

Tutorial 1 Discussions

Please use the following format to present your answers:

Question X - your name

Xxx
xx
xx

Question 1 - Foo Hui Yi - ok

- 1a) Computer system can be used as a **Point of Sales (POS) system** to process customer's orders in an **organized manner** so that workers can prepare food efficiently.
- b) Computer system can be used to **process customer's transactions quickly** without human interaction. An example would be an **ATM machine**.
- c) Computer system can be used as an **intranet** for schools so that students or the school staff have **easy access to school resources** as well as able to **communicate or share information** with each other.

Question 2 - ok

(a) **Input Devices:**

Mouse is used to **control the cursor** on the screen so that you are able to **point on objects** and make selections or issue instructions.

Keyboard is used to **type in letters, words and numbers** into the computer as well issuing instructions.

Output Devices:

Monitor is used to **display tutorial answers created on the screen**.

Printer is used to generate the answers in **hardcopy**

Communication Devices:

Network Interface Card (NIC) is used to allow the computers to **connect to each and another** in a LAN for communication and data sharing.

Router is used to **forward the network packets** destined to other networks

(b) **4 key functions of IPC/IPO model**

Input: capturing / collecting raw ideas and suggestions

Processing: compile, edit and summarize the ideas then format them in proper arrangement and display to increase the readability

Output: ready / compiled Tutorial Answers

Storage: save a copy for future retrieval

Question 3 - Chee Ching Hong - ok

- a) 4.70 GHz (CPU's speed)
- b) 16GB (RAM capacity)
- c) WINDOW 11 HOME (OS)

Question 4

Refer to your notes

Question 5 - Kelvin Chong Khai Shen - ok

a) Bus Topology

b) -All computer and other devices are connected by a single central cable

-Terminators are installed at both ends of the cable of the network to prevent bouncing of signals that cause interference.

- Signals can be sent in both directions, hence collisions might occur.

c) Advantages :

- inexpensive and easy to install
- Nodes can be attached to or detached from the bus without disturbing the network.
- Failure of one node does not affect the network.

Disadvantages :

- Failure of bus will affect the entire network, single point failure, not suitable for critical system
- Difficult to troubleshoot problems

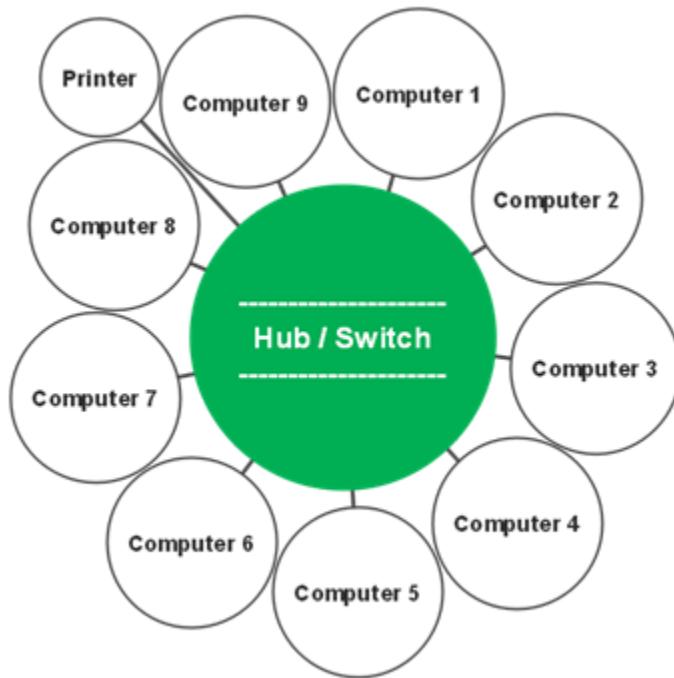
Question 6 - LIM FANG CHERN - acceptable although BUS is more preferable

a) Star Topology

- Its installation and maintenance are both easy.

- The computers can connect to the central devices such as the hub, switch.
- Computer/printer can be added to and removed from the network anytime easily without any disruption towards the network.
- The other computers / printer will not be affected if one of the computers / printer fails to connect the network.

b)



Q7 - Chin Gian Terng - ok

a) Client-Server Architecture.

Servers are more powerful computers that are able to manage and share content or a function since resource sharing is centralized. All 10000 students can request for a function at the same time

b) Peer-to-Peer Architecture.

Each node (computer) has equivalent responsibilities and can be both a server and a client. Each department can share and receive files from other departments, and does not require expensive infrastructure to set up.

c) Client-Server Architecture.

Servers are more powerful computers that are able to manage and share content or a function since resources must be centralized. Customers and employees can access information and data stored on the server efficiently and easily, with more security since the data is centralized.

Questions 8 - Yee Guan Jun - ok

There are a few reasons why I participate in online shopping. Firstly, by participating online shopping surely will **avoid crowds**. Nowadays there are many items which are in trend and will bring crowds. So to avoid crowds and not be left out from the trend I will mostly choose to buy it online.

Next, I also feel it is **convenient** to shop online. This is because by shopping online I do not need to go out from my house to get what I need.

Furthermore, I feel that online shopping can always get a **cheaper price and more variety** compared to physical shopping.

Tutorial 2 Discussions

SECTION A: NUMBERING SYSTEMS

Question 1a) b) c) - Tan Jia Wen - ok

(A) Numbering Systems

1a) $3D7_{16} \rightarrow \text{binary, octal \& dec}$

$$\begin{aligned}3D7_{16} &= (3 \times 16^3) + (D \times 16^2) + (7 \times 16^0) \\&= (3 \times 16^2) + (13 \times 16^1) + (7 \times 16^0) \\&\approx 983_{10}\end{aligned}$$

$$\begin{array}{r} 2 \overline{)983} \\ 2 \overline{)491} \\ 2 \overline{)245} \\ 2 \overline{)122} \\ 2 \overline{)61} \\ 2 \overline{)30} \\ 2 \overline{)15} \\ 2 \overline{)7} \\ 2 \overline{)3} \\ \quad \quad \quad | \\ \quad \quad \quad 1 \end{array}$$

Binary: 1111010111,

$$\begin{array}{r} 8 \overline{)983} \\ 8 \overline{)122} \\ 8 \overline{)15} \\ \quad \quad \quad | \\ \quad \quad \quad 7 \end{array} \quad \text{oct: } 1727_8$$

b) $1100010100100001_2 \rightarrow \text{oct, dec, hexadecimal}$

$$\begin{array}{ccccccc} 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ & C & 5 & 2 & 1 & & & & & & & & & \end{array}$$

hexadecimal: C52116

$$\begin{aligned}C521_16 &= (C \times 16^3) + (5 \times 16^2) + (2 \times 16^1) + (1 \times 16^0) \\&= (12 \times 16^3) + (5 \times 16^2) + (2 \times 16^1) + (1 \times 16^0) \\&= 50465_{10}.\end{aligned}$$

Dec = 50465₁₀

$$\begin{array}{r} 8 \overline{)50465} \\ 8 \overline{)6308} \\ 8 \overline{)788} \\ 8 \overline{)98} \\ 8 \overline{)12} \\ \quad \quad \quad | \\ \quad \quad \quad 4 \end{array} \quad \text{oct: } 142441$$

1c) $7098_{10} \rightarrow$ binary, oct, hexadecimal

$$\begin{array}{r} 2 | 7098 \\ 2 | 3549 \quad 0 \\ 2 | 1774 \quad 1 \\ 2 | 887 \quad 0 \\ 2 | 443 \quad 1 \end{array}$$

Binary: 0001101110111010

$$\begin{array}{r} 2 | 221 \quad 1 \\ 2 | 110 \quad 1 \\ 2 | 55 \quad 0 \\ 2 | 27 \quad 1 \\ 2 | 13 \quad 1 \\ 2 | 6 \quad 1 \\ 2 | 3 \quad 0 \\ 1 \quad 1 \end{array}$$

$$\begin{array}{r} 8 | 7098 \\ 8 | 887 \quad 2 \\ 8 | 110 \quad 7 \\ 8 | 13 \quad 6 \\ 1 \quad 5 \end{array}$$

Oct: 15672

$$\begin{array}{r} 16 | 7098 \\ 16 | 443 \quad 10 \\ 16 | 27 \quad 11 \end{array}$$

hexa: 1BBA

JIN HUEI WONG - ok

21d) Convert $136+2_8$ to binary, decimal and hexadecimal numbers respectively.

$$\text{Decimal} : 1 \times 8^4 + 3 \times 8^3 + 6 \times 8^2 + 1 \times 8^1 + 2 \times 8^0$$

$$= 6026_{10}$$

$$\text{Binary} : 1011110001010_2$$

$$\text{Hexadecimal} : 178A,$$

$$\begin{array}{r} 2 \overline{)6026} \\ 2 \overline{)3013} \quad 0 \\ 2 \overline{)1506} \quad 1 \\ 2 \overline{)753} \quad 0 \\ 2 \overline{)376} \quad 1 \\ 2 \overline{)188} \quad 0 \\ 2 \overline{)94} \quad 0 \\ 2 \overline{)47} \quad 0 \\ 2 \overline{)23} \quad 1 \\ 2 \overline{)11} \quad 1 \\ 2 \overline{)5} \quad 1 \\ 2 \overline{)2} \quad 1 \\ 2 \overline{)1} \quad 0 \end{array}$$

e) Convert 210102_3 to decimal numbers.

$$\text{Decimal} : 2 \times 3^5 + 1 \times 3^4 + 0 \times 3^3 + 1 \times 3^2 + 0 \times 3^1 + 2 \times 3^0$$

$$= 578_{10}$$

f) Convert 105_{10} to base - 5 number.

$$105_{10} = 4105$$

$$\begin{array}{r} 5 \overline{)105} \\ 5 \overline{)21} \quad 0 \\ 5 \overline{)4} \quad 1 \\ 0 \quad 4 \end{array}$$

2a) (JUN HONG) - ok

$$101101101 + 10011011 + 10010011$$

$$\begin{array}{r} 101101101 \\ + 10011011 \\ \hline 1000001000 \\ + 10010011 \\ \hline 1010011011 \end{array}$$

Ans : 1010011011 B

2b) 1FF9H + AC

Ans : illogical, H doesn't exist in base 16

2c) 7702 - 577

$$\begin{array}{r} 7702 \\ - 577 \\ \hline \end{array}$$

1. $2-7 \Rightarrow (8+2)-7$
= 3
2. $(8-1)-7 = 0$ Ans : 7103 Oct
3. $(7-1)-5 = 1$

Please complete (d) (e) (f)

(d) $\begin{array}{r} 3 \\ 2 | 6 \\ - 2 | A \\ \hline 2 | f 8 \end{array}_{12}$

$$\begin{aligned} \textcircled{1} \quad 6 - A &= 6 - 10 \\ &\Rightarrow (12+6) - 10 \\ &= 8 \\ \textcircled{2} \quad (A-1) - 2 &= 7 \\ \textcircled{3} \quad 2 \end{aligned}$$

450744

1FF9H₁₆

(e) 1101

$$\begin{array}{r} \times 1011 \\ \hline 1101 \\ 1101 \\ 0000 \\ 1101 \\ \hline 10001111 \end{array}$$

(f) illogical, there are no
5 & F in base-5

Jeff - ok

2g) The expression is illogical. H does not exist in hexadecimal.

