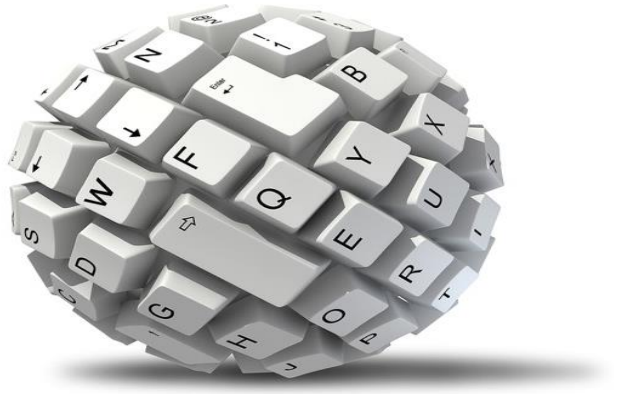
UNIVERSITI TENAGA NASIONAL

Chapter 3:
Operating System Overview



Operating System

- Definition:
  - ❑A program that controls the execution of application programs
  - ❑An interface between applications and hardware

❑An OS is a program which acts as an interface between computer system users and the computer hardware. It provides a user-friendly environment in which a user may easily develop and execute programs. Otherwise, hardware knowledge would be mandatory for computer programming. So, it can be said that an OS hides the complexity of hardware from uninterested users.

Operating system Objectives

- 1. Convenience
  - Makes the computer more convenient to be use – as a user/computer interface
- 2. Efficiency
  - Allows computer system resources to be used in an efficient manner – as a resource manager
- 3. Ability to evolve
  - Permit effective development, testing, and introduction of new system functions without interfering with service.

Obj. 1: OS as a User/Computer Interface

- Obj. 1: OS as a User/Computer Interface
- Hardware and software used in providing applications to a user can be viewed in a
- layered or hierarchical f n figure below:

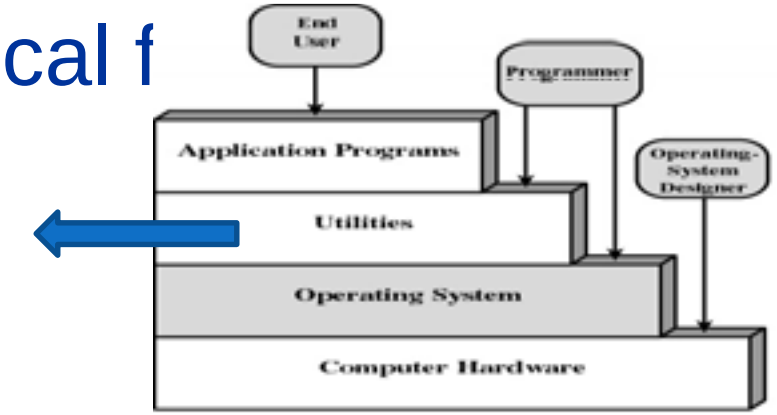


Figure 2.1 Layers and Views of a Computer System

- a set of system programs used
- in assisting program creation, management of
- files and controlling I/O. Application running and
- will call this utilities to perform function.

OS as a User/Computer Interface

- OS in the 3rd layer:

❖Act as a mediator, making it easier for the programmer and for application programs to access and use those facilities and services.

OS as a User/Computer Interface..

- OS typically provides services in the following areas:

❑ Program development

- Provides editors and debuggers (as utilities program) to assist programmer in creating programs.

❑ Program execution

- A number of steps need to be performed to execute a program; loaded data to memory, initializes the I/O devices and files, prepared the processor. OS will handled these scheduling duties for the user.

OS as a User/Computer Interface..

❑Access to I/O devices

- Provides a uniform interface where programmer can access the I/O using simple read and write

❑Controlled access to files

- Provide protection mechanisms to control access to the files.

❑System access

- For shared or public systems OS controls access to the system.

OS as a User/Computer Interface..

