إن اصبت فمن الله وإن اخطأت
فمن نفسي والشيطان .

Note 1

- operating system : A program that acts as an intermediary/interface between a user of a computer and the computer hardware

- Computer system can be divided into four components:
Hardware , Operating system , Application programs , Users

- “The one program running at all times on the computer” is the kernel

- Main memory – only large storage media that the CPU can access directly : volatile

- Secondary storage – extension of main memory that provides large nonvolatile storage capacity

- Caching – copying information into faster storage system

- Advantages of Multiprocessors include:
1. Increased throughput
2. Economy of scale
3. Increased reliability – graceful degradation or fault tolerance

- A computer cluster consists of a set of loosely or tightly connected computers that work together

- Dual-mode : operation allows OS to protect itself and other system components

- Program is a passive entity, process is an active entity

- Direct Memory Access : Used for high-speed I/O devices able to transmit information at close to memory speeds

- Protection – any mechanism for controlling access of processes or users to resources defined by the OS

- Security – defense of the system against internal and external attacks

Timer to prevent infinite loop / process hogging resources

Note 2

One set of operating-system services provides functions that are helpful to the user :

1- User interface
2- Program execution
3- I/O operations
4- File-system manipulation
5- Communications
6- Error detection
7- Resource allocation
8- Accounting - To keep track of which users use how much and what kinds of computer resources
9- Protection and security



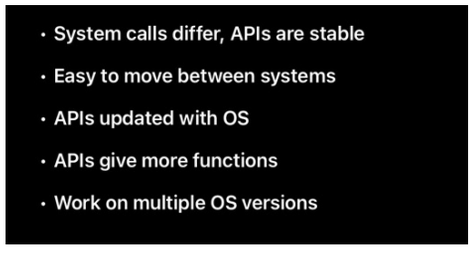
Block and stack methods do not limit the number or length of parameters being passed

System Calls : Programming interface to the services provided by the OS , Typically written in a high-level language

API vs. System Calls

1- The API usually provides more useful functionality than the system call directly.

2- System calls differ from platform to platform ,API it is easier to migrate your software to different platforms



Types of System Calls :

Process control :
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

File management :
create file, delete file
open, close file
read, write, reposition
get and set file attributes

Device management :
request device, release device
read, write, reposition
get device attributes, set device attributes
logically attach or detach devices

Information maintenance :
get time or date, set time or date
get system data, set system data
get and set process, file, or device attributes

Communications :
create, delete communication connection
send, receive messages
transfer status information
attach and detach remote devices

Goals:

- **User goals:** convenient, easy to learn, reliable, safe, fast.
- **System goals:** easy to design, implement, maintain; flexible, reliable, error-free, efficient.

Microkernel System Structure

Benefits:

- Easy to add or change
- Easy to move to other systems
- More stable
- More safe

هذا اختصار فقط
ارجعي للسلايدات

Drawback:

- Slower (because of communication)



System programs provide a convenient environment for program development and execution. They can be divided into:

1- File management
2- Status information
3- File modification
4- Programming-language support
5- Program loading and execution
6- Communications

Debugging is finding and fixing errors, or bugs





Failure of an application can generate core dump file capturing memory of the process

Operating system failure can generate crash dump file containing kernel memory

Small piece of code – bootstrap loader, locates the kernel, loads it into memory, and starts it Bootstrap program: stored in ROM, locates and loads the kernel into memory, starts execution

Virtual machine : An abstract hardware of a single computer into several different executing in the system

🔥 الخلاصة السريعة:

الـ Virtual Machine:

- تقسم جهاز واحد إلى عدة أجهزة وهمية
- كل جهاز وهمي يشتغل كنظام مستقل
- يعطي وهم أنه عنده CPU وMemory خاصة فيه

Note 3

Multiple parts

- The program code, also called text section
- Current activity including program counter, processor registers
- Stack containing temporary data
  - Function parameters, return addresses, local variables
- Data section containing global variables
- Heap containing memory dynamically allocated during run time

Program is passive entity stored on disk (executable file), process is active

- Program becomes process when executable file loaded into memory {Multi}

Process – a program in execution





possible states for process :
new , running , waiting , ready , terminated

So far, process has a single thread of execution

Maintains scheduling queues of processes

Job queue – set of all processes in the system

Ready queue – set of all processes residing in main memory, ready and waiting to execute

Device queues – set of processes waiting for an I/O device

Long-term scheduler (or job scheduler) – selects which processes should be brought into the ready queue ← Controls degree of multiprograming

