CS222 final exam 462

انا هذا الي اتذكره بس حوسه
المقالي كله تجميعات

1-security, protection
2-least privilege
3-lo performance
4- security violation categories
5- signature based , behaviour based.

الا سؤالين
سؤال عرفي ال mmu عليه ٣ درجات
سؤال عددي السيكورتي لفلز كل واحد مع مثال له عليه ٤ درجات

المسأله sjf non premtiv

وحق البيست والوورست فيت

الاختياري والصح والخطا عليهم ١٠ درجات
الاختياري:
1-تعريف Paravirtualization
2-تعريف man in the middle
3- each Ip adress have corresponding >media access control(mac) address.
4- simplest way to pass parameters > register.
5-LAN definition

الصح والخطا 5
1- System secure if resources used and accessed as intended under all circumstances.T
2- logical address can access / locate the physical address directly without the need of MMU . F
3 - PROCESS must be brought from disk
4- multilevel page table reduce memory usage
5- all OS have the same IO sub system . F

19, 3:18 PM مايو

خيارات:
اسهل طريقة لتمرير البراميترز
مو نموذج Passing parameters in regester

مو نموذج MCI اخو الTCP و الIP

نموذج Man in the middle

تعريف بارافيرتولايزنق او شي زي كذا الي يستعمله تايب ون من الvmm
مو نموذج

مو نموذج LAN

صح وخطا:
ان السكيورتي يكون كويس اذا الريسورزسز زي ما هم طول الوقت او شي زي كذا مو نموذج

اننا نقدر نحول العنوان المنطقي للعنوان الفيزيائي بدون MMU
مو نموذج

تعريف الmultilevel page table او شي زي كذا ما اعرفها بس كان واضح انها خطا مو نموذج

ان كل الانبوت والاوتبوت سيستم يتشابهون بالانظمه نموذج بس مع تريك

ان البروقرام يجي من الديسك لين الميموريي عشان يشتغل او شي زي كذا من نموذج

مقالي:
اول شي الي من النماذج:
معادلة النون بريمتف سجف
معادلة الفيرست بيست ورست
فرق بين سكورتي وبروتكشن
كيف تحسنين الانبوت والاوتبوت
تعداد انواع التعدي على السكيورتي الي هو تعدي على CIA واخوانه
فرق السقنتشر بايسد والبهايفير
شرح الليست بريفلج
يمكن جاء الفرق بين فيزيكال ولوجيكال مدري ما اذكر

اثنين مو من النماذج:
سكيورتي لفلز ومثال لكل واحد
تعريف MMU

19, 3:27 PM مايو



# 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
  - Huge range, including denial-of-service, worms, viruses, identity theft, theft of service
- Systems generally first distinguish among users, to determine who can do what
  - User identities (**user IDs**, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (**group ID**) allows set of users to be defined and controls managed, then also associated with each process, file
  - **Privilege escalation** (increase) allows user to change to effective ID with more rights





# Principles of Protection

- Guiding principle – **principle of least privilege**
  - Programs, users and systems should be given just enough **privileges** to perform their tasks
  - Limits damage if entity has a bug, gets abused
  - Can be static (during life of system, during life of process)
  - Or dynamic (changed by process as needed) – **domain switching**, **privilege escalation**
  - "Need to know" a similar concept regarding access to data
- Must consider "grain" aspect
  - Rough-grained privilege management easier, simpler, but least privilege now done in large chunks
    - For example, traditional Unix processes either have abilities of the associated user, or of root
  - Fine-grained management more complex, more overhead, but more protective
    - File ACL lists, RBAC
- Domain can be user, process, procedure



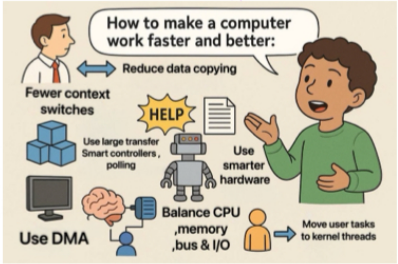


# How to Improving Performance in I/O?

مهم جدًا تعداد/مهم/مهم مقالي

- Reduce number of context switches
- Reduce data copying
- Reduce interrupts by using large transfers, smart controllers, polling
- Use DMA
- Use smarter hardware devices
- Balance CPU, memory, bus, and I/O performance for highest throughput
- Move user-mode processes / daemons to kernel threads

From Tala ملخص تيلا








List Dive of

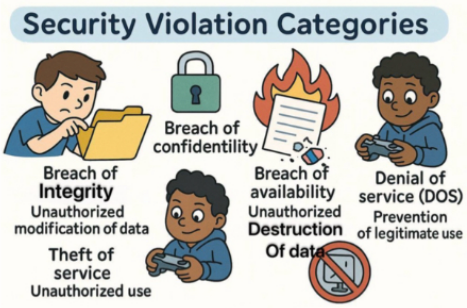
# Security Violation Categories

If system has CIA it will Be Secure

- **Breach of confidentiality** — C → قراءة البيانات بدون تصريح
  - Unauthorized reading of data
- **Breach of integrity** — I → تعديل البيانات بدون تصريح
  - Unauthorized modification of data
- **Breach of availability** — A → تدمير الوصول هنا تدمير البيانات
  - Unauthorized destruction of data
- **Theft of service** — سرقة الموارد
  - Unauthorized use of resources
- **Denial of service (DOS)** — منع المستخدمين الشرعيين من استعمال الخدمات
  - Prevention of legitimate use