2h) $1011 * 11 = 100001$

$$\begin{array}{r} 1011 \\ \times 11 \\ \hline 1011 \\ + 1011 \\ \hline 100001 \end{array}$$

$$1110001 - 100001 = 1010000$$

$$\begin{array}{r} 1110001 \\ - 100001 \\ \hline 1010000 \end{array}$$

SECTION B: SIGNED AND UNSIGNED NUMBERS

Question 1 -ok

$$- 11001100_2$$

1a) signed decimal value (int)

i) flip $\rightarrow 00110011_2$

ii) $+1 \rightarrow 00110100_2$

iii) convert binary to decimal (Multiply with weight method)
 $\rightarrow 2^5 + 2^4 + 2^2 = 52$

iv) $\therefore f.a = -52$

1b) unsigned decimal value (char)

i) convert binary to decimal (Multiply with weight method)
 $\rightarrow 2^7 + 2^6 + 2^3 + 2^2 = 204$

ii) $\therefore f.a = 204$

Question 2 - please upload the answers

$$-27 + (-58) = -85$$

$$27 = 00011011 \quad 58 = 00111010$$

$$\begin{array}{r} 00011011 \\ + 00111010 \\ \hline 11100101 \end{array}$$

$$\begin{array}{r} 11100101 \\ - 11000101 \\ \hline -1100110 \end{array}$$

$$11100101 \quad (-27)$$

$$+ 11000110 \quad (-58)$$

$$\begin{array}{r} 11000110 \\ \times 10101011 \\ \hline \end{array} \quad \text{2nd Method}$$

To verify:	$-27 + 2^5 + 2^3 + 2^1 + 2^0$
$\begin{array}{r} 01010100 \\ + 1 \\ \hline 01010101 \end{array}$	$= -128 + 32 + 8 + 2 + 1$
	$= -85 \#$
	$01010101: 2^6 + 2^4 + 2^2 + 2^0 = 85$
	$\therefore f.a = -85$

Question 3 - Qi Lun - ok

-75 - 25

a) And b)

$$\begin{array}{r} 75 \\ \hline 2 | 37 \\ 2 | 18 \\ 2 | 9 \\ 2 | 4 \\ 2 | 2 \\ \hline & 1 \end{array}$$

$$\begin{array}{r} 25 \\ \hline 2 | 12 \\ 2 | 6 \\ 2 | 3 \\ \hline & 1 \end{array}$$

$$75_{10} \Rightarrow 01001011_2 \quad 25_{10} \Rightarrow 00011001_2$$

$$\begin{array}{r} 75 \Rightarrow 01001011 \\ 75 \Rightarrow 10110100 \\ + \hline 10110101 \end{array}$$

$$\begin{array}{r} 25 \Rightarrow 00011001 \\ 25 \Rightarrow 11100110 \\ + \hline 11100111 \end{array}$$

$$\begin{array}{r} 10110101 \\ + 11100111 \\ \hline 110011100 \end{array}$$

↑ carry ignored

$$\begin{array}{r} 10011100 \\ \text{To verify} \Rightarrow 01100011 \\ + \hline 01100100 \end{array}$$

$$\text{Ans} = 01100100_2$$

$$\begin{aligned} \text{Ans in dec} &= 2^6 + 2^5 + 2^2 \\ &= 100, \end{aligned}$$

c) The answer is valid since the answer of both binary and decimal are the same/ because overflow does not occur / the expected result falls within the representable range of 8-bit data.

d)

	Carry flag	Overflow flag
Definition	occurs when the result of an arithmetic operation exceeds the fixed number of bits allocated. Extra “1” bit generated	occurs when the results of the calculation does not fit into the value range available.
Occur in signed or unsigned number?	Can occur in both operations	Only signed number operation
How to detect?	Extra ‘1’ bit is generated at the front after calculation is finished.	When the sign of the result is opposite of the sign of both operands.
Example	$ \begin{array}{r} 1100 \\ + 1110 \\ \hline 11010 \end{array} $	$ \begin{array}{r} 0100 \\ + 0110 \\ \hline 1010 \end{array} $
Based on your answer obtained in Q2 a), does carry or/and overflow occur?	Yes, carry occur.	No, overflow does not occur.

SECTION C: FLOATING POINT NUMBERS

Question 1 - Toong Shi Lin - ok

Section C

Q1 a) Convert $B.AC_{16}$ to base-8 number.

$$\begin{array}{r} \text{" } \overset{10}{\cancel{1}} \overset{12}{\cancel{2}} \\ B.AC_{16} \end{array} \Rightarrow 1011.10101100_{16}$$

$$00|011.10|011000_{16} \Rightarrow 13.530_8$$

b) Convert 56.7_{10} to base-5 number.

$$56.7_{10} = 56.0_{10} + 0.7_{10}$$

$$56.7_{10} \Rightarrow 211.3222_5$$

②

$$\begin{array}{r}
 5 | \begin{array}{r} 56 \\ 11 \\ 2 \\ 0 \end{array} & \begin{array}{r} 1 \\ 1 \\ 2 \\ 0 \end{array} \\
 \times 5 & \times 5 \\
 \hline
 0.5 & 0.5 \\
 \times 5 & \times 5 \\
 \hline
 2.5 & 2.5 \\
 \times 5 & \times 5 \\
 \hline
 0.5 & 0.5 \\
 \end{array}$$

Saved & Dropped

c) Convert 67.89_8 to base-2 number.

Illogical, there are no 8 and 9 in base-8.

d) $110.11_2 + 111.111_2 = 1110.101_2$.

$$\begin{array}{r}
 111.110 \\
 + 111.111 \\
 \hline
 1110.101
 \end{array}$$

Question 2 - Travis Owen Stothard - ok, just lack of steps

- (1) 4 for positive and 8 for negative sign
- (2) the implied decimal point is at the beginning of the mantissa
- (3) excess-80 is applied

SEEMMMMM → Value in exponential form

4 84 84848

$$= 0.84848 \times 10^4$$

$$= 8484.8$$

Full steps:

$x \cdot x \longrightarrow \text{SEEMMMMM}$

?

$\begin{array}{r} \text{S E M} \\ 484 | 84848 \\ \hline \end{array}$

sign = + (positive)
exponent: $84 - 80 = +4$
mantissa: $84848 \rightarrow 0.84848$
 $\therefore + 0.84848 \times 10^4$

Question 3 - Marcus - (a) is correct, (b) is incorrect, © is incorrect

3010 + 2.2222₁₀
provide 0 exponent: $+2.2222 \times 10^0$
adjust the point: $+0.22222 \times 10^1$
exponent: $50 + 1 = 51$
sign: positive (2)
SM: 22222
SEEMMMMM: 25122222

iid - 0.2468₁₀
provide 0 exponent: -0.2468×10^0
adjust point: -0.2468×10^0
exponent: $50 + 0 = 50$
sign: negative (1)
SM: 24680
SEEMMMMM: 150.24680

b) $25122222 - 151024680$
 $0.22222 - 0.024680$
 $= 0.19754$

c) $51 + 50 = 101$ 0.22222
 $101 - 50 = 51$ $\times 0.24680$
 $= 0.054843896$
 $= 0.54843896 \times 10^{-1}$
 $= 0.54844$

Ans: - 25054844

Question 4-Wei Jia - ok

a)

385805

$$\text{Q4(a)} \quad \begin{array}{r} \text{S E M} \\ \begin{array}{r|rr} 5 & 50 & 20311 \\ +153 & \hline 75321 \end{array} \end{array} \rightarrow +0.20311 \times 10^5$$

$$\rightarrow -0.75321 \times 10^3$$

① adjust the exponent: $\begin{array}{r} 5 \\ +1 \end{array} \begin{array}{l} [53]00020311 \\ [53]75321 \end{array}$

② add the mantissa: $0.00020311 + (-0.75321) = 0.75306(689)$

③ $\text{SEEMMMMM} = 15375301$

b)

$$(b) \quad \begin{array}{r} \text{S E M} \\ \begin{array}{r|rr} 1 & 5 & 176323 \\ \times & 154 & 85496 \end{array} \end{array} \quad \begin{array}{l} -0.76323 \times 10^5 \\ -0.85496 \times 10^6 \end{array}$$

① add the exponent: $51 + 54 = 105$
 $\Rightarrow 105 - 50 = 55$

② multiply the mantissa: $-0.76323 \times -0.85496 = 0.65253(N)$

③ $\text{SEEMMMMM} = 55565253 = +0.65253 \times 10^5$

C)

5 51 52295
1 52 56608

Adjust the exponent

5 52 052295
1 52 56608

Subtract the mantissa

$$0.052295 - (-0.56608) = 0.618375$$

$$\text{seemmmmm}=55261838=0.61838 \times 10^2$$

Question 5 - ok

Convert the binary number 11011.0011011_2 into IEEE 754 single precision format using excess-127 notation. (You are required to show your conversion steps clearly.) //carmaine

i) provide 0 exponent $\rightarrow 11011.0011011_2 \times 2^0$

ii) normalize format $\rightarrow 1.10110011011_2 \times 2^4$

iii) exponent = $127 + 4 = 131$

iv) convert decimal exponent to binary (division method)

$$\begin{array}{cccccccccc} 1 & \leftarrow & 2 & \leftarrow & 4 & \leftarrow & 8 & \leftarrow & 16 & \leftarrow & 32 & \leftarrow & 65 & \leftarrow & 131 \\ 1 & & 0 & & 0 & & 0 & & 0 & & 0 & & 1 & & 1 \end{array}$$

v) mantissa in 23 bits $\rightarrow 10110011011000000000000$

vi) sign = positive (0)

vi) IEEE754 representation $\rightarrow 0\ 10000011\ 10110011011000000000000$

Question 6 - ok

-15.25 = -1111.010 - missing conversion steps

Provide 0 exponent = -1111.01* 2^0

Normalize format = -1.11101* 2^3

exponent : $127 + 3 = 130 = 10000010$

Sign : negative (1)

23m : 11101 (23-bit)

IEEE754= 1 10000010 11101...0(32bit)

Question 7 - ok

1 10000010 110110100000000000000000000000

Sign = 1 → negative

Exponent = 10000010 = 130

$$\rightarrow 130 - 127 = 3$$

Mantissa = 11011010000000000000000000

$$\rightarrow 1.110110100000000000000000000000$$

The original format = -1.1101101×2^3

$$= -1110.1101 \text{ (base-2)}$$

$$= -(2^3 + 2^2 + 2^1 + 2^{-1} + 2^{-2} + 2^{-4})$$

$$= -14.8125 \text{ (base-10)}$$

Question 8 - WeiJun - ok

a)

i) $-5/16 = -0.3125 \text{ (base-10)}$

$$= -0.00101 \text{ (base-2)}$$

b) IEEE754 single precision (32-bit format)

i) Provide 0 exponent: -0.00101×2^0

ii) Normalize: $-1.01 \times 2^{(-3)}$

iii) Exponent: $127 + (-3) = 124$
 $= 01111100 \text{ (base-2)}$

iv) Mantissa in 23-bit: 01(23-bit)

v) Sign: negative (1)

vi) IEEE754 representation: 1 01111100 01(32-bit)

c) IEEE754 double precision (64-bit format)

i) Provide 0 exponent: -0.00101×2^0

ii) Normalize: $-1.01 \times 2^{(-3)}$

iii) Exponent: $1023 + (-3) = 1020$
 $= 0111111100 \text{ (base-2)}$

iv) Sign: negative (1)

v) Mantissa in 52-bit: 01..... (52-bit)

vi) IEEE754 representation: 1 01.....(64-bit)

