

MODUL SISTEM OPERASI



Struktur Sistem Operasi



STRUKTUR SISTEM OPERASI

Komponen Sistem

Struktur Sistem

Layanan Sistem Operasi

Virtual Machines

System Calls

System Design & Implementation

Program Sistem

System Generation



KOMPONEN SISTEM

Manajemen Proses

Manajemen Memori Utama

Manajemen File

Manajemen I/O Sistem

Manajemen Secondary Storage

Command-Interpreter System

Jaringan

Sistem Proteksi





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**





MANAJEMEN PROSES

- Sistem operasi bertanggung jawab terhadap aktifitas yang berhubungan dengan manajemen proses:
 - **Process creation & deletion.**
 - **Process suspension (block) & resumption.**
 - **Mekanisme:**
 - Sinkronisasi antar proses
 - Komunikasi antar proses
 - Penanganan Deadlock





MANAJEMEN MEMORY UTAMA

- Memori sebagai tempat penyimpanan instruksi/data dari program
 - Storage yang cepat sehingga 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)





MANAJEMEN MEMORY UTAMA

- Umumnya main memory bersifat “volatile” – tidak permanen. Isinya akan hilang jika komputer dimatikan.
- Manajemen memori:
 - Melacak pemakaian memori (siapa dan berapa besar?).
 - Memilih program mana yang akan di-load ke memori.
 - Alokasi dan De-alokasi memori fisik untuk program.





MANAJEMEN FILE

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





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 data dari/ke I/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 SECONDARY STORAGE

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





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.





JARINGAN

- ❑ Dukungan terhadap komunikasi data antar perangkat komputer.
- ❑ Sekumpulan prosesor yang tidak berbagi memori atau *clock* diatur oleh sistem operasi komputer host untuk penggunaan prosesor dan alokasi tempat penyimpanan serta mekanisme pendistribusian data maupun proses yang dilakukan.



- Sistem terdistribusi menyediakan akses penggunaan sumber daya sistem, sehingga menyebabkan :
 - ✓ Proses komputasi semakin cepat
 - ✓ Peningkatan ketersediaan data
 - ✓ Peningkatan kemampuan



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 belum.
- ✓ Menetapkan sistem pengaturan yang digunakan.
- ✓ Menyiapkan alat yang digunakan.





LAYANAN SISTEM OPERASI

1. Pembuatan Program

- ✓ Sistem operasi menyediakan fasilitas dan layanan untuk membantu para pemrogram untuk menulis program

2. Eksekusi Program

- ✓ Kemampuan sistem untuk me'load' program ke memori dan menjalankan program yang dikehendaki user maupun sistem



3. Operasi I/O

- ✓ Sistem operasi menyediakan mekanisme untuk melakukan operasi I/O atas nama pengguna.

4. Sistem Manipulasi Berkas

- ✓ Kemampuan program untuk operasi pada berkas (membaca, menulis, membuat, dan menghapus berkas yang berupa file atau direktori)

5. Komunikasi

- ✓ Pertukaran data/informasi antar dua atau lebih proses yang berada pada satu komputer (atau lebih).

6. Deteksi Error

- ✓ Kegiatan untuk menjaga kestabilan sistem dengan mendeteksi error, perangkat keras maupun operasi yang dilakukan.

7. Deteksi dan Pemberian Tanggapan

- ✓ Memberikan tanggapan yang menjelaskan kesalahan yang terjadi serta dampaknya terhadap aplikasi yang sedang berjalan.

8. Accounting

- ✓ Kegiatan merekam aktivitas pengguna, report pemakaian sumber daya, mampu mengumpulkan data statistik penggunaan beragam sumber daya dan memonitor parameter kinerja

9. Efisiensi Penggunaan Sistem

- ✓ **Resource Allocator** : mengalokasikan sumber daya hardware maupun software ke beberapa pengguna atau mengalokasikan job yang berjalan pada saat bersamaan ke beberapa komputer dalam jaringan.
- ✓ **Proteksi Sistem** untuk menjamin akses ke sistem sumber daya yang aman, dikendalikan oleh

sistem sehingga pengguna dikontrol aksesnya ke sistem.



SYSTEM CALLS

- ❑ 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.