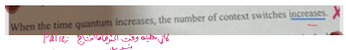
Short-term scheduler (or CPU scheduler) – selects which process should be executed next and allocates CPU

from ch5: selects a process from ready queue and allocate the CPU

Medium-term scheduler can be added if degree of multiple programming needs to decrease

I/O-bound process – spends more time doing I/O than computations, many short CPU bursts

CPU-bound process – spends more time doing computations; few very long CPU bursts



When CPU switches to another process, the system must save the state of the old process and load the saved state for the new process via a context switch

Context of a process represented in the PCB

Context-switch time is overhead; the system does no useful work while switching

- The more complex the OS and the PCB → longer the context switch

unbounded-buffer places no practical limit on the size of the buffer

bounded-buffer assumes that there is a fixed buffer size

Processes within a system may be independent or cooperating

Process Control Block (PCB)
(also called task control block):
Process state
execram counter
CPU registers
CPU scheduling information
Memory-management information
Accounting information
1/O status information

Interprocess Communication:
Mechanism for processes to communicate and to synchronize their actions

Many web browsers ran as **single process** (some still do)

- If one website causes trouble, entire browser can hang or crash

- Cooperating processes need interprocess communication (IPC)
- Two models of IPC
  - Shared memory
  - Message passing

Advantages of process cooperation:

- Information sharing
- Computation speed-up
- Modularity
- Convenience

Independent process cannot affect or be affected by the execution of another process

Cooperating process can affect or be affected by the execution of another process

Message system – processes communicate with each other without resorting to shared variables

Interprocess Communication : Mechanism for processes to communicate and to synchronize their actions

Tybe message passing:

Properties of direct communication :

1-Links are established automatically
2-A link is associated with exactly one pair of communicating processes
3-Between each pair there exists exactly one link
4-The link may be unidirectional, but is usually bi-directional

Properties of indirect communication :

Link established only if processes share a common mailbox
A link may be associated with many processes
Each pair of processes may share several communication links
Link may be unidirectional or bi-directional

- Operations
  - create a new mailbox
  - send and receive messages through mailbox
  - destroy a mailbox

Note 4

Consider a system consisting of n processes {P0, P1, ..., Pn−1}. Each process has a segment of code, called a critical section

in critical section no two processes are executing in their critical sections at the same time.

The critical-section problem is to design a protocol that the processes can use to cooperate. Each process must request permission to enter its critical section

solution of critical section is :

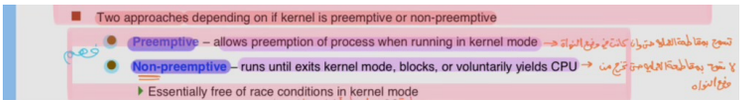
1- **Peterson's Solution** It is a good algorithmic description for solving the problem, but only two processes.
2- Mutual Exclusion
3- Progress
4- Bounded Waiting
5- Simplest is : mutex lock
   this solution requires busy waiting Also called spinlock

② If process Pi is executing in its critical section, then no other processes can be executing in their critical sections

③ If no process is executing in its critical section and there exist some processes that wish to enter their critical section, then the selection of the processes that will enter the critical section next cannot be postponed indefinitely.

④ A bound must exist on the number of times that other processes are allowed to enter their critical sections
before and after

A synchronization tool that does not require busy waiting



Counting semaphore – integer value can range over an unrestricted domain

Binary semaphore – integer value can range only between 0 and 1

Checkpoints shorten the log and reduce recovery time

Note 5

Maximum CPU utilization is achieved through multiprogramming

Dispatcher module gives control of the CPU to the process selected by the short-term scheduler

Dispatch latency – time it takes for the dispatcher to stop one process and start another running

Scheduling Criteria:

CPU utilization – keep the CPU as busy as possible

Throughput – of processes that complete their execution per time unit

Turnaround time – amount of time to execute a particular process

Waiting time – amount of time a process has been waiting in the ready queue

Response time – amount of time it takes from when a request was submitted until the first response is produced

CPU scheduling decisions may occur when a process:

- Switches from running → waiting
- Switches from running → ready
- Switches from waiting → ready
- Terminates





goal of Scheduling techniques :
Max CPU utilization
Max throughput
Min turnaround time
Min waiting time
Min response time

Convoy effect - short process behind long process] Problem to FCFS

SJF is optimal – gives minimum average waiting time for a given set of processes has Two schemes:

1- nonpreemptive – once CPU given to the process it cannot be preempted until completes its CPU burst

2- preemptive – if a new process arrives with CPU burst length less than remaining time of current executing process, preempt.

