Computer Science

95/20

Midterm (second)

BIRZEIT UNIVERSITY
College of Information Technology
Computer Science Department

Name: Ayah Belal Number: 1100985

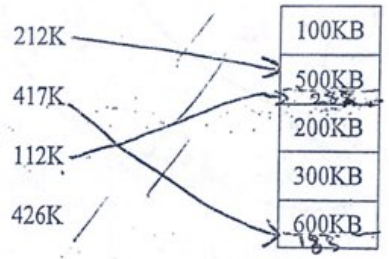
Comp431 **Midterm Exam** Fall 13/14

---

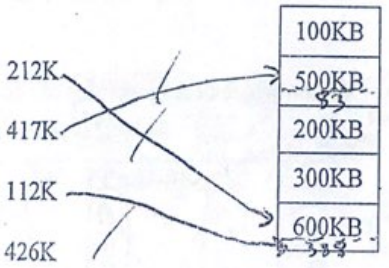
[1] In dynamic partitions memory management, given the *holes* :
100KB, 500KB , 200KB , 300KB , 600KB (in order)

Given the processes 212K , 417K , 112K , 426K (in order) . Draw arrows showing where you place each process in the following allocation algorithms.

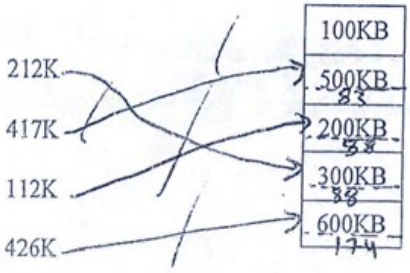
1) First-Fit



2) Worst-Fit



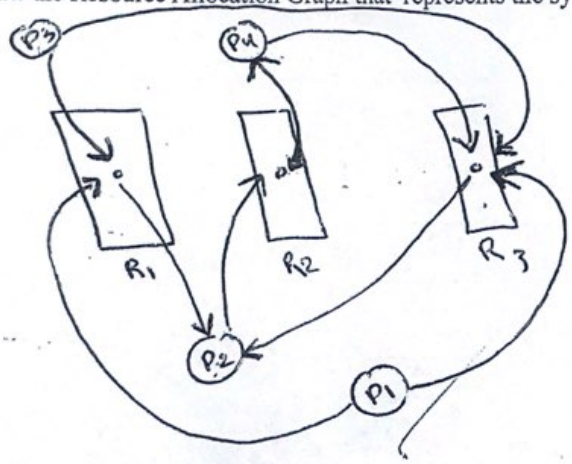
3) best-Fit



[2] A system is composed of four processes, p_1, p_2, p_3, and p_4, and three types of resources, R_1, R_2, R_3. There is one unit of each resource type available.

- p_2 assigned a unit of R_1 and a unit of R_3.
- p_4 assigned a unit of R_2 and requests one unit of R_3.
- p_2 requests one unit of R_2.
- p_1 requests a unit of R_1 and a unit of R_3.
- p_3 requests a unit of R_1 and a unit of R_3.

(a) Draw the Resource Allocation Graph that represents the system state.



(b) Is the system is safe ? If not, which of the processes are deadlocked?

safe state ⇒ safe sequencess لازم تكون
: sequencess of process.

it is not a safe state, there is a deadlock.

the deadlock is :- $P_2 \rightarrow R_2 \rightarrow P_4 \rightarrow R_3 \rightarrow P_2$.

- if the process 2 does not hear then deadlock will not be a deadlock.

[3] (a) If the LA is 20 bits long and given the LA =(00000101000100000111)$_2$, and page size = 4096 = 2^{12} word, and given the page table :