- Error detection and response
  - Errors occur while computer run:
    - Internal and external hardware errors
    - Software errors
    - OS cannot grant request of application
  - OS provide response that clear the error condition with least impact to running applications
    - Ending program that cause error
    - Retrying the operation
    - Reporting error to application

OS as a User/Computer Interface..

- Accounting
  - -collect statistics
  - -monitor performance
  - -used to anticipate future enhancements

Obj. 2: OS as Resource Manager

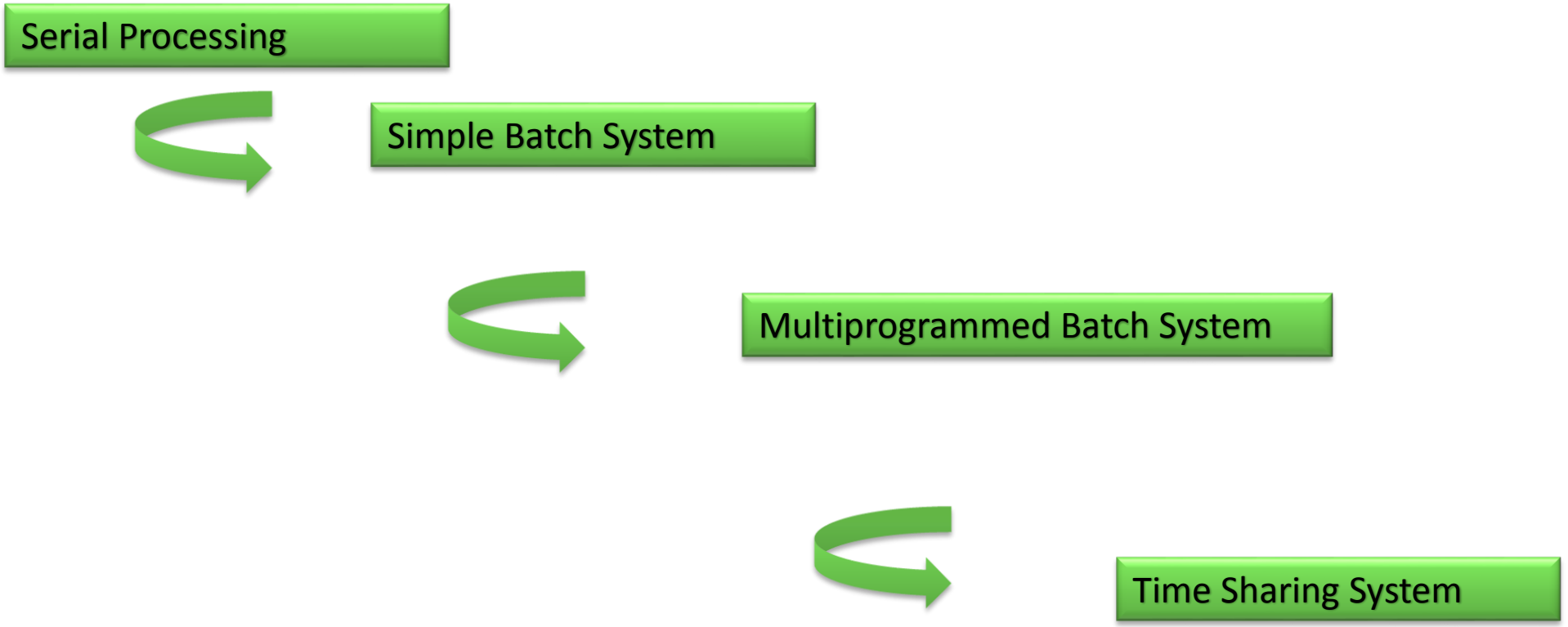
- A computer is a set of resources for the movement, storage and processing of data and for the control of these functions.

- OS is responsible for managing the resources.
- In the alternative view, the job of the operating system is to provide for an orderly and controlled allocation of the processors, memories, and i/o devices among the various programs competing for them.

Obj. 3: Ease of Evolution of an OS

- Major OS will evolved over the time because:
  - Hardware upgrades and new types of hardware
  - New services

Evolution of OS



Serial Processing

- No operating system
- User interact directly with computer hardware.
- Machines run from a console with display lights and toggle switches, input device, and printer
- This mode of operation could be termed serial processing because users have access to the computer in series.