D: BITWISE LOGICAL OPERATION

Question1-(Wei Zhun) - ok

Bitwise Logical Operations								Date.....
Q1	X = 6BH	Y = 45H	Z = 17H					
	= 0110 1011	= 0100 0101	= 0001 0111					
a) W = X + Z								
	0 1 1 0 1 0 1 1 (X)							
	OR 0 0 0 1 0 1 1 1 (Z)							
	0 1 1 1 1 1 1 1 (W)							
	W = 0111 1111 ₂							
	= 7FH							
b) W = (X · Y)' ⊕ Z And then Not then XOR								
	0 1 1 0 1 0 1 1 (X)							
	AND 0 1 0 0 0 1 0 1 (Y)							
	0 1 0 0 0 0 0 1 (X · Y)							
	NOT 0 1 0 0 0 0 0 1 (X · Y)'							
	1 0 1 1 1 1 1 0 (X · Y)'							
	XOR 0 0 0 1 0 1 1 1 (Z)							
	1 0 1 0 1 0 0 1							
	W = 1010 1001 ₂							
	= A9H							

Question 2-Weng Zhen (a) (b) are correct, please double check © → should be 20H

2) a) $A = 3b_{16} \rightarrow A = 0011010_2$ $B = 9A_{16} \rightarrow B = 10011010_2$ $C = BB_{16} \rightarrow C = 10111011_2$

$$A \cdot B = \begin{array}{r} 0011010 \\ 10011010 \\ \hline 00010010 \end{array}$$
$$= 12_{16} \rightarrow 10001100_2$$

b) $B + C' \rightarrow C'$ higher precedence

$$C' = 01000100_2$$
$$+ B + C' = \begin{array}{r} 10011010 \\ 01000100 \\ \hline 11011110 \end{array}$$
$$= D5_{16} \rightarrow 10100101_2$$

c) $A + CB \oplus C \rightarrow A + C(B \oplus C) \leftarrow \begin{array}{r} 10011010 \\ 10111011 \\ 00100001 \\ \hline 00100001 \end{array}$

$$A + CB \oplus C = \begin{array}{r} 10011010 \\ 00110100 \\ 00100001 \\ \hline 00110100 \end{array}$$
$$= 00110100_2$$

answering back to last slide $\rightarrow 37_{16}$

Question 3 - please update the answers

Tutorial 3 Discussions

Q1- you - ok

Machine Cycle (Instruction Cycle)

Fetching: To get the instruction from the main memory RAM (*with the helps of PC, MAR and MDR*)

Decoding: The CPU's control units *interpret and identify* the program's instruction (*with the help of IR*)

Executing: The CPU's Arithmetic logical unit operates on the operand / *carry out the instruction* (*with the helps of A*)

Storing: The processed data is sent to the main memory RAM or stored temporarily in the CPU registers for later retrieval

Q2-Tang Yan Chun - ok

LOAD

i) PC → MAR ; MAR = 20 → Fetch cycle
MDR → IR ; IR = 5 60

IR[address] → MAR ; MAR = 60 → Execute cycle
MDR → A ; A = 422
PC + 1 → PC ; PC = 21

ii) ADD

PC → MAR ; MAR = 21 → Fetch cycle
MDR → IR ; IR = 1 61

IR[address] → MAR ; MAR = 61 → Execute cycle
A + MDR → A ; A = 422 + 008 = 430
PC + 1 → PC ; PC = 22

iii) STORE

PC → MAR ; MAR = 22 → Fetch cycle
MDR → IR ; IR = 3 60

IR[address] → MAR ; MAR = 60 → Execute cycle
A → MDR ; MDR = 430
PC + 1 → PC ; PC = 23

Q3

a)

Function of Instruction Pointer (IR) is to **hold the current instruction** get from the memory that is currently executed.

Function of Program Counter (PC) is to **hold/keep track of the instruction address** that is going to be executed.

Function of Memory Address Register (MAR) is to **hold the address of memory location where the data is to be fetched or stored.**

Function of memory Data Register (MDR) is to **hold the data which is taken / store from / to the memory location that currently activated by the MAR**

Function of Accumulator (A) is to **store data/results for arithmetic operations.**

b)

The answers are incorrect, please reupload

IR: 5338

PC:342

MAR : 338

MDR: BB16

A: FA16 - BB16=3F16

3b) Subtract - correct

Step 1 : PC → MAR ; MAR = 341

Step 2: MDR → IR ; IR = 5 338

Step 3: IR[address] → MAR ; MAR = 338

Step 4: A - MDR → A ; A = FA - BB = 3F

Step 5: PC + 1 → PC ; PC = 342

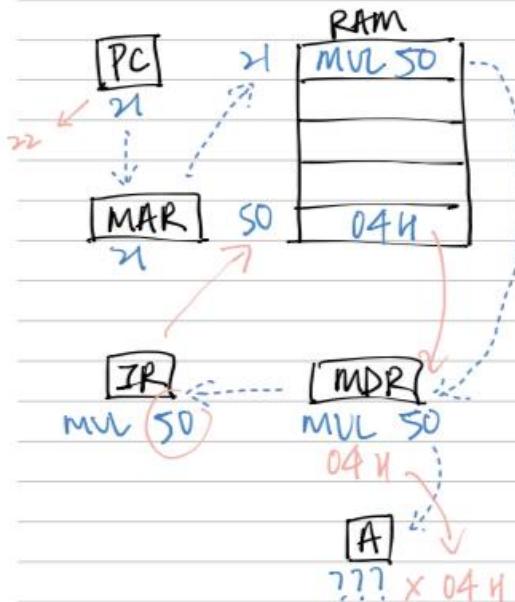
c)

PC > MAR	Students refer to Google Classroom for Assignment question
MDR > IR	Student visit to Google Classroom to download, interpret and understand the Assignment question.
IR [Address] > MAR	Based on the Assignment question, students do research from reading material / websites, etc
MDR > A	Students record the findings into their local drive for future reference
PC + 1 > PC	Students completed the research and ready for next action.

Question 4 - Chong Yu Jing -ok

4. With the Little Man Computer, explain the FIVE (5) steps of how the instruction MUL 50 is executed in the CPU. Given the Program Counter (PC) is now 21. The contents of location 21 and location 50 are MUL 50 and 04H respectively.

Program counter (PC) = 21 Accumulator : ???
 Memory [location 21] : MUL 50
 Memory [location 50] : 04H

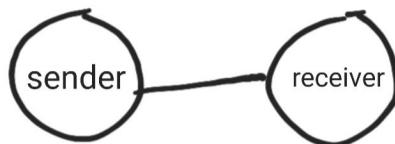


$\text{PC} \rightarrow \text{MAR}; \text{MAR} = 21$
 $\text{MDR} \rightarrow \text{IR}; \text{IR} = \text{MUL } 50$
 $\text{IR}[\text{add}] \rightarrow \text{MAR}; \text{MAR} = 50$
 $A \times \text{MDR} \rightarrow A; 04H \times \text{value in } A \text{ and the result of multiplication will be stored in } A$
 $\text{PC} + 1 \rightarrow \text{PC}; \text{PC} = 22$

Q5(cheong ze xiang) - ok

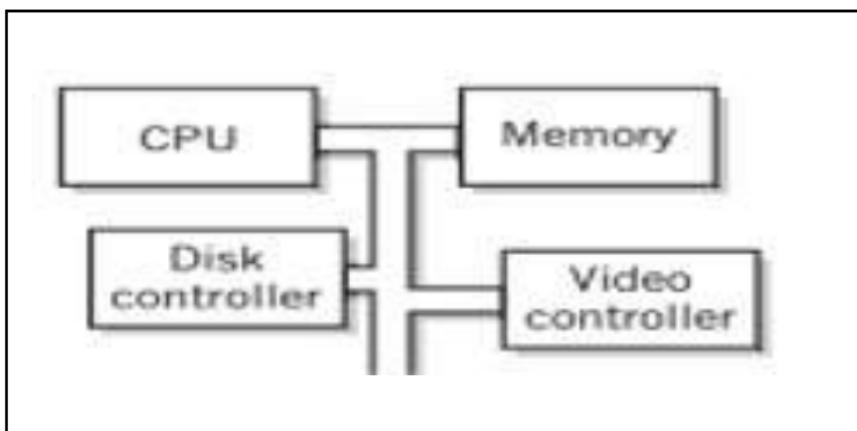
Point-to-point bus

- Signals in point to point bus are carried from a **specific sender to a specific receiver.**
- Point-to-point bus consists of **Data lines, Power lines, Control lines.** Address Lines are not needed as the receiver is known.



Multipoint Bus

- Signals in multipoint bus are carried **from a sender to several receivers.** Bus that used to connect several points together.
- When a sender sends the data, it is **broadcasted to everyone** in the connection.
- **Addressing signal is needed** in order to identify who is the actual receiver.



Q6.CCC - ok

a) This is a multipoint bus.

Multipoint is a bus topologies that carries signals to several destinations. It is also known as multidrop bus (MDB) or broadcast bus. Multipoint bus usually requires addressing signals on the bus to identify the addressed destination.

b) Advantage:

Multiple buses permit several devices to work simultaneously / connected together

Disadvantage:

When the backbone bus is down, all the connected devices will be affected

Q7-Chee Ching Hong

- a) RISC (Reduced Instruction Set Computer) is the name of the computer processor architecture for most modern computers which provide a large number of general purpose registers and very few memory-access instructions.

