

## 2. STRUKTUR SISTEM OPERASI

# TUJUAN PEMBELAJARAN

- Menjelaskan komponen pada sistem operasi
- Menjelaskan layanan sistem operasi terhadap user, proses, dan sistem lainnya.
- Membahas berbagai cara penataan sebuah sistem operasi.
- Menjelaskan bagaimana sistem operasi diinstal dan dikustomisasi dan bagaimana proses bootingnya.

# Komponen Dasar Sistem Komputer

- **Hardware / Peripheral**

- Penyedia sumber daya untuk komputasi dasar

- Memory, CPU, I/O

- **Software**

- Operating system

- Mengontrol dan mengkoordinasi penggunaan hardware antara berbagai aplikasi dan pengguna

- Application programs – mendefinisikan cara penggunaan sumber daya sistem dan persoalan komputasi yang dialami pengguna

- Word processors, compilers, web browsers, database systems, video games

- **Brainware**

- orang, mesin, dan komputer lain

# Menurut Karakteristiknya

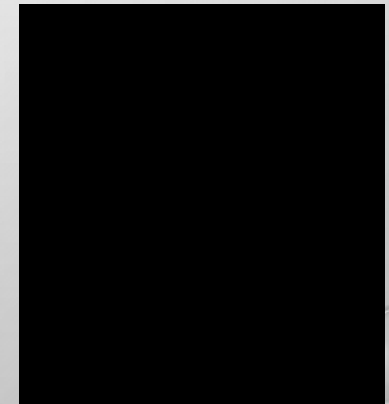
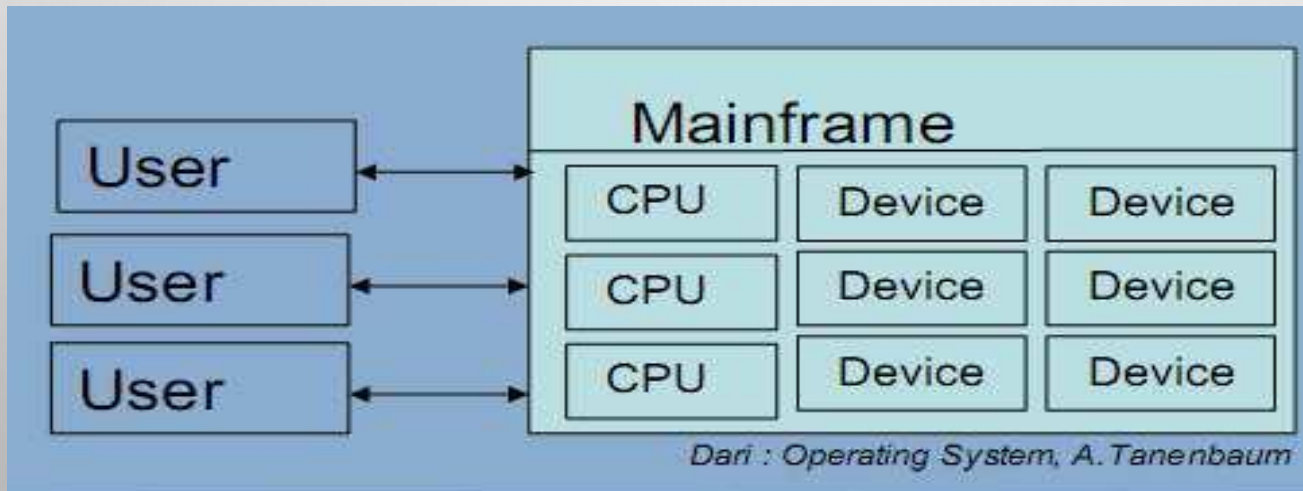
- Single processor
  - Komputer yang hanya memiliki satu prosesor
- Multiprocessor
  - Komputer yang memiliki lebih dari satu prosesor
- Personal Computer
  - Komputer yang digunakan oleh hanya satu orang dalam satu waktu (umumnya)
- Distributed System
  - Komputasi yang dikerjakan dengan beberapa prosesor

# Menurut karakteristiknya

- Clustered System
  - Gabungan dari beberapa sistem individu yang saling berbagi tempat penyimpanan data (**storage / SAN**) dan saling terhubung dalam jaringan **local**
- Real Time System
  - Sebuah sistem yang mengutamakan **ketepatan waktu** dalam eksekusi satu buah tugas

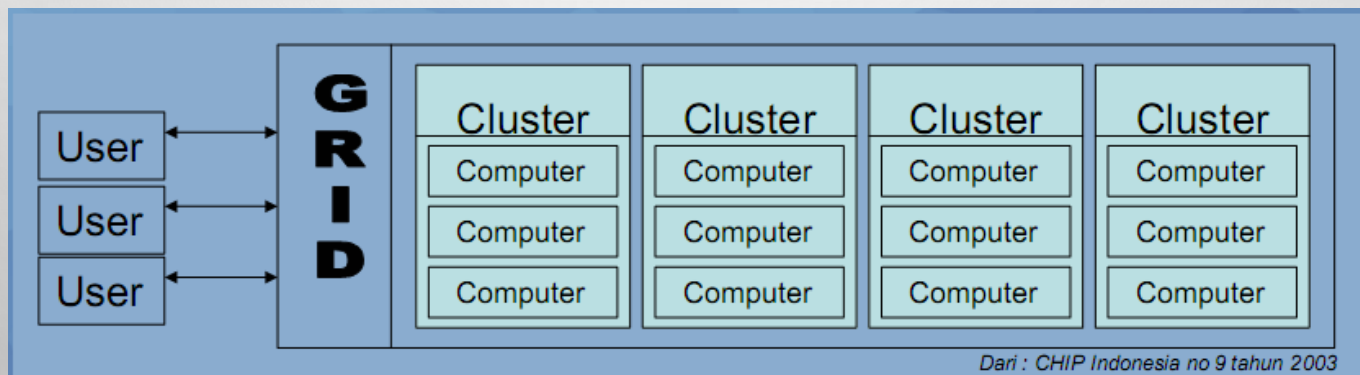
# Mainframe System

- Kumpulan dari device-device & CPU yang berfungsi sama atau berbeda yang disatukan dalam sebuah sistem yang saling berbagi (sharing)
- Mengumpulkan job-job yang **mirip**
- Secara **otomatis** berpindah dari satu job ke job yang lain



# Grid Computer/Super Computer

- Kumpulan dari banyak komputer yang dikembangkan dalam sebuah *computer-farm*
- Pengembangan dari Mainframe dan Desktop
- Menggunakan **banyak** CPU untuk menghasilkan output maksimal
- Kumpulan **cluster** system, bisa tidak dalam 1 lokasi



# Tipe alokasi job

- **Batch System**

- Job yang mempunyai **kesamaan** kebutuhan sumber daya dikumpulkan – Ketika komputer dalam keadaan siap maka CPU menjalankan masing-masing kumpulan job tersebut sebagai sebuah kelompok.

