46

جامعة بيرزيت
BIRZEIT UNIVERSITY

Computer Science Department

NAME : [redacted] NUMBER : [redacted]

Comp 431 Exam # 2 25/05/2004

Question 1:

(1) The difference between internal fragmentation in paging and segmentation is :

* internal fragmentation in paging : In MFT it caused because ever req take only one process but in MVT we can put more than on process in region but in two cases there is internal fragmentation.

* internal fragmentation in segmentation : is caused because in one segment we can put more than one process so after a while we notice that there are a number of holes but we can solve this problem by compaction these holes during execution

(2) The difference between the dirty bit and the Read-Only bit is :

* dirty bit : bit with each page table indicates if page table is modified or not

* Read_only bit : bit with each page table indicates if page table is read only or not (for protection)

(3) Two advantages of paging are :

① decrease external fragmentation

② sharing paging

(4) - The page replacement algorithm which gives the ideal (optimum) solution is optimal Algorithm and its main disadvantage is [redacted]

- The LRU page replacement algorithm is to replace the page which Least recent use page table

- The best implementation of LRU is using using queue and put any page used in the rear of the queue.

(5) The allocation methods in MVT are:

i) Best fit

ii) available fit

iii) worst fit

Also, the poorest one is worst fit

(6) Assume a memory of m frames (initially empty), the page references during execution is of length n, and among them k references are distinct, $k \leq n$.

- The minimum number of page faults is K
- The maximum number of page faults is K (n-1)

(7) In a demand paging system where the degree of multiprogramming is fixed at 4. the status of the system is measured, in each of the following cases, what is happening, can we increase or decrease the degree of multiprogramming, explain:

a. CPU utilization 15%, Disk utilization 95%

in this case cpu utilization is low, because rate of page fault is high and we can increase cpu utilization by

① decrease the degree of multiprograming

② Add a bigger memory

③ Increase pagesize

b. CPU utilization 15%, Disk utilization 5%.

In this case cpu utilization is also low so increase cpu utilization we should increase Disk utilization and decrease the degree of multiprograming and Increase page size.

Question 2:

In a demand paging system where the page table is kept in memory with a set of associative registers. 90% of all accesses in the associative registers, page transfer time is 10 mils, memory access is 100 mics. 20% of all accesses cause a page fault. Compute the EAT ignoring the lookup time in the association register.

EAT = (1 − h)(m) + h(page fault * 2 tranfertime + (1-pagefault) * transfer time + m)

= (1 − 0.9)(100) + 0.9(0.2 * 2(10) + (1-0.2) * (10 + 100))

= (0.1)(100) + 0.9(4 + 8 + 100)

= 10 + 0.9 (112)

= 10 + 100.8

= 110.8

$t = 10 \times 10^{3} = 1000$

$m = 100$

$p = 20\%$

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

EAT = (1-p)m + p(cost fault + access time)

cost fault = swap in + swap out

EAT = 46 + 25 = 71

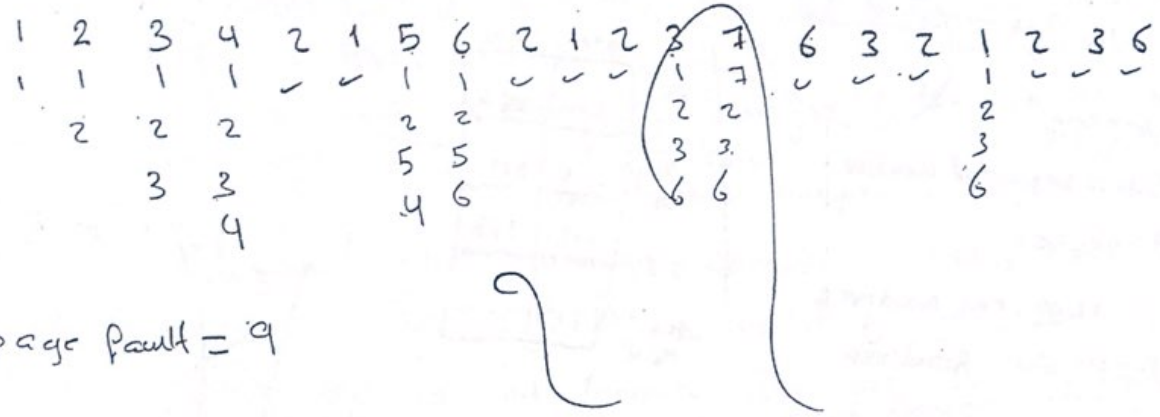
Question 3:

In executing a certain process, we have the following page references:

1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6

How many page faults there is with 4 frames of memory:

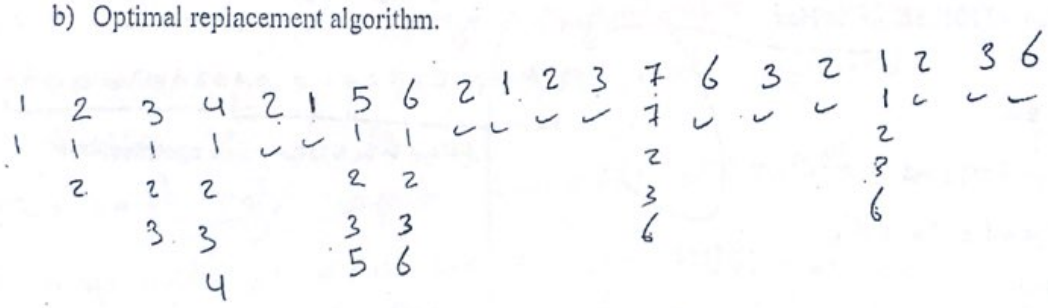
a) LRU replacement algorithm.



page fault = 9

10

b) Optimal replacement algorithm.

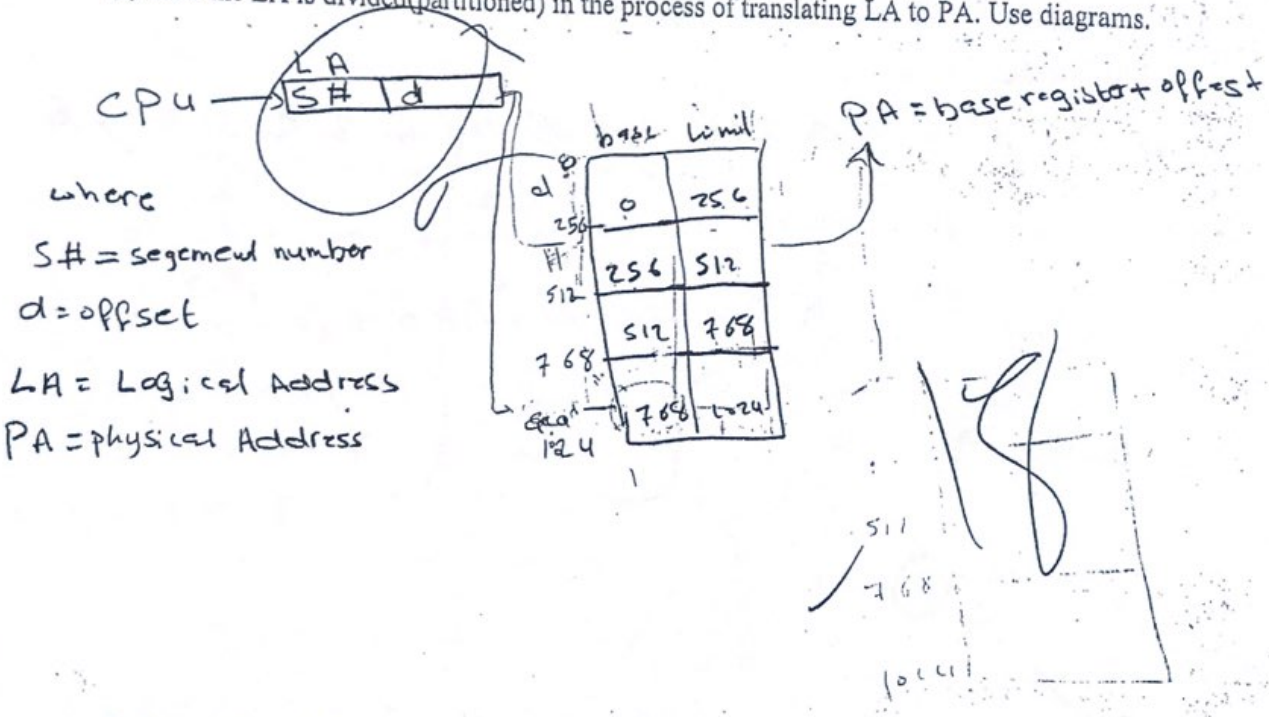


Page fault = 8

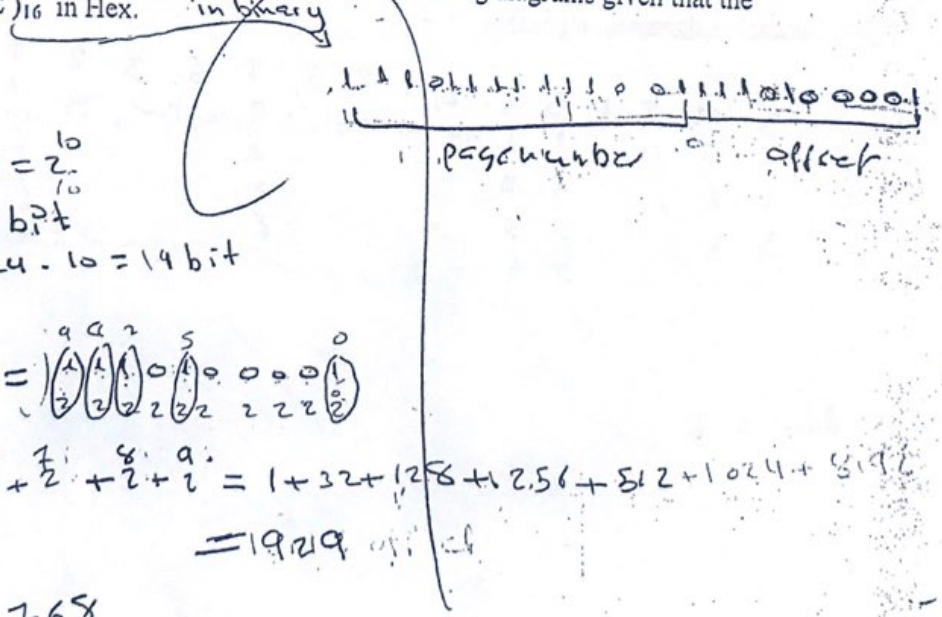
Question 4:

A computer system with 24 bits logical address and page seize = 1024 Byte. Assume segmentation with paging (paging the segments) is used with equal segments size of 256 kb.

(a) How the LA is divided(partitioned) in the process of translating LA to PA. Use diagrams.



(b) Explain how the LA is translated into PA in that case using diagrams given that the LA = $(10185E)_{16}$ in Hex.



$LA = 2^{24}$

$Page\ size = 1024 = 2^{10}$

So offset = 10 bit

page number = 24 - 10 = 14 bit

offset = $1\,0\,0\,0\,0\,1\,0\,1\,1\,1\,0$

$2^1 + 2^2 + 2^3 + 2^7 + 2^8 + 2^9 = 1 + 32 + 128 + 256 + 512 + 1024 = 1953$

$= 1929$

So base = 768

So

PA = 768 + offset

= 768 + 929

= 1697

[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 ? Explain your answer in full.

(i) Install a faster CPU.

Installing a faster CPU will not improve CPU utilization because as given in the question CPU utilization is somehow low and so no need to install a faster CPU because CPU utilization is low, also, we are concerned with decreasing the disk utilization so as to improve CPU utilization and so installing a faster CPU won't help us in this case because swapping is high.

(ii) Install a bigger paging disk ?