- b) Reduced Instruction Count:

High-level Abstraction:

Reduced Memory Access:

Kindly refer to the LECTURE's note Chap3 for the 5 advantages of RISC

CISC	VS	RISC
Complex Instruction Set Computer		Reduced Instruction Set Computer
Consist of many instruction (around 120-350) Most of them are not frequently used	Instruction	Simplifying instruction set of CPU (less than 100)
Hardware	Emphasise	Software
Requires additional hardware complexity		Emphasise on frequently used instruction eg. BRANCH, LOAD, STORE
Longer	Execution Time	Shorter
Require more step to complete	For a simple instruction,	Supposed to be executed within the same clock interval
Intel x86, AMD	Strong supporter	Apple, Android-based system (ARM Cores)

Tutorial 4 Discussions

Please use the following format to present your answers:

Question X - your name

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Question 1 - WINNIE CHIONG YAN -ok

128-bit (16-byte)

Question 2 - Kelvin Chong Khai Shen

RAM	Aspect	Register
Ranges from some GB	Capacity	Ranges from 32-bits register to 64-bit register
Hold the data that will be required for processing reverse -byte sequence (little endian order) Use reverse-byte sequence in which the low order byte refers to a low memory address whereas high order byte refers to high memory address.	Data Storage	Holds the data that the CPU is currently processing Normal sequence (big endian order) -Use normal byte sequence -use absolute address method which is a direct reference method, - use segment;offset method in which the distance will be stored in the pointer and index register whereas starting address will be stored in segment register.
To refer a data in the RAM we need to know the memory address Eg. 00001H	Reference method	To refer a data in the register, we need to know the name of the register Eg. AX, BX, CX, DX

Question 3 - LIM FANG CHERN

RAM			
	Memory Location	Memory Data	
VAR1	0150H	41H / 0100 0001B	"A"
VAR2	0151H	E7H / 1110 0111B	2023
	0152H	07H / 0000 0111B	
VAR3	0153H	41H	202305
	0154H	16H	
	0155H	03H	
	0156H	00H	
VAR3	0157H	24H	1024H
	0158H	01H	
	0159H	01H	
	015AH	00H	
VAR4	015BH	3DH	98H 3DH
	015CH	98H	
	015DH	00H	
	015EH	00H	
	015FH	00H	
	0160H	00H	
	0161H	00H	
	0162H	00H	

Question 4 - GT - OK

Assuming all Flag registers are cleared (value = 0) initially.

	Instructions	Working (Binary)	OF	SF	ZF	CF
a)	MOV AX, B014H	a) B014 H = 1011 0000 0001 0100 B	0 (OFF)	0	0	0
b)	ADD AX, B014H	b) B014 H + B014 H = 1011 0000 0001 0100 B + 1011 0000 0001 0100 B ----- carry occurred → 0110 0000 0010 1000 B = 6028 H	1 (ON)	0	0	1
c)	XOR AX, 6028H	⊕ 0110 0000 0010 1000 B 0110 0000 0010 1000 B ----- 0000 0000 0000 0000 B = 0000 H	1	0	1	1

Question 5 - Guan Jun - OK

CS: 2788₁₆

Convert CS to 20 bits

IP: 1705₁₆

2788 → 27880 H

Absolute address = 27880

$$\begin{array}{r}
 + \\
 \hline
 27880 \\
 1705 \\
 \hline
 28F854
 \end{array}$$

Question 6 - Hui Yi

a) Code Segment - ok

b) CS:IP - ok

CS stores the starting address of Code Segment

IP stores the distance of a specific instruction calculated from the starting location

c) 2BC8:FFFD - ok

d) done correction

$16\text{-bit} \rightarrow 2BC8$	
$20\text{-bit} \rightarrow 2BC80H$	
$\begin{array}{r} 2BC80H \\ + FFFDH \\ \hline DH \end{array}$	$\textcircled{1} \quad 8+F = 8+15 \\ = 23 - (1 \times 16) \\ = 7$
	$C+F+1 = 12+15+1 \\ = 28 - (1 \times 16) \\ = 12 \text{ (C)}$
$\therefore 3BC7\#$	$B+F+1 = 11+15+1 \\ = 27 - (1 \times 16) \\ = 11 \text{ (B)}$
	$2+1 = 3$

Question 7 - Jia Wen

Question 7

7a) $73_8 + 25_8$

$$73_8 + 25_8 = 120_8$$

$$= (1 \times 8^2) + (2 \times 8^1) + (0 \times 8^0)$$

$$= 80_{10}$$

$$80_{10} = 16 \underline{180} \\ 16 \underline{150} \\ 05$$

$$= 50_{16}$$

$$= 0050H$$

In Ax register

00	50	0
----	----	---

0000 0000	0101 0000	0
-----------	-----------	---

In memory

0000	0001
50	00

- 00 (high order byte)

- 50 (low order byte)

0000 0000	0101 0000
-----------	-----------

2. $1111_2 \times 111_2 \times 11_2$

$$1111_2 \times 111_2 \times 11_2 = 0001 \ 0011 \ 1011_2$$

$$= (1 \times 2^8 + 0 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0)$$

$$= 315_{10}$$

$$315_{10} = 16 \underline{315} \\ 16 \underline{19} \ 11 \\ 1 \ 3$$

$$= 13B_{16}$$

In Ax register

01	3B	0
----	----	---

In memory

0000	0001
3B	01

- 01 (high order byte)

- 3B (low order byte)

Q8 - jinhuei - ok

a) What is cache memory?

Cache memory is **extremely fast memory** between **CPU and the main memory** that is used to **duplicate** frequently data/instructions from the RAM.

b) How does a cache memory work?

Every memory request goes to the cache controller which checks the request against each tag. **If there is a hit, the cache location is used instead of memory.** If the required data cannot be found in the cache memory, this is called a “miss”. **“Miss” requires the cache controller to select a line for replacement from memory.** After which, the new line in cache is treated as before. In the case when the cache is full, **Least Recently Used (LRU)** will be used to decide which cache block to be replaced.

c) What could be the impact of a cache miss?

Cache misses can **significantly impact system performance**. When a cache miss occurs, the system must fetch the requested data from the main memory or another lower-level cache, which is a slower process compared to retrieving data from the cache.

Q9 - Jun Hong - ok

Based on the question:

P1 - 370 KB	J1 - 75 KB
P2 - 256 KB	J2 - 125 KB
P3 - 120 KB	J3 - 398 KB
P4 - 50 KB	J4 - 225 KB
P5 - 400 KB	

(a)(a) First fit allocation algorithm

Memory partition	Memory partition size	Job	Job size	Internal fragmentation
P1	370 KB	J1	75 KB	370 - 75 = 295 KB
P2	256 KB	J2	125 KB	256 - 125 = 131 KB
P3	120 KB			
P4	50 KB			
P5	400 KB	J3	398 KB	400 - 398 = 2 KB
Total:				428 KB

J4 needs to wait. Total internal fragmentation will be 428KB.

(a)(b) Best fit allocation algorithm

Memory partition	Memory partition size	Job	Job size	Internal fragmentation

P1	370 KB	J4	225KB	$370 - 225 = 145\text{KB}$
P2	256 KB	J2	125KB	$256 - 125 = 131\text{ KB}$
P3	120 KB	J1	75 KB	$120 - 75 = 45\text{KB}$
P4	50 KB			
P5	400 KB	J3	398KB	$400 - 398 = 2\text{KB}$
Total:				323 KB

(a)(c) Worst fit allocation algorithm

Memory partition	Memory partition size	Job	Job size	Internal fragmentation
P1	370 KB	J2	125KB	$370 - 125 = 245\text{KB}$
P2	256 KB	J4	225KB	$256 - 225 = 31\text{KB}$
P3	120 KB			
P4	50 KB			
P5	400 KB	J1	75KB	$400 - 75 = 325\text{KB}$
Total:				601 KB

J3 needs to wait. Total internal fragmentation will be 601KB.

(b) Best fit memory allocation algorithm. All jobs can be loaded to the RAM. Total internal fragmentation will be 323KB (which is the least in all three algorithms)

Q10 - Jeff - ok

a)

First fit - J1

Best fit - J4

Worst fit - J1

b)

Best fit memory allocation algorithm

c)

$P1 = 600\text{KB} - 550\text{KB} = 50\text{KB}$

$P2 = 0\text{KB}$

$P3 = 100\text{KB} - 50\text{KB} = 50\text{KB}$

$P4 = 400\text{KB} - 350\text{KB} = 50\text{KB}$

$P5 = 300\text{KB} - 250\text{KB} = 50\text{KB}$

Total internal fragmentation = 50KB + 50KB + 50KB + 50KB
= 200KB

Tutorial 5 Discussions

Please use the following format to present your answers:

Question X - your name

Xxx
xx
xx

Question 1 - Carmaine (the 3 components should be the CPU, I/O Module and I/O device)

(i) I/O ports & Address

I/O communicates with the CPU through specific I/O ports or addresses. These ports and addresses are used by I/O to send and receive data. Moreover, these ports and addresses are also used by the CPU to send command, data and request information.

(ii) I/O Controllers

I/O controllers are the **intermediaries between CPU and the I/O devices**. It interacts with the CPU using I/O ports and addresses, and **provides a standardized interface for the CPU to interact with different kinds of I/O devices**. The controller also manages and stores data in a buffer, streamlining the communication between I/O devices and CPU.

(iii) Interrupts & Polling

Interrupts and polling are two different mechanisms to ensure timely or sequential communication between CPU and I/O devices. When an I/O device needs to communicate with CPU, it sends an interrupt request (IRQ), this causes CPU to break what it's doing and save its progress, then it will execute an interrupt service routine (ISR) specifically to the device. On the other hand, polling mechanism ensures I/O devices get CPU's check in at regular intervals, the downside of this mechanism will be the wasting of the CPU's resources, as the CPU will still check in I/O devices even when I/O devices do not have the need to communicate with CPU.

Question 2 - Daniel (the question is asking for functions of CPU interface and Device interface of I/O modules, please reupload the answers)

(i) CPU interface

- Accept commands from the CPU
- Data transfer
- Sending interrupts and status signals

- (ii) Device interface
- Control a particular device that a CPU command refers to

Question 3 - Qi Lun - ok

- a) The sender of the interrupt is the I/O device(USB) and receiver is the CPU.
- b) As an external event (device) notifier ~ Freeing CPU from performing polling input dat
In this scenario, the purpose of this interrupt is to alert the CPU that this USB requires
immediate attention and please carry out the action for this interrupt.

Will be good if you provide multiple interrupts examples and show how prioritization works

- c) Before someone/something interrupts me, I will focus on my assignment report writing.
Until suddenly a notification from the phone interrupts me, I will stop and save my
assignment report in google document, then proceed to check on that notification. After
dealing with the notification, I will get back to my assignment report writing at which part
that I left off.

Question 4 - Shi Lin - ok

Programmed I/O is **slow** since a full instruction fetch-execute cycle must be performed
for every data word transferred. The data transferred in PIO is **used primarily on simple
character based**. PIO is suitable for **sending small amount of data** and the **CPU is fully involved**
for the whole transfer operation.

Hard Disk Drive or Graphic Display involve **block-oriented data transmission**, So DMA is more
suitable when the I/O devices used is hard disk drive or graphics display,

Question 5 - Travis - ok

Direct Memory Access (DMA) is not suitable to facilitate the data transfer from
a keyboard because **keyboards produce slow and character-based input**. Since **DMA is
optimized for speed and efficiency, it doesn't provide significant benefits for the relatively slow
and intermittent character-by-character input from a keyboard**. This type of input is better
managed by the CPU, which can handle the low data rate without introducing any noticeable
performance issues.

Question 6 - Marcus - - ok

A computer system determines the interrupting device by using a process mechanism
known as the IRQ, or an Interrupt Request Mechanism.

2 ways of determining interrupting devices.