- Dieksekusi secara **bergantian**

- User yang mengoperasikan

- Tidak real-time



## **Multiprogramming:**

- Job/process disimpan di **main memory** pada waktu yang sama
- CPU dipergunakan bergantian oleh job-job/process tersebut
- Dikerjakan oleh OS
- Berupa *background proses*
- Lebih cepat drpd Batch

## **• Dibutuhkan Kemampuan OS:**

- Penyediaan I/O routine
- Pengaturan memory, untuk mengalokasikan memory untuk masing2 Job
- Penjadwalan CPU, untuk memilih job mana yang akan dijalankan
- Pengalokasian untuk hardware lain

## Time-Sharing System/Multitasking - Interactive Computing

- CPU digunakan bergantian oleh job-job **di memori** dan **di disk**
- Waktu dibatasi
- Response time harus  $< 1$  detik
- CPU dialokasikan hanya pada job yg ada di memory = **CPU scheduling**
- Job dipindahkan dari dan ke disk (konsep **swapping** dan **virtual memory**)
- Terjadi komunikasi antara user dan sistem operasi
- Setelah sistem operasi menyelesaikan satu perintah ia menunggu perintah berikutnya dari user

# Berdasarkan Hubungan Sistem

- Single Process
  - Sistem hanya memiliki satu CPU
- Multi Process:
  - Sistem memiliki lebih dari satu CPU untuk memproses satu atau lebih program
  - Resource digunakan bersama-sama
  - Sering disebut **Tightly Coupling System**

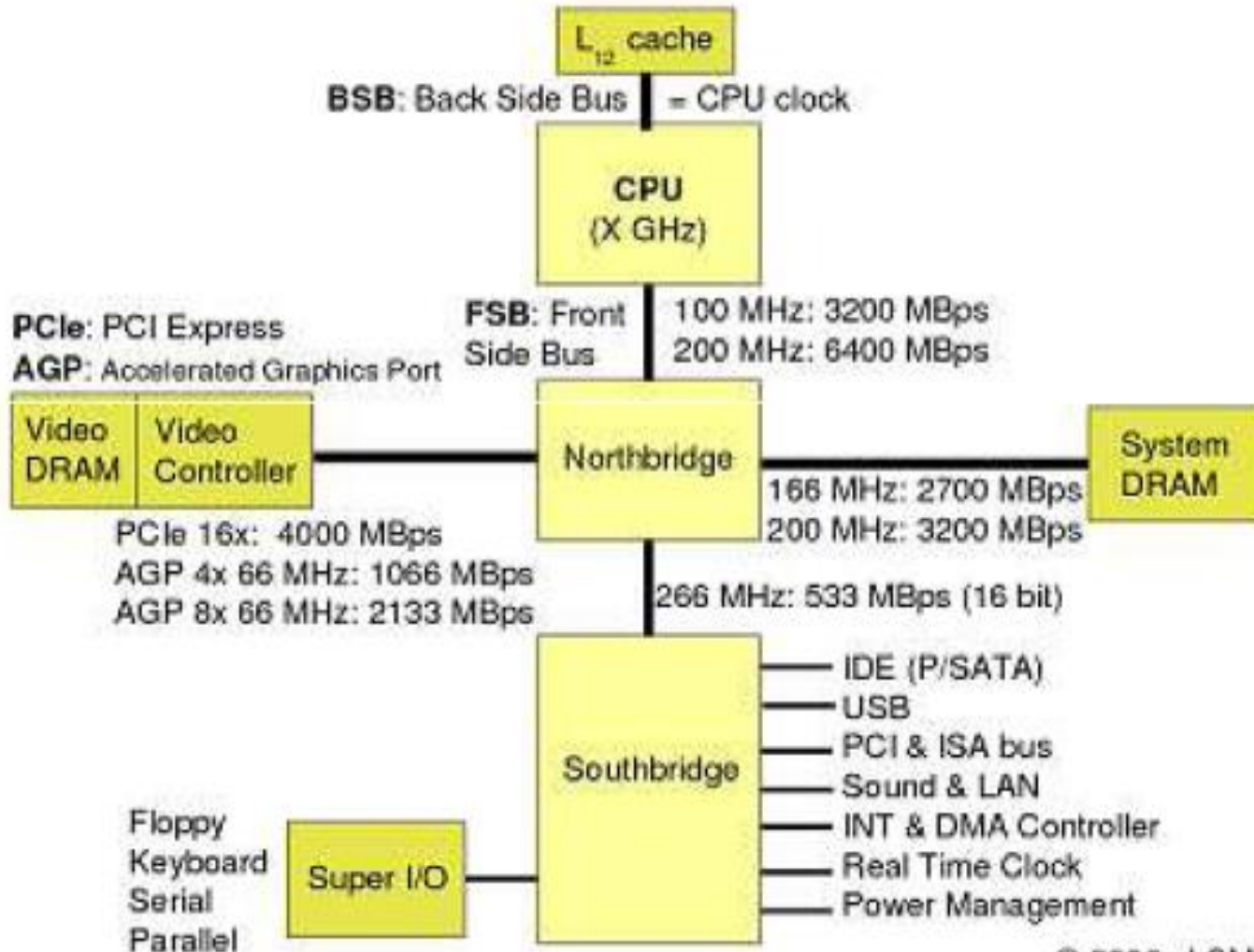
# • Distributed System

- Melaksanakan komputasi secara terdistribusi diantara beberapa prosesor, tidak digunakan bersamaan.
- **Loosely coupling system** – setiap prosesor mempunyai local memory / resource.
  - Komunikasi terjadi melalui bus atau jalur telepon / LAN
- Keunggulan:
  - Pembagian sumber daya
  - Komputasi lebih cepat
- Model
  - Client-Server Systems
  - Peer-to-peer (P2P) System

# Peer to Peer vs Client Server

- P2P **tidak** membedakan client dan server
  - Semua node yang terhubung disebut dengan **peer**
  - Satu node bersifat sebagai client dan sekaligus sebagai server
  - Node harus terhubung dalam jaringan P2P
- Menggunakan service melalui **discovery protocol**

# PC Modern Architecture



# Komponen Sistem

- Manajemen proses
- Manajemen memori utama
- Manajemen file
- Manajemen sistem I/O
- Manajemen penyimpan sekunder
- Sistem jaringan (terdistribusi)
- Sistem proteksi
- Sistem *command interpreter*.