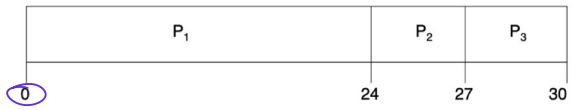
Note: waiting time = (finish time - Arrival time) - Burst time

بكل الحالات نشوف اذا فيه Arrival اعتبريه 0 وتوضع

First-Come, First-Served (FCFS) Scheduling

Process	Burst Time
P_1	24
P_2	3
P_3	3

- Suppose that the processes arrive in the order: P_1, P_2, P_3
  The Gantt Chart for the schedule is:



- Waiting time for P_1 = 0; P_2 = 24; P_3 = 27
- Average waiting time: (0 + 24 + 27)/3 = 17

Suppose that the processes arrive in the order:

P_2, P_3, P_1

- The Gantt chart for the schedule is:

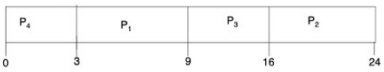


- Waiting time for P_1 = 6; P_2 = 0, P_3 = 3
- Average waiting time: (6 + 0 + 3)/3 = 3
- Much better than previous case
- Convoy effect - short process behind long process
  - Consider one CPU-bound and many I/O-bound processes

SJF

Process	Burst Time
P_1	6
P_2	8
P_3	7
P_4	3

- SJF scheduling chart



- Average waiting time = (3 + 16 + 9 + 0) / 4 = 7

- اول في اي أبلاوت
- اذا متساوين بالطول اختار الاقصر SJF

Non Preemptive

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

- SJF (non-preemptive)

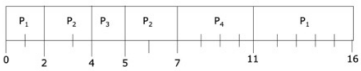


- Average waiting time = (p1+p2+p3+p4)
  = (0 + 6 + 3 + 7)/4
  = 4

Preemptive

Process	Arrival Time	Burst Time
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

- SJF (preemptive)



- Average waiting time = (9 + 1 + 0 +2)/4 = 3

- Now we add the concepts of varying arrival times and preemption to the analysis

Process	Arrival Time	Burst Time
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

- *Preemptive* SJF Gantt Chart



- Average waiting time = [(10-1)+(1-1)+(17-2)+5-3)]/4 = 26/4 = 6.5 msec

---

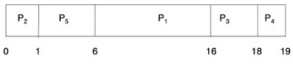
Example of Priority Scheduling

(Non preemptive)

Process	Burst Time	Priority
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

$P_1 = 6$
$P_2 = 0$
$P_3 = 16$
$P_4 = 18$
$P_5 = 1$

- Priority scheduling Gantt Chart

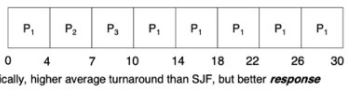


- Average waiting time = (P2+ .... +p4) = 8.2 msec

Example of RR with Time Quantum = 4

Process	Burst Time
P_1	24
P_2	3
P_3	3

- The Gantt chart is:



- Typically, higher average turnaround than SJF, but better ***response***
- q should be large compared to context switch time
- q usually 10ms to 100ms, context switch < 10 usec

تربح تدور كل مدة

* اي شي ما اقسمه waiting time = (Start - arrive) أو ماعي استخدمي الطريقة الاساسية كلهم نفس الجواب

Process	Duration (Burst time)
P1	6 → 3
P2	2 ✓
P3	4 → 1

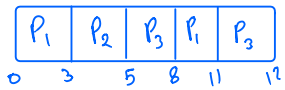
Question 5, ~6 Marks

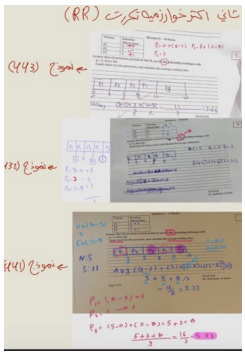
Assume that all processes arrived at time 0, use the RR scheduling technique with q = 3, draw the Gantt chart for the processes, and calculate the average waiting time.