MCQ

From Tala ملخص الويب








# Implementing Security Defenses

مهم الدكتور قال لازم تحفظ / مهم جدا

- **Defense in depth** is most common security theory – multiple layers of security
- Security policy describes what is being secured
- Vulnerability assessment compares real state of system / network compared to security policy
- Intrusion detection endeavors to detect attempted or successful intrusions
  - **Signature-based** detection spots known bad patterns — يكشف الهجمات بناءً على انماط معروفة مسبقًا
  - **Anomaly detection** spots differences from normal behavior — يكشف اي انحراف لسلوك النظام
    - Can detect **zero-day** attacks
  - **False-positives** and **false-negatives** a problem
- Virus protection
- Auditing, accounting, and logging of all or specific system or network activities

مهم جدا

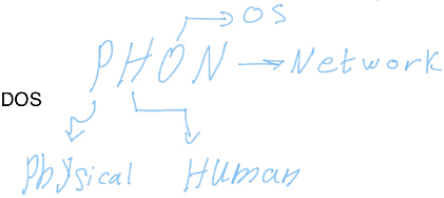




# Security Measure Levels

- Impossible to have absolute security, but make cost to perpetrator sufficiently high to deter most intruders
- Security must occur at four levels to be effective:
  - **Physical**
    - Data centers, servers, connected terminals
  - **Human**
    - Avoid **social engineering**, **phishing**, **dumpster diving**
  - **Operating System**
    - Protection mechanisms, debugging
  - **Network**
    - Intercepted communications, interruption, DOS
- Security is as weak as the weakest link in the chain
- But can too much security be a problem?

أعرفوا الفرق بينهم








# Implementation of VMMs (cont.)

- Other variations include:
  - **Paravirtualization** - Technique in which the guest operating system is modified to work in cooperation with the VMM to optimize performance
  - **Programming-environment virtualization** - VMMs do not virtualize real hardware but instead create an optimized virtual system
    - Used by Oracle Java and Microsoft.Net
  - **Emulators** – Allow applications written for one hardware environment to run on a very different hardware environment, such as a different type of CPU
  - **Application containment** - Not virtualization at all but rather provides virtualization-like features by segregating applications from the operating system, making them more secure, manageable
    - Including Oracle Solaris Zones, BSD Jails, and IBM AIX WPARs
- Much variation due to breadth, depth and importance of virtualization in modern computing





# Security Violation Methods

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










# Example: TCP/IP

- The transmission of a network packet between hosts on an Ethernet network
- Every host has a unique IP address and a corresponding Ethernet **Media Access Control** (**MAC**) address
- Communication requires both addresses
- **Domain Name Service** (**DNS**) can be used to acquire IP addresses
- **Address Resolution Protocol** (**ARP**) is used to map MAC addresses to IP addresses
  - **Broadcast** to all other systems on the Ethernet network
- If the hosts are on the same network, ARP can be used
  - If the hosts are on different networks, the sending host will send the packet to a router which routes the packet to the destination network





# Memory-Management Unit (MMU)

mca

- Hardware device that maps virtual to physical address
  - In MMU scheme, the value in the relocation register is added to every address generated by a user process at the time it is sent to memory

Does the user program see physical addresses?
No, it only sees logical addresses.

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

قطعة هاردوير تحول العناوين المنطقية الى فيزيائية
والبرنامج ما يدري عن العناوين الفيزيائية لأن الـ MMU
تحول العناوين تلقائيًا فهو يتعامل مع عناوين منطقية





# System Call Parameter Passing

- Often, more information is required than simply identity of desired system call
  - Exact type and amount of information vary according to OS and call
- Three general methods used to pass parameters to the OS
  - Simplest: pass the parameters in *registers*
    - In some cases, may be more parameters than registers
  - Parameters stored in a *block*, or table, in memory, and address of block passed as a parameter in a register
    - This approach taken by Linux and Solaris
  - Parameters placed, or *pushed*, onto the *stack* by the program and *popped* off the stack by the operating system
  - Block and stack methods do not limit the number or length of parameters being passed





# Network Structure

- **Local-Area Network (LAN)** – designed to cover small geographical area
  - Multiple topologies like star or ring
  - Speeds from 1Mb per second (Appletalk, bluetooth) to 40 Gbps for fastest Ethernet over twisted pair copper or optical fibre
  - Consists of multiple computers (mainframes through mobile devices), peripherals (printers, storage arrays), routers (specialized network communication processors) providing access to other networks
  - Ethernet most common way to construct LANs
    - Multiaccess bus-based
    - Defined by standard IEEE 802.3
  - Wireless spectrum (**WiFi**) increasingly used for networking
    - I.e. IEEE 802.11g standard implemented at 54 Mbps