(i) vectored interrupt

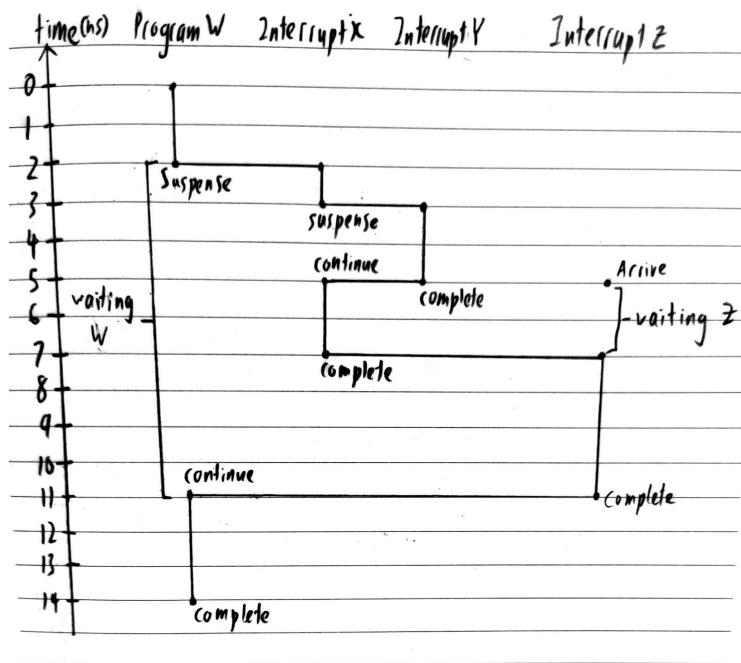
- the address of the interrupting device is included as part of the interrupt.

(ii) general (polled) interrupt

- uses the general interrupt that's shared by all devices.
CPU polls each device to identify the interrupt source.

Question 7-Wei Jia

7a)



7b) Program W(9ns of waiting time); because its priority is the lowest so it needs to wait for the other higher priority task to complete before it continues.

- W is interrupted at the 2nd ns and only able to continue in the 11th ns. Hence, the waiting time = 11 - 2 = 9ns which is the longest among all processes.

Tutorial 6 Discussions

Question 1 - WeiJun - ok

Operating system(OS) is a software that acts as an **intermediary between computer user and computer hardware.**

OS provides the **interface** (eg. GUI, CLI) needed for the user to issue instructions to the computer hardware

OS also provides the **platform** needed in order to **run the application program**

OS **allocates resources** (eg. RAM, CPU, I/O facilities) needed for different operations.

Examples of OS used for desktop are Microsoft Windows and Linux.

Examples of OS used for mobile devices are IOS and Android.

Question 2 - WeiZhun ok

Yes, the operating system is mandatory in every computer system.

This is because the operating system **manages the computer's memory and processes**, as well as all of its **software** and **hardware**.

Operating systems are like an **intermediary** between the computer user and the computer. Operating systems **provide the interface needed** to allow the user to **communicate** with the computer without knowing the computer's language.

Operating systems also allocate the resource to the computer operations.

Question 3 - Weng Zhen ok

First key function:

OS makes computer system easy to use. It **provides interface** for the user to have a better outlook the computer system. For instance, OS provides graphical user interface (GUI) or command-line interface (CLI) that allows users to interact with the computer. Through these interfaces,users are able to **launch applications, manage files, configure system settings**, and perform various tasks without needing to understand the intricacies of the hardware.

Second key function:

OS is significant in the way that it enables us to **use the computer hardware and control I/O operations** in an efficient manner. The OS includes device drivers, which are software components that enable communication between the OS and hardware devices like printers, graphics cards, and network adapters. These **drivers ensure that hardware devices are recognized and can be used** by the computer system and applications.

Third key function:

The OS **allocates resources** to specific programs. To elaborate this point, the OS manages system resources such as CPU time, memory, storage devices, and peripherals. It ensures that these resources are allocated efficiently to running processes and applications, preventing conflicts and resource contention. This **resource management** is crucial for ensuring that multiple programs can run concurrently without causing system crashes or slowdowns.

Question 4 - Xin Jie

a) Batch os.

Batch system is **executing a series of jobs all at one time**. Transactions are done in groups. **No interaction with the user while the program is being executed**. Suitable for applications that have a long execution time and do not need response.

b) Embedded System - refer to Chap6 - Types of OS

c) Time Sharing Os.

Time sharing systems is a multiprogramming operating system that allows active user interactions. Multiple programs are executed by the CPU by switching between them, but the switches occur so frequently. Thus, the user can receive an immediate response.

d) Embedded Os.

Embedded systems are placed inside other products to add features and capabilities. Designed for specific purposes and not changeable among systems.

e) Interactive OS

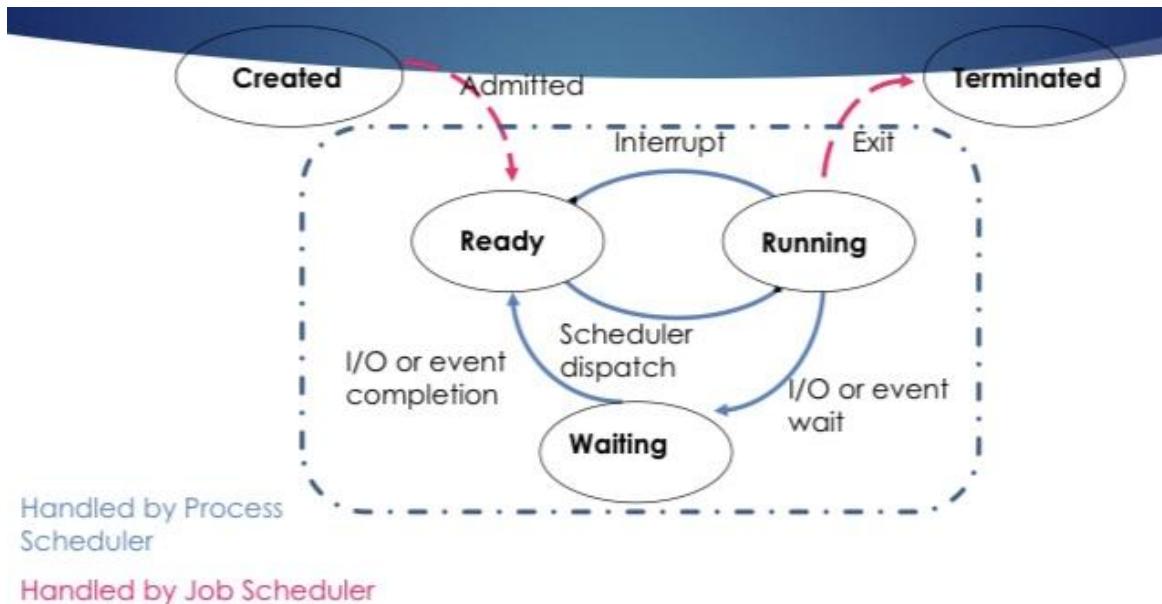
User interacts directly with the operating system via commands entered from the keyboard. Operating system provides immediate feedback to the user. Response time can be measured in fractions of a second

Tutorial 7 Discussions

Question 1 - Xuan You - ok

- PCB is a **data structure** that stores **information about a job** (from the time it is created until the time the job is terminated) - *similar to a passport to a traveler.*
- PCB consists of the following information:
 - Process identification
 - Process status (e.g. HOLD, READY, RUNNING, WAITING)
 - Process state (process word, register contents, main memory info, resources, process priority)
 - Accounting (CPU Time, total amount of time, I/O operations, number input records read, etc.)

Question 2 - Yan Chun - ok



CREATED → READY: The job is accepted by the system and is arranged in proper sequence (ie. balance mixed of I/o bound activities and CPu bound activities). This transition is done by Job Scheduler.

READY → RUNNING: The job is ready to be processed by the CPU and will be assigned with the CPU resource using some predefined process scheduling algorithm (eg. FCFS, RR, PS etc). This transition is done by Process Scheduler.

RUNNING → READY: in some occasions, a job may be interrupted when a predefined time limit or other criterion is met. The job will be put back to the READY queue. This transition is done by Process Scheduler.

RUNNING → WAITING: When a job initiates an I/O operation, it will be put into WAITING state, waiting for available I/O resources. This transition is done by Process Scheduler.

WAITING → READY: When an I/O request has been satisfied and the job wishes to continue with CPU processing. This transition is done by Process Scheduler.

RUNNING → FINISH: When a job is completed / error occurs and needs to be terminated prematurely. This transition is done by Job Scheduler.

Question 3 - ok

Context switching is the acts of saving a job's processing information in its PCB so the current job can be swapped out of memory and loads the processing information from the PCB of another new coming or interrupted job into the appropriate registers.

Advantages: -

- Hardware mechanism saves almost all of the CPU state
- Software can be more selective and save only that portion that actually needs to be saved & reloaded.
- It permits better control over the validity of the data that is being loaded

Disadvantages: -

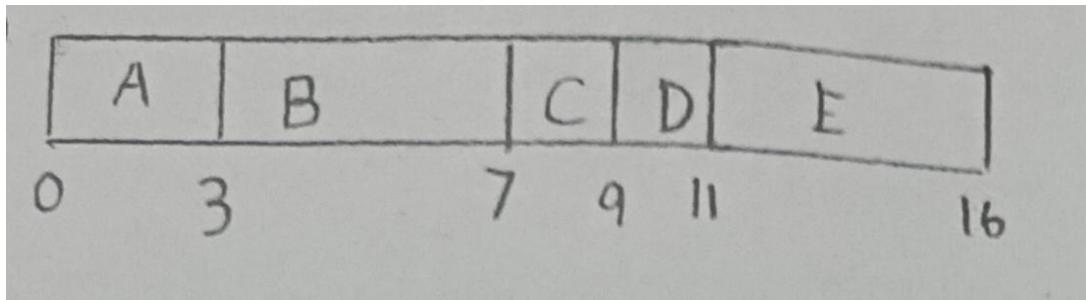
- In the process of switching the CPU to another (context switch), it requires saving the state of the old process and loading the saved state for the new process.
- **Increase overhead** and requires considerable processor time

Question 4 - ok

	Preemptive policy	Non-preemptive policy
Interrupt allowed?	Allow interrupts to the processing of a job and transfers the CPU to another job.	Process scheduling functions without external interrupts
Algorithm	(a) Shortest Remaining Time (SRT) (b) Round Robin (RR) (c) Priority Scheduling (PS)	(a) First Come First Serve (FCFS) (b) Shortest Job First (SJF) (c) Priority Scheduling (PS)

