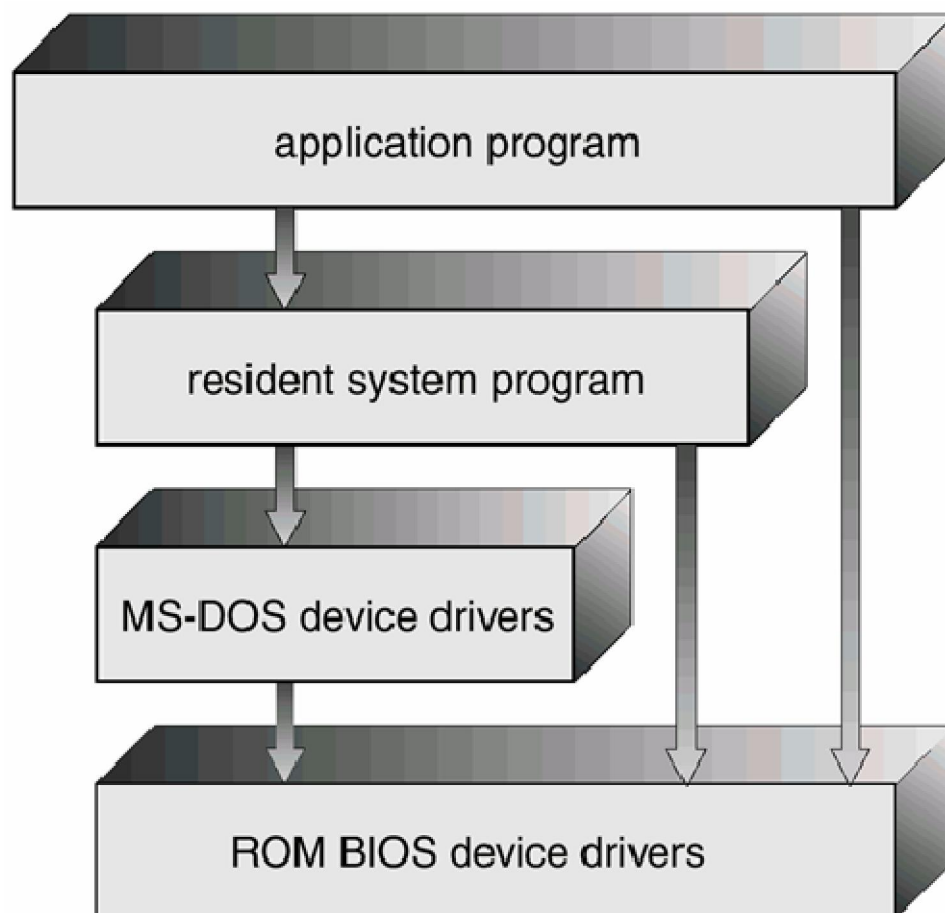
Program Sistem

- Program sistem menyediakan kemudahan pembangunan program dan eksekusi.
 - Manipulasi File
 - Informasi status
 - Modifikasi File
 - Dukungan bahasa pemrograman
 - Loading dan eksekusi program
 - Komunikasi
 - Aplikasi program
- Kebanyakan user memandang sistem operasi sebagai program sistem, bukan sebagai “actual system calls”.

Struktur Sistem Operasi

- Struktur Sistem Operasi
 - Metode untuk mengorganisasi dan membangun sistem operasi
 - Contoh: MS-DOS
 - Saat dirancang kemampuan PC sangat minimal
- § Prosesor: 8086 (10 MHz), Max. memory: 640 Kb
- MS-DOS – dibuat dengan menyediakan “fungsional” dari OS sebanyak mungkin pada resources yang sangat terbatas (memori)
 - Tidak dalam bentuk modul => monolithic (satu kesatuan):
 - § MS-DOS menjadi satu kesatuan besar tanpa batasan jelas– fungsional dan interface
 - § Terdapat struktur yang sangat sederhana dan “proteksi” yang longgar (single user system)