Serial Processing..

- How it works?
  - Programs in machine code were loaded via the input device (e.g.. Card Reader)
  - If there is an error (indicating by the lights) → halt the program.
  - Then the programmer will proceed to examine processor registers and main memory to determine the cause of error.
  - If the program proceeds to normal completion, the output will be printed.

Serial Processing..

- Has TWO main problems:
  1. Scheduling
  2. Setup Time

- processor idle most of the time
- » Doing nothing, waiting to execute a job

Serial Processing..

- **Scheduling**
  - How user “used” the machine
  - User reserve a machine time
    - – Used a hardcopy sign-up sheet
  - Problem arise when user sign up more that the time needed or can’t complete their job with the time reserve.
- – Reserved for 1 hour, but completed execute job for 20 minutes → 40 minutes is a processor idle time
- – Reserved for 1 hour, but job did not completed.

  because of error occurs →need to reserve another time

Serial Processing..

- **Setup time**
  - Many steps involve in running a single program/job
    - – load the compiler + source program to memory
    - – saving the compiled program
    - – load and link the object program and common functions
    - – each of these steps involving mounting or dismounting tapes
  - If error occurred, user had to go back to the beginning setup step.
  - Much of the time is spending on setting up program.

Serial Processing..

- Over the time, various system software tools were developed to attempt to make serial processing more efficient.
  - Libraries of common functions, linkers, loaders, debuggers and I/O driver routines.

Simple Batch Systems

- Early machine was so expensive so it is important to maximize machine utilization → min the idle time of the processor.
- To improve utilization, the concept of a batch OS was developed.
- Central idea behind the simple batch processing scheme was the used of software known as the **monitor**.

Simple Batch Systems

- With monitor user have no longer direct access to the machine.
- Monitors:
  - Software that controls the running programs
  - Batch jobs together
  - Program branches back to monitor when finished
  - A part of the monitor call “Resident monitor” is in main memory and available for execution.

Simple Batch Systems

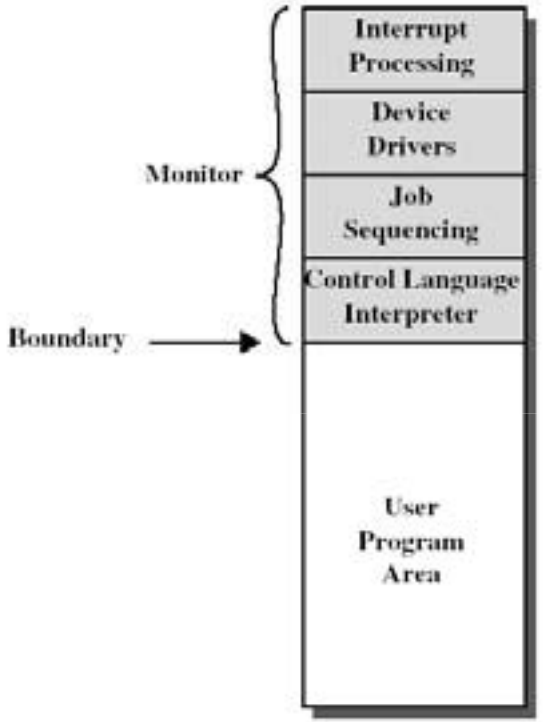


Figure 2.3 Memory Layout for a Resident Monitor

Simple Batch Systems

- How monitor works:
  - Monitor reads jobs one at a time from the input device
  - Monitor places a job in the user program area
  - A monitor instruction branches to the start of the user program
  - Execution of user program continues until
    - End of program occurs
    - Error occurs
  - This will cause the CPU to fetch its next instruction from monitor

Simple Batch Systems

- With each job, instructions are included in a primitive form of job control language (JCL)
  - Special type of programming language used to provides instruction to the monitor
    - what compiler to use
    - what data to use