# Manajemen Proses

- Proses → Program yang sedang dieksekusi
- Tanggung jawab sistem operasi pada aktifitas-aktifitas manajemen proses:
  - pembuatan/penghapusan proses oleh user atau sistem
  - menghentikan proses sementara dan melanjutkannya
  - menyediakan mekanisme sinkronisasi dan komunikasi proses
- Proses → resource (waktu CPU, memori, file, I/O device)



# Manajemen Memori Utama

- Memori → array besar berukuran word atau byte yang mempunyai alamat tertentu
- Memori bersifat volatile (RAM)
- Tanggung jawab sistem operasi pada aktifitas-aktifitas manajemen memori:
  - Menjaga dan memelihara bagian memori yang sedang digunakan dan dari yang menggunakan
  - Memutuskan proses tertentu yang harus dipanggil ke memori
  - Mengalokasikan dan mendealokasikan ruang memori

# Manajemen File

- File → kumpulan informasi yang saling berhubungan (user)
- Tanggung jawab sistem operasi pada aktifitas-aktifitas manajemen file:
  - Membuat/ menghapus file
  - Membuat/ menghapus direktori
  - Pemetaan file ke memori sekunder
  - Backup file ke media penyimpanan yang stabil

# Manajemen I/O

- Tanggung jawab sistem operasi pada aktifitas-aktifitas manajemen I/O:
  - Sistem buffer-caching
  - Antarmuka device-driver secara umum
  - Driver untuk device hardware-hardware tertentu

# Manajemen Penyimpanan Sekunder

- Tanggung jawab sistem operasi pada aktifitas-aktifitas manajemen penyimpanan sekunder:
  - Pengaturan ruang bebas
  - Alokasi penyimpanan
  - Penjadwalan disk

# Sistem Jaringan (Terdistribusi)

- Tidak menggunakan memori atau clock bersama-sama
- Mengakses resource yang beragam
- Keuntungan:
  - Meningkatkan kecepatan komputasi
  - Meningkatkan ketersediaan data
  - Meningkatkan kehandalan sistem

# Sistem Proteksi

- Mekanisme untuk mengontrol akses oleh program, proses atau user pada sistem maupun resource dari user
- Mekanisme:
  - Membedakan antara penggunaan yang sah dan yang tidak sah
  - Menentukan kontrol yang terganggu
  - Menetapkan cara pelaksanaan terproteksi

# Sistem Command Interpreter

- Perintah yang dimasukkan ke sistem operasi menggunakan pernyataan kontrol digunakan untuk manajemen proses, penanganan I/O, manajemen penyimpan sekunder, manajemen memori utama, akses sistem file, proteksi, dan jaringan
- Shell → Command Line Interpreter (CLI) dan Graphical User Interface (GUI)

# Layanan Sistem Operasi

- Eksekusi program
- Operasi-operasi I/O
- Manipulasi sistem file
- Komunikasi
- Pendeteksi kesalahan



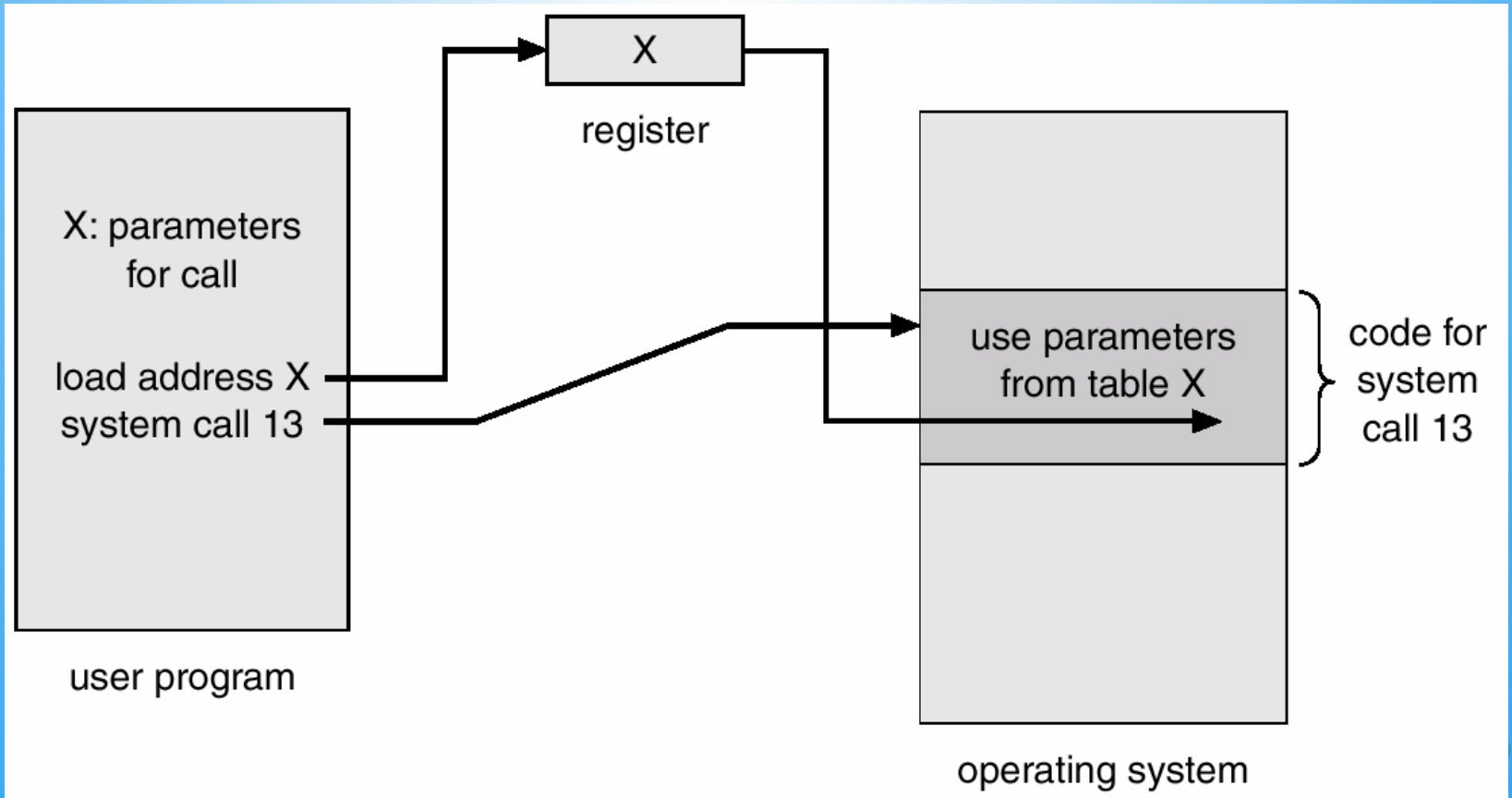
# Layanan Sistem Operasi (cont.)