- What is p and d without using the / and % operations.
- compute PA

0	10
1	150
2	840
3	122
4	40
5	20
	10
	4096
	.
	100

page table

① page size = 4096 = 2^{12}

⇒ d = 12

LA = P + d ⇒ 20 = P + 12

⇒ P = 8

LA: | 8bit (P) | 12bit (d) |

5	263
P	d

② PA = S + F + d

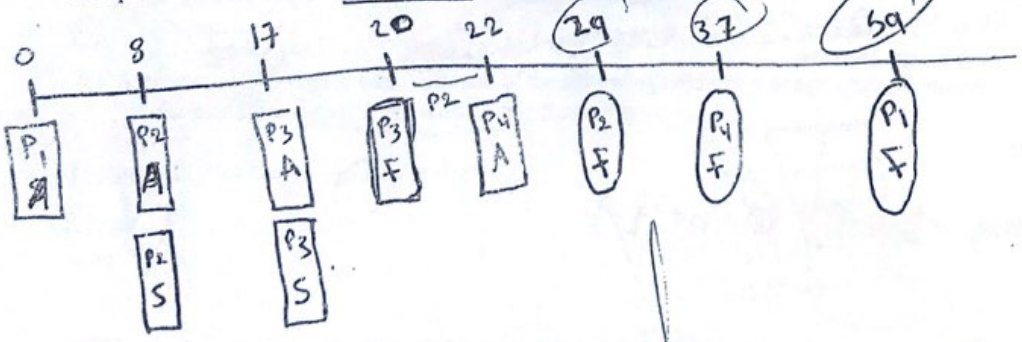
PA = 4096 * 20 + 263 = 2^{12} * 20 + 263

= 82183.

(b) Assume the ready queue looks like:

Process	CPU Burst	Arrival Time
P_1	22	0
P_2	9	8
P_3	3	17
P_4	2	22

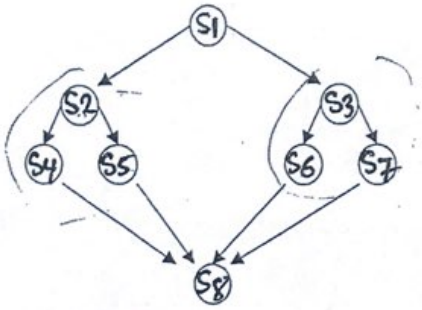
Compute the AWT in the preeptive SJF scheduling algorithm.



AWT = $\frac{(59-0-30)+(29-8-20)+(20-17-3)+(37-22-8)}{4}$

AWT = $\frac{29+1+0+7}{4} = \frac{37}{4} = 9.25$

[4] Given the precedence graph:



(a) Write an equivalent code using **parbegin & parend**

S1;
Parbegin
begin
S2;
Parbegin
S4;
S5;
Parend
end
begin
S3;
Parbegin
S6;
S7;
Parend.
end
Parend
S8;

(b) Given the following statements which will be executed in the same order:

S_1 : **int x=1 , y=100;**
S_2 : **x += 10;**
S_3 : **y++;**
S_4 : **int z;**
S_5 : **z = x+y;**
S_6 : **cout <<z+x;**
printf.

(a) compute :
The write set , $W(S_6)$ = ∅ or { }.

The read set , $R(S_5)$ = {x, y}.

[5] (a) Give
int sem

[5] (a) Given the only data structure declaration:
int semaphore S = 1;

Define the *wait(S)* and *signal(S)* instructions.

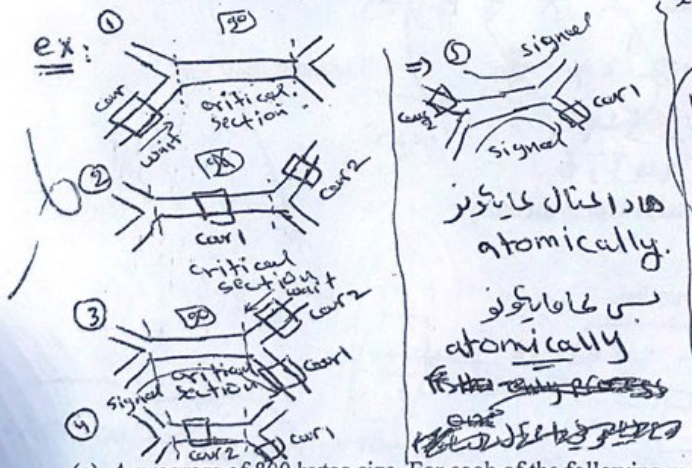
wait → entry section

Signal → exit section

```
wait(S){
  S = S - 1
}
```

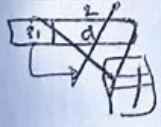
```
signal(S)
{
  S = S + 1
}
```

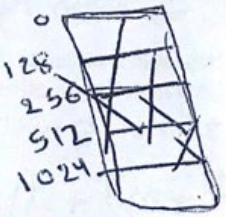
(b) Show that if the *wait(S)* and *signal(S)* semaphore operations are not executed **atomically**, then mutual exclusion may be violated.