$P_1 = 0 + (8-3)$ $P_2 = 5 + (11-8)$

$P_2 = 3$

نفسه تكرر ٣ مرات





waiting time : $P_1 = 5$ $P_2 = 3$, $P_3 = 8$

Average waiting time : $\frac{16}{3} = 5.33$

---

2. SJF Preemptive (SRTF):

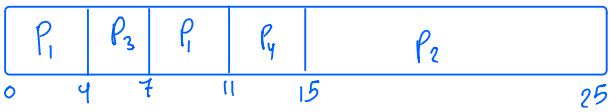
P1: arrival=0, burst=8 4 ← short

P2: arrival=4, burst=10 ←

P3: arrival=4, burst=3 ← short

P4: arrival=10, burst=4

Draw the Gantt chart and calculate average waiting time.



$P_1 = 3$ $P_2 = 11$ $P_3 = 0$ $P_4 = 1$

$= 15/4 = 3.75$

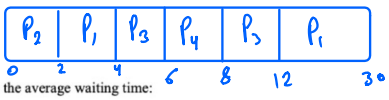
Consider the following process table:

Process	Burst Time	Arrival Time
P1	18 ~~20~~	1
P2	2	0 ← ما انتبهت
P3	4 ~~6~~	4 ← ما انتبهت
P4	2	6

Use the Shortest-Remaining-Time-First (SRTF) algorithm:

a) Draw the Gantt chart for the processes:

Time →



b) Calculate the average waiting time:

P1 waiting time = 9

P2 waiting time = 0

P3 waiting time = 2

P4 waiting time = 0

Average Waiting Time = (P1 + P2 + P3 + P4) / 4 = 11 / 4 = 2.75

---

3. Non-preemptive SJF:

P1(burst=21), P2(burst=3), P3(burst=6), P4(burst=2)

All arrive at time 0.

Draw the Gantt chart and calculate the average waiting time



$P_1 = 11$ $P_2 = 2$ $P_3 = 5$ $P_4 = 0$

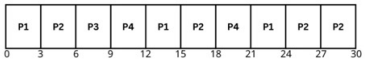
$18/4 = 4.5$

Q5- Consider the following table [6 Marks]

Assume all processes arrived at time 0. Use the Round Robin (RR) scheduling technique with q = 3, draw the Gantt Chart, and calculate the average waiting time.

Process	Duration (Burst time)
P1	9
P2	12
P3	3
P4	6

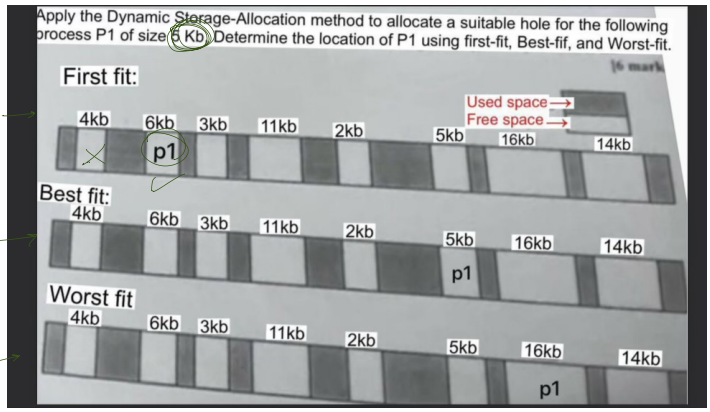
Gantt Chart:



average waiting time:

P1 --> 24 - 9 = 15 : P2 --> 30 - 12 = 18 : P3 --> 9 - 3 = 6 : P4 --> 21 - 6 = 15

$$\frac{15 + 18 + 6 + 16}{4} = 13.5$$



12. If the CPU generated a logical address = 501 and the value of the relocation register = 1320, the physical address will be:

a. 2740	b. 1821	c. 902	d. 819

relocation register + logic = physcal

مثال بسيط

- Base = 100
- Limit = 50

المسموح:

- Logical Address 49 إلى 0 من

لو البرنامج طلب:

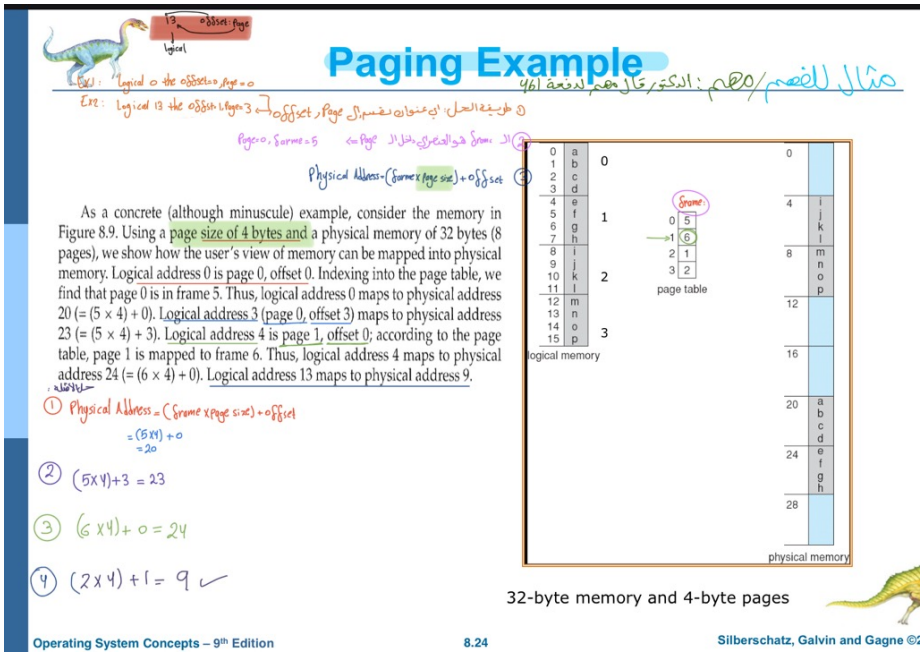
- Logical = 20

✓ Physical = 100 + 20 = 120

لو طلب:

- Logical = 60

❌ خطأ (تعدى limit)



in the FCFS seheduling technique, processes cannot be preempted

Problem Starvation – low priority processes may never execute
Solution Aging – as time progresses increase the priority of the process

Round Robin : Each process gets a small unit of CPU time (time quantum q), usually 10-100 milliseconds

Note 6

Program must be brought (from disk) into memory and placed within a process for it to be run

Main memory and registers are only storage CPU can access directly

Cache sits between main memory and CPU registers

Memory-Management Unit : Hardware device that maps virtual to physical address

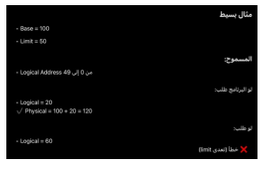
relocation register + logic = physcal

* The user program deals with logical addresses; it never sees the real physical addresses

Backing store – fast disk large enough to accommodate copies of all memory images for all users

Base register contains value of smallest physical address
Limit register contains range of logical addresses – each logical address must be less than the limit register (start from 0)





Reduce external fragmentation by compaction

* MMU maps logical address dynamically

Divide physical memory into fixed-sized blocks called frames

Which of the following has different size?
Program segment

Divide logical memory into blocks of same size called pages

* Dynamic Storage-Allocation Problem.
1- First-fit : first hole that big enough
2- Best-fit : smallest hole " "
3- worst-fit : largest " must also search entire list.

type of page table :
Hierarchical Paging
Hashed Page Tables
Inverted Page Tables

Segmentation : Memory-management scheme that supports user view of memory

Protection
— File owner/creator should be able to control:
• what can be done
• by who
— Types of access
• **Read**
• **Write**
• **Execute**
• **Append (attach)**
• **Delete**
• **List**

Note 7

File Attributes :
Name , Identifier , Type , Location , Size , Protection , Time, date, and user identification

File Operations :
File is an abstract data type , Create , Write , Read , Reposition within file , Delete , Truncate

Operations Performed on Directory :
Search for a file , Create a file , Delete a file , List a directory , Rename a file ,Traverse the file system

Organize the Directory to Obtain :
Efficiency , Naming , Grouping

Information kept in the directory structure

Directory Structure : A collection of nodes containing information about all files

Entity containing file system known as a volume

A file system must be mounted before it can be accessed

Sharing files may be done through a protection scheme

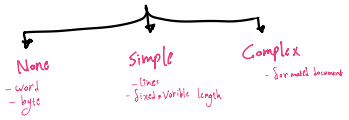
Protection
— File owner/creator should be able to control:
• what can be done
• by whom
— Types of access
• **Read**
• **Write**
• **Execute**
• **Append (attach)**
• **Delete**
• **List**

Note 8

Note 7:

File Concept
- Contiguous logical address space
- Types:
  - Data
    1. numeric
    2. character
    3. binary
  - Program

File structure



Who desied structure?
- OS
- Program

File attributes



File Operations
- File is an abstract data type (ADT is a set of objects together with a set of operations.)
- Create ,Write ,Read ,Reposition within file ,Delete ,Truncate (cut off)

