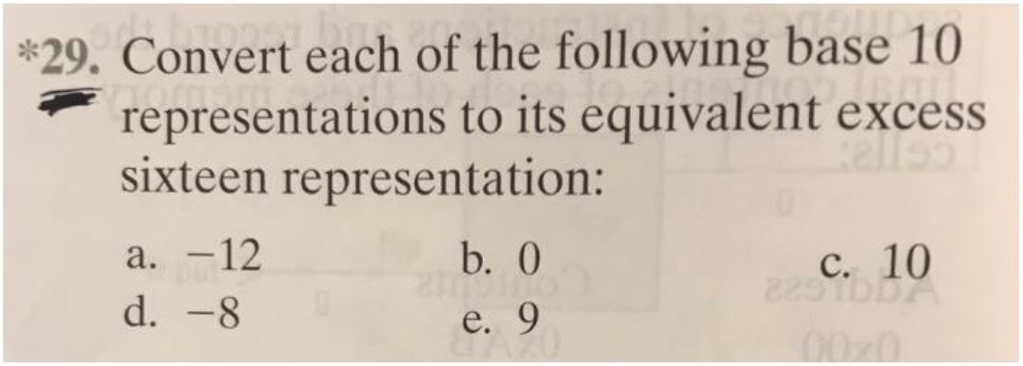
cs03

Main Memory

- Cell: A unit of main memory
  - (typically **8 bits** which is one **byte**)
- **Address** (記憶體位址)
  - CPU知道要存取的是哪塊記憶體
- **Value** (記憶體所存的值)
  - CPU真正要處理的值

hw02

#29



*29. Convert each of the following base 10 representations to its equivalent excess sixteen representation:

a. −12 b. 0 c. 10
d. −8 e. 9

Answer:

29)10 分 每個 2

A	00100
B	10000
C	11010
D	01000
E	11001

#36

Convert 85.125 from Decimal to IEEE 754 Floating Point Representation (**single precision 32 bits**)

Answer:

36) 6 分

0 1000 0101 0101 0100 1000 0000 0000 000

hw3

#7

Identify a logical operation (along with a corresponding mask) that, when applied to an input string of 8 bits, produces an output string of all 0s if and only if the input string is 10000001.

7. 兩種不同答案皆可 6 分

Ans1: XOR 10000001

Ans2: AND 00000000

#9

Summarize the difference between a CISC architecture and a RICS architecture.

9. 8 分

CISC	RISC
Emphasis on hardware	Emphasis on software
Multiple instruction sizes and formats	Instructions of same set with few formats
Less registers	Uses more registers
More addressing modes	Fewer addressing modes
Extensive use of microprogramming	Complexity in compiler
Instructions take a varying amount of cycle time	Instructions take one cycle time
Pipelining is difficult	Pipelining is easy

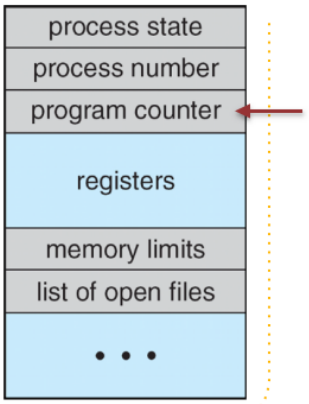
cs06

A program vs. a process

- **Program**:
  - a set of instructions
- **Process**:
  - the **activity of executing a program**
  - **Process Status table**
- A program can be run multiple times, each instance/activity called a process
- **Interprocess communication (IPC)**
  - The communication between processes from running one or more programs

Context (process state)

- **Process Control Block** (PCB)
- Snapshot of the current status of a process
  - A process **identifier**, or **PID**
  - Register values, **Program Counter** value
  - The memory space, I/O, files for the process
  - **State of the process**
    - Ready: ready for execution.
    - Waiting: waiting for some I/O.
    - Complete: finished process.



Process Control Block

Deadlock

- Processes **block** each other from continuing because each is waiting for a resource that **is allocated to another**
- Conditions required for **deadlock**
  - 1. **Mutual exclusion**:
    - Competition for non-sharable resources
  - 2. **Hold and wait**:
    - Resources requested on a **partial basis**
  - 3. **No preemption** :
    - An allocated resource can not be **forcibly retrieved**
  - 4. **Circular wait**

Remove any one of the conditions can resolve the deadlock

hw4

#3
What is virtual memory?

- **Virtual memory**:
  - Employ the physical memory and disk space
  - Create the **illusion** of a larger memory space
  - To facilitate the mapping, memory is grouped into **pages** (the basic memory unit).
  - Paging:
    - shuffle **pages** between main memory and disk.

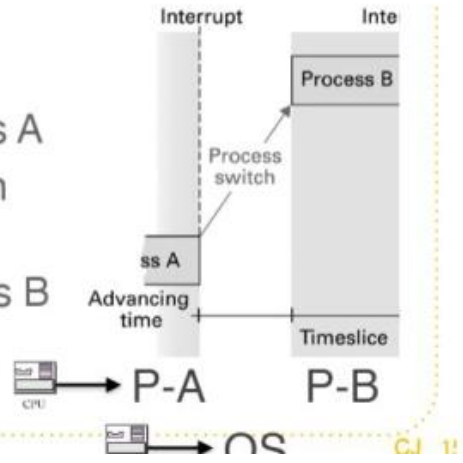
#7
What is a context switch?