- ❑ 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 sistim 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.



JENIS SISTEM CALLS

Manajemen Proses

Manajemen Berkas

Manajemen Piranti

Informasi/Pemeliharaan

Komunikasi





MANAJEMEN 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





MANAJEMEN BERKAS

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





MANAJEMEN PIRANTI

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





INFORMASI / PEMELIHARAAN

- ✓ 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 dilakukan dengan melewatkan 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)



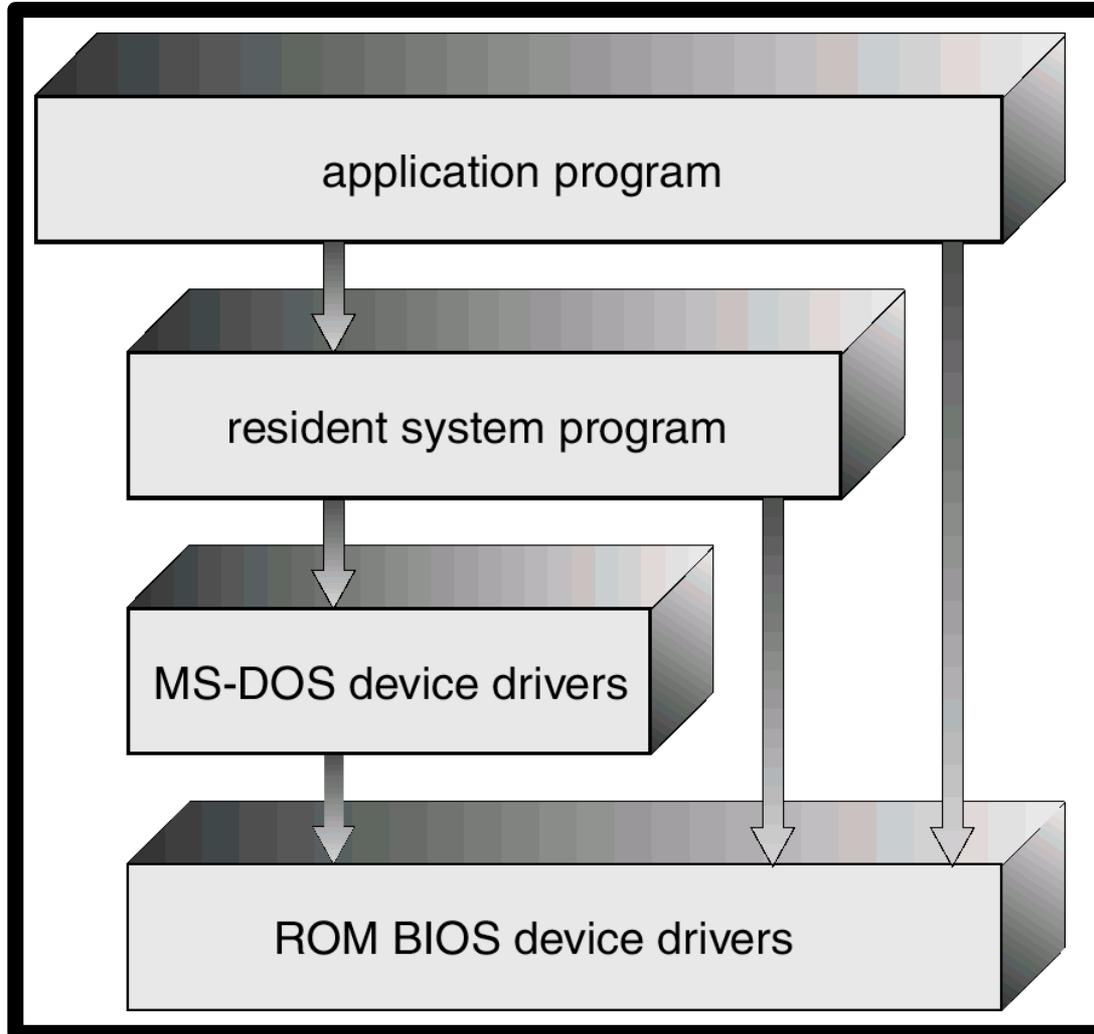
➤ Tidak dalam bentuk modul =>

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 SISTEM OPERASI

Sistem Monolithic

Sistem Berlapis

Sistem Mesin Virtual

Sistem Client-Server

Sistem Berorientasi Obyek





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:
 - timeout (timer interrupt)
 - kembali ke OS (service)
 - Interrupt (hardware)