Question 5

(a) First Come First Served (FCFS) -ok

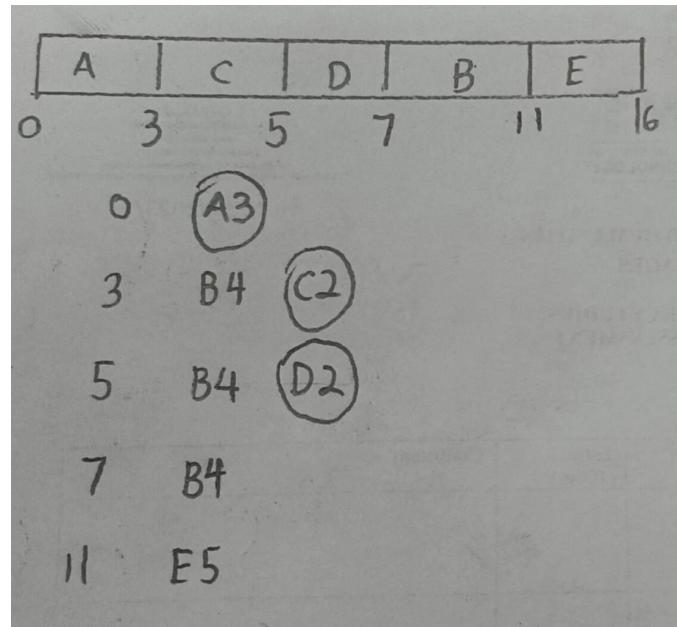


Turnaround Time = Finish Time - Arrival Time

Waiting Time = Finish Time - Arrival Time - CPU cycle

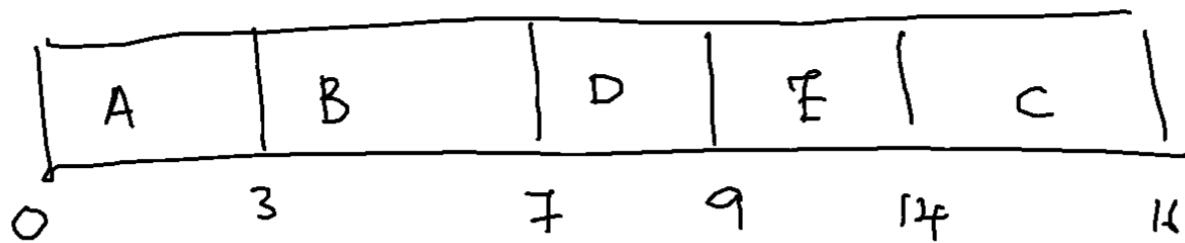
Process	Arrival time	Processing time	Finish time	Turnaround time (ns)	Waiting time (ns)
A	0	3	3	$3 - 0 = 3$	$3 - 0 - 3 = 0$
B	2	4	7	$7 - 2 = 5$	$7 - 2 - 4 = 1$
C	3	2	9	$9 - 3 = 6$	$9 - 3 - 2 = 4$
D	5	2	11	$11 - 5 = 6$	$11 - 5 - 2 = 4$
E	8	5	16	$16 - 8 = 8$	$16 - 8 - 5 = 3$
Average:				$28 / 5 = 5.6$	$12 / 5 = 2.4$

(b) Shortest Job First (SJF) -ok



Process	Arrival time	Processing time	Finish time	Turnaround time	Waiting time
A	0	3	3	$3 - 0 = 3$	$3 - 0 - 3 = 0$
B	2	4	11	$11 - 2 = 9$	$11 - 2 - 4 = 5$
C	3	2	5	$5 - 3 = 2$	$5 - 3 - 2 = 0$
D	5	2	7	$7 - 5 = 2$	$7 - 5 - 2 = 0$
E	8	5	16	$16 - 8 = 8$	$16 - 8 - 5 = 3$
Average:				$24 / 5 = 4.8$	$8 / 5 = 1.6$

(c) Priority Scheduling (PS) - Non-preemptive ok



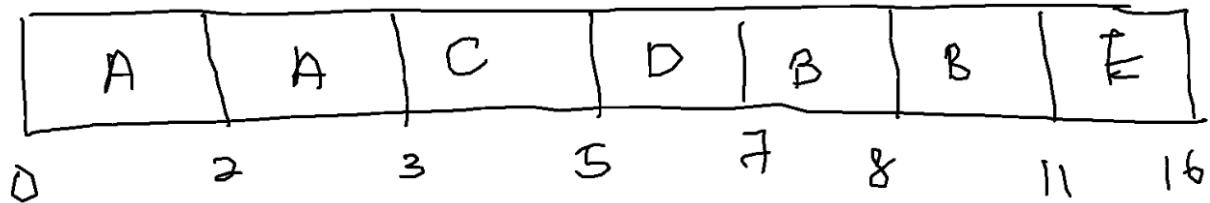
Process	Arrival time	Processing time	Finish time	Turnaround time	Waiting time
A (5)	0	3	3	$3 - 0 = 3$	$3 - 0 - 3 = 0$
B (2)	2	4	7	$7 - 2 = 5$	$7 - 2 - 4 = 1$
C (4)	3	2	16	$16 - 3 = 13$	$16 - 3 - 2 = 11$
D (1)	5	2	9	$9 - 5 = 4$	$9 - 5 - 2 = 2$
E (3)	8	5	14	$14 - 8 = 6$	$14 - 8 - 5 = 1$
Average:				$31 / 5 = 6.2$	$15 / 5 = 3$

(d) Priority Scheduling (PS) - Preemptive ok



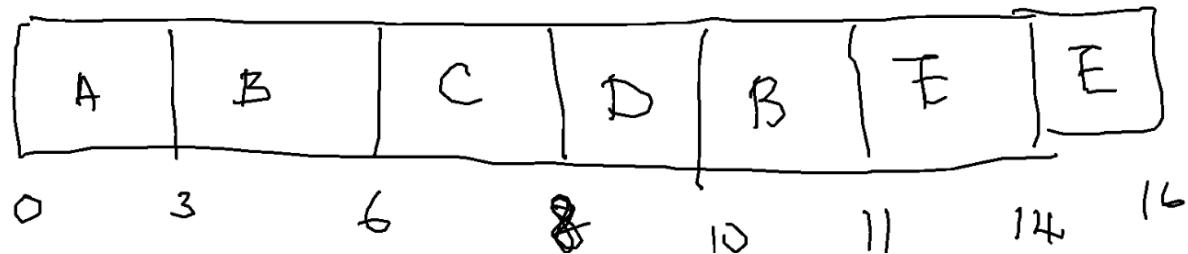
Process	Arrival time	Processing time	Finish time	Turnaround time	Waiting time
A (5)	0	3	16	16 - 0 = 16	16 - 0 - 3 = 13
B (2)	2	4	8	8 - 2 = 6	8 - 2 - 4 = 2
C (4)	3	2	15	15 - 3 = 12	15 - 3 - 2 = 10
D (1)	5	2	7	7 - 5 = 2	7 - 2 - 5 = 0
E (3)	8	5	13	13 - 8 = 5	13 - 8 - 5 = 0
Average:				41 / 5 = 8.2	25 / 5 = 5

(e) Shortest Remaining Time Scheduling (SRT) -ok



Process	Arrival time	Processing time	Finish time	Turnaround time	Waiting time
A	0	3	3	$3 - 0 = 3$	$3 - 0 - 3 = 0$
B	2	4	11	$11 - 2 = 9$	$11 - 2 - 4 = 5$
C	3	2	5	$5 - 3 = 2$	$5 - 3 - 2 = 0$
D	5	2	7	$7 - 5 = 2$	$7 - 5 - 2 = 0$
E	8	5	16	$16 - 8 = 8$	$16 - 8 - 5 = 3$
Average:				$24 / 5 = 4.8$	$8 / 5 = 1.6$

(F) Round Robin scheduling (RR) - (Assuming the quantum given is 3 time slices.) ok



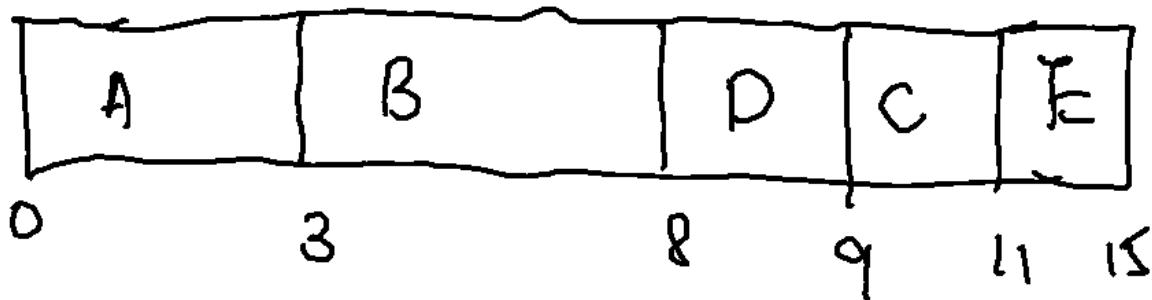
Process	Arrival time	Processing time	Finish time	Turnaround time	Waiting time
A	0	3	3	3 - 0 = 3	3 - 0 - 3 = 0
B	2	4	11	11 - 2 = 9	11 - 2 - 4 = 5
C	3	2	8	8 - 3 = 5	8 - 3 - 2 = 3
D	5	2	10	10 - 5 = 5	10 - 5 - 2 = 3
E	8	5	16	16 - 8 = 8	16 - 8 - 5 = 3
Average:				$30 / 5 = 6$	$14 / 5 = 2.8$

Question 6a-Ching Hong - ok

The factor that needs to be considered is **turnaround time** to make sure how long it takes to execute the process. The lower the turnaround time, the better the policy is.

The other factor is **wait time**. This is because the process needs to wait before being processed by the CPU. The lower the waiting time, the better the policy is.

Question 6(b)-ok

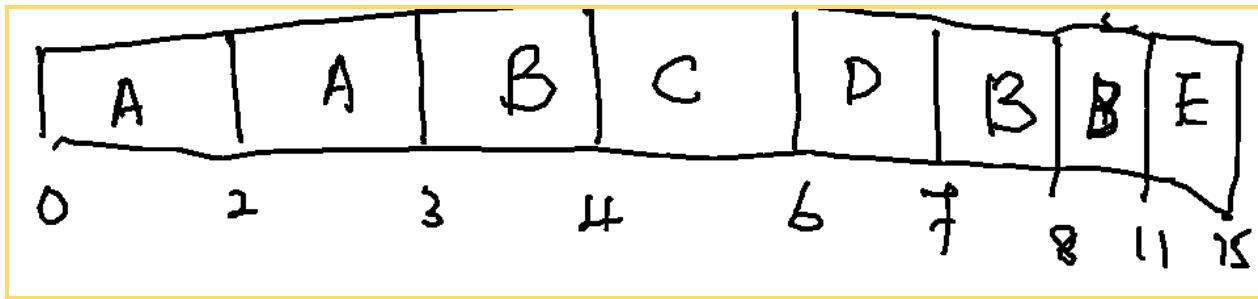


Shortest Job First (SJF)

Process	Arrival time	Processing time	Finish time	Turnaround time	Waiting time
A	0	3	3	$3 - 0 = 3$	$3 - 0 - 3 = 0$
B	2	5	8	$8 - 2 = 6$	$8 - 2 - 5 = 1$
C	4	2	11	$11 - 4 = 7$	$11 - 4 - 2 = 5$
D	6	1	9	$9 - 6 = 3$	$9 - 6 - 1 = 2$
E	8	4	15	$15 - 8 = 7$	$15 - 8 - 4 = 3$
Average:				$26 / 5 = 5.2$	$11 / 5 = 2.2$

Message: CPU Processing Time = CPU Cycle

Shortest Remaining Time (SRT) - Preemptive - ok



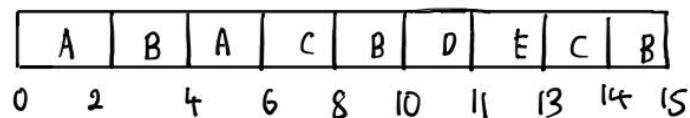
Process	Arrival time	Processing time	Finish time	Turnaround time	Waiting time
A	0	3	3	3 - 0 = 3	3 - 0 - 3 = 0
B	2	5	11	11 - 2 = 9	11 - 2 - 5 = 4
C	4	2	6	6 - 4 = 2	6 - 4 - 2 = 0
D	6	1	7	7 - 6 = 1	7 - 6 - 1 = 0
E	8	4	15	15 - 8 = 7	15 - 8 - 4 = 3
Average:				22 / 5 = 4.4	7 / 5 = 1.4

SRT is more efficient in this scenario as average turnaround time (4.4) and average waiting time (1.4) are lesser than what SJF generates.