- Tambahkan layanan sistem operasi:
  - Mengalokasikan resource
  - Accounting
  - Proteksi

# System Calls

- Menyediakan antar muka program yang sedang berjalan dengan sistem operasi
- Dulunya menggunakan bahasa assembly, sekarang bahasa tingkat tinggi (C atau C++)
- Diakses via Application Programming Interface (API)
- Metode untuk melewatkan parameter antara program yang sedang berjalan:
  - Melalui register
  - Menyimpan parameter pada tabel memori
  - Menyimpan parameter ke stack dan mengambil isi stack

# Parameter Passing via Tabel



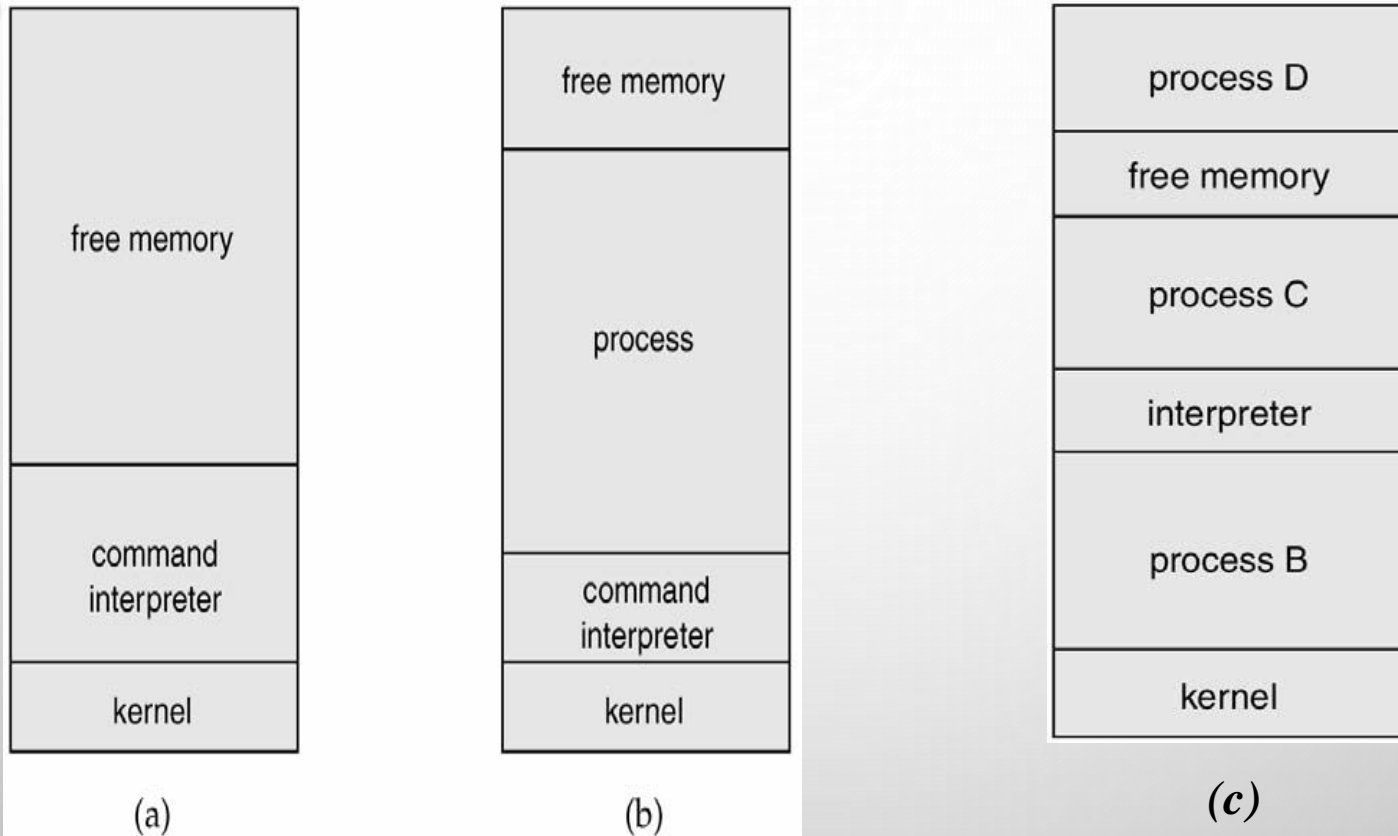
# Contoh System Call

	Windows	Unix
<b>Process Control</b>	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
<b>File Manipulation</b>	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
<b>Device Manipulation</b>	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
<b>Information Maintenance</b>	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
<b>Communication</b>	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()
<b>Protection</b>	SetFileSecurity() InitializeSecurityDescriptor() SetSecurityDescriptorGroup()	chmod() umask() chown()

# Jenis System Calls

- Kontrol Proses:
  - 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

# Jenis System Calls (cont.)



*Sistem MSDOS : (a) pada saat startup (b) pada saat running*  
*Sistem UNIX: (c) bisa lebih dari 1 proses*

# Jenis System Calls (cont.)

- Manajemen File:
  - create file, delete file
  - open, close file
  - read, write, reposition
  - get and set file attributes
- Manajemen Device:
  - request device, release device
  - read, write, reposition
  - get device attributes, set device attributes
  - logically attach or detach devices

# Jenis System Calls (cont.)

- Information Maintenance:
  - get time or date, set time or date
  - get system data, set system data
  - get and set process, file, or device attributes
- Komunikasi:
  - create, delete communication connection
  - send, receive messages
  - transfer status information
  - 2 model: message-passing dan shared-memory



# System Programs

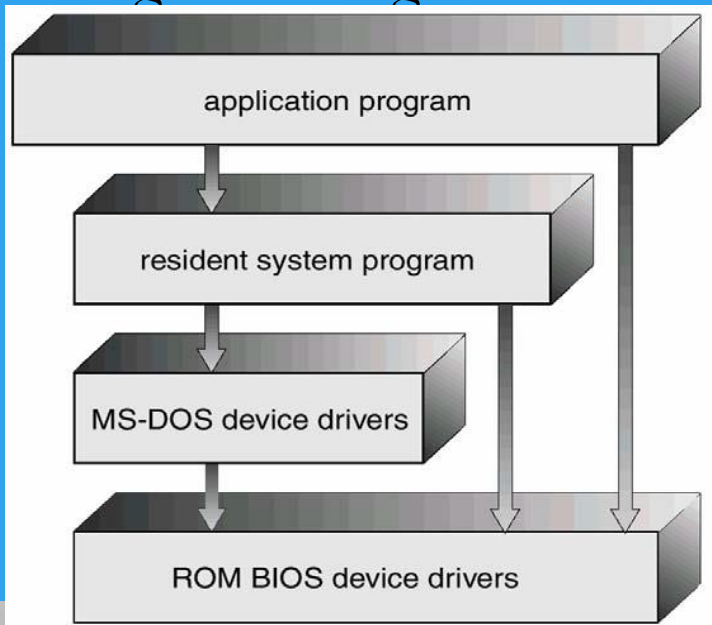
- Menyediakan lingkungan yang nyaman untuk pengembangan dan eksekusi program.
- Kategori:
  - Manipulasi File
  - Status Informasi
  - Modifikasi File
  - Bahasa pemrograman yang mendukung
  - Pemanggilan dan eksekusi program
  - Komunikasi
  - Program-program aplikasi