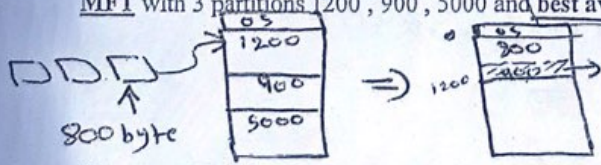
(c) A program of 800 bytes size. For each of the following memory management, Compute the amount **internal fragmentation** if there is any :

Paging with page size 2^6 → 64 bytes

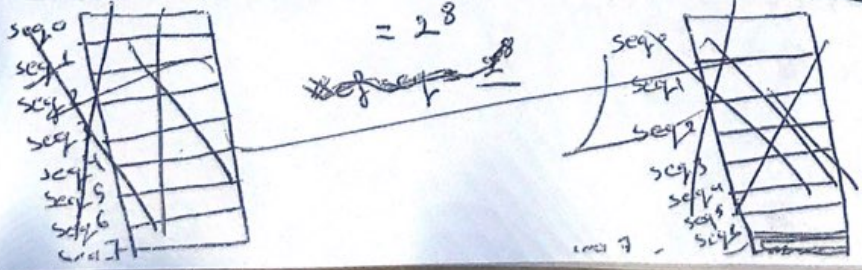




MFT with 3 partitions 1200 , 900 , 5000 and **best available fit** only algorithm.

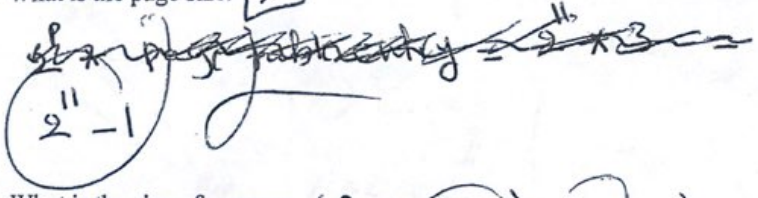


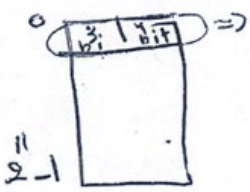
Segmentation with equal segments size = 256 bytes



[6] (a) On IBM 370 OS with paging memory management, the logical address is 24 bits, 11 bits for the page number and 13 bits for the offset. The page table entry is 3 bytes long containing 20 bits for the frame number, and 4 bits for the **legal/illegal** bit, **V/I** bit, **dirty** bit, and **reference** bit.

1. What is the page size. 2^{13}





2. What is the size of memory. (P memory)

Size of memory $= 2^{13} * 20$

$= 163840$.

3. What is the size of page table.

Size of page table $= 2^{p} * 3$ bytes

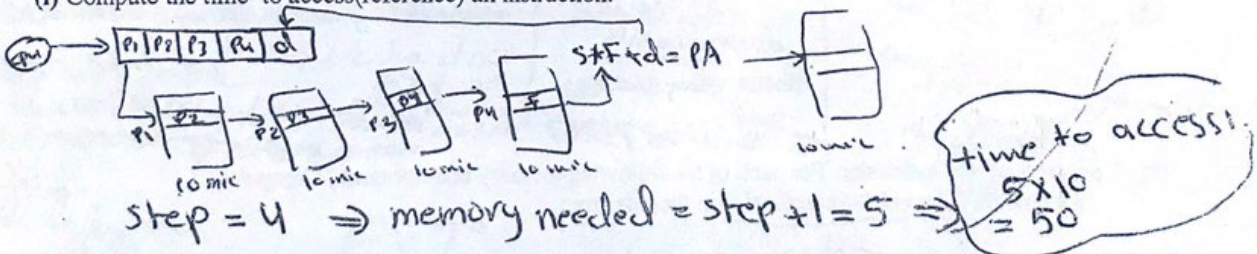
$= 2^{11} * 3$

$= 6144$.

$= 14336$

(b) In demand paging system with 4-levels page table kept in **memory**,
Assume memory access = 10 mics

(i) Compute the time to access(reference) an instruction.



step = 4 $\Rightarrow$ memory needed = step + 1 = 5 $\Rightarrow$

time to access:
5 x 10
= 50

(ii) If an associative registers table is added to the system with **hit ratio** 98%. Compute the EAT given that the lookup time in the associative registers is 1 mics.