Question 7 - kelvin - ok

Round Robin (A quantum size of 2 time slices)

7. a)



Quantum size = 2

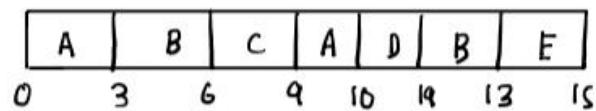
Process	Arrival Time	CPU cycle	Finish time	Waiting time	turnaround time
A	0	4	6	2	6
B	1	5	15	9	14
C	3	3	14	8	11
D	5	1	11	5	6
E	7	2	12	4	6

$$\text{Average Waiting time} = \frac{2+9+8+6+4}{5} \\ = 5.6$$

$$\text{Average turnaround time} = \frac{6+14+11+6+6}{5} \\ = 8.6$$

Round Robin (A quantum size of 3 time slices)

b)



Quantum size = 3

Process	Arrival time	CPU cycle	Finish time	Waiting time	turnaround time
A	0	4	10	6	10
B	1	5	13	7	12
C	3	3	9	3	6
D	5	1	11	5	6
E	7	2	15	6	8

$$\text{Average Waiting time} = \frac{6+7+3+5}{5} \\ = 5.4$$

$$\text{Average turnaround time} = \frac{10+12+6+6+8}{5} \\ = 6.4$$

RR with quantum size of 3 slices will give a more efficient CPU allocation as it has lower average waiting time and lower turnaround time.

Question 8 - LIM FANG CHERN - ok

It is said that the Round Robin scheduling algorithm is one of the most appropriate to be implemented for an interactive system. Do you agree? Justify your answer.

Yes. I strongly agree with the statement above.

Firstly, it has been used intensively in interactive operating systems because it is easy to implement.

Secondly, Round Robin is a preemptive scheduling algorithm. Operating systems can interrupt the currently executing process and allocate CPU time to another process when its time quantum expires or when a higher priority becomes available.

Round robin is a kind of equal CPU sharing and it is not monopolised by any job. It aims to provide equal CPU time to each active process in the system.

Systems are able to respond in a faster period of time as Round Robin scheduling dramatically improves average response time and ultimately it saves more time.

Tutorial 8 Discussions

Question 1 - Gian Terng - ok

Demand Paging	Segmentation
Each job is divided into blocks of the same size	Each job is divided into blocks of different sizes
Each block is known as a Page	Each block is known as a Segment
Page Map Table (PMT) is used to store the mapping information when a job is loaded to the RAM	Segment Map Table (SMT) is used to store the mapping information when a job is loaded to the RAM
Paging may lead to internal fragmentation	Segmentation may lead to external fragmentation

Question 2 - Guan Jun - ok

Advantages of Virtual Memory:

- Increased RAM Capacity:** Virtual memory allows a computer to use more memory than physically available by utilizing a portion of the hard drive as additional RAM. This extends the capacity of the system to run larger applications and handle more data.
- Multi-Tasking Support:** Virtual memory enables multitasking by providing a way for multiple programs to share limited physical memory. It allows the operating system to allocate memory efficiently to different processes, enhancing overall system performance.
- Isolation and Security:** Virtual memory provides memory protection, isolating processes from one another. This prevents one misbehaving program from accessing or modifying the memory space of another, enhancing system stability and security.

Disadvantages of Virtual Memory:

- Performance Overhead:** Utilizing virtual memory incurs a performance penalty because accessing data from the hard drive is slower than accessing it from physical RAM. Frequent swapping of data between RAM and the disk can lead to performance bottlenecks → Thrashing
- Complex Memory Management:** Virtual memory systems are more complex to manage than systems with only physical memory. This complexity can lead to issues like fragmentation, where free memory spaces become scattered, potentially wasting resources.

3. **Storage Space Usage:** Virtual memory uses a portion of the hard drive to store data temporarily, which can consume significant disk space. This might be a concern on systems with limited storage capacity, especially if large swap files are created.

Question 3 - Hui Yi - ok

3a) Demand Paging Memory Allocation

b)

Simplicity

Fixed-size paging is a **relatively simple** memory management technique. It divides physical memory and virtual memory into fixed-sized chunks or pages. This simplicity makes it easier to implement and manage.

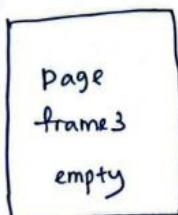
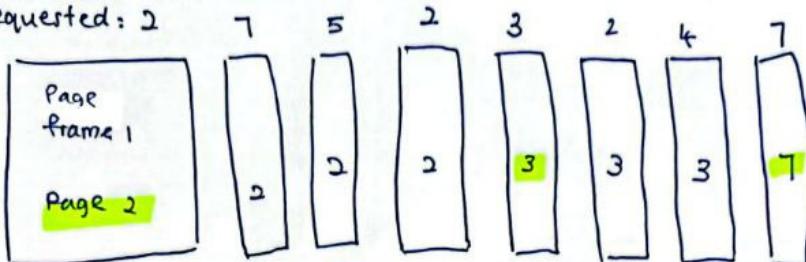
Eliminate External Fragmentation

It also completely **eliminates external fragmentation** because with fixed-size paging, each page is of the same size, and they are allocated and deallocated independently.

Question 4 - Jia Wen

Question 4 (FIFO)

Page
requested: 2



Interrupts: • • • • • • • •

Time Snapshot: 1 2 3 4 5 6 7 8

∴ when page 7 is requested again, it's already in memory (frame 2). In the FIFO algorithm, the page that was loaded earliest and is at the front of the queue (oldest) should be replaced when a page fault occurs. Therefore, Page 3, which was the first to be loaded into frame 1, is the first page to be swapped out.

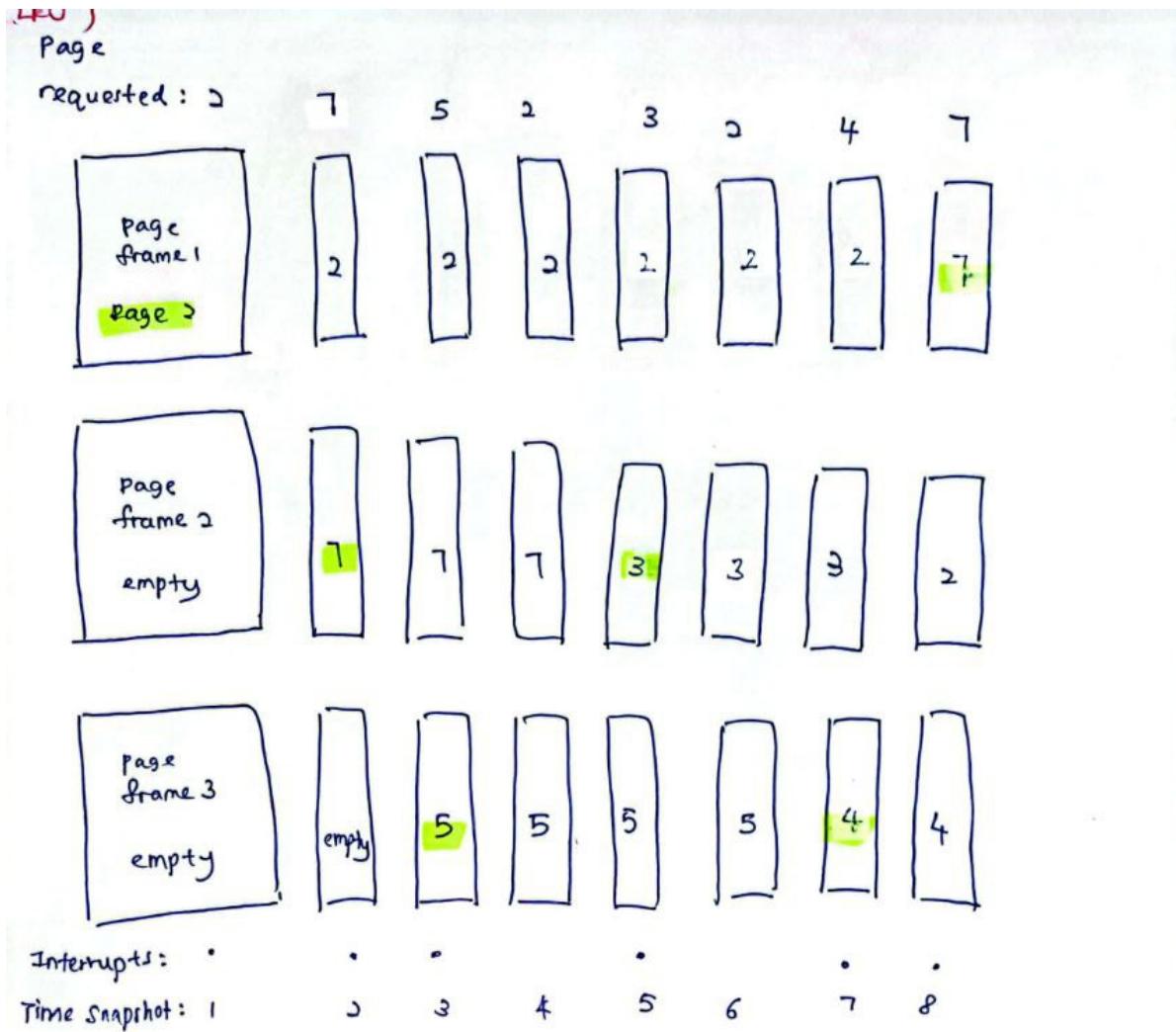
$$\text{Failure rate} = \frac{7}{8} = 87.5\% \quad \text{success rate} = 12.5\%$$

(a) FIFO - ok

Page 2 is the first page to swap out.

8 requests with 7 failures

hence , the failure rate = 87.5%, success rate = 12.5%



\therefore The first page would be swapped out by LRU is page 5 because it was the least recently used page among those in memory when a page fault occurred.

$$\text{Failure rate} = \frac{6}{8} = 75.0\%, \text{ success rate} = 25.0\%$$

(b) LRU - ok

Page 7 is the first page to swap out.

8 requests with 6 failures

hence , the failure rate = 75%, success rate = 25%

c) - ok

Efficiency in page replacement algorithms can be assessed in terms of the number of page faults generated. In this specific sequence, we can see LRU is more effective than FIFO which generated the higher number of page faults.