❑ Struktur terbatas pada dua layer

- Systems programs: bagian OS yang dibangun di atas kernel – extended machine
- Kernel
 - ✓ Operasi vital yang penting dan melindungi resources hardware
 - ✓ Semua service untuk user proses melalui mekanisme system call

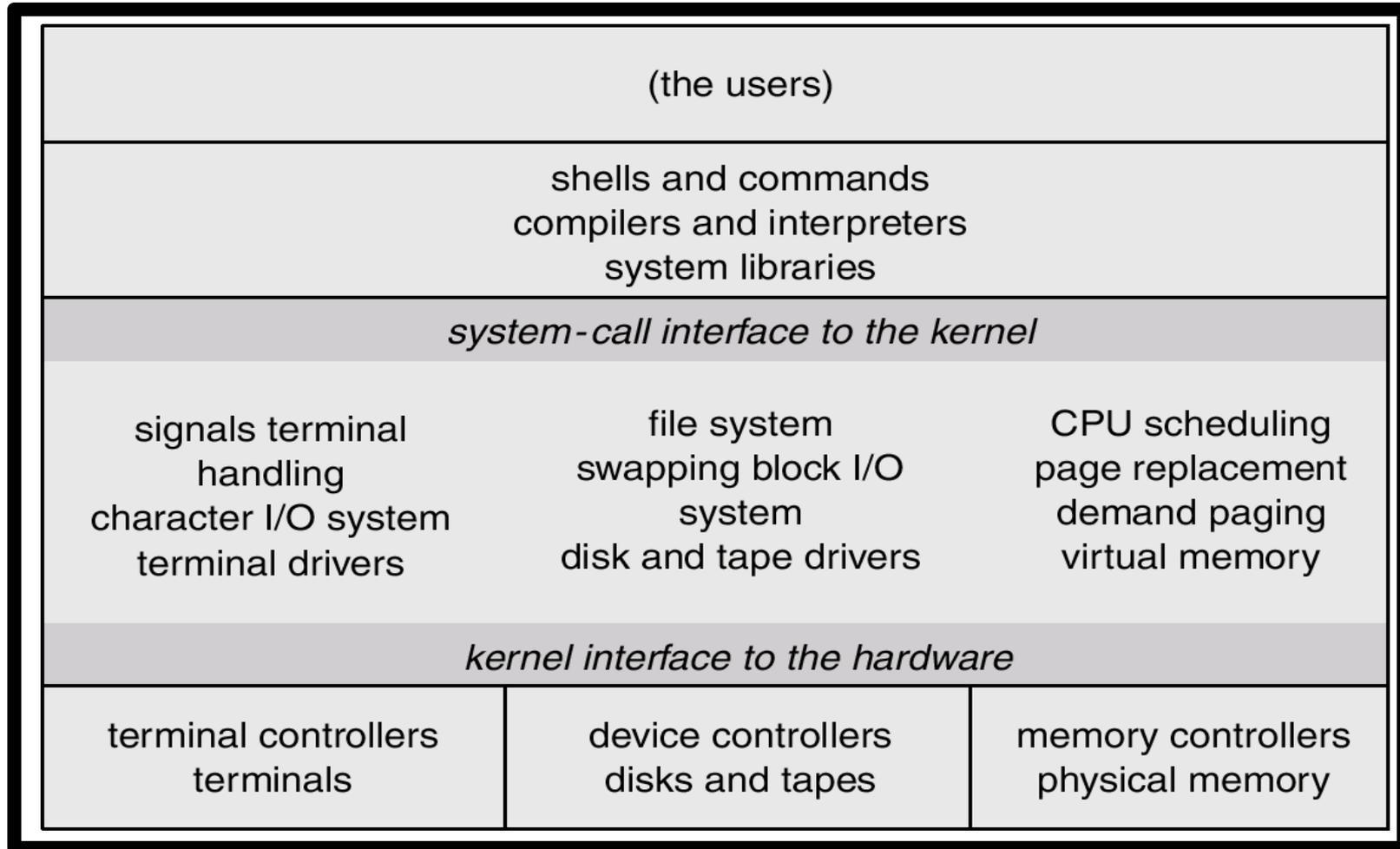
✓ Tugas utama kernel menyediakan fasilitas untuk: multiprogramming/multitasking – dimana proses-proses dapat berjalan serentak (concurrent) dan terpisah

