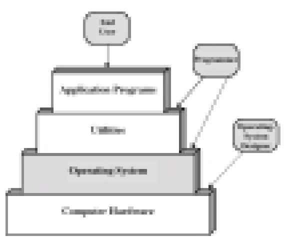
Operating Systems

Operating System Support

OS - is a **program** that

- Manages the computer's resources
- Provides services for programmers
- Schedules the execution of other programs
- Acts as an interface between user and computer (hardware)

OS As User/Computer Interface



OS provides these Services

- Program creation
- Program execution
- Access to I/O devices
- Controlled access to files
- System access
- Error detection and response
- Accounting

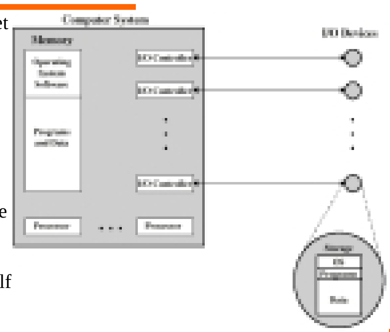
OS As Resource Manager

A computer is a set of resources for

- Data movement
- Data storage
- Data processing

OS is responsible for managing these resources.

The processor itself is a resource



Types of Operating Systems

Interactive - user interacts directly with the computer (via keyboard and display terminal) to request the execution of a job (program)

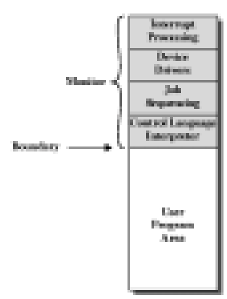
Batch - multiple programs batched together and submitted by an operator

Multiprogramming - the computer works on more than one program at a time, also known as multitasking

Uniprogramming - works on one program at a time

Simple Batch System

The user submit the job on cards or tapes to a program operator, who batches the jobs together sequentially and places the entire batch on an input device, for use by the monitor (program)



Simple Batch System: The monitor

- The monitor controls the sequence of events
  - Much of the monitor always in main memory
  - The monitor reads in jobs one at a time
  - The current job placed in user program area
  - Control passed on this this job
  - When the job is completed, it returns control to the monitor
  - The monitor reads the next job ….

Simple Batch System: The processor

At a certain point in time:

- The processor is executing instructions from the monitor part of memory
- These instructions cause the next job to be read (into user program part of the memory)
- (After finish reading the job) The processor encounter a branch instruction (in monitor) -
- Instruct the processor to execute program at the start of the user program.
- At the end of the user program or encounter an error, the processor fetches the next instruction from the monitor program

Simple Batch System: Desirable features

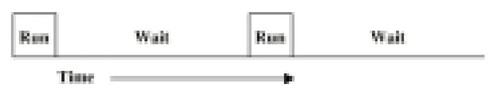
The monitor - batch OS is a computer program
The processor fetches instruction from various portions of the memory in order to seize or relinquish control

Following hardware features are desirable:

- Memory protection
  To protect the Monitor
- Timer
  To prevent a job monopolizing the system
- Privileged instructions
  Only executed by Monitor, e.g. I/O
- Interrupts
  Allows for relinquishing and regaining control

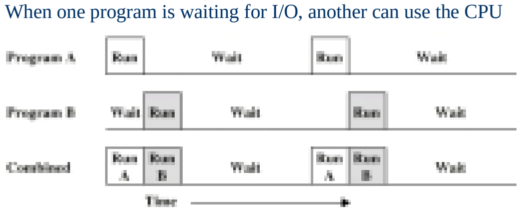
Uniprogramming (single tasking) Systems

I/O devices very slow



Multi-programmed (multitasking) Systems

When one program is waiting for I/O, another can use the CPU



Time Sharing Systems

In a time sharing system, multiple users simultaneously access the system through terminals, with the operating system interleaving the execution of each user program in a short burst of computation

Scheduling

The key to multitasking is scheduling.

Process

Many definitions:

A program in execution

The " animated spirit" of a program

The entity to which a processor is assigned.