Struktur Layer MS-DOS



Struktur Monolithic

- Tidak terstruktur secara jelas
 - Kumpulan program yang menggunakan langsung resources hardware
 - Terdapat program-program yang mewakili fungsi OS: processor management, memory management
 - § OS awal: satu kesatuan proses, dimana kontrol berpindah dari program-program tersebut (“procedure calls”)
 - § Program user (proses): menjadi satu bagian rutin dari (loop) program utama jika tidak melakukan fungsi OS
 - § User program dijalankan “call” dari OS => eksekusi pada user mode akan berhenti:
 - ü timeout (timer interrupt)
 - ü kembali ke OS (service)
 - ü Interrupt (hardware)

Pendekatan Sederhana (Kernel)

- Struktur terbatas pada dua layer
 - Systems programs: bagian OS yang dibangun diatas kernel –extended machine
 - Kernel
 - § Operasi vital yang penting dan melindungi resources hardware
 - § Semua service untuk user proses melalui mekanisme system call
 - § Tugas utama kernel menyediakan fasilitas untuk: multiprogramming / multitasking –dimana proses-proses dapat berjalan serentak (concurrent) dan terpisah
 - § UNIX (1978)
 - § Menggunakan pendekatan rancangan sederhana dengan dukungan H/W yang terbatas (PDP-11)

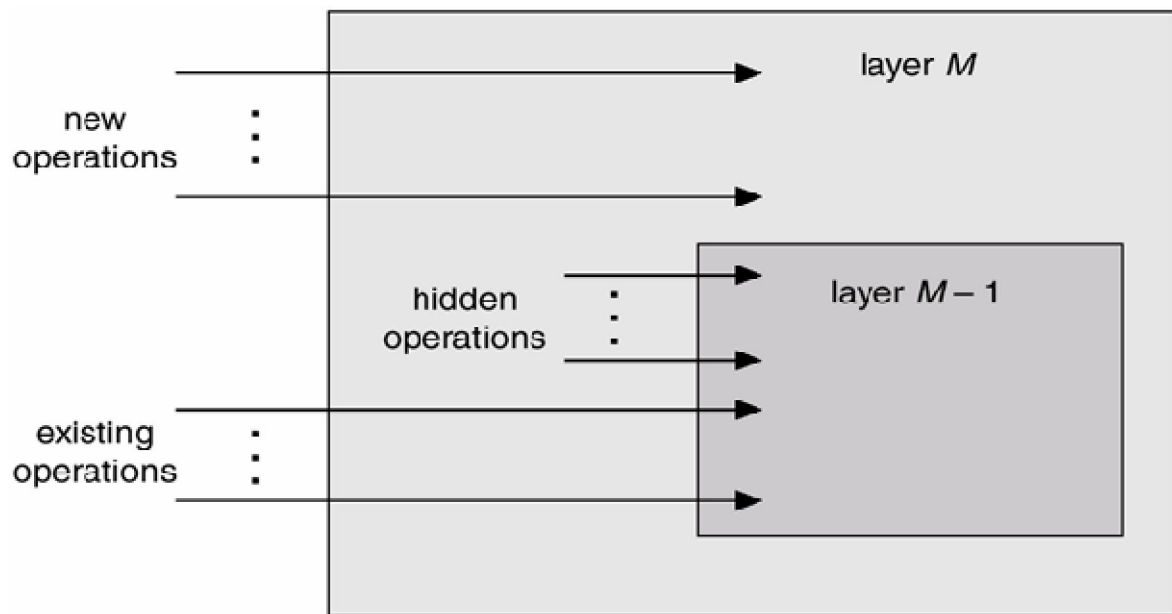
Struktur Sistem UNIX

(the users)		
shells and commands compilers and interpreters system libraries		
<i>system-call interface to the kernel</i>		
signals terminal handling character I/O system terminal drivers	file system swapping block I/O system disk and tape drivers	CPU scheduling page replacement demand paging virtual memory
<i>kernel interface to the hardware</i>		
terminal controllers terminals	device controllers disks and tapes	memory controllers physical memory