7.令多個 process 能共享單一 CPU 資源

#8
Summarize the steps performed by the CPU when an interrupt occurs.

The steps of Context Switch

- 1. Get an **interrupt** from timer
- 2. Go to the interrupt handler
  - Save the context of process A
  - Find a process **ready** to run (Assume that is process B)
  - **Load** the context of process B
- 3. Start (continue) process B



Carrier Sense Multiple Access (CSMA) Algorithm

- Carrie Sense
  - 在傳送資料之前會**先偵測網路媒體**中是否有訊號傳遞
  - **送出後也一直偵測**是否有人也傳輸
- Multiple Access
  - 同時會有**多個網路節點**連接到這個網路上
    - 為了能夠指定資料是給哪一個節點**接收**，
    - 為了讓接收者能夠知道這份資料是由誰所**發出**，
    - 所以乙太網路使用 **48 bits** 的資料作為**定址**

CSMA/CD & CA

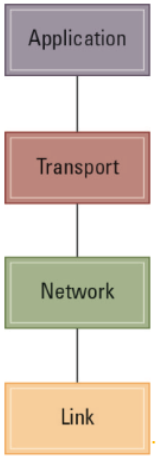
- CSMA/CD (**Collision Detection**)
  - used in **Ethernet**
  - both machines **stop and wait** for a independent, **random** time
- CSMA/CA (**Collision Avoidance**)
  - used in **WiFi**, where **not all machines** can hear each other
    - **Hidden terminal problem**
  - give advantage to the machine that has already been waiting

Transmission Control Protocol (TCP)

- Transport layer protocols
- Provides services to the **application layer** and receives services
- A **process-to-process** connection
- **Reliable** delivery
- **Connection-oriented** delivery
  - **three-way handshake**
- **Flow control** and **congestion control**
- User Datagram Protocol (UDP)
  - unreliable and Connectionless protocol

因為IP對於封包
是不可靠的
不保證封包一定會送達
不保證送達順序

Internet Software Layers



- **Application**:
  - Constructs message with address
- **Transport**:
  - Chops message into packets
- **Network**:
  - Handles routing through the Internet
- **Data Link**:
  - Handles actual transmission of packets

cs09

Domain name service (DNS)

- To identify an entity (**Host or Server**)
  - TCP/IP protocols use the **IP address**
- A **directory** system that can **map** a **name** to an **address**
- **DNS is a protocol** that can be used in different platforms.
- The domain name space (tree) was originally divided into three different sections:
  - **generic** domains,
  - **country** domains
  - **inverse** domain

hw05

#1
What are some differences between a transport layer based on the TCP protocol and another based on the UDP protocol?

#2

1. 12分

	TCP	UDP
可靠性	可靠	不可靠
速度	慢	快
傳輸方式	封包按順序傳輸	封包以串流方式傳輸
錯誤檢查與修正	有	無
壅塞控制	有	無
確認	有	只有檢查碼
適用服務	要求可靠傳輸的服務，例如電子郵件、網頁瀏覽、檔案傳輸	即時服務，例如串流媒體、網路電話、網路遊戲

#2
What is DNS?

DNS: domain name system

people: many identifiers:
- SSN, name, passport #

Internet hosts, routers:
- IP address (32 bit) - used for addressing datagrams
- "name", e.g., www.yahoo.com - used by humans

Q: how to map between IP address and name, and vice versa ?

Domain Name System:
- *distributed database* implemented in hierarchy of many *name servers*
- *application-layer protocol:* hosts, name servers communicate to *resolve* names (address/name translation)
  - note: core Internet function, implemented as application-layer protocol
  - complexity at network's "edge"

#8
In what way could TCP be considered a better protocol for implementing the transport layer than UDP? In what way could UDP be considered better than TCP?

8. 12分
TCP:需要高穩定時使用，資料不能接受遺失:例如文件傳輸

UDP:需要快速傳輸且不太需要穩定:例如影像傳輸

Maintaining data **integrity**

- Simultaneous access problems
  - Incorrect summary problem
  - Lost update problem
- **Locking**
  - preventing others from accessing data being used by a transaction
    - **Shared lock**: used when reading data
    - **Exclusive lock**: used when altering data

Locking

- Preventing others from accessing data being used by **a transaction**
  - **Shared lock (Read lock)**
    - used when reading data
    - Only **read**
  - **Exclusive lock (Write lock)**
    - used when altering data (Update data)
    - **No one can access**

Transaction: **ACID**

- 原子性 **Atomicity**
  - 所有操作，或者**全部完成**，或者**全部不完成**，不會結束在中間某個環節
- 一致性 **Consistency**
  - 結束以後，資料庫的完整性沒有**被破壞**
- 事務隔離 **Isolation**
  - 防止多個事務並發(同時)執行時由於交叉執行而導致數據的**不一致**
- 持久性 **Durability**
  - 對數據的修改就是**永久**的

Transaction Log (Journal)

- A **non-volatile record** of each transaction's activities
  - **Commit point**:
    - The point at which a transaction has been recorded in the log
  - **Roll-back**:
    - The process of undoing a transaction