$t = 1$ mics, $m = 10$ mics, $h = ,98$

$EAT = h(t+m) + (1-h)(t+5m)$.

$= ,98(1+10) + (1-,98)(1+5*10)$.

$= 10,78 + 0,02 * 51$

$= 10,78 + 1,02$.

$= 11,8$.

86

Second

جامعة بيرزيت

second

Birzeit University - Computer Science Dept.

Name مصعب البواجة Number 9630320

Comp 431 Exam. #2 29/01/2007

Question (1): Define the following :

(a) Dirty Bit : a bit added to the page table entry which indicates weather the page is modified or not.

(b) Valid/Invalid Bit : a bit add to page table entry which show legal address and if the used page is in memory or

(c) PTBR : page table base register: contains the address of beginning of page table in memory.

(d) Page Fault Rate

the percentage of time of that the page will not be in the associative registers. memory

[6] (a) On IB
the page
bits for

Question (2)

In Paging memory management , if virtual address is 20 bits long with page size 256 bytes. If physical memory is 64MB and given the following page table:

2^8

(a) What is the maximum size of programs executed ?

12	8

$2^{12} \times 2^8 = 2^{20} = 1MB$

Virtual

0	10
1	150
2	810
3	16
4	2480
5	5000
6	66
.	.
.	.
.	.
100	

(b) Compute the size of the page table ?

$d = 8$, $p = 12$

Ph. mem: $\frac{2^{12} * 64}{2^8} = \frac{2^{12} \times 2^6}{2^8} = 2^{10}$ ⇒ to bits

2^{10} ⇒ 10 bits ⟶ need 2 bytes

$2^{10} \times 2 = 2^{11} = 2K$

1557

(c) Given the virtual address in Octal $(3025)_8$. Compute the Physical Address (PA) .

0 11 000 010 101

$d = 00010101 = 21$

$p = 011 = 6$

1557

$Ph.A = 66 * 256 + 21$

(P): m divs

(d): m mods

1024
512
21
1557

[6] (a) On IBM 370 OS with **paging** memory management, the logical address is 24 bits, 11 bits for the page number and 13 bits for the offset. The page table entry is 3 bytes long containing 20 bits for the frame number, and 4 bits for the **legal/illegal** bit, **V/I** bit, **dirty** bit, and **reference** bit.

24 bit
LA = 24

1. What is the page size.

page size = 2^d

$\Rightarrow d = 13 \Rightarrow$ page size = $2^{13} = 8192$

2. What is the size of memory.

Memory size = 2^{LA}

$\Rightarrow 2^{24} = 16777216$ bit

3. What is the size of page table.

Size of page table = $2^{11} * 20$

$= 40960$ bit

$= 2^P *$ entry bits $= 40960$

In A

(b) ~~A demand~~ **paging** system with **4-levels** page table kept in **memory**,
Assume memory access = 10 mics

(i) Compute the time to access(reference) an instruction.

The Time = $10 * 4 = 40$ unit
↳ 40 mics

(ii) If an associative registers table is added to the system with **hit ratio** 98%. Compute the EAT given that the lookup time in the associative registers is 1 mics.