Simple Batch Systems

- Example of job format using FORTRAN language:
  - $JOB
  - $FTN
  - …
  - FORTRAN Program
  - …
  - $LOAD
  - $RUN
  - …
  - DATA
  - …
  - $END

- Instruction start with a $ is a JCL language

Simple Batch Systems

- With JCL:
  - Each read instruction (in user program) causes one line of input to be read
  - Causes input routine (OS) to be invoke
    - Checks for not reading JCL line
    - Skip to the next JCL line at completion of user program

Simple Batch Systems

- In summary, the monitor or batch OS:
  - It is a computer program
  - Relies on the:
    - ability of the processor to fetch instructions from the various portions of the main memory to alternately seize and relinquish control.

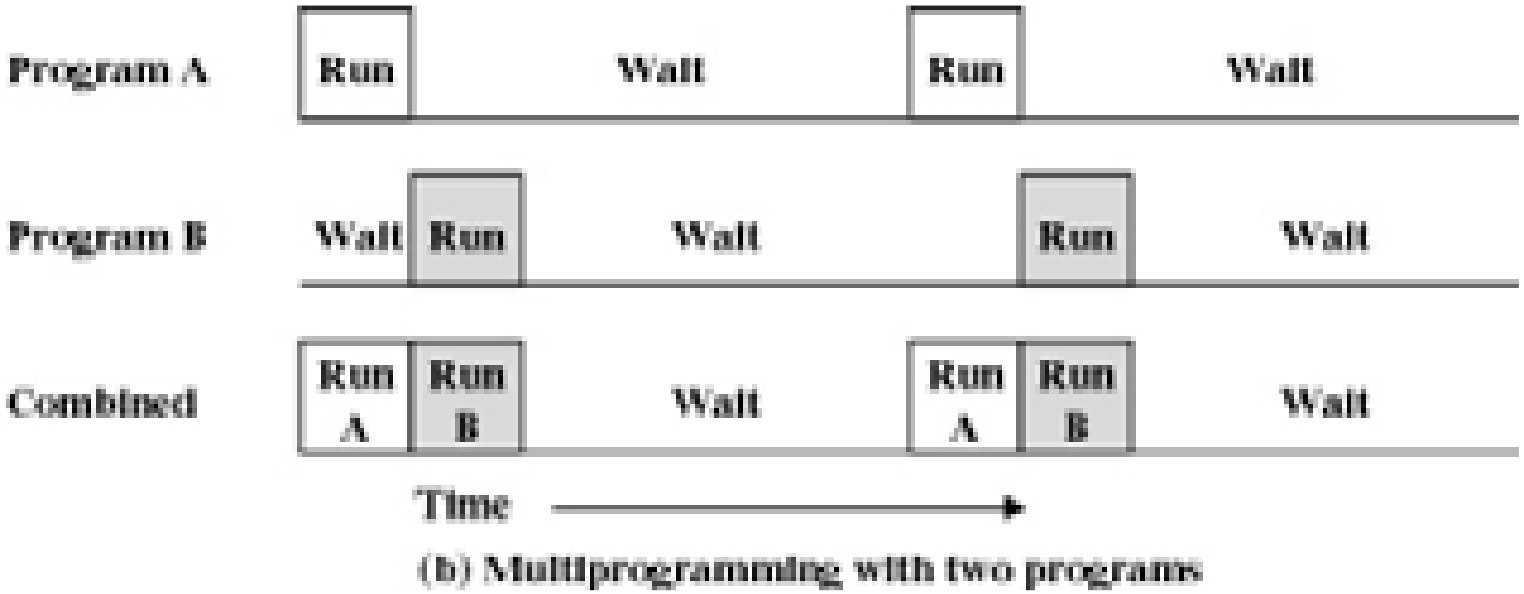
Multiprogrammed Batch System

- Simple Batch OS implement uniprogramming technique.

- Uniprogramming:
  - Processor execute a job, when job need for I/O, processor must wait for I/O instruction to complete before proceed to execute the other job.