File Locking

- **Mandatory** – access is denied depending on locks held and requested
- **Advisory** – processes can find status of locks and decide what to do

Disk Structure
- Disk can be subdivided into **partitions**
- Disks or partitions can be **RAID** protected against failure
- Disk or partition can be used **raw** – without a file system, or **formatted** with a file system
- Partitions also known as minidisks, slices
- Entity containing file system known as a **volume**
- Each volume containing file system also tracks that file system's info in **device directory** or **volume table of contents**

Operations Performed on Directory
- Search for a file
- Create a file
- Delete a file
- List a directory
- Rename a file
- Traverse the file system (tree search)

Organize the Directory to Obtain :
Efficiency , Naming , Grouping

File System Mounting
- A file system must be **mounted** before it can be accessed
- A unmounted file system (i.e., Fig. 10-11(b)) is mounted at a **mount point**

Protection
- File owner/creator should be able to control:
  - what can be done
  - by whom
- Types of access
  - **Read**
  - **Write**
  - **Execute**
  - **Append (attach)**
  - **Delete**
  - **List**

Information kept in the directory structure
Directory Structure : A collection of nodes containing information about all files
Entity containing file system known as a volume
A file system must be mounted before it can be accessed
Sharing files may be done through a protection scheme

I/O management Important :

1- aspect of computer opreation
2- I/O devices vary greatly
3- Various methods to control them
4- Performance management
5- New types of devices frequent

Incredible variety of I/O devices :

Storage - Transmission - Human-interface

Common concepts – signals from I/O devices interface with computer

Port - Bus - Controller

Connect to various devices

Device drivers encapsulate device details
• Present uniform device-access interface to I/O subsystem

Devices have addresses, used by
Direct I/O instructions
Memory-mappedI/O

Polling For each byte of I/O

1. Read busy bit from status register until 0
2. Host sets read or write bit
3. Host sets command-ready bit
4. Controller sets busy bit
5. Controller clears busy bit when transfer done

Polling can happen in 3 instruction cycles

1- Read status
2- logical-and to extract status bit
3- branch if not zero

interrupts الحل هو ال

—Used for time-sensitive processing, frequent, must be fast

OS writes DMA command block into memory

1- Source and destination addresses
2- Read or write mode
3- Count of bytes
4- Writes location to DMA controller
5- When done, interrupts to signal completion

Bus mastering of DMA controller – grabs bus from CPU

Direct Memory Access
—Used to avoid programmed I/O (one byte at a time) for large data movement
—Requires DMA controller
—Bypasses CPU to transfer data directly between I/O device and memory

Each OS has its own I/O subsystem structures and device driver frameworks

Subtleties of devices handled by device drivers

اختلاف الاجهزة

Devices vary in many dimensions

1- read-write, read only, or write only
2- Speed of operation
3- Sequential or random-access
4- Character-stream or block
5- Sharable or not

Broadly I/O devices can be grouped by the OS into

1- Block I/O
2- Character I/O
3- Memory-mapped file access
4- Network sockets

Subtleties of devices handled by device drivers

Block devices include disk drives

Commands include read, write, seek

Character devices include keyboards, mice, serial ports

~~Commands include get(), put()~~

Blocking : process suspended until I/O completed

1- Easy to use
2- Insufficient for some needs

Nonblocking : I/O call returns as much as available

1- Implemented via multi-threading
2- Returns quickly

Asynchronous : process runs while I/O executes

1- Difficult to use
2- I/O subsystem signals process when I/O completed

Kernel I/O Subsystem :

1- Scheduling : some I/O request ordering via per-device queue
2- Buffering : store data in memory while transferring between devices
3- Caching : faster device holding copy of data
4- Spooling - hold output for a device
5- Device reservation : provides exclusive access to a device

Error Handling : OS can recover from disk read - device unavailable - transient write failures

Kernel keeps state info for I/O components, including open file tables - network connections - character device state

STREAM : a full-duplex communication channel between a user-level process and a device in Unix System V and beyond

Each module contains a read queue and a write queue

A STREAM consists of:

1- STREAM head interfaces with the user process

2- driver end interfaces with the device

3- zero or more STREAM modules between them

List five:

Improving I/O Performance :

1. **Reduce number of context switches**
2. **Reduce data copying**
3. **Reduce interrupts**
4. **Use DMA**
5. **Use smarter hardware devices**

Note 9

Protection and Security

— Protection – any mechanism for controlling access of processes or users to resources defined by the OS

— Security – defense of the system against internal and external attacks

Goals of Protection

Each object has a unique name and can be accessed through a well-defined set of operations

Capability list for domain is list of objects together with operations allows on them

Each object has list of unique bit patterns, called locks

Each domain as list of unique bit patterns called keys

Principles of Protection :

- Principle of least privilege: Programs, users, and systems should be given just enough privileges to perform their tasks.