Scheduling

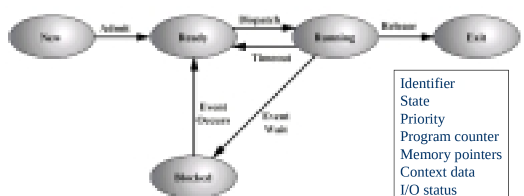
Long Term Scheduling: The decision to add to the pool of *processes* to be executed

Medium term scheduling: The decision to add to the number of *processes* that are partially or fully in memory

Short term scheduling: The decision as to which available *process* will be executed

I/O Scheduling: The decision as to which *process's* pending I/O request shall be handled by an available I/O devices.

Process States

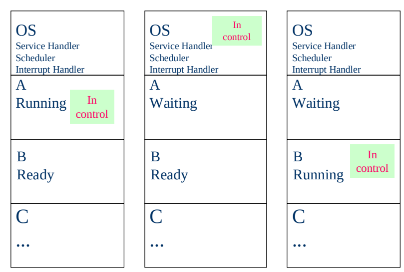


Identifier
State
Priority
Program counter
Memory pointers
Context data
I/O status
Accounting information
………

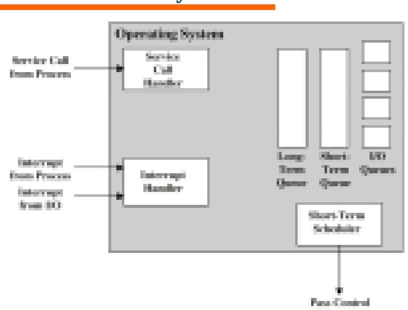
Process control block is maintained by the OS to indicate the state of the process and other info

Process Control Block

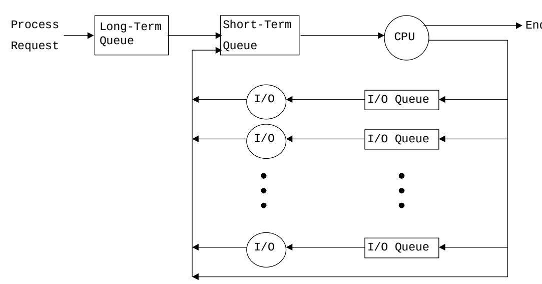
Scheduling Techniques



Key Elements of OS

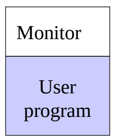


Process Scheduling

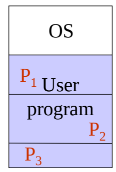


Memory Management

In single tasking system the memory is divided into two parts:

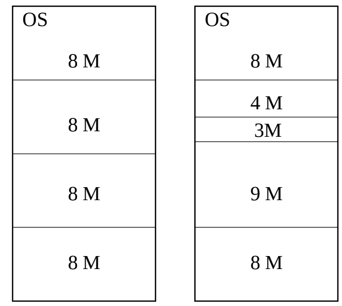


In multitasking system, the user part of the memory is divided to accommodate multiple processes:



Memory Management

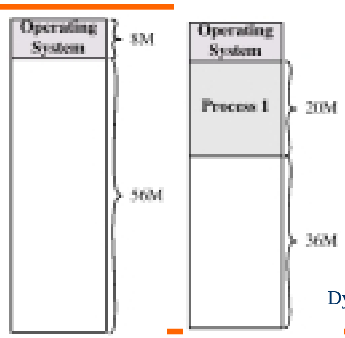
Partitioning



Fixed

Memory Management

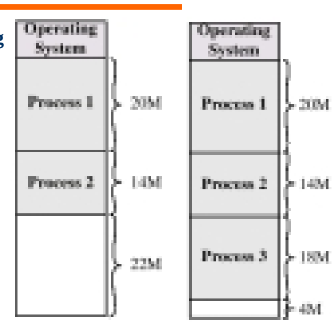
Partitioning



Dynamic

Memory Management

Partitioning



Dynamic

Logical and Physical Addresses

Logic address: Expressed as a location relative to the beginning of the program. Instructions in the program only contains logical addresses.