Therefore, based solely on the given sequence, LRU appears to be more efficient than the FIFO. However, it's important to note that the efficiency of these algorithms can vary depending on the access patterns of different programs and sequences of requests.

Question 5 - Jin Huei - incorrect, please reupload

Three page frames implementation :

Page request	3	1	1	2	4	5	5	4	8	2
Newest Page	3	1	1	2	2	2	2	4	4	4
		3	3	3	3	5	4	4	4	2
Oldest Page				1	4	4	5	5	8	8

The total number of page fault of three page frames implementation is 9.

Four page frames implementation :

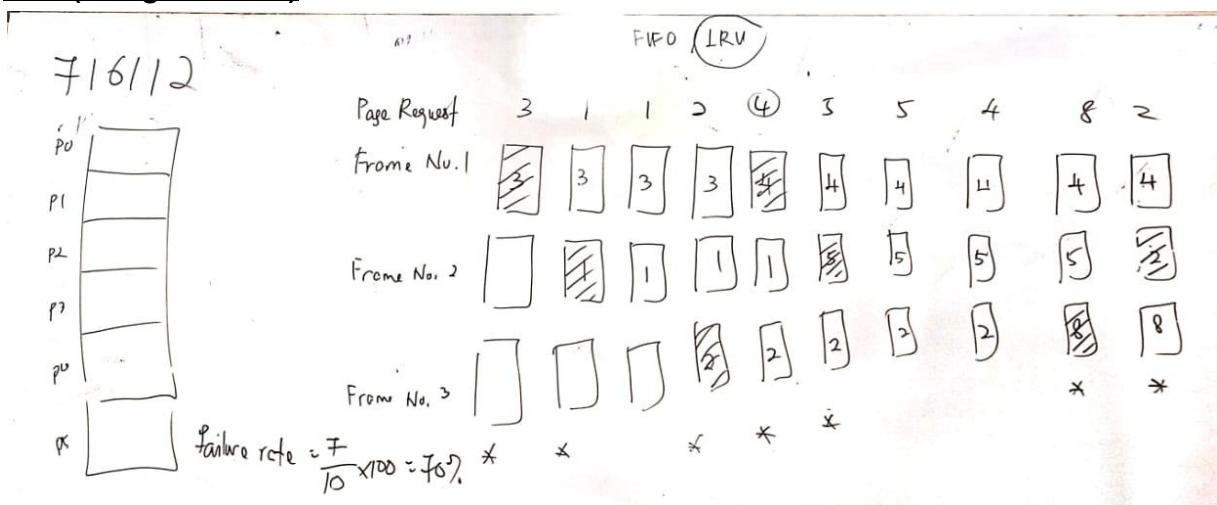
Page request	3	1	1	2	4	5	5	4	8	2
Newest Page	3	1	1	1	1	1	1	4	4	4
		3	3	3	3	3	3	3	8	8
				2	2	2	2	2	2	2
Oldest Page					4	5	5	5	5	5

The total number of page fault of four page frames implementation is 8.

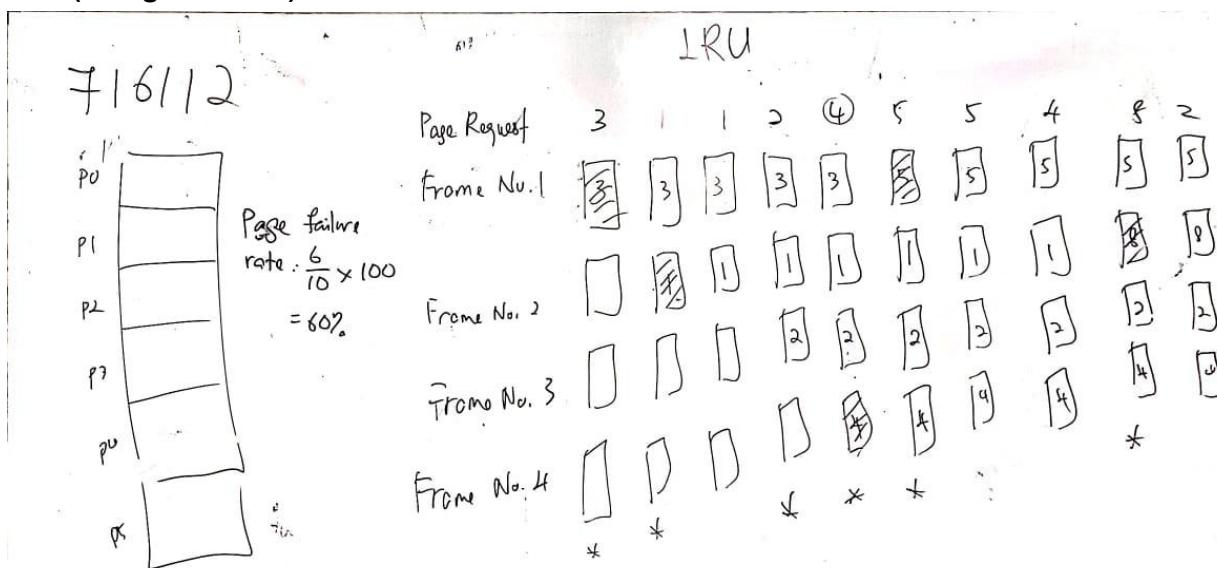
Three page frames implementation : $9/10 * 100\% = 90\%$

Four page frames implementation : $7/10 * 100\% = 70\%$

LRU (3 Page Frames)



LRU (4 Page Frames)



Question 6 - Jun Hong

Page size given = 100 bytes

Page Number = Virtual Address / Page size

1. $102 / 100 = 1$
2. $324 / 100 = 3$
3. $397 / 100 = 3$
4. $532 / 100 = 5$
5. $643 / 100 = 6$
6. $443 / 100 = 4$
7. $623 / 100 = 6$
8. $554 / 100 = 5$
9. $743 / 100 = 7$
10. $488 / 100 = 4$

FIFO:

Page request	1	3	3	5	6	4	6	5	7	4
Newest Page	1	3	3	5	6	4	4	4	7	7
	1	1	3	5	6	6	6	6	4	4
Oldest Page				1	3	5	5	5	6	6

The total number of page faults of the FIFO algorithm is 6.

Percentage of page faults occurred:

$$(6/10) * 100\% = 60\%$$

LRU:

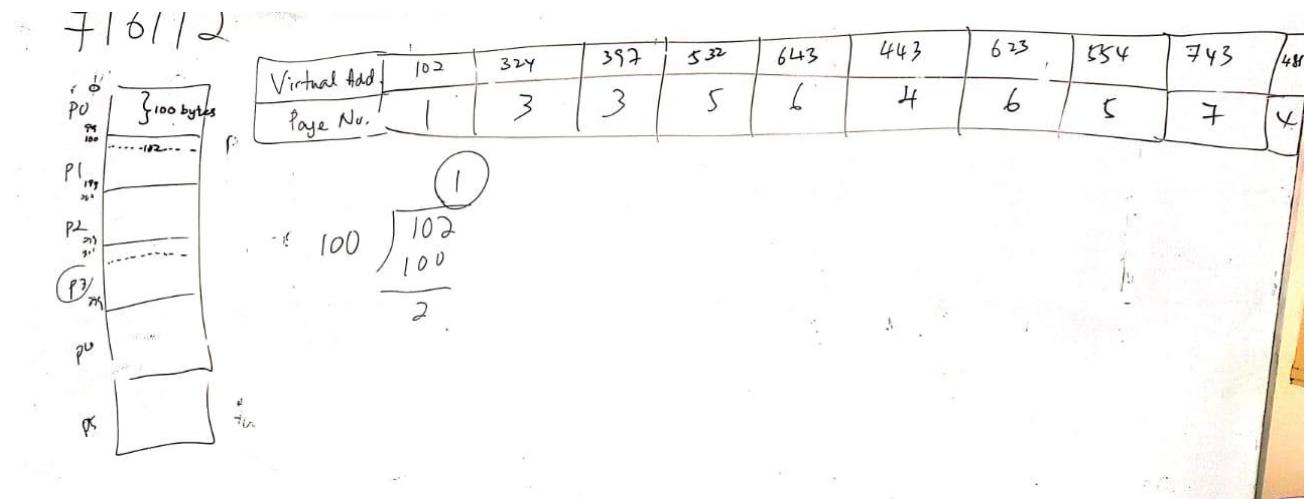
Page request	1	3	3	5	6	4	6	5	7	4
Newest Page	1	3	3	5	5	5	5	5	5	5
	1	1	3	3	4	4	4	4	7	7
Oldest Page				1	6	6	6	6	6	4

The total number of page faults of the LRU algorithm is 7.

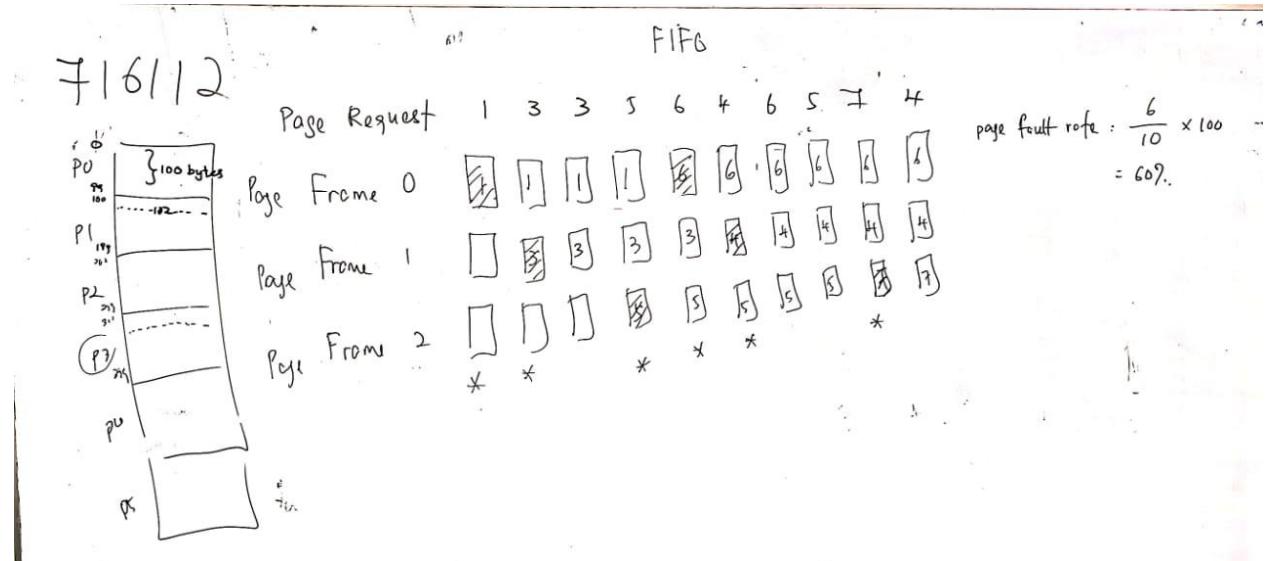
Percentage of page faults occurred:

$$(7/10) * 100\% = 70\%$$

Page Number Computation



FIFO



LRU