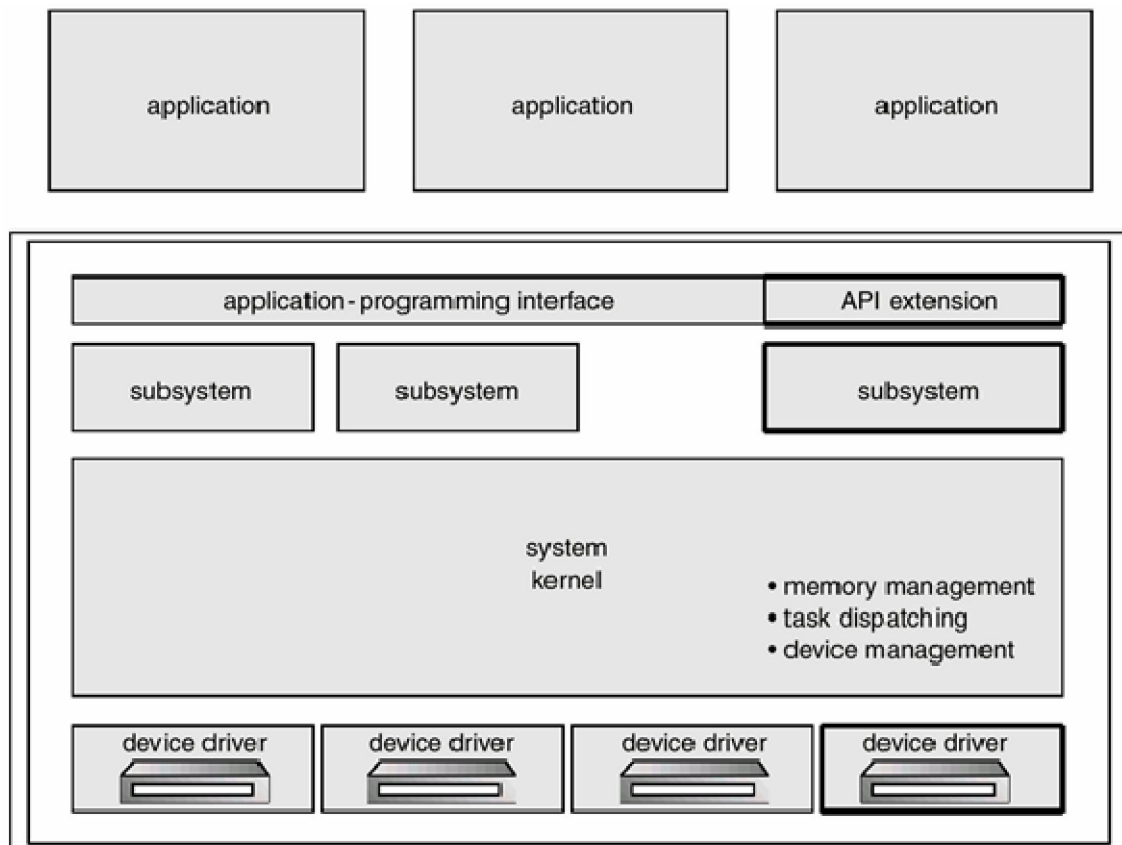
Pendekatan Berlapis (layer approach)

- Susunan berlapis:
 - OS dibagi atas sejumlah (lebih dari 2) layer
 - Setiap layer melingkupi layer dibawahnya (kendali, akses)
 - Layer paling bawah (0) => hardware
 - Layer paling atas (N) => user interface
- Rancangan moduler:
 - Layer disusun sehingga setiap fungsi / operasi layer atas akan menggunakan “services” pada layer bawah.