This will not improve CPU utilization because in this case the disk utilization will be very high, resulting in more and more swapping and so will increase the page fault rate and so CPU utilization might even be decreased or reduced instead of getting improved ?

(iii) Decrease the degree of multiprogramming.

This will improve CPU utilization because if degree of multiprogramming got decreased, this will lead to less swapping and so decreasing the page fault rate and eventually CPU utilization will get higher and improved

(iv) Increase page size.

This will not improve the CPU utilization because at this time things will get even worse, the swap time will increase and so more page fault overhead is occurring, which will of course decrease the CPU utilization.

(v) Install a bigger memory.

This will increase CPU utilization, because if memory size increased this will lead to more pages being in it and not swapped from the backing store and so in this case swapping rate will decrease and page fault will decrease and eventually CPU utilization is increased and improved

[5]

In a certain computer, the virtual address is 28 bits with page size is 4096 bytes. The page table is implemented using two-levels with 6 bits for the outer page table and 10 bits for the inner.
Given the virtual address in binary: 0010 0000 0001 1101 0000 1001 0110

Explain using diagrams how the physical address is computed from the virtual address.

LA is 28 bits

page size = 4096 bytes

two levels of page Tables.

6 bits for outer page table

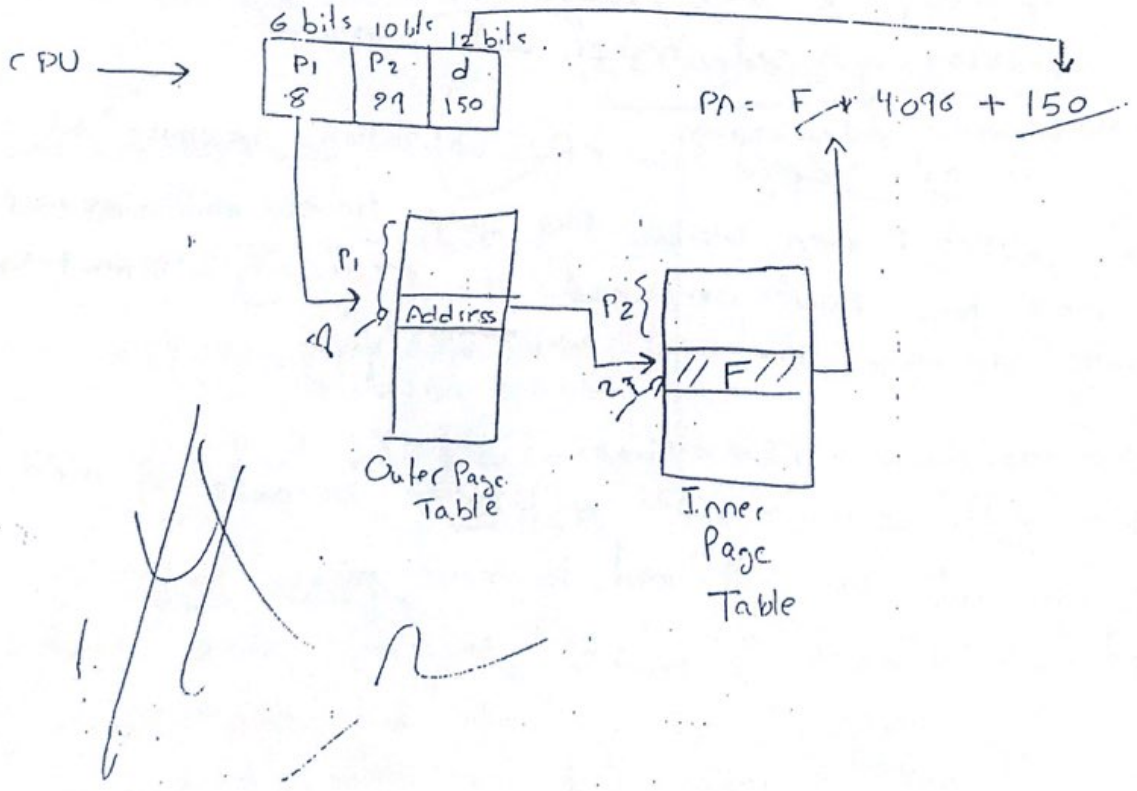
10 bits for inner page table.

