

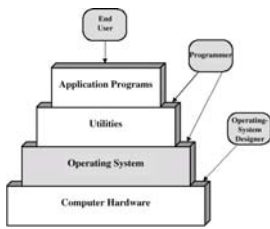
# Operating Systems

## 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

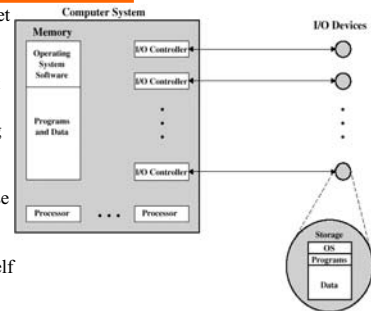
## OS As Resource Manager

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)

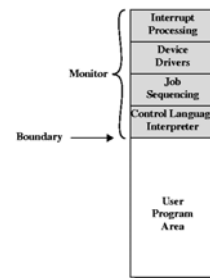
**Batch** - multiple programs batched together and submitted by an operator

**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

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



School of Computer Science GS1CSA

7

### Simple Batch System: The processor

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



School of Computer Science GS1CSA

8

### 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:

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

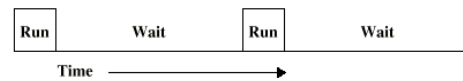


School of Computer Science GS1CSA

9

### Uniprogramming (single tasking) Systems

I/O devices very slow

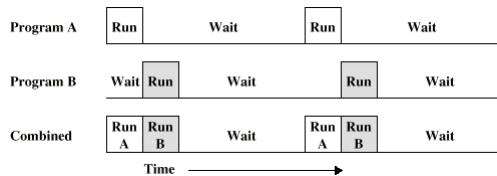


School of Computer Science GS1CSA

10

### Multi-programmed (multitasking) Systems

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



School of Computer Science GS1CSA

11

### 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



School of Computer Science GS1CSA

12

## 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.



School of Computer Science GS1CSA

13

## 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

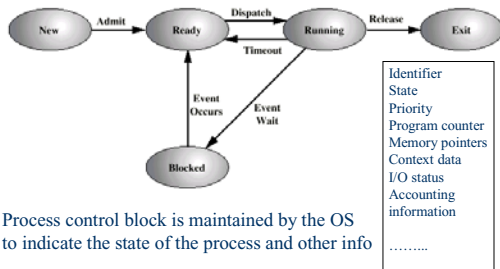
**I/O Scheduling:** The decision as to which *process*'s pending I/O request shall be handled by an available I/O devices.



School of Computer Science GS1CSA

14

## Process States



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

Process Control Block

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



School of Computer Science GS1CSA

15

## Scheduling Techniques

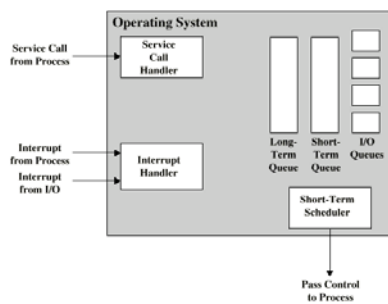
OS Service Handler Scheduler Interrupt Handler	OS Service Handler Scheduler Interrupt Handler	OS Service Handler Scheduler Interrupt Handler
A Running In control	A Waiting	A Waiting
B Ready	B Ready	B Running In control
C ...	C ...	C ...



School of Computer Science GS1CSA

16

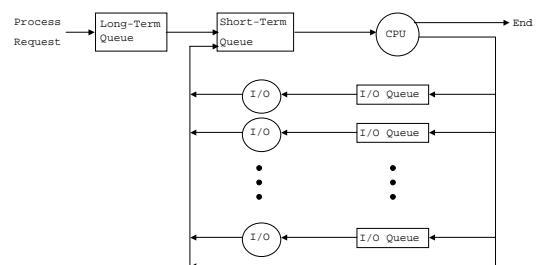
## Key Elements of OS



School of Computer Science GS1CSA

17

## Process Scheduling



School of Computer Science GS1CSA

18

### Memory Management

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

Monitor

User program

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

OS

P<sub>1</sub> User program

P<sub>2</sub>

P<sub>3</sub>

School of Computer Science GS1C5A
19

### Memory Management

**Partitioning**

OS
8 M
8 M
8 M
8 M

OS
8 M
4 M
3M
9 M
8 M

**Fixed**

School of Computer Science GS1C5A
20

### Memory Management

**Partitioning**

Operating System	
8M	
}	
56M	

Operating System	
8M	
Process 1	
20M	
}	
36M	

**Dynamic**

School of Computer Science GS1C5A
21

### Memory Management

**Partitioning**

Operating System
20M
Process 1
14M
Process 2
22M

Operating System
20M
Process 1
14M
Process 2
18M
Process 3
4M

**Dynamic**

School of Computer Science GS1C5A
22

### 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

School of Computer Science GS1C5A
23

### 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

School of Computer Science GS1C5A
24

### 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

Free Frame List	Main Memory
13	14
14	15
15	16
18	17
20	18
	19
	20
	21
	...

School of Computer Science GS1CSA

### Paging

**Page Table:** records the frame location of each page.

Process A

Page Table

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

13	page 0 of A
14	page 1 of A
15	page 2 of A
16	In Use
17	In Use
18	page 3 of A
19	In Use
20	
21	
22	

School of Computer Science GS1CSA

### Logical and Physical Addresses Conversion

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

School of Computer Science

### 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

School of Computer Science GS1CSA

### 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

School of Computer Science GS1CSA

### Page Fault

Englander: The Architecture of Computer Hardware and Systems Software, 2nd edition Chapter 16, Figure 16-19

Englander: The Architecture of Computer Hardware and Systems Software, 2nd edition Chapter 16, Figure 16-19 continued

School of Computer Science GS1CSA

## 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

