1.1 Computer System Overview

Sistem Operasi adalah penghubung antara userengan hardware.

Fungsi sistem operasi :

- Memanfaatkan sumber daya hardware dari satu atau lebih processor
- Menyediakan berbagai layanan terhadap system users
- Mengatur memori sekunder dan I/O devices

Basic Element :

1. Processor.
    a. Processor terdiri dari 2 internal register
    MAR→ menyimpan address untuk read dan write.
    MBR → meyimpan data yang tertulis di memory atau menerima data yang dibaca dari memory
    b. I/O address register dan I/O buffer register
2. Main Memory
    Biasa disebut sebagai real memory atau primary memory
3. I/O modules
    Terdiri dari secondary memory devices, communications equipment, terminals.
4. System Bus
    Penghubung /komunikator antar processor, main memory, dan I/O modules

Processor Register

1. User Visible Register
    - Biasa disebut sebagai machine language dan tersedia untuk semua program
    - Terdiri dari Address Register :
        - Index register: Adding an index to a base value to get the effective address
        - Segment pointer: When memory is divided into segments, memory is referenced by a segment and an offset
        - Stack pointer: Points to top of stack-push/pop (no address)
    - Fungsi : Membantu programmer dalam meminimalkan penggunaan main memory dengan mengoptimasi penggunaan dari register.

2. Control and Status Register
    - Digunakan oleh processor untuk mengontrol penjalanan processor
    - Digunakan oleh privileged OS untuk mengontrol jalannyan eksekusi program
    - Terdiri dari :
        - Program Counter → Mengandung address dari instruksi yang akan diambil
        - Instruction Register → Mengandung instruksi yang baru saja diambil
        - Program status word → Mengandung informasi status (interupt, kernel/user mode bit, dll)
        - Condition code or flags→bit ser dari processor hardware sebagai result dari operasi

Instruction Execution

Terdiri dari 2 Step :

1. Processor membaca/mengambil instruksi dari memory.
2. Processor mengeksekusi/menjalankan setiap instruksi yang telah di baca.

Instruction Fetch and Execute

1. Processor membaca/mengambil instruksi dari memory
2. Program Counter mengambil address dari instruksi yang akan dibaca selanjutnya.
3. Program Counter increment setelah setiap pembacaan

Instruction Register

1. Mengambil instruksi dan memasukkannya ke Instruction Register.
2. Mengkategorikan setiap instruksi (processor-memory, processor I/O, data processing, control)

Interrupts

Interrupts merupakan request dari external device untuk mendapatkan respon dari procesesor dikarekan I/O devices cenderung lebih lambat dibandingkan proccessor sehingga processor harus di pause untuk menunggu devices.

- Jenis – Jenis Interrupts :
  1. Program → Dibuat oleh sebuah kondisi yang muncul atas respon terhadap eksekusi sebuah instruksi (arithmetic overflow, divison by zero, attempt to execute an illegal machine instruction, etc)
  2. Timer → Dibuat oleh timer yang ada di processor untuk membuat Operating System dalam menjalankan berbagai fungsi dalam keadaan regular
  3. I/O → Dibuat oleb I/O controller untuk memberikan signal normal terhadap penyelesaian sebuah operasi atau signal dari variasi error dari sebuah kondisi
  4. Hardware Failure → Dibuat oleh kegagalan seperti power failure, atau parity error
- Interrupt Stage :
  1. Processor mengecek terhadap intterupts
  2. Jika terdapat interrupt, Tunda eksekusi dari program dan eksekusi interrupt-handler routine

Multiprogramming

- Processor memiliki lebih dari satu program untuk dieksekusi
- Urutan darimana program tersebut akan dieksekusi bergantung terhadap prioritas relative mereka dan apakah mereka sedang menunggu untuk I/O
- Setelah sebuah interupt handler selesai, control mungkin tidak kembali ke program yang telah di eksekusi pada saat di interupsi

Memory Hierarchy

- Faster access time → Greater cost per bit

- Greater capacity → Smaller cost per bit
- Greater Capacity → Slower access speed

Going Down the Hierarchy

- Decreasing cost per bit
- Increasing capacity
- Increasing access time
- Decreasing frequency of access to the memory by the processor

Secondary Memory

- Memory pembantu
- Bersifat eksternal
- Tidak mudah berubah ubah
- Digunakan untuk menyimpan program dan file data

Chace Memory

- Speed processor lebih cepat dibanding dengan speed memory access
- Menggunakan prinsip dari locality with a small fast memory

Chace Principles :

1. Mengandung copy dari bagian main memory
2. Processor pertama mengecek cache
3. Jika data yang diinginkan tidak ditemukan, block of memory yang relevan membaca cache
4. Karena referensi dari locality, sangat memungkinkan memory yang akan data akan direferensikan di block tersebut.