# Struktur Sistem Operasi

- Struktur sistem operasi:
  - Simple Structure (MS-DOS, UNIX)
  - Layered Approach (THE, Venus)
  - Microkernel (Minix)
  - Modules (Solaris)
  - Hybrid (Apple Mac OS X, iOS, Android)

# Sistem MS-DOS

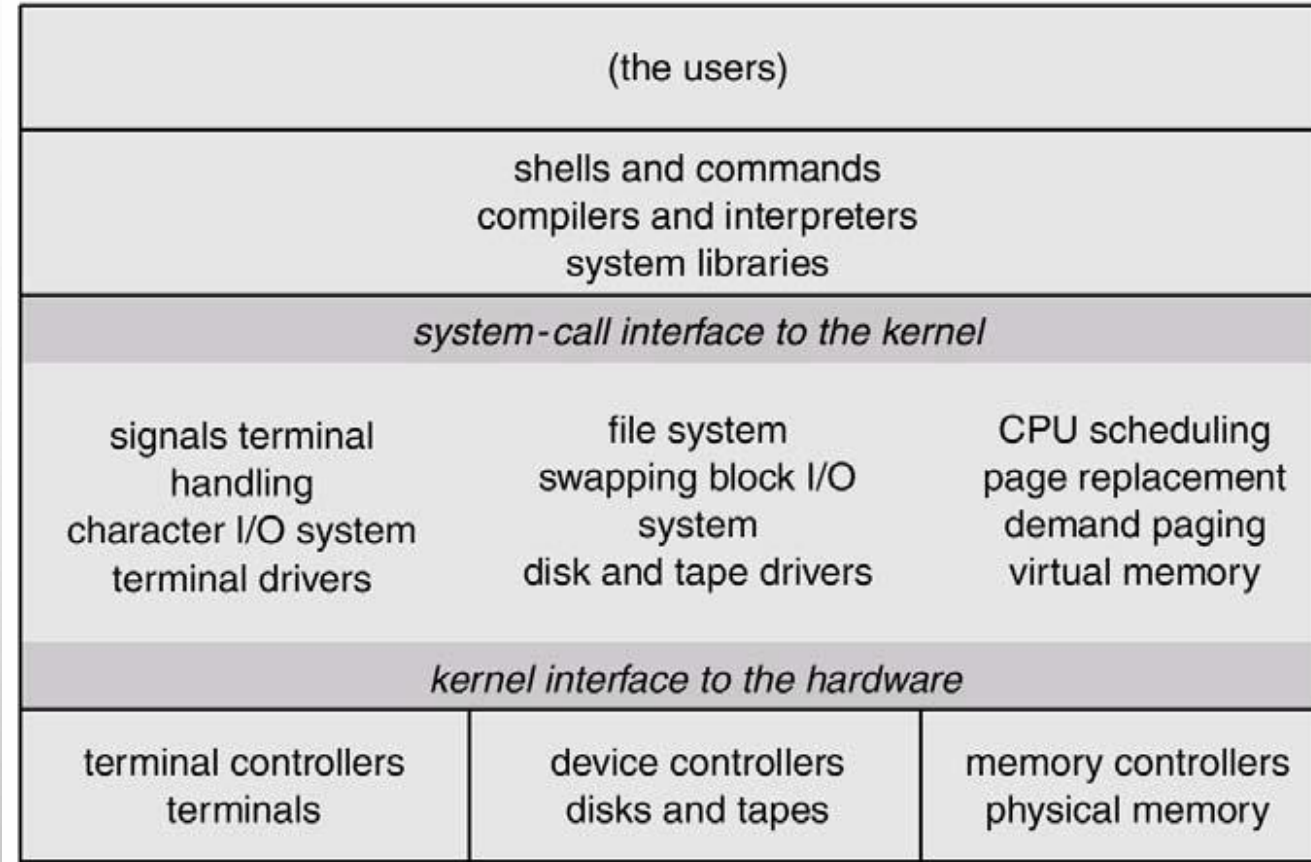
- Kecil dan terbatas
- Tidak terbagi menjadi modul-modul
- Meski mempunyai beberapa struktur, antar muka dan tingkatan fungsionalitas tidak terpisah secara baik



# Sistem UNIX

- 2 bagian UNIX: Kernel dan System Program
- Kernel terdiri dari antar muka system call dan hardware atasnya
- Kernel menyediakan sistem file, penjadwalan CPU, manajemen memori