✓ UNIX (1978)

Menggunakan pendekatan rancangan sederhana dengan dukungan H/W yang terbatas (PDP-11)



STRUKTUR SYSTEM UNIX

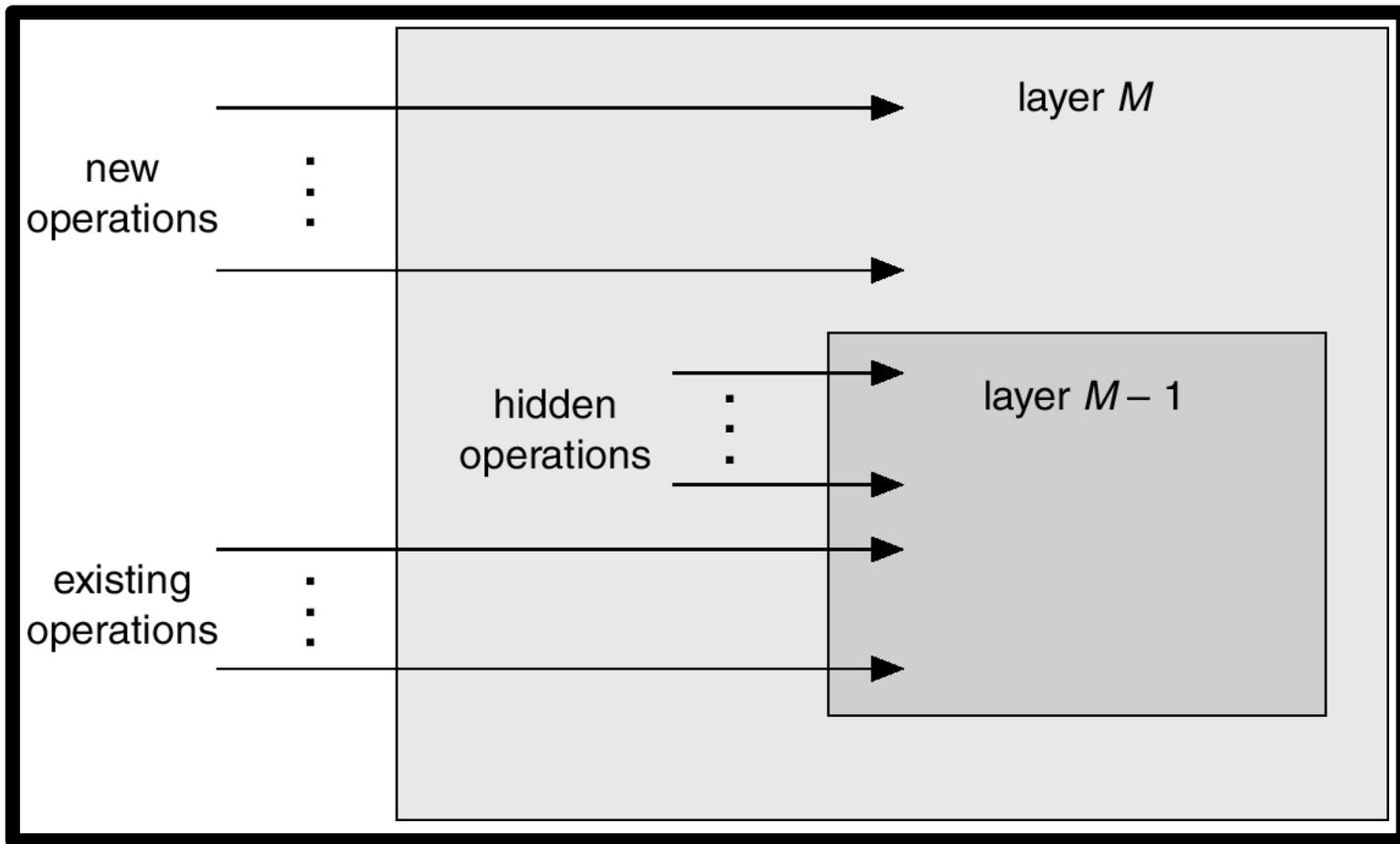




SISTEM BERLAPIS

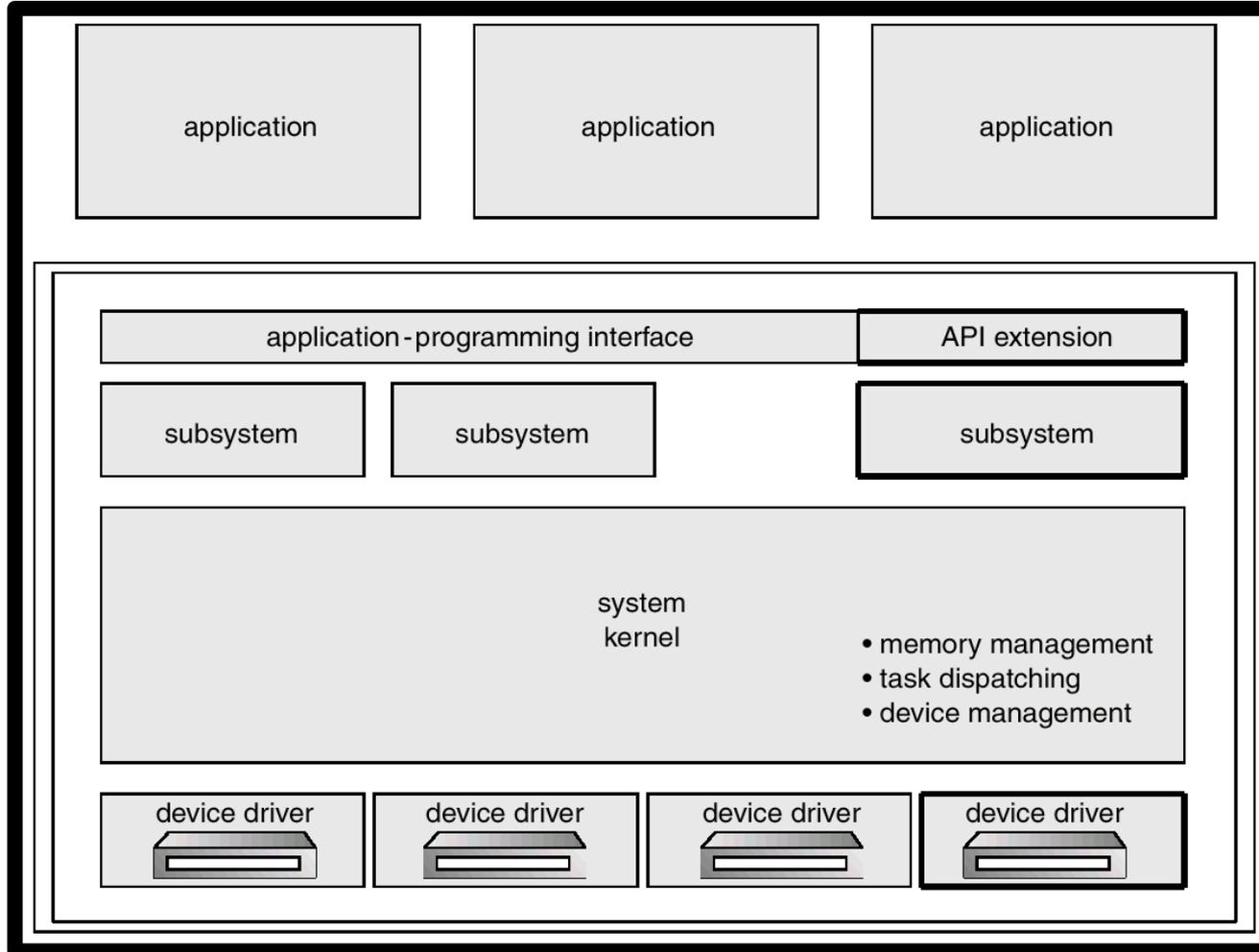
- OS dibagi atas sejumlah (lebih dari 2) layer
- Setiap layer melingkupi layer di bawahnya
(kendali, akses)
- Layer paling bawah (0) => hardware
- Layer paling atas (N) => user interface
- Layer disusun sehingga setiap fungsi/operasi layer atas akan menggunakan “services” pada layer bawah.