1.2 Process Description and Control

Process

- Process merupakan sebuah unit aktifitas yang dikategorikan oleh eksekusi dari sebuah set urutan instruksi, kondisi sekarang, dan memiliki hubungan dengan instruksi sistem.
- Biasa disebut dengan program yang sedang dijalankan/dieksekusi
- Atau sebuah entitas yang bisa ditugaskan dan menjalankan/mengeksekusi pada processor

Process Elements

- Suatu process terdiri dari :
    - Program code
    - A set of data
- Element yang terdapat pada process :
    - Identifier
    - State
    - Priority
    - Program Counter
    - Memory pointers
    - Context data
    - I/O status information
    - Accounting information

Process Control Back

- Terdiri dari process element
- Dibuat dan diatur oleh sistem operasi
- Mendukung banyak proses sekaligus

Trace of the Process

- Trace merupakan list dari urutan instruksi yang sedang dijalankan/eksekusi yang merupakan kebiasaan dari sebuah process individual.
- Dispatcher merupakan program kecil/sederhana yang menukar processsor dari sebuah proses ke proses lainnya. (dapat membuat keputusan untuk proses mana saja yang akan di eksekusi selanjutnya.)

Two-State Processor Model

- Process bisa teradapat pada 2 kondisi :
  1. Running
  2. Not Running
- Sistem operasi harus dapat mengatur proses dari sebuah eksekusi dimana hal ini menyangkut dalam menentukan pola untuk menyusun sebuah eksekusi dan alokasi sumber kepada proses

Process creation and termination

Creation	Termination
New batch job	Normal Completion
Interactive Login	Memory unavailable
Created by OS to provide a service	Protection error
Spawned by existing process	Operator or OS Intervention

1. Process Creation
   - Sistem operasi membuat sebuah struktur data untuk mengatur process
   - Secara sederhana, sistem operasi membuat semua process, tetapi terdapat process yang telah berjalan membantu dalam membuat process lain (disebut sebagai **Process Spawning**).
   - Process spawing terdiri dari :
     - Parent Process → Original
     - Child Process → Process baru yang dibuat oleh parent process.

2. Process Termination
   - Harus terdapat cara untuk mengindikasikan selesainya sebuah process.
   - Hal tersebut bisa berupa :
     - A HALT instruction yang membuat peringatan intterupts ke OS
     - A user action (log off, quitting an application)
     - A fault or error
     - Parent process terminating

Five-State Process Model

1. **New**: It means a new process that has been created but has not yet been admitted by the OS for its execution.
2. **Ready**: It means a process that is prepared to execute when given the opportunity by the OS.

3. **Running**: It means a process that is currently being executed.
4. **Blocked/Waiting**: It means that a process cannot continue executing until some event occurs
5. **Exit/Terminate**: A process or job that has been released by the OS, either because it is completed or is aborted for some issue.

Suspended Processes

- Processor is faster than I/O so all processes could be waiting for I/O
  - Swap these processes to disk to free up more memory and use processor on more / other processes
- Blocked state becomes **suspend** state when swapped to disk (secondary storage)
- Two new states
  - Blocked/Suspend … from memory → disc
  - Ready/Suspend … from disc → memory
- Reason for Suspension :

Reason	Comment
Swapping	The OS needs to release sufficient main memory to bring in a process that is ready to execute.
Other OS Reason	OS suspects process of causing a problem.
Interactive User Request	e.g. debugging or in connection with the use of a resource.
Timing	A process may be executed periodically (e.g., an accounting or system monitoring process) and may be suspended while waiting for the next time.
Parent Process Request	A parent process may wish to suspend execution of a descendent to examine or modify the suspended process, or to coordinate the activity of various descendants.

Processes dan Resources

- The OS controls events within the computer system. It schedules and dispatches processes for execution by the processor, allocates resources to processes, and responds to requests by user processes for basic services.
- Fundamentally, we can think of the OS as that entity that manages the use of system resources by processes.

Operating System Control Structutes

- For the OS is to manage processes and resources, it must have information about the current status of each process and resource.
- Tables are constructed for each entity the operating system manages

Memory Tables

- Memory tables are used to keep track of both main and secondary memory.
- Must include this information:
  - Allocation of main memory to processes
  - Allocation of secondary memory to processes
  - Protection attributes for access to shared memory regions
  - Information needed to manage virtual memory

I/O Tables

- Used by the OS to manage the I/O devices and channels of the computer.

- The OS needs to know
    - Whether the I/O device is available or assigned
    - The status of I/O operation
    - The location in main memory being used as the source or destination of the I/O transfer

File Tables

- These tables provide information about:
    - Existence of files
    - Location on secondary memory
    - Current Status
    - other attributes.
- Sometimes this information is maintained by a file management system

Process Tables

- To manage processes the OS needs to know details of the processes
    - Current state
    - Process ID
    - Location in memory
    - Etc
- Process control block
    - ***Process image*** is the collection of program. Data, stack, and attributes

Process Attributes

- We can group the process control block information into three general categories:
    - Process identification
    - Processor state information
    - Process control information

Process Identification

- Each process is assigned a unique numeric identifier.
- Many of the other tables controlled by the OS may use process identifiers to cross-reference process tables

Processor State Information

- This consists of the contents of processor registers.
    - User-visible registers
    - Control and status registers
    - Stack pointers
- Program status word (PSW)
    - contains status information
    - Example: the EFLAGS register on Pentium processors

Process Control Information

- This is the additional information needed by the OS to control and coordinate the various active processes.