$EAT = h * (t+m) + (1-h) * (t + 5m)$

$= 0.98(11) + 0.02 * 51$

$10.78 + 1.02$

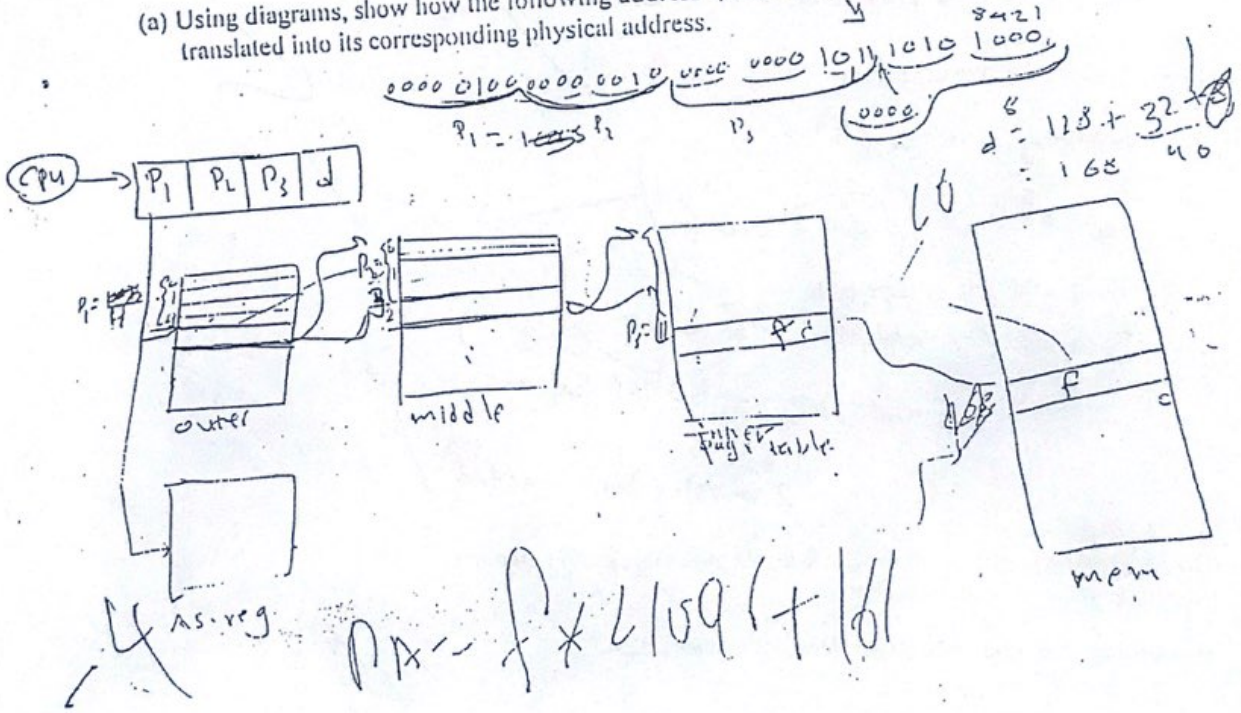
$EAT = 11.8$ unit
↳ 11.8 mics

Question (3)

In a paging system, if the virtual address is 40 bits long with page size 4096 bytes. Assume with 3-levels page table, 8 bits are used for the first outer page table, 8 bits for the middle page table, and the remaining number of bits for the page table entries.

8	8	12	12

(a) Using diagrams, show how the following address $(0A0200B0A1)_{16}$ in Hex is translated into its corresponding physical address.



(b) If associative registers are used with Lookup time 10 mics with hit ratio 95%. Assume memory access is 5 mils. Compute the Effective Access Time (EAT).

$h*(t+m) + (1-h)*(t+2m)$

$EAT = 0.95(10 + 5000) + 0.05(10 + 2*(5000))$

Question (4)

Given the following page references:

2 3 1 3 2 7 3 1 3 2 5 2 1

Compute the number of page faults with 3 frames of memory in the following replacement algorithms:

(a) LRU

2	3	1	3	2	7	3	1	3	2	5	2	1
2	2	2	✓	✓	2	✓	1	✓	1	5	✓	2
	3	3			3		3		3	3		1
		1			7		7		7	7		2

8 page faults.

(b) Optimal

2	3	1	3	2	7	3	1	3	2	5	2	1
2	2	2	✓	✓	7	✓	✓	✓	2	2	✓	✓
	3	3			3				3	5		
		1			1				1	1		

6 page faults

74/100

جامعة بيرزيت

Birzeit University
Computer Science Dept.

NAME : Malik Taha . NUMBER : 980008
Comp 431 Midterm #2 20/06/2001

[1]
(a) Given the following page references:

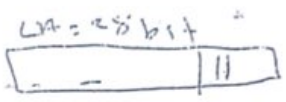
1 2 3 1 3 4 2 5 3 6 5 1 2 1

With 3 frames ,compute the number of page faults using the Optimal replacement algorithm:

1	2	3	1	3	4	2	5	3	6	5	1	2	1
1	1	1	-	-	4	-	5	-	5	-	1	-	-
	2	2			2		2		2		2		
		3			3		3		6		6		

7 Page faults rate.

1
1 1
2 2
3 3



16 8 4 2 1

20 2

b) In a system that uses Segmentation with Paging (paging the segments), if the virtual address is 28 bits. Page size is 2048 bytes. Segments size is 1 MB

Given the virtual address 050A846 in Hex. Using diagram show how the given virtual address Is translated into physical address. Fill any missing details you need.