- Limits damage and can be static or dynamic

- Grain aspect: Rough-grained privilege management is easier but less protective, while fine-grained management is more complex but more protective.

Access Matrix : View protection as a matrix Rows represent domains Columns represent objects

- is a table that defines access permissions between specific subject and object ?

Ans: Access matrix

- View Protection as a matrix Rows Represented domains, Columns represented objects?

Ans: Access matrix

Implementation of Access Matrix :

Option 1 – Global table

Option 2 – Access lists for objects

Option 3 – Capability list for domains

Capability list for domain is list of objects together with operations allows on them

Option 4 – Lock-key

Process in a domain can only access object if domain has key that matches one of the locks

Comparison of Implementations :

- **Global Table** → ✅ Simple | ❌ Large
- **Access List** → ✅ Object-focused | ❌ Domain search difficult
- **Capability List** → ✅ Domain-focused | ❌ Object search difficult
- **Lock-Key** → ✅ Flexible | ❌ Complex

1- Global table is simple, but can be large

2- Access lists correspond to needs of users

Determining set of access rights for domain non-localized so difficult

Every access to an object must be checked

3- Capability lists useful for localizing information for a given process

4- Lock-key effective and flexible, keys can be passed freely from domain to domain, easy revocation

Most system use Combine of access list and cability.

Revocation of Access Rights :

Various options to remove the access right of a domain to an object :

Immediate vs delayed

Selective vs general

Partial vs total

Temporary vs permanent

Access List – Delete access rights from access list

Capability List – Scheme required to locate capability in the system before capability can be revoked

Note 10

– System secure if resources used and accessed as intended under all circumstances

Threat is potential security violation

Attack is attempt to breach security , can be accidental or malicious

Digital Certificates : Proof of who or what owns a public key

برنامج يبدأ حادثة أمنية تحت ظروف معينة

Logic Bomb : Program that initiates a security incident under certain circumstances

Cryptography : Means to constrain potential senders (sources) and/or receivers(destinations) of messages

Note symmetric cryptography based on transformations, asymmetric based on mathematical functions

Security Violation Categories :

1- Breach of confidentiality : Unauthorized reading of data
2- Breach of integrity : Unauthorized modification of data
3- Breach of availability : Unauthorized destruction of data
4- Theft of service : Unauthorized use of resources
5- Denial of service (DOS) : Prevention of legitimate use

Security Violation Methods :

1- Masquerading : Pretending to be an authorized user to escalate privileges
2- Replay attack : As is or with message modification
3- Man-in-the-middle attack : Intruder sits in data flow, masquerading as sender to receiver and vice versa
4- Session hijacking : Intercept an already-established session to bypass authentication

Security must occur at four levels to be effective :

1- Physical => servers
2- Human => Avoid social engineering
3- Operating System => Debugging
4- Network => DoS

Program threats :

1- Trojan Horse
2- Trap Door
3- Logic Bomb
4- Stack and Buffer Overflow
5- Viruses
6- Virus dropper

categories of viruses :

1- Boot / memory
2- Macro
3- Source code
4- Stealth
5- Multipartite

System and Network Threats :

1- Worms
2- Port scanning
3- Denial of service

What the difference between encryption and description

Encryption for generating ciphertexts from messages

Decryption for generating messages from ciphertexts

Explain Authentication تعيين المرسلين المحتملين للرسالة Constraining set of potential senders of a message

Algorithm components :

set K of keys , set M of messages , set A of authenticators

Function S for generating authenticators from messages

Function V for verifying authenticators on messages

1- Signature-based : detection spots known bad patterns

2- Anomaly detection : spots differences from normal behavior

Can detect zero-day attacks

3- False-positives and false-negatives a problem

Note 11:

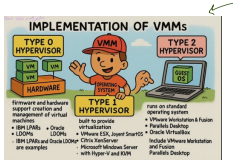
— Fundamental idea – abstract hardware of a single computer into several different execution environments » Virtual machine