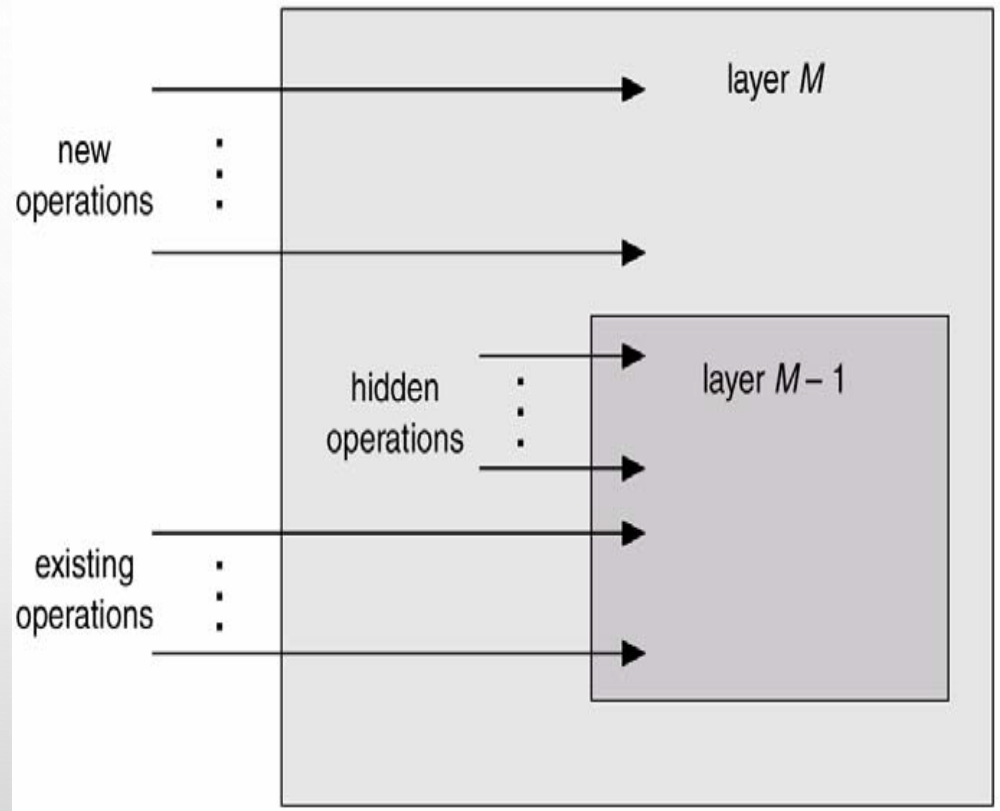
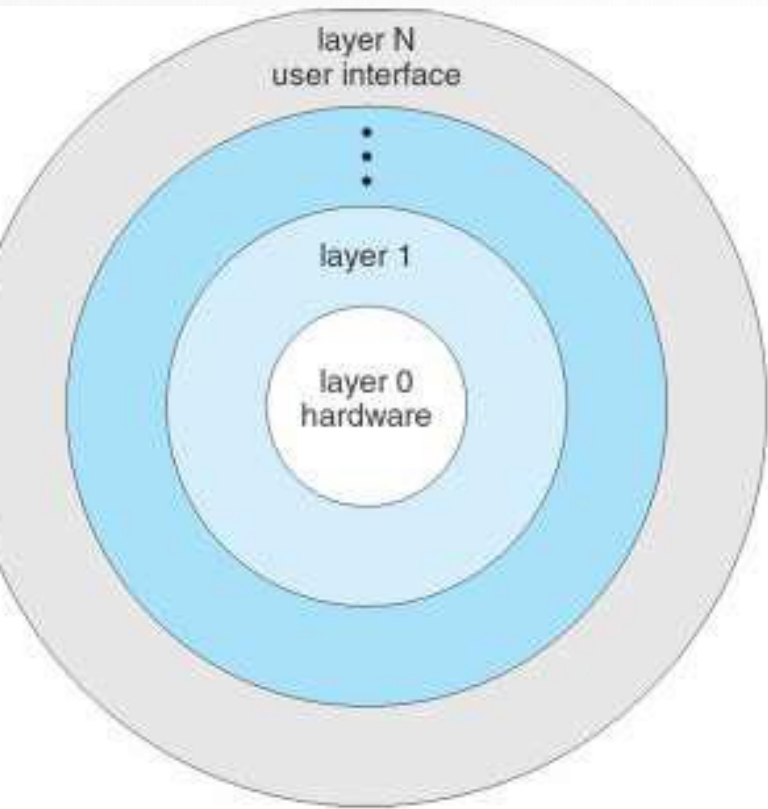
# Sistem UNIX (cont.)

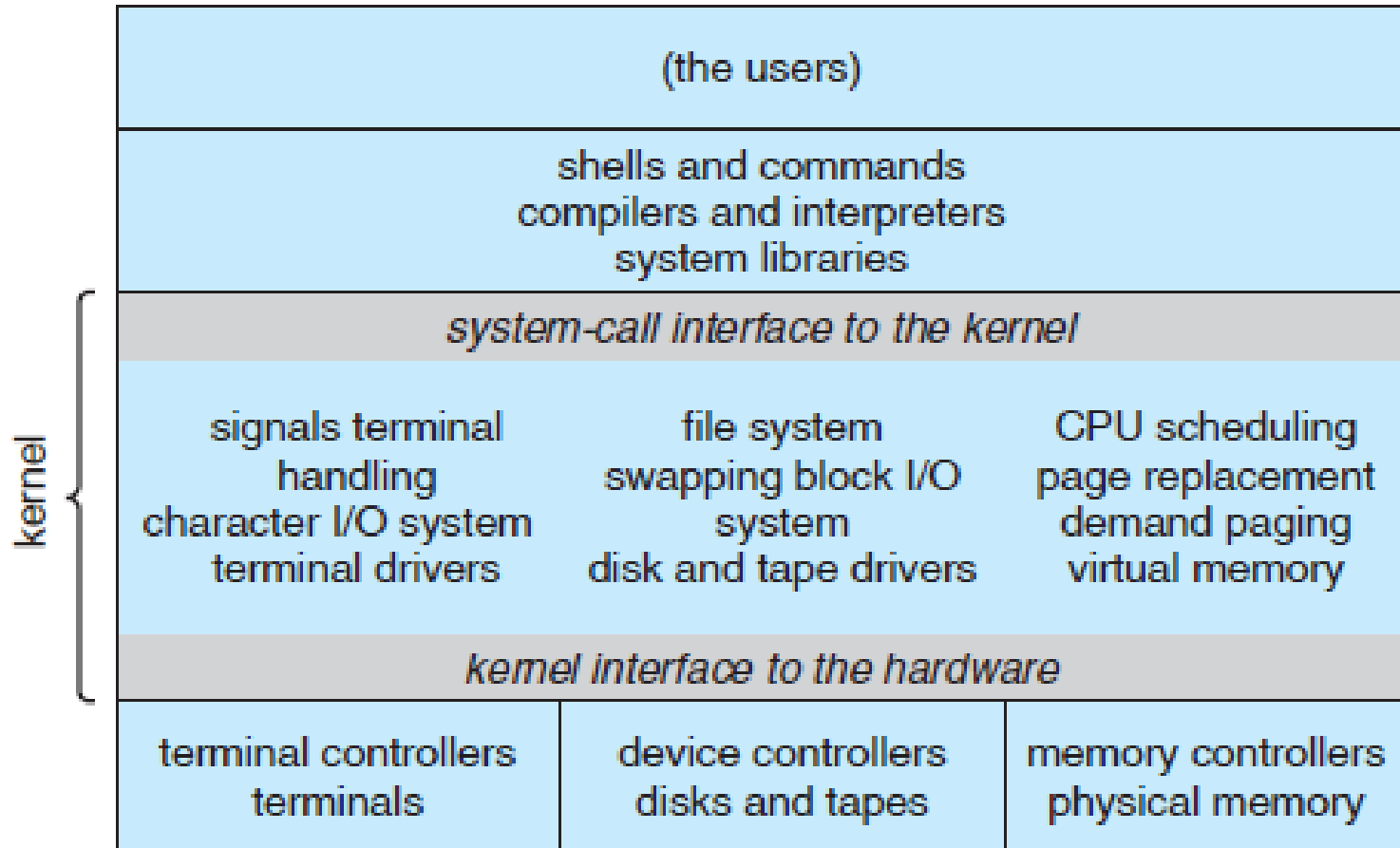


# Layered Approach

- Metode top-down
- Semua fungsi ditentukan dan dibagi menjadi komponen-komponen
- Modularisasi → memecah menjadi beberapa tingkat
- Layer terendah (layer 0) hardware, layer teratas (layer N) user interface