LA = 28 bits

Page size = 2048 = 2^{11}

segment size = 1 MB

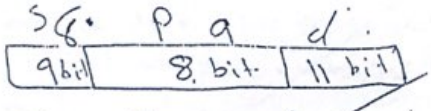
LA = 050A846

= 0000 0101 0000 1010 1000 0100 0110

d = 11 bits

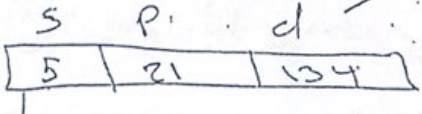
segment size = 1 MB $\Rightarrow$ # of segments = $\frac{1\text{ MB}}{2^{11}} = \frac{2^{20}}{2^{11}} = 2^{9}$

need 9 bits then



d = 70 = 000 0100 0110

= 134

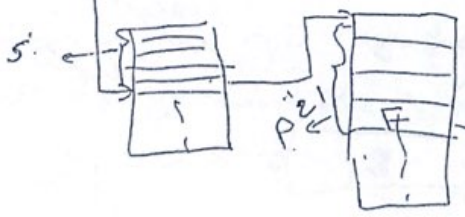


P = 21

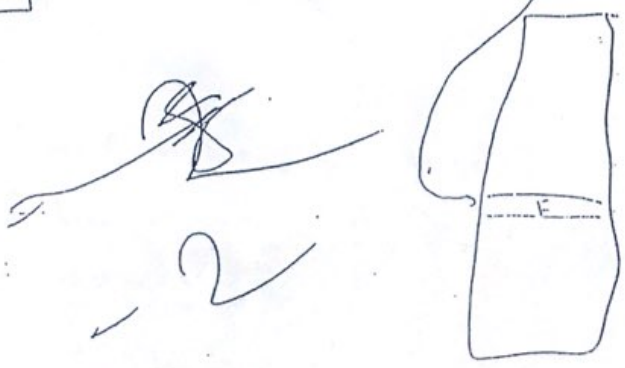
d = 5

16 8 4 2 1

0 0 0 1 0 1 0



PA = F x 2^{11} + 134



[2]

(a) Explain why the page size is a power of 2 (2^n). Support your answer by an example.

when the cpu generate an address it divided it to two component page # and page offset it calculate:

P = LA div page size

d = LA mod //

but div and mod operation is very cost by the OS and take time the OS take advantage from size 2^n because

d = n order low Bit from LA

P = the remaining bit from high order from LA

ex) LA = 20 bit

sizeof page = 256 = 2^8 then

d = 8 bits

P = 12 bits

example

[3] In a demand paging system, the system status look like:

CPU utilization is 20%.
Disk utilization is 98%.
Other devices 5%

Which of the following do you think will improve CPU utilization ? Justify your answer?

(i) Install a faster CPU.

not ~~necessarily~~ increase the cpu utilization because it depend on memory management if the page fault rate is very large then the cpu work only swap pages I and out then we will get Low cpu utilization

(ii) Install a bigger paging disk ?

not increase cpu utilization

?? why

(iii) Increase the degree of multiprogramming.

not increase the cpu utilization because if we increase the degree of multiprogramming then the page fault rate is increase and cpu doing swaping In and out.

(iv) decrease the degree of multiprogramming.

increase the cpu utilization because the rate of fault to page will decrease and the cpu work afew swapping In and out.

(v) Install a bigger memory.

increase the cpu utilization because this ~~Solution~~ case most of the page in memory then the fault Rate is small amor swapping out and In is little

[4]

scuss briefly MVT and MFT regarding:

(i) main disadvantage?

MFT: Internal fragmentation: the remaining unused space inside a allocated region.

MVT: 1) external fragmentation: the set of holes that is very small not fit any Job.

2) Internal fragmentation.

(ii) degree of multiprogramming.

In MFT: The degree of multiprogramming is boun by the number of fixed region.

In MVT: not bounded by the number of region

(iii) Hardware support.

It needs base and limit register in two MVT and MFT.

(iv) Job scheduling.

In MFT: 1) a Queue of wating Job for each reg generally served by FCFS.