First, to calculate the (d) offset.

page size = 4096 = 2^{12}

so d = 12 bits

P1	P2	d
0 0 1 0	0 0 0 0 0 0 0 1 1 0 1	0 0 0 0 1 0 0 1 0 1 1 0
= 8	= 29	= 150



[4] In a demand paging system with page table stored in memory.

(a) If memory reference is 500 nanS. How long does it take to reference an instruction.

memory-access = 500 nans

EAT = ??

$\frac{100}{100} \times 2 \times 500 = 1000$ nans

it's 100% accessing the memory and no hit ratio is involved.

(b) If we add associative registers, and 80% of all page references are in the associative registers. Compute the **Effective Access Time** assuming the associative registers lookup time is **T**.

hit ratio (h) = $\frac{80}{100} = 0.8$

Effective Access Time EAT = ??

EAT = $h(\text{mem-acc} + \epsilon) + (1-h)(2 \times \text{mem-acc} + \epsilon)$

$= 0.8(500 + T) + 0.2(2 \times 500 + T)$

$= 400 + 0.8T + 200 + 0.2T$

$= 600 + T$ nans

So the effective access time in terms of register lookup time = $600 + T$ nans

where T is register lookup time.

96/100

BIRZEIT UNIVERSITY

College of Engineering and Technology

Computer Science Department

Name Amin Onba Number 1111935 Section (10-11) 1-2

Comp431 Short Exam 1 Fall 15/16

Question (1) : 48 points

Fill in the blanks below:

(1) The OS acts as an ---intermidate (interface)--- between the running program and the bare hardware machine.

(2) The software that interprets the machine language instructions is --Microprogramming--

(3) In spooling the I-O of one job is --overlape to another job-- overlap to excution another job

(4) Two types of multiprocessor systems are --Tightly coupled-- and --distributed coupled / (oosly coupled)--

(5) The table generally stored in low-memory that contains pointers to some service routines is called --Interputs vector--

(6) The type of I/O in which after the I/O starts and control returns to the user program without waiting for the I/O to complete is called --Asynchronous--

(7) The OS services are provided in two basic methods, --system call--, and --system program--

(8) The major parts of the process are ----data section----, ---code section---, and ----stack----.

4

Question (2) : 27 points

(a) Define a **privileged** instruction ?

priviley instruction that can exeute only on the monitow mode.

Which of the following is a **privileged** instruction?

Read the computer clock Non privileged

Set the computer clock Non privileged

Set the computer timer --privileged

Switch from user to monitor mode --privileged

Clear memory --privileged

(b) **Cache** memory is:

copying information and data into high speed device. more performance

Examples of cache in the system :

(1) Register cash for memory

(2) memory is cash the hard disk

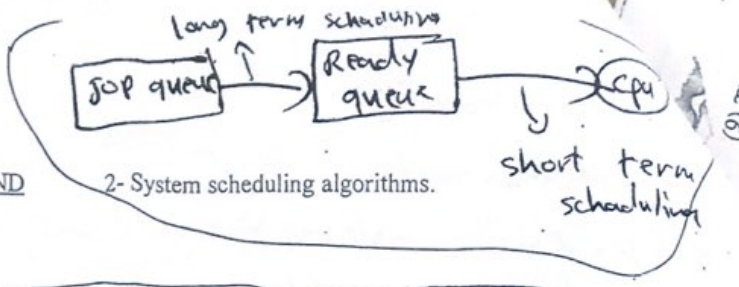
(c) What is a **trap** ? Give example .