# Layered Approach (cont.)





**Figure 2.13** Traditional UNIX system structure



# Layered Approach (cont.)

- Contoh OS: THE, Venus
- Lapisan 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

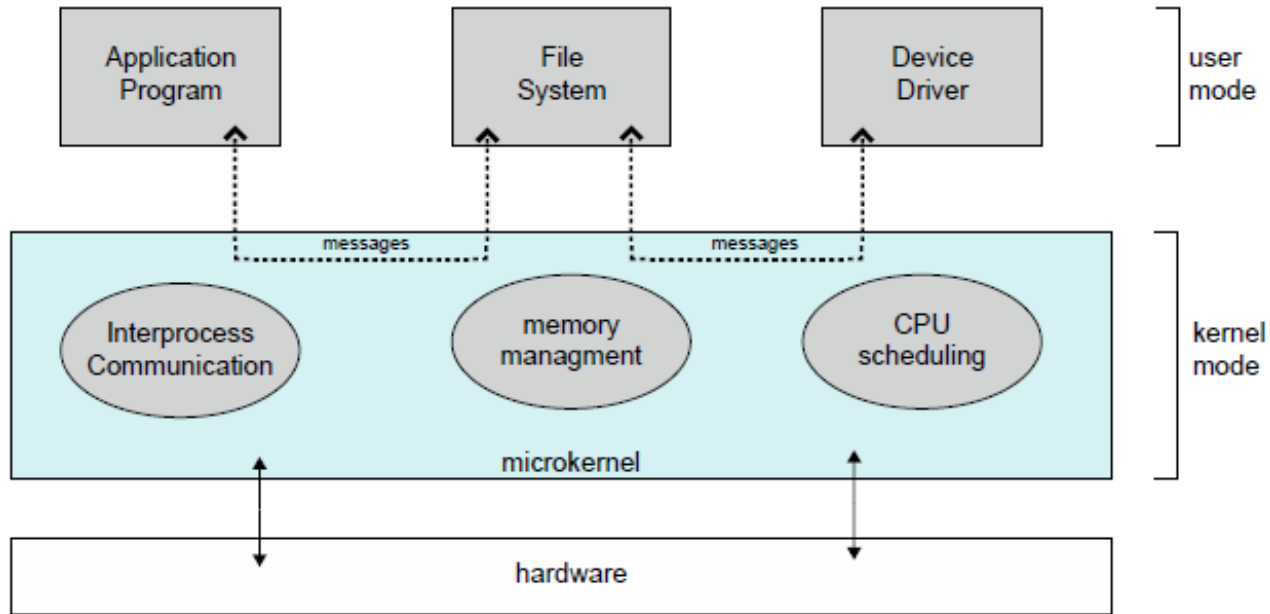
# Layered Approach (cont.)

- Lapisan Venus:
  - Lapis-6 : user program
  - Lapis-5 : *device driver* dan sceduler
  - Lapis-4 : virtual memory
  - Lapis-3 : I/O channel
  - Lapis-2 : penjadwalan CPU
  - Lapis-1 : instruksi interpreter
  - Lapis-0 : hardware

# Microkernel

- Menghapus komponen yang tidak penting dari kernel dan mengimplementasikannya dalam sistem dan user-level program.
- Menyediakan proses dan manajemen memori yang minimal.
- Komunikasi terjadi antara modul user menggunakan message passing.
- Keuntungan (lebih secure, lebih handal, mudah untuk memperluas sebuah microkernel, mudah diubah ke arsitektur baru).
- Kekurangan (kinerja akan berkurang selagi bertambahnya fungsi-fungsi yang digunakan).
- Contoh: Mach dan Minix

# Microkernel (cont.)



# Modules

- Kernel mempunyai kumpulan komponen-komponen inti dan secara dinamis terhubung pada penambahan layanan selama waktu boot atau waktu berjalan
- Sistem operasi yang modern saat ini menggunakan loadable kernel module.
- Contoh: Linux, Solaris, Windows