Sistem Operasi Berlapis



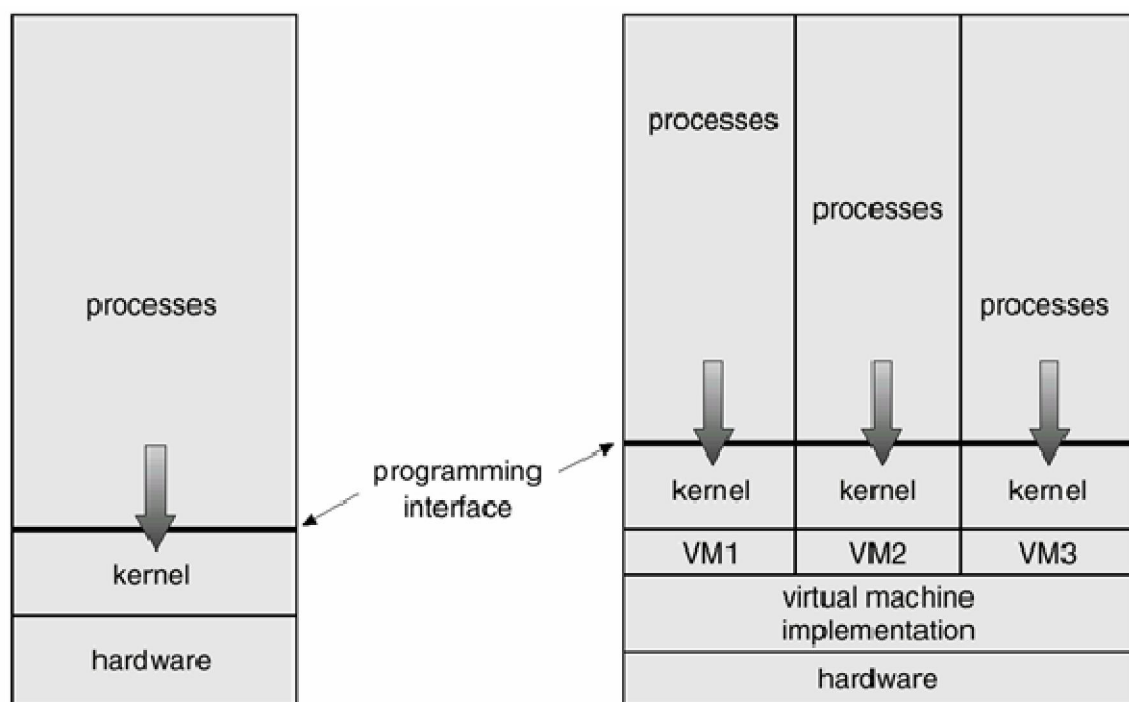
S



Virtual Machines

- Misalkan terdapat system program => control program yang mengatur pemakaian resources hardware.
- Control program = trap system call + hardware acces.
- Control program memberikan fasilitas keproses user
 - Mendapatkan jatah CPU dan memori.
 - Menyediakan interface “identik” dengan apa yang disediakan oleh hardware
=> sharing devices untuk berbagai proses.
- Virtual machine => control program yang minimal
 - VM memberikan ilusimultitasking: seolah-olah terdapat prosesor dan memori eksklusif digunakan (virtual machine).
 - VM memilah fungsi multitasking dan implementasi extended machine (tergantung user proses) => flexible dan lebih mudah untuk maintained (proteksi).

Model Sistem VM



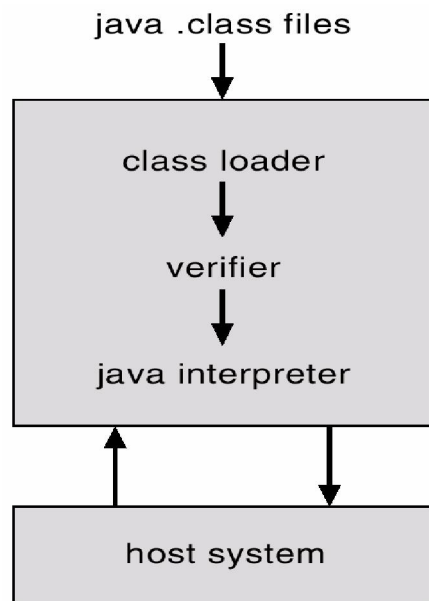
Virtual Machines (Cont.)

- Jika setiap user diberikan satu virtual machine => user bebas untuk menjalankan OS (kernel) yang diinginkan pada VM tersebut.
 - Potensi lebih dari satu OS dalam satu komputer.
 - Contoh :
 - ü IBM VM370: menyediakan VM untuk berbagai OS: CMS (interaktif), MVS, CICS, dll.
 - Problem :
 - ü Sharing disk => setiap OS mempunyai file system yang mungkin berbeda.
 - ü IBM : virtual disk (minidisk) yang dialokasikan untuk user melalui VM.