Role of the Process Control Block

- The most important data structure in an OS
    - It defines the state of the OS
- Process Control Block requires protection
    - A faulty routine could cause damage to the block destroying the OS's ability to manage the process
    - Any design change to the block could affect many modules of the OS

Modes of Execution

- Most processors support at least two modes of execution
- User mode
    - Less-privileged mode
    - User programs typically execute in this mode
- System mode
    - More-privileged mode
    - Kernel of the operating system

Process Creation

- Once the OS decides to create a new process it:
    - Assigns a unique process identifier
    - Allocates space for the process
    - Initializes process control block
    - Sets up appropriate linkages
    - Creates or expand other data structures

Switching Processes

- Several design issues are raised regarding process switching
    - What events trigger a process switch?
    - We must distinguish between mode switching and process switching.
    - What must the OS do to the various data structures under its control to achieve a process switch?
- When to Switch Processes

Mechanism	Cause	Use
Interrupt	External to the execution of the current instruction	Reaction to an asynchronous external event
Trap	Associated with the execution of the current instruction	Handling of an error or an exception condition
Supervisor call	Explicit request	Call to an operating system function

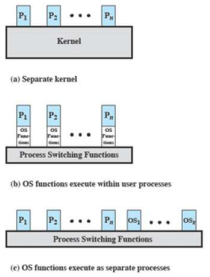
- Change of Process State

  The steps in a process switch are:
    - Save context of processor including program counter and other registers
    - Update the process control block of the process that is currently in the Running state
    - Move process control block to appropriate queue – ready; blocked; ready/suspend
    - Select another process for execution
    - Update the process control block of the process selected
    - Update memory-management data structures
    - Restore context of the selected process

Is the OS a Process?

- If the OS is just a collection of programs and if it is executed by the processor just like any other program, is the OS a process?
- If so, how is it controlled?
    - Who (what) controls it?

Execution of the Operating System



(a) Separate kernel

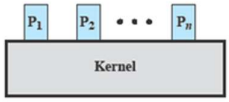
(b) OS functions execute within user processes

(c) OS functions execute as separate processes

Figure 3.15 Relationship Between Operating System and User Processes

Non-process Kernel

- Execute kernel outside of any process
- The concept of process is considered to apply only to user programs
  - Operating system code is executed as a separate entity that operates in privileged mode



(a) Separate kernel

Execution *Within* User Processes

- Execution Within User Processes
  - Operating system software within context of a user process
  - No need for Process Switch to run OS routine

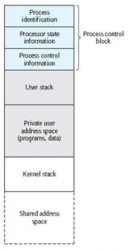
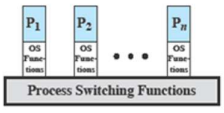


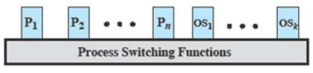
Figure 3.16 Process Image: Operating System Executes within User Space



(b) OS functions execute within user processes

Process-based Operating System

- Process-based operating system
  - Implement the OS as a collection of system process



(c) OS functions execute as separate processes

Security Issues

- An OS associates a set of privileges with each process.
  - Highest level being administrator, supervisor, or root, access.
- A key security issue in the design of any OS is to prevent anything (user or process) from gaining unauthorized privileges on the system

    - Especially - from gaining root access.

System Acces Threats

- Intruders
    - Masquerader (outsider)
    - Misfeasor (insider)
    - Clandestine user (outside or insider)
- Malicious software (malware)

Countermeasures: Intrusion Detection

- Intrusion detection systems are typically designed to detect human intruder and malicious software behaviour.
- May be host or network based
- Intrusion detection systems (IDS) typically comprise
    - Sensors
    - Analyzers
    - User Interface

Countermeasures: Authentication

- Two Stages:
    - Identification
    - Verification
- Four Factors:
    - Something the individual ***knows***
    - Something the individual ***possesses***
    - Something the individual ***is*** (static biometrics)
    - Something the individual ***does*** (dynamic biometrics)

Countermeasures: Access Control

- A policy governing access to resources
- A security administrator maintains an authorization database
    - The access control function consults this to determine whether to grant access.
- An auditing function monitors and keeps a record of user accesses to system resources.

Countermeasures: Firewalls

- Traditionally, a firewall is a dedicated computer that:
    - interfaces with computers outside a network
    - has special security precautions built into it to protect sensitive files on computers within the network.