

Implementation of Access Matrix

مهم: الدكتور قالها لدورة 471

أعرفوا الفرق بينهم

- Generally, a sparse matrix
- Option 1 – Global table → جدول يخزن الصلاحيات كلها لكنه يحتاج مساحة كبيرة في التخزين والذاكرة
  - Store ordered triples < *domain, object, rights-set* > in table
  - A requested operation M on object O_j within domain D_i -> search table for < D_i, O_j, R_k >
    - with M ∈ R_k
  - But table could be large -> won't fit in main memory
  - Difficult to group objects (consider an object that all domains can read)
- Option 2 – Access lists for objects → قائمة وصول خاصة بالأوبجكتس ، كل أوبجكت له نوع: صلاحيات توضعا / تقدر تضيف بالإمكان اختراقهما، سجلة
  - Each column implemented as an access list for one object
  - Resulting per-object list consists of ordered pairs < *domain, rights-set* > defining all domains with non-empty set of access rights for the object
  - Easily extended to contain default set -> If M ∈ default set, also allow access

1:

is a table that defines access Permissons between specific subjects and objects?

Ans: Access matrix

→ السؤال جاء بنفوذ 4529 خيارات التعريف موحد ونوع

OV:

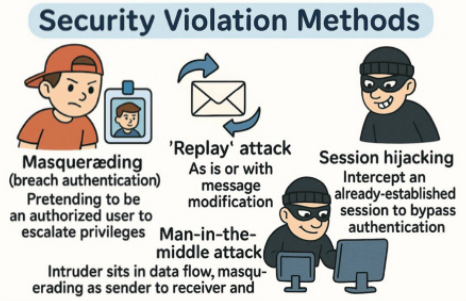
View Protection as a matrix Rows Represent domains, Columns represent objects? Ans: Access matrix





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







Overview

4:
MCQ

تقسيم الجهاز الواحد اي خدة بيئات مختلفة متوهمك جهاز واحد تقدر تشغل ويندوز ولينكس وماك على نفس الجهاز

- Fundamental idea – abstract hardware of a single computer into several different execution environments → Virtual machine
  - Similar to layered approach
  - But layer creates virtual system (**virtual machine**, or **VM**) on which operation systems or applications can run

الطبقة تعني الجهاز الواحد بيشتغل انظمة أو برامج كأنها اجهزة منفصلة بدون ما تتداخل مع بعض

- Several components

الجهاز الحقيقي

  - **Host** – underlying hardware system
  - **Virtual machine manager** (**VMM**) or **hypervisor** – creates and runs virtual machines by providing interface that is ***identical*** to the host

برنامج يقدم واجهة عمل مطابقة للجهاز الحقيقي

    - (Except in the case of paravirtualization)
  - **Guest** – process provided with virtual copy of the host
    - Usually an operating system

النظام او العملية اللي تشتغل على نسخة من الـ host
بالغالب يكون الـ OS

- Single physical machine can run multiple operating systems concurrently, each in its own virtual machine





List like 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 مثل 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

5: MCQ

From Tala ملخص تيلا







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





Program Threats (Cont.)

- **Logic Bomb**
  - Program that initiates a security incident under certain circumstances
- **Stack** and **Buffer Overflow**
  - Exploits a bug in a program (overflow either the stack or memory buffers)
  - Failure to check bounds on inputs, arguments
  - Write past arguments on the stack into the return address on stack
  - When routine returns from call, returns to hacked address
    - Pointed to code loaded onto stack that executes malicious code
  - Unauthorized user or privilege escalation





Semaphore Usage

- **Counting semaphore** – integer value can range over an unrestricted domain
- **Binary semaphore** – integer value can range only between 0 and 1
  - Then a **mutex lock**
- Can implement a counting semaphore ***S*** as a binary semaphore
- Can solve various synchronization problems
- Consider P_1 and P_2 that require S_1 to happen before S_2

```
P1:
    S1;
    signal(synch);
P2:
    wait(synch);
    S2;
```





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





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**





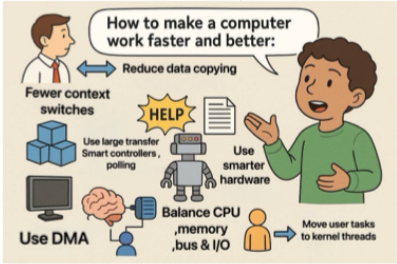
How to Improving Performance in I/O?