2) one Queue for region served by
   a) FCFS with without skip.
   b) Best fit
   c) Best available fit

In MVT: a) First fit: allocated the first hole that is Big enough to fit the
b) best fit: Smallest region that fit the Job
c) worst fit.

[4] In a demand paging system, memory access is 10 mics. Pages contain 1024 words, the transfer rate is 2^{20} words/sec.
10% of all instructions executed access a page other than the current page.
80% of instructions that access another page are already in memory
70% of the replaced pages are modified.

Compute the EAT

m = 10 μs

EAT = (1-p) * m + p * (page fault overhead)

the transfere time per = $2^{10} \times \frac{1}{2^{20}}$ = 980 μs

= 0.8 X 10 + 0.2 (980 + 0.7 X 980 + 10)

= 8 + 335.2

= 343.2 Ms

83

BIRZEIT UNIVERSITY

Computer Science Deptartment

NAME: Laila Rimawi NUMBER: 990327

Comp 431 **Midterm #2** 1/06/2003

[1]

(a) Briefly explain the difference between logical and physical address.

Logical address: It is an address generated by the CPU, it's also called virtual address

Physical address: It's the address that is seen by the Memory unit.

(b) When the process CPU burst is finished, the process looses the CPU to another process. Describe two more cases where the process looses the CPU.

(1) The process may loose CPU if a higher priority process or job arrives.

(2) When the context switch time is over, so the CPU switches to another job.

(3) when swapping is need to resume execution.

(c) Given the page size = 1024 words, physical memory is 64 frames. Then,

The LA contains ---------- 15 ---------- Bits.

The PA contains ---------- 16 ---------- Bits.

LA: $1024 = 2^{10} \Rightarrow d = 10$

program size = 32 pages

$32 = 2^5 \Rightarrow p = 5$

so LA = 5 + 10 = 15

64 frames

$64 = 2^6$

so F = 6 bits and d = 10

$\Rightarrow$ PA = 6 + 10 = 16 bits

(d) In simple words, what does the page fault means ?

Page fault means. The page to be referenced is not in memory, it involves swapping. In page fault at least one swapping is involved. It could be 2 swapping (in + out) if a victim is selected.

(e) Rank the following page replacement algorithm from bad to good.

LRU , Second Chance , Optimal , FIFO

FIFO

LRU

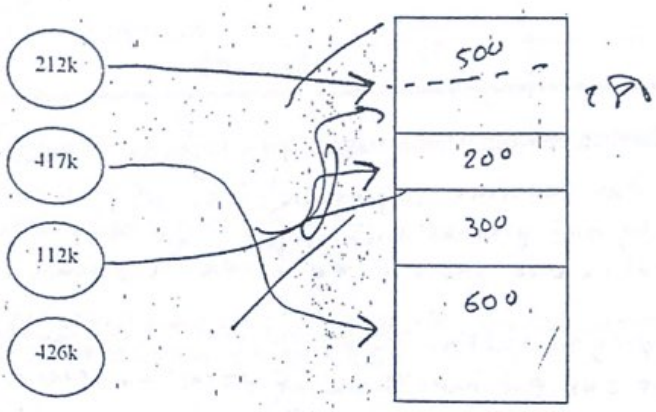
Second Chance

Optimal

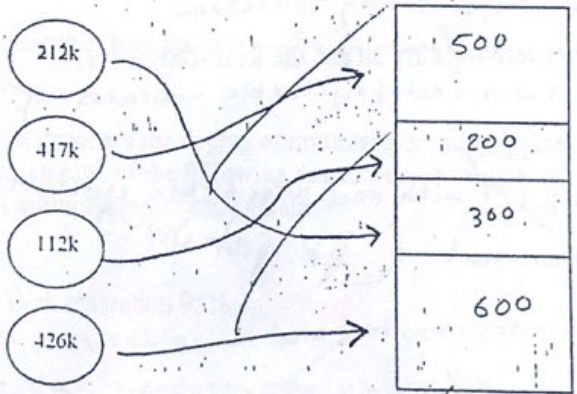
[2]

In MVT (Dynamic Regions), assume we have the following holes: 500k , 200k , 300k , 600k and also assume we have the following processes 212k . 417k , 112k , 426k in order. Show using diagrams only how you place these processes using :

(a) First Fit



(b) Best Fit



(c) Worst Fit