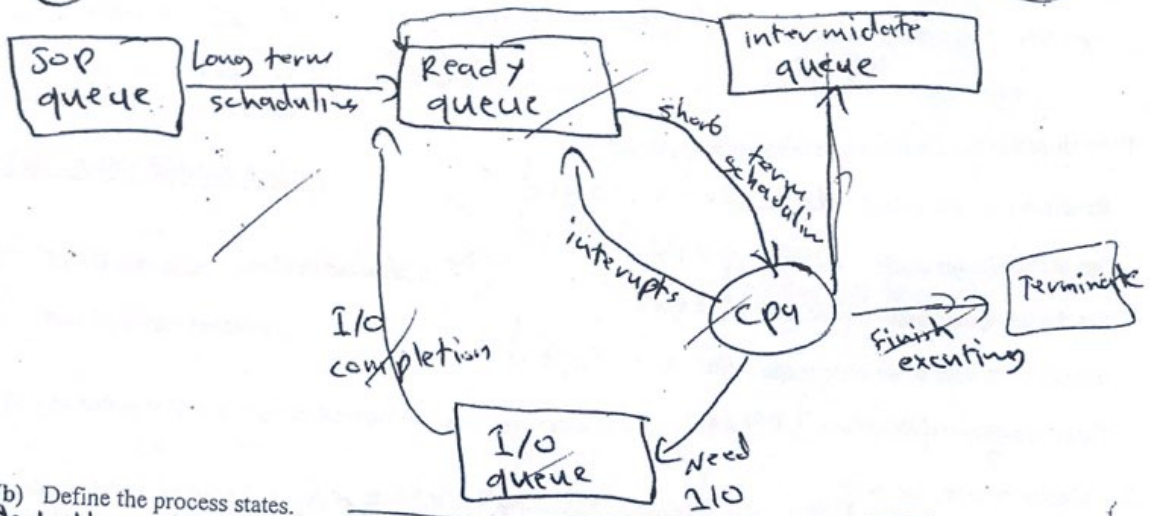
trap is asoftware there ~~generate~~ genarate Intrupter when an error occure such as division by zero.

ojeua

Question (3) : 25 points

(a) Draw **one** diagram that shows:
1- System queues. AND 2- System scheduling algorithms.



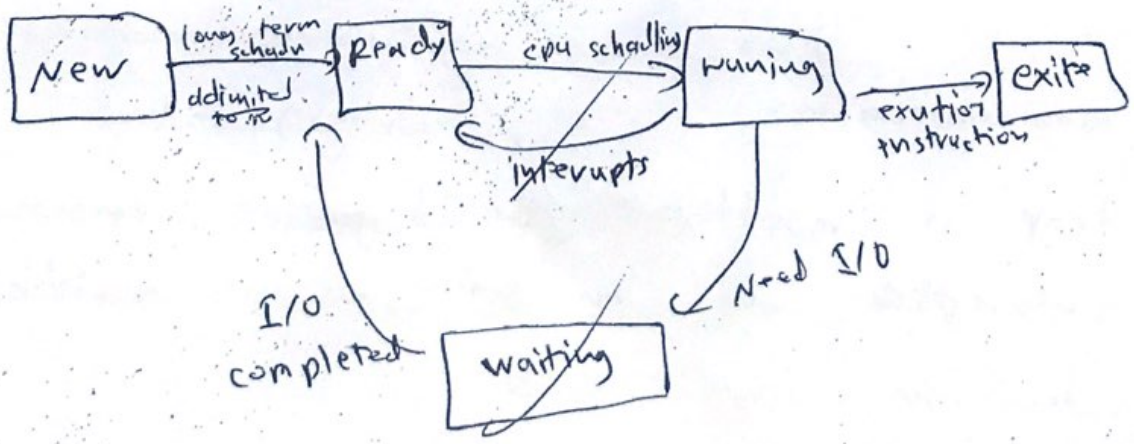


(b) Define the process states.

That the process consist of five main state that is
1. New ⇒ admited process to the memory by long term schadling
2. Ready ⇒ The process spend in queue and cpu selecte the next process to run
3. running :. The process have the time to run or executed.
4. waiting :. The process have the time and wait to take I/O needed or completion
5. exite :. the state when finished excuted instruction then terminated.

1,2,3 ⇒ major state ~~process contain data section, code section~~ the 1,2,3 state ~~are~~ a major state ~~structure~~

(c) Using diagrams only, show the Process States.