Physical Address: Actual location in the memory. When executing a program, the logical addresses are automatically converted into physical addresses

Virtual Memory

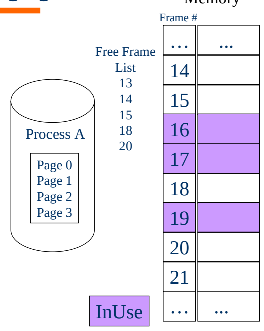
- We do not need all of a process in memory for it to run
- We can swap in pages as required
- So - we can now run processes that are bigger than total memory available!
- Main memory is called real memory
- User/programmer sees much bigger memory (that which is allocated on disk) - virtual memory

Paging

Partitioning the memory into small fixed sized chunks - **Frame**

Partitioning the process into small fixed sized chunks - **Page**

The wasted memory is a fraction of the last frame

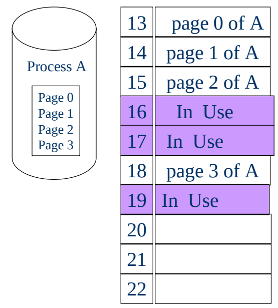


Paging

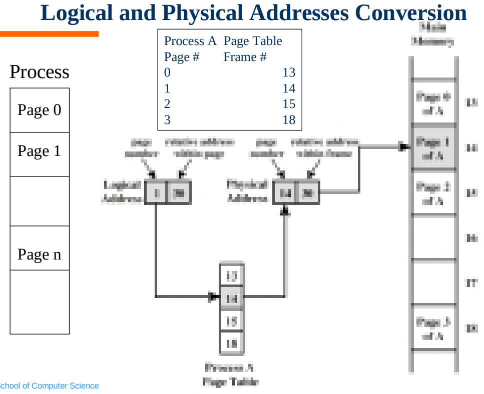
Page Table: records the frame location of each page.

Process A
Page Table

Page #	Frame #
0	13
1	14
2	15
3	18



Logical and Physical Addresses Conversion



Logical and Physical Addresses Conversion

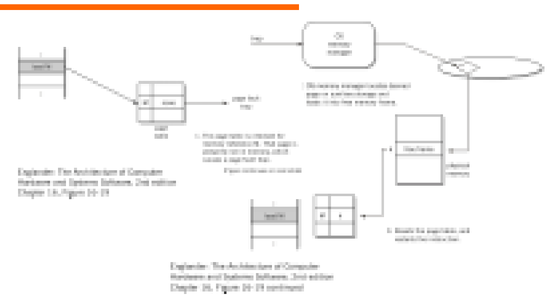
Example: Suppose the page table for the process currently executing on the processor looks like following. All numbers are decimal, everything is numbered starting from zero, and all addresses are memory byte addresses. The page size is 1024 bytes. What physical address, if any, would each of the following virtual addresses correspond to? (i) 1052, (ii) 2221, (iii) 5499

Virtual page #	Frame #	Valid bit
0	4	1
1	7	1
2	--	0
3	2	1
4	--	0
5	--	0

Page Fault

- Page fault
  - Required page is not in memory
  - Operating System must swap in required page
  - May need to swap out a page to make space
  - Select page to throw out based on recent history

Page Fault



Page Replacement Algorithms

- First-In, First-Out page replacement (FIFO)
- Least recently used page replacement (LRU)
- Least frequently used page replacement (LFU)

Thrashing

- Thrashing
  - Too many processes in too little memory
  - Operating System spends all its time swapping
  - Little or no real work is done
  - Disk light is on all the time
- Solutions
  - Good page replacement algorithms
  - Reduce number of processes running
  - Fit more memory

An example

Assume that a program is to be executed on a computer with virtual storage. The machine supports 10,000 words of logical memory overall, broken into pages of 100 words each. This particular machine contains 400 physical memory locations. Suppose that the machine starts to execute a program. The page table is initially empty, and is filled as necessary. Suppose that the program references the following sequence of memory locations:

start 951, 952, 4730, 955, 2217, 3663, 2217, 4785, 957, 2401, 959, 2496,3510, 962 *end*

Indicate the points at which page faults will occur and show the page table at the end of the sequence for each of the following demand page replacement algorithms:

FIFO LRU LFU