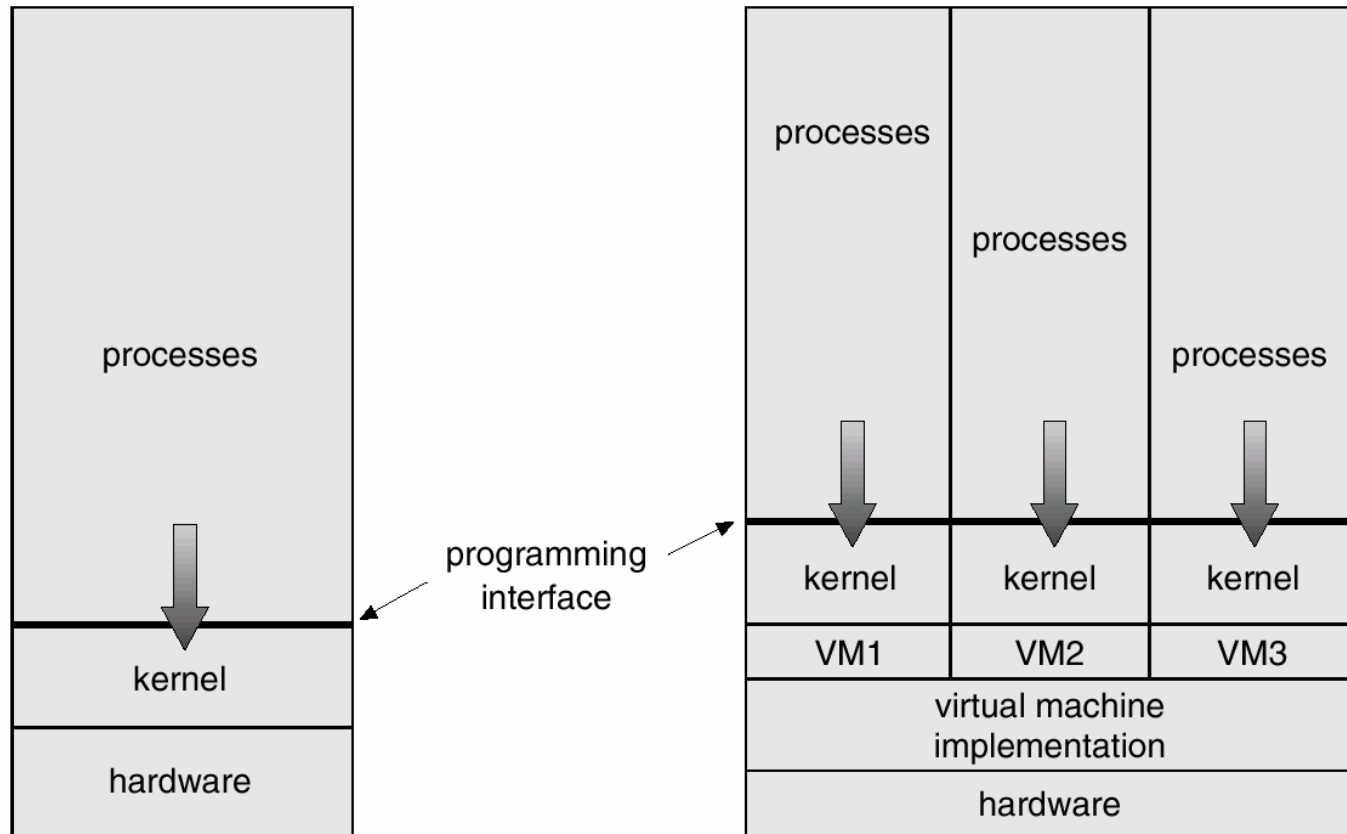
# Hybrid

- Sebagian besar sistem operasi modern tidak dalam satu model yang asli
- Menggabungkan beberapa struktur yang berbeda (kinerja, keamanan, kegunaan)
- Misal: Linux dan Solaris (monolithic dan juga modular), Windows (monolithic dan juga microkernel)

# Mesin Virtual

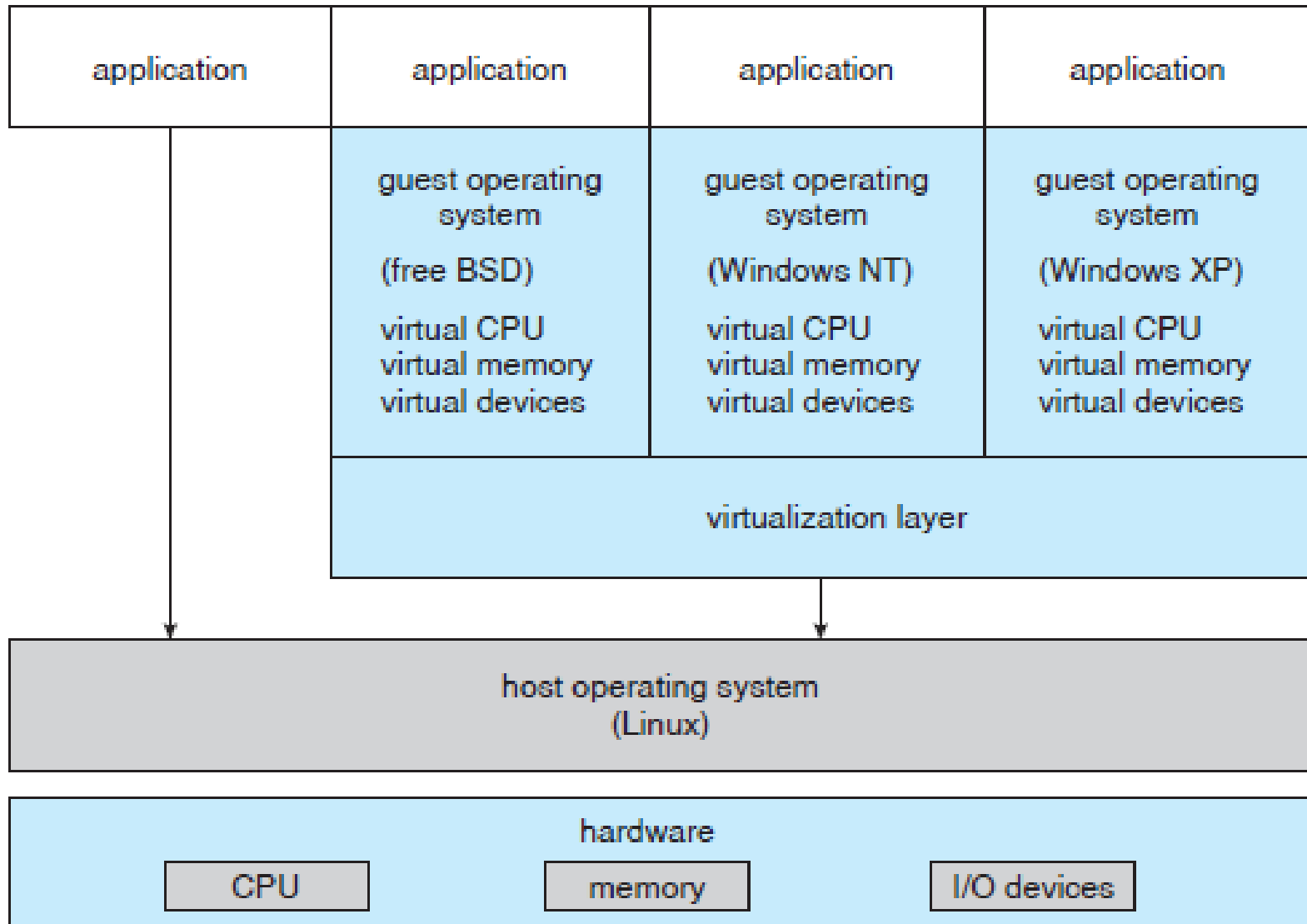
- Menyediakan antar muka yang identik untuk perangkat keras yang ada
- Sistem operasi membuat ilusi untuk beberapa proses, masing-masing mengeksekusi prosesor masing-masing untuk memori (virtual) masing-masing.
- Resource (physical mode) dibagi membuat untuk mesin virtual
- CPU scheduling → user mempunyai prosesor sendiri

# Mesin Virtual (cont.)





# VM Ware



# System Boot

- Prosedur menghidupkan komputer dengan memuat kernel disebut booting
- Ketika power diinisialisasi pada sistem, eksekusi dimulai pada lokasi memori yang tetap
- Sistem operasi harus tersedia untuk hardware agar dapat memulai proses booting
- Potongan kecil dari kode dinamakan bootstrap program atau bootstrap loader
- Bootstrap loader yang umum digunakan (GRUB) memungkinkan pemilihan kernel dari beberapa disk, versi, dan opsi kernel