Host – underlying hardware system

• Virtual machine manager (VMM) or hypervisor – creates and runs virtual machines by providing interface that is identical to the host



• Guest – process provided with virtual copy of the host
- Usually an operating system



Implementation of VMMs

Type 0	Type 1	Type 2
VM VM hardware	VM VM hypervisor hardware	VM VM hypervisor program operating system hardware
firmware and hardware support creation and management of virtual machine	- Operating system-like software built to provide virtualization	Application that run on standard OS but provide VMM features to guest OS. - Commonly used in Computer.
- IBM LPARs - Oracle LDOMs	- VMware ESX - Joyent SmartOS - Microsoft Windows Server	- Parallels Desktop. - Oracle VirtualBox.

• Paravirtualization - Technique in which the guest operating system imodified to work in cooperation with the VMM to optimize performance Does not fit the definition of virtualization

Programming-environment virtualization - Ex: Java Virtual Machine (JVM

Emulators – Allow applications written for one hardware environment to run on a very different hardware environment,

• Application containment - Not virtualization at all but rather provides virtualization-like features by segregating applications from the operating system, making them more secure, manageable ?

Only one kernel running – host OS

Clone by creating copy and running both original and copy

Live migration – move a running VM from one host to another!

in Xen;

Each device has circular buffer shared by guest and VMM via shared memory

Guest uses hypercall (call to hypervisor) when page-table changes needed

overcommitment
- VMM can use standard scheduling algorithms to put threads on CPUs
- Some add fairness aspect

Note 12

Processors variously called nodes, computers, machines, hosts

• Siteis location of the processor

Reasons for Distributed Systems

- Resource sharing
- Communication
- Computing speed-up
- Smaller computers

Types of Distributed Operating Systems

— Network Operating Systems

— Distributed Operating Systems

Network-Operating Systems

— Users are aware of multiplicity of machines

— Access to resources of various machines is done explicitly by:

- Remote loging
- Remote Desktop
- Transferring Data from Remote to Local machines

Distributed-Operating Systems

— Users not aware of multiplicity of machines

• Access to remote resources similar to access to local resources

— Data Migration – transfer data by transferring entire file, or transferring only those portions of the file necessary for the immediate task

— Computation Migration – transfer the computation, rather than the data, across the syste

Distributed-Operating Systems

— Process Migration – execute an entire process, or parts of it, at different sites

• Load balancing

• Computation speedup

• Hardware preference

• Software preference

• Data access

— Local-Area Network (LAN) – designed to cover small geographical area

• Multiple topologies like star or ring

Wide-Area Network (WAN) – links geographically separated sites

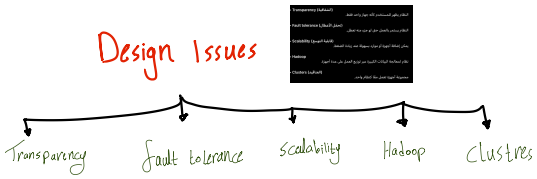
Point-to-point connections over long-haul lines

Communication Structure

The design of a communication network must address four basic issues:

- Naming and Name resolution
- Routing Strategies
- Connection Strategies
- Contention

— Every host has a unique IP address and a corresponding Ethernet Media Access Control (MAC) address



— Distributed file system (DFS) – a distributed implementation of the classical time-sharing model of a file system, where multiple users share files and storage resources

— Naming– mapping between logical and physical objects

— Multilevel mapping– abstraction of a file that hides the details of how and where on the disk the file is actually stored

— A transparent DFS hides the location where in the network the file is stored

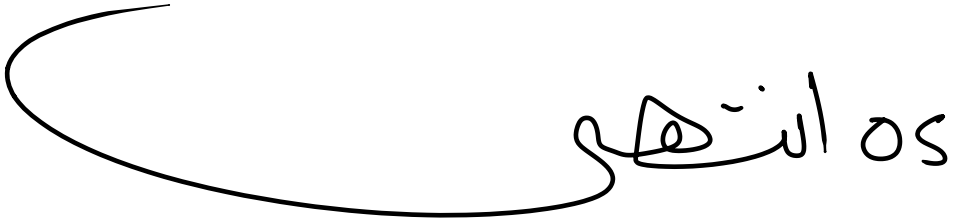
Naming Structures

— Location transparency

— Location independence

Cache-consistency problem– keeping the cached copies consistent with the master file

* Client-initiated approach
* Server-initiated approach



"الحمدلله على ما جرى قضاها الله"