SISTEM BERLAPIS OS/2



❑ Contoh : Sistem Operasi THE (Technique Hoogesschool Endicche), Djikstra, 1968

| Lapisan | Fungsi |
|---------|---|
| 5 | The Operator |
| 4 | User Programs |
| 3 | I/O Management |
| 2 | Operator Proccess Communication |
| 1 | Memory & Drum Management |
| 0 | Processor Allocation & Multiprogramming |



SISTEM MESIN VIRTUAL

- ❑ Misalkan terdapat system program => control program yang mengatur pemakaian resources hardware.
- ❑ Control program = trap system call + hardware acces.
- ❑ Control program memberikan fasilitas ke proses user :
 - Mendapatkan jatah CPU dan memori.
 - Menyediakan interface “identik” dengan apa yang disediakan oleh hardware => sharing devices untuk berbagai proses.

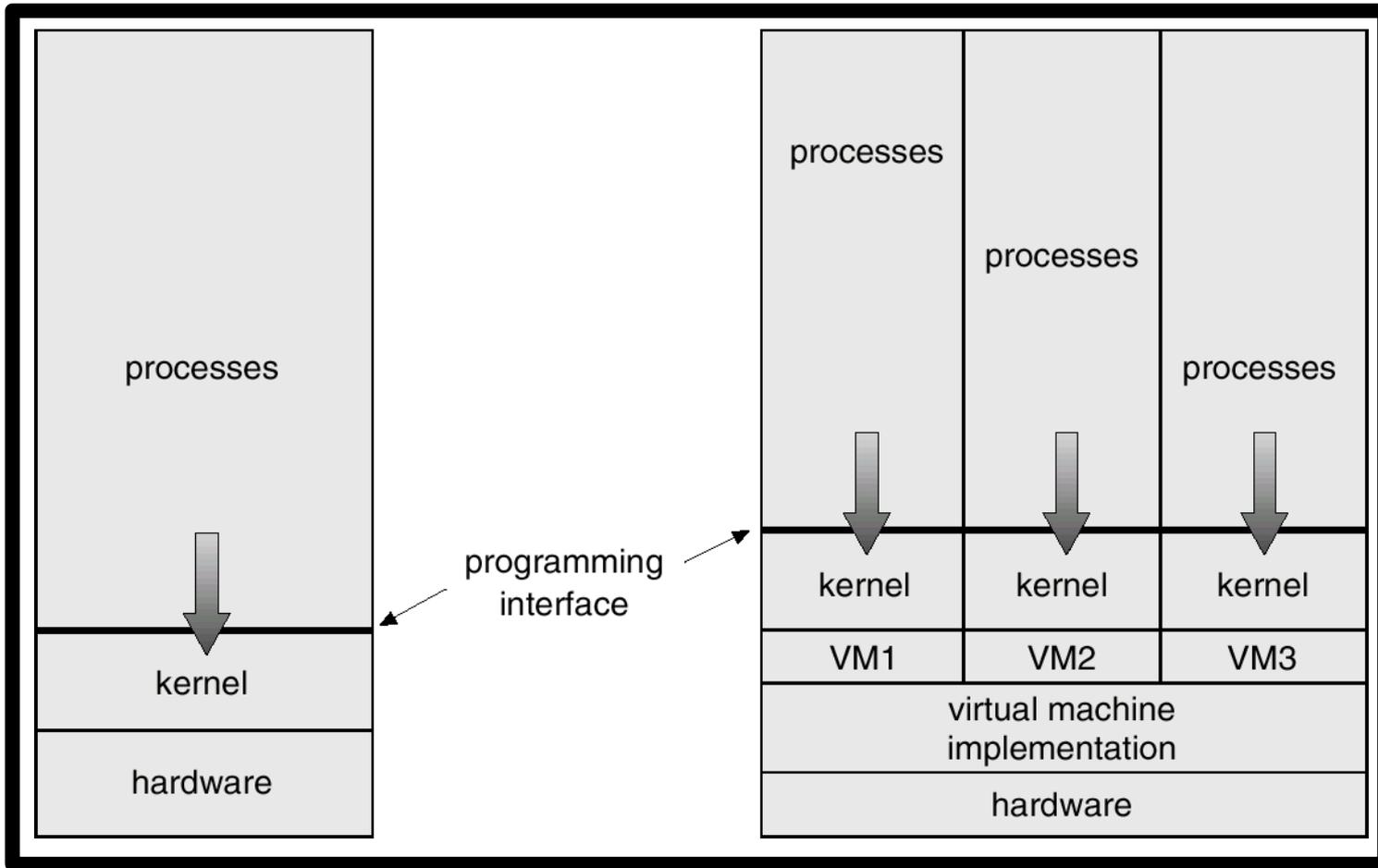


❑ Virtual machine => control program yang minimal

- ✓ VM memberikan ilusi multitasking: 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

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:
IBM VM370: menyediakan VM untuk berbagai OS: CMS (interaktif), MVS, CICS, dll.

- ✓ Problem:
Sharing disk => setiap OS mempunyai file system yang mungkin berbeda.
IBM: virtual disk (minidisk) yang dialokasikan untuk user melalui VM.