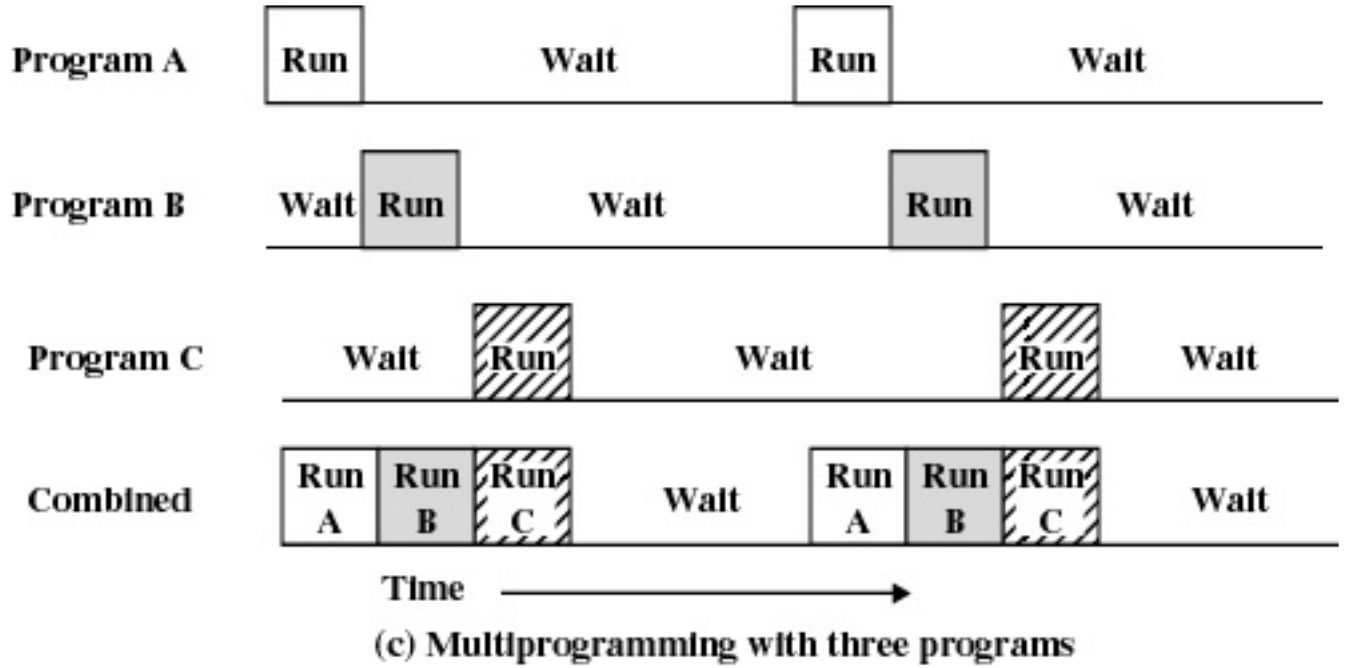
Multiprogrammed Batch System..

- Multiprogramming:
  - Processor execute a job, When job needs to wait for I/O, the processor can switch to the other job



(b) Multiprogramming with two programs

Multiprogrammed Batch System..



(c) Multiprogramming with three programs

Time Sharing

- A need to provide a mode in which the user interacts directly with the computer.
  - Transaction processing
- Batch multiprogramming does not support interactions with users.
- Time sharing extends multiprogramming to handle multiple interactive jobs.
- It is called time sharing because the processor's time is shared among multiple users.

Evolution of Operating Systems..

- Both batch OS and time sharing implement multiprogramming.
- The differences between batch OS and time sharing is shown below.

Major Achievements

- OS is the most complex pieces of software ever developed.
- Five major theoretical advances in the development of OS:
  - Process
  - Memory Management
  - Information protection and security
  - Scheduling and resource management
  - System structure

Major Achievements – Process

- Introduced to obtain a systematic way of monitoring and controlling program execution.
- Definitions of process:
  - A program in execution
  - An instance of a program running on a computer
  - The entity that can be assigned to and executed on a processor
  - A unit of activity characterized by a single sequential thread of execution, a current state, and an associated set of system resources

Major Achievements – Process..