مهم جدًا تعداد/مهم/مهم متلي

I. list Five:

- 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







Protection and Security

2. Define:

- **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





Implementing Security Defenses

3. Defference:

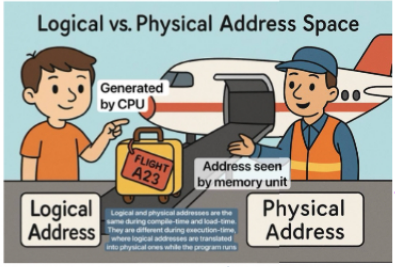
- **Defense in depth** is most common security theory – multiple layers of security
- Security policy describes what is being secured
- Vulnerability assessment compares real state of system / network compared to security policy
- Intrusion detection endeavors to detect attempted or successful intrusions
  - **Signature-based** detection spots known bad patterns
  - **Anomaly detection** spots differences from normal behavior
    - 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



Logical vs. Physical Address Space

1. Defference:

- The concept of a logical address space that is bound to a separate **physical address space** is central to proper memory management
  - **Logical address** – generated by the CPU; also referred to as **virtual address**
  - **Physical address** – address seen by the memory unit
- Logical and physical addresses are the same in compile-time and load-time address-binding schemes; logical (virtual) and physical addresses differ in execution-time address-binding scheme



Physical address = logical address : load time & Compile time بال-1

Physical address ≠ logical address: Execution time بال-2

From Tala

Principles of Protection

2. Explain :

- 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





Operations Performed on Directory

١٠ list Five: / مهم / تعداد جدا مهم

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

From: Tala ملخص الطب



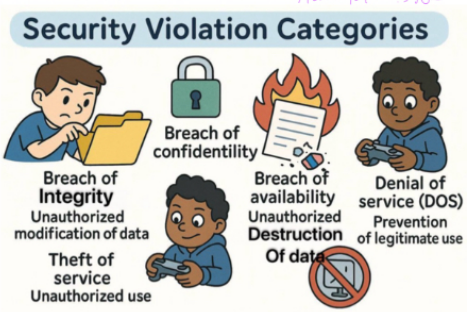




Security Violation Categories

2. list Five:

- **Breach of confidentiality**
  - Unauthorized reading of data
- **Breach of integrity**
  - Unauthorized modification of data
- **Breach of availability**
  - Unauthorized destruction of data
- **Theft of service**
  - Unauthorized use of resources
- **Denial of service (DOS)**
  - Prevention of legitimate use





1. Consider the following table:

Process	Duration (Burst time)	Priority
P1	21	2
P2	3	1
P3	6	4
P4	2	3

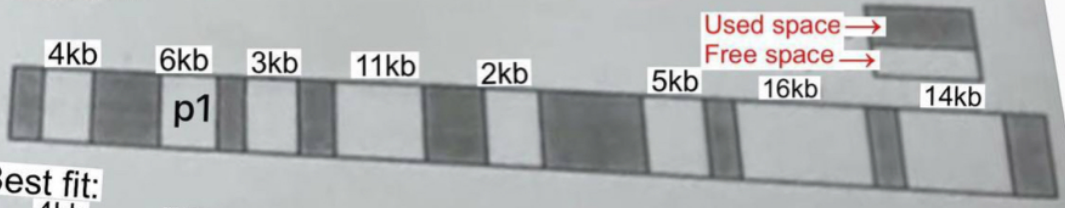
Assume that all processes arrived at time 0, use the Non-Preemptive Priority scheduling algorithm, draw the Gantt chart for the processes, and calculate the average waiting time.

[6 marks]

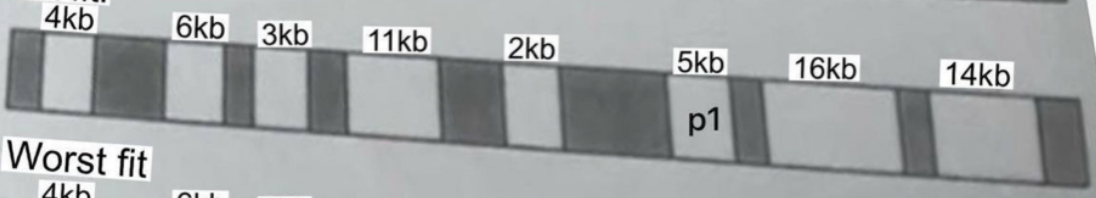
Apply the Dynamic Storage-Allocation method to allocate a suitable hole for the following process P1 of size 5 Kb, Determine the location of P1 using first-fit, Best-fif, and Worst-fit.

[6 mark

First fit:



Best fit:



Worst fit