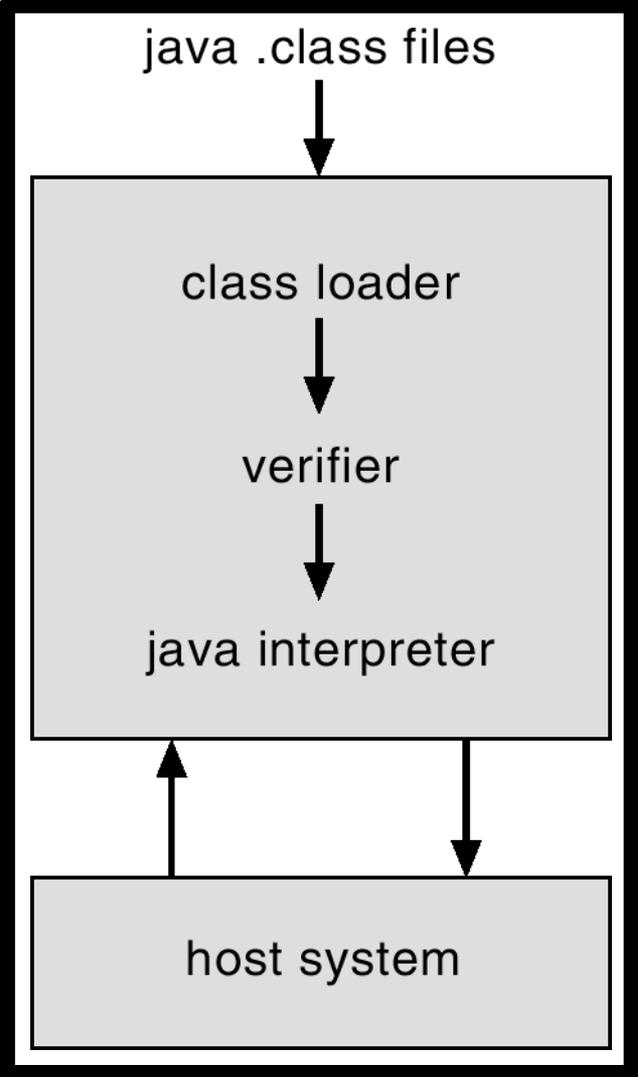




JAVA VIRTUAL MACHINES

- ❑ 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







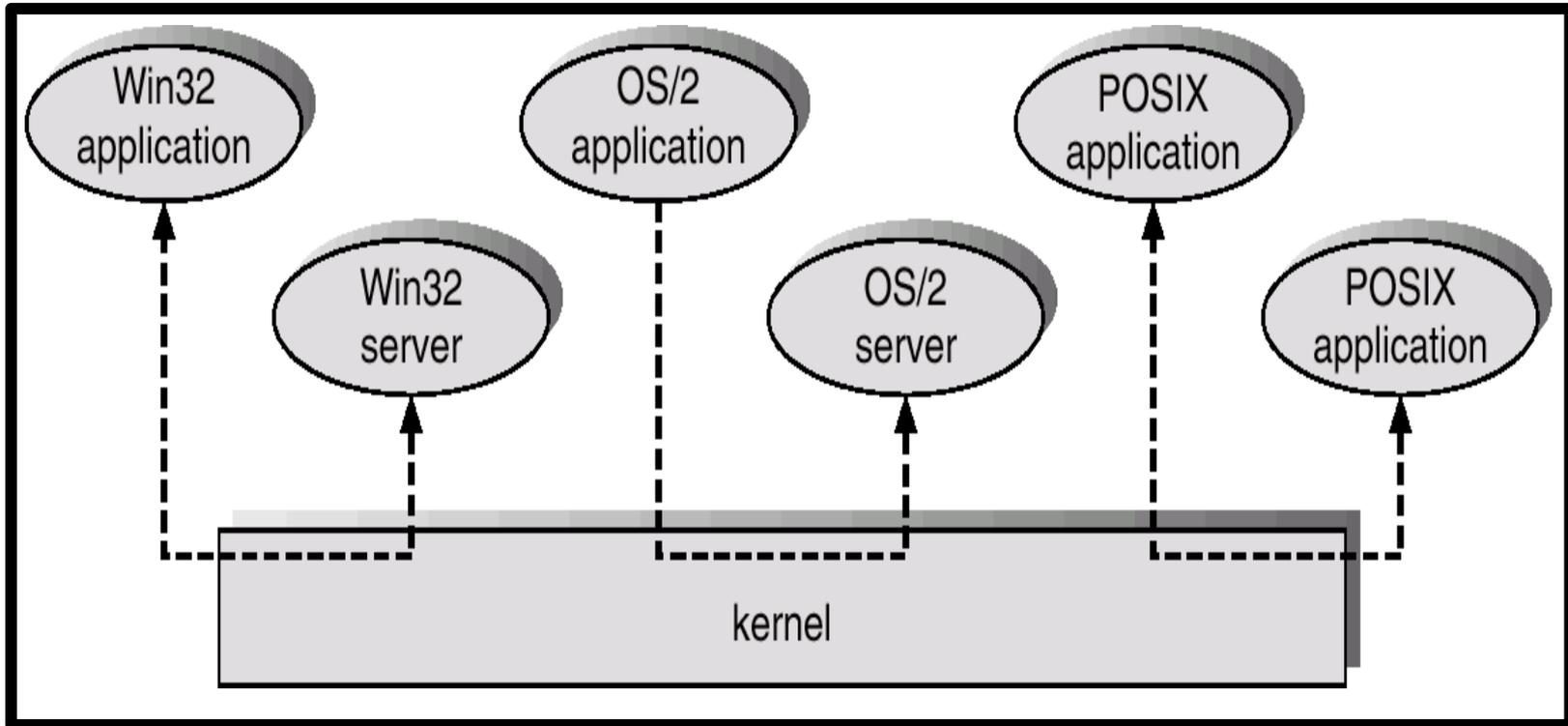
SISTEM CLIENT-SERVER

- ❑ 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.





STRUKTUR WINDOWS NT CLIENT-SERVER





SISTEM BERORIENTASI OBYEK

- Layanan diimplementasikan sebagai obyek-obyek.
- Model ini terstruktur dan memisahkan antara layanan dan implementasinya.
- Contoh :
- Sistem Operasi X-Kernel
- Sistem Operasi MS Windows NT sudah mengadopsi beberapa teknologi berorientasi obyek, tetapi belum secara keseluruhan.





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

- ✓ 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





SYSTEM GENERATION

- ✓ *Booting* –awal komputer diaktifkan dengan melakukan loading kernel.
- ✓ *Bootstrap program* – kode yang disimpan di ROM yang dapat ditempatkan pada kernel, di load ke memori dan memulai eksekusi.





KUIS

1. Jelaskan perbedaan sistem operasi berbasis Command Line Interface (CLI) dan Graphic User Interface (GUI)?
2. Sebutkan Keutamaan dari CLI dan GUI?
3. Jelaskan pemahaman anda apa yang dimaksud dari Sistem Operasi Berbasis Open Source?