- Difficulties with Designing System Software
  - Improper synchronization
    - ensure a process waiting for an I/O device receives the signal
  - Failed mutual exclusion
    - Must permit only one program at a time to perform a transaction on a portion of data.
  - Non-determinate program operation
    - program should only depend on input to it, not relying on common memory areas

Major Achievements – Process..

- Deadlocks
  - Two or more programs wait endlessly after each other to perform an operation.
- Consists of three components
  - An executable program
  - Associated data needed by the program
  - Execution context of the program
    - All information the CPU needs to execute the process
    - All information the operating system needs to manage the process

Major Achievements –
Memory Management

- Five principle storage management responsibilities:
  - Process isolation
  - Automatic allocation and management
  - Support for modular programming
  - Protection and access control
  - Long-term storage
- The key contribution is virtual memory and file system facilities.

Major Achievements – Memory Management

- Virtual memory:
  - Virtual memory is a feature of an operating system (OS) that allows a computer to compensate for shortages of physical memory by temporarily transferring pages of data from random access memory (RAM) to disk storage.
  - It allows programs to address memory from a logical point of view without regard to the amount that is physically available
  - While a program is running only a portion of the program and data is kept in (real) memory

Major Achievements – Memory Management

- File System:
  - Implements long-term store (often on disk)
  - Information stored in named objects called files
    - a convenient unit of access and protection for OS
  - Files (and portions) may be copied into virtual memory for manipulation by programs

Major Achievements – Memory Management

- Paging:
  - Allows process to be comprised of a number of fixed-size blocks, called pages
  - Virtual address is a page number and an offset within the page
  - Each page may be located any where in main memory
  - Real address or physical address in main memory

Major Achievements –

Memory Management

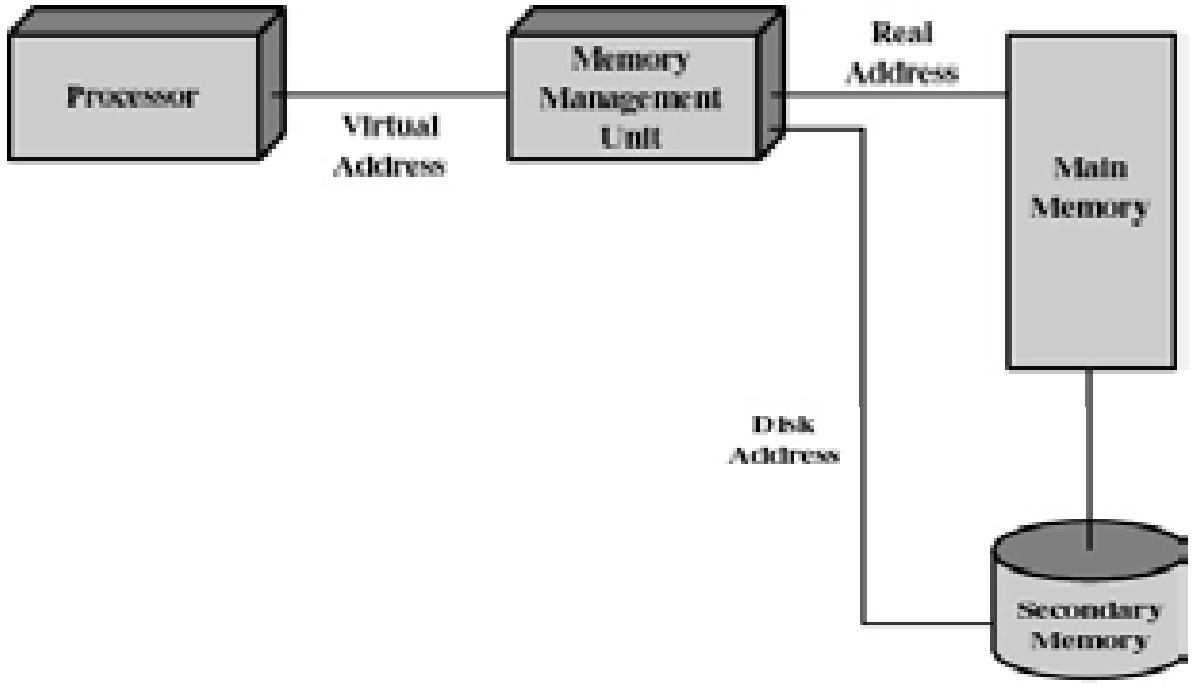


Figure 2.10 Virtual Memory Addressing

Major Achievements – Information Protection and Security..