# Background

- 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
  - Register access in one, CPU clock (or less)
  - Main memory can take many cycles
  - **Cache** sits between main memory and CPU registers
  - Protection of memory required to ensure correct operation





# The Security Problem

- System **secure** if resources used and accessed as intended under all circumstances
  - Unachievable
- Intruders (crackers) attempt to breach security
- **Threat** is potential security violation
- **Attack** is attempt to breach security
- Attack can be accidental or malicious
- Easier to protect against accidental than malicious misuse





# Application I/O Interface

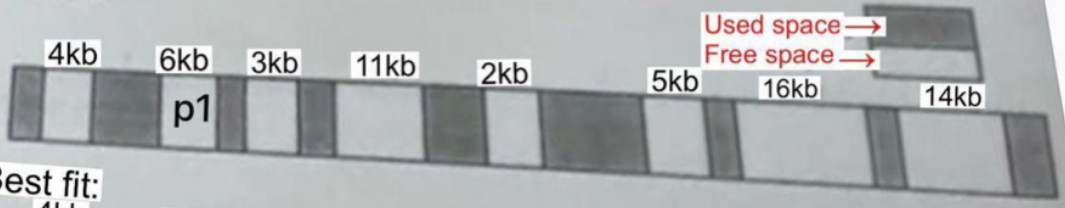
- I/O system calls encapsulate device behaviors in generic classes
- Device-driver layer hides differences among I/O controllers from kernel
- New devices talking already-implemented protocols need no extra work
- Each OS has its own I/O subsystem structures and device driver frameworks
- Devices vary in many dimensions
  - **Character-stream** or **block**
  - **Sequential** or **random-access**
  - **Synchronous** or **asynchronous** (or both)
  - **Sharable** or **dedicated**
  - **Speed of operation**
  - **read-write**, **read only**, or **write only**



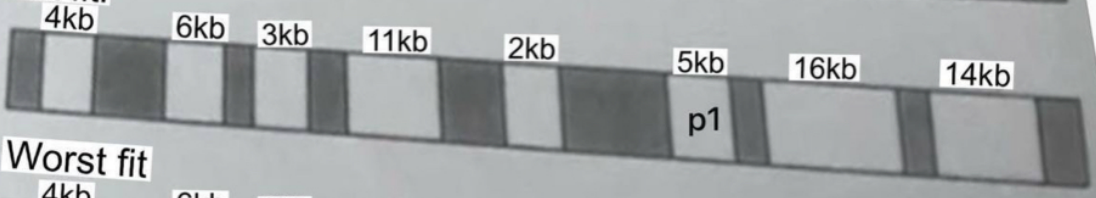
Apply the Dynamic Storage-Allocation method to allocate a suitable hole for the following process P1 of size 5 Kb, Determine the location of P1 using first-fit, Best-fif, and Worst-fit.

[6 mark

First fit:




Best fit:




Worst fit



# 3rd Case : SJF (short job first) non-Preemptive

**Draw Gantt Chart and calculate the average waiting time using the given table ??**

| Process | Burst Time | Arrival Time |
|---|---|---|
| P2 | 12 | 0 |
| P3 | 8 | 3 |
| P4 | 4 | 5 |
| P1 | 10 | 10 |
| P5 | 6 | 12 |

Waiting time : start time – arrival time

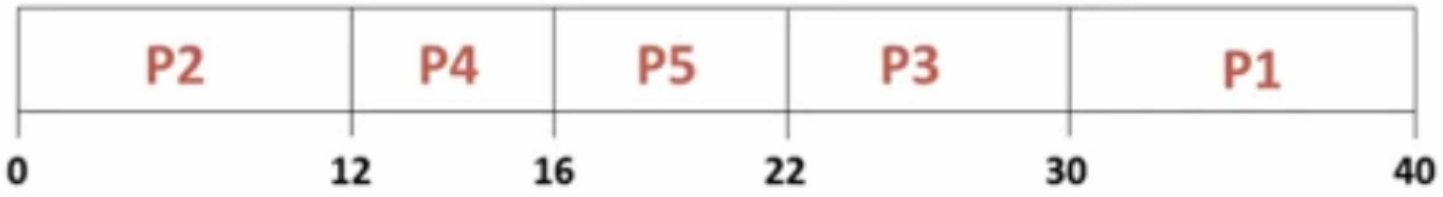
P1 = 30 – 10 = 20

P2 = 0 – 0 = 0

P3 = 22 – 3 = 19

P4 = 12 – 5 = 7

P5 = 16 – 12 = 4




**Average waiting time = (20 + 0 + 19 + 7 + 4) / 5 = 50 / 5 = 10**