JAVA VIRTUAL MACHINE

- Program Java dikompilasi pada platform-neutral bytecodes yang dieksekusi oleh Java Virtual Machine (JVM)
- JVM terdiri dari :
 - class loader
 - class verifier
 - runtime interpreter
- Kompiler Just-In-Time (JIT) meningkatkan kinerja

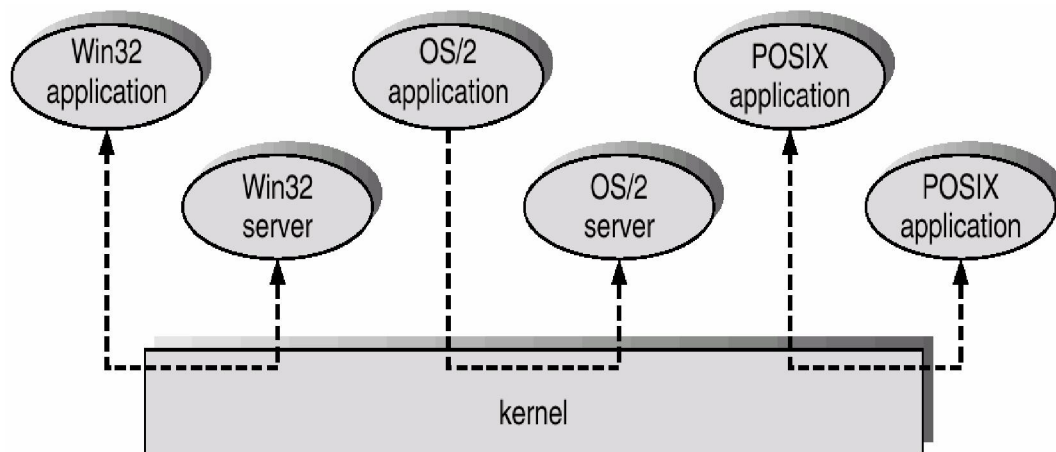
JAVA VIRTUAL MACHINE



MODEL CLIENT-SERVER

- Implementasi fungsi OS (extended machine) dapat menjadi bagian dari user proses (client).
 - Request service ke server proses (kernel).
 - Kernel : Terdiri dari server (file, memory, I/O dll) yang melayani request dari client proses.
 - Akses ke hardware harus melalui server proses dari kernel yang mengontrol hardware tersebut.
- Proses : Partisi dan struktur yang sangat jelas (interface dan fungsional).
- Konsekuensi : Lambat (komunikasi antar client dan server), tidak efisien dalam menggunakan resources.

WINDOWS NT CLIENT-SERVER STRUCTURE



PERANCANGAN SISTEM

Tujuan Perancangan Sistem :

- Tujuan User : Sistem operasi mudah digunakan, mudah dipelajari, handal, aman dan cepat.
- Tujuan Sistem : Sistem operasi mudah didisain, diimplementasikan dan dikelola, memiliki fleksibilitas, handal, bebas kesalahan dan efisien.

IMPLEMENTASI SISTEM

- Secara tradisional ditulis dalam bahasa assembly, saat ini sistem operasi dapat ditulis dalam bahasa tingkat tinggi.
- Kode yang ditulis dalam bahasa tingkat tinggi :
- Code written in a high-level language :
 - Dapat ditulis secara cepat.
 - Lebih compact.
 - Lebih mudah dipahami dan debug.
- Sistem operasi jauh lebih mudah untuk di port (dipindahkan ke hardware lain) jika ditulis dalam bahasa tingkat tinggi).

SYSTEM GENERATION (SYSGEN)

- Sistem operasi didisain untuk dijalankan pada berbagai kelas mesin. Sistem harus dikonfigurasi untuk setiap komputer secara spesifik.
- Program SYSGEN memiliki informasi dalam mengkonfigurasi sistem hardware secara spesifik.
- Booting –awal komputer diaktifkan dengan melakukan loading kernel.
- Bootstrap program – kode yang disimpan di ROM yang dapat ditempatkan pada kernel,

C. SOAL LATIHAN/TUGAS

1. Jelaskan komponen sistem?
2. Jelaskan layanan sistem operasi?
3. System calls?

D. DAFTAR PUSTAKA

Buku

Bambang Hariyanto. 1997. Sistem Operasi, Bandung:Informatika Bandung.

Dali S. Naga. 1992. Teori dan Soal Sistem Operasi Komputer,Jakarta: Gunadarma.

Silberschatz Galvin. 1995. 4 Edition Operating System Concepts: Addison Wesley.

Sri Kusumadewi. 2000. Sistem Operasi. Yogyakarta: J&J Learning.

Tanenbaum, A.1992. Modern Operating Systems.New York: Prentice Hall

Link and Sites:

<http://www.ilmukomputer.com>
<http://vlsm.bebas.org>
<http://www.wikipedia.com>