- Protection of information is needed since the growth in the use of time-sharing systems and
- computer network.
- Work in security and protection as it relates to OS can be roughly grouped into four
- categories.
  - Availability
    - Protecting the system against interruption
  - Confidentiality
    - Assures that user cannot read data for which access is unauthorized

Major Achievements – Information Protection and Security..

- Data Integrity
  - Protection of data from unauthorized modification
- Authenticity
  - Proper verification of identity of users and the validity of messages or data

Major Achievements –
Scheduling and Resource Management

- A key task of the OS is to manage the various resources available and to schedule their use by the various active process.
- Resource allocation and scheduling policy must consider three factors:
  - Fairness
    - give equal and fair access to all processes
  - Differential responsiveness
    - discriminate between different classes of jobs

Major Achievements –
Scheduling and Resource Management

❖Efficiency

- maximize throughput, minimize response time, and accommodate as many uses as possible

Major Achievements –
System Structure

- Because of it's enormous complexity, the OS can be viewed as a series of levels
- Each level performs a related subset of functions
- Each level relies on the next lower level to perform more primitive functions
- Well defined interfaces: one level can be modified without affecting other levels
- This decomposes a problem into a number of more manageable sub problems

Characteristics of Modern OS

- New design elements were introduced recently.
- In response to new hardware development
  - multiprocessor machines
  - high-speed networks
  - faster processors and larger memory
- In response to new software needs
  - multimedia applications
  - Internet and Web access
  - Client/Server applications

Characteristics of Modern OS

- Changes not only in modification and enhancement to existing architectures but also in new ways of organizing the operating system.
- Much of the works fits into:
  - 1. Microkernel architecture
  - 2. Multithreading
  - 3. Symmetric multiprocessing
  - 4. Distributed OS
  - 5. Object-oriented design.

Characteristics of Modern OS - Microkernel Architecture

- Only a few essential functions in the kernel
  - primitive memory management (address space)
  - Inter process communication (IPC)
  - basic scheduling
- Other OS services are provided by processes running in user mode (servers)
  - Such as device drivers, file system, virtual memory
- Provide more flexibility, extensibility, portability

Characteristics of Modern OS - Multithreading

- A process is a collection of one or more threads that can run simultaneously.
- Useful when the application consists of several tasks that do not need to be serialized.
- Gives the programmer a greater control over the timing of application-related events.
- All threads within the same process share the same data and resources and a part of the process's execution context

Characteristics of Modern OS - Symmetric Multiprocessing

- Refers to a computer hardware architecture and also to the operating system behavior
- that reflects that architecture.
- Standalone computer system with the following characteristics:
  - A computer with multiple processors
  - These processors share the same main memory and I/O facilities, interconnected by a communication bus or other internal connection scheme.

Characteristics of Modern OS - Symmetric Multiprocessing..

- ❖ All processor can perform the same functions.
- ❖ Existence of multiple processors is transparent to the user.

❑ Advantages over uniprocessor architecture:

- 1. Performance
- 2. Availability
- 3. Incremental Growth
- 4. Scaling

Characteristics of Modern OS -

Distributed OS

- DOS is a collection of independent computers that appears to the users of the system as a single computer.
- Provides the illusion of a single main memory and single secondary memory space, i.e. appearing as a single system.
- E.g.. Cluster computer.

Characteristics of Modern OS - Object Oriented Design

- Used for adding modular extensions to a small kernel
- Enables programmers to customize an operating system without disrupting system integrity

THE END