(9) The time required for the CPU to change (switch) from one process to another is called

swiching time

(10) A **thread** consists of:

Register set , stack

program counter

(11) The **thread** share with its peer threads:

data section , code section

OS resourse

(12) **Two** advantages of using **threads**.

parrallel system ; and Increas cpu utilizatior

Also **one** disadvantage you can think of each proses need Kernal for ib because it parrallel system

(13) The job with **many** short cpu bursts is called I/o Bound ,

but the job with **few** long cpu bursts is called cpu Bound

92

BIRZEIT UNIVERSITY

Computer Science Department

NAME : Raid qaleboh NUMBER : 1001212

Comp 431 Midterm # 2 11/01/2004

[1] (a) In a demand paging system, the system status is as shown below, briefly explain what is happening,

and what do you suggest :

(1) CPU utilization is 95% and paging disk utilization is 10%.

the CPU is hight utilization, and CPU does process for the process and lettel page fault and then little swap in and out and thus mostk the CPU at all time is process (doing job) and because the page fault and swap in and out is lettel then paging disk is low utilization.

① Increas degree of multiprogramming

② Decress size of page and thus mak number of page high and then we have swap in and out.

(2) CPU utilization is 10% and paging disk utilization is 95%.

CPU utilization is low Because the System is doing Swap in and out at most time and that happen Because the page fault is very hight. and to reduse page fault and then Improve CPU utilization we can use

① add new memory to the system and this decres number of page fa

② decres degree of Mutt programing

③ Increas size of page and thus reduse numbe of pages,

(b) If the logical address space contains 256 pages with 1024 words each. Given that physical memory is 4096 frames. Then,

(1) Compute how many bits the LA and PA contains,

logical address ⇒ d = 2^{10} ⇒ 10 Bit ⇒ Logical addre = 10+8 = 18 Bits

p = 2^8 8 Bit

physical addres ⇒ size of frame = 1024 = 2^{10} = 10 Bit

number of fram = 4096 = 2^{12} = 12 Bit ⇒ 10+12 = 22 Bits

(2) Compute the size of the page table.

~~Size of memory = $2^2 \times 2^{10} = 2^{12} \times 2^{20} = 4$ MB~~

num of frame = 2^{12} ⇒ 12 bit needed, number of entry = 2^8

Size of page table = $2^8 \times 12 = 256 \times 12$ words

[2] In a demand paging system with page table stored in memory and of two-levels, if memory access = 100 nanosecond

(a) how long does it take to access(reference) an instruction.

In demand page table system with two level we need 3 memory accesses ① access the outer page table ② access the inner page table ③ fetch the instruction.

it take = 3 X 100 nanoSecond = 300 nanoSecond

(b) If an associative registers table is added to the system with hit ratio 95%. Compute the EAT given that the lookup time in the associative registers is 10 nanasecond.

EAT = .95(100 + 10) + 0.05(3 * 100 + 10)

= 95 (110) + 0.05(310)

= 104.5 + 15.5

= 120 nanoSecond.

[3] (a) Given the LA = 020A9 in Hexadecimal , and page size = 2048 word, and given the page table below , compute:

- p and d .

$10 \rightarrow 2^{11}$

LA = 0000 0010 0000 1010 1001

page size = 2048 = 2^{11} ⇒ d = 11 Bit

p = 9 Bit

⇒ p = 0000 0001 00 = 4

⇒ d = 000 1010 1001 = 169

0	10
1	150
2	840
3	16
4	200
5	5000
6	66
.	.
.	.
.	100

- PA

200 | offset 169

PA = 200 × page size + offset

= 200 * 2048 + 169 = 409769

(b) In a demand paging system where page table is kept in registers. If servicing a page fault takes 10 millisecond if a free frame is available or the page is not modified and 20 millisecond if page is modified. If memory access is 100 nanosecond, and 60% of the time the page is modified. Compute the page fault rate p, assuming the EAT is 200 nanosecond.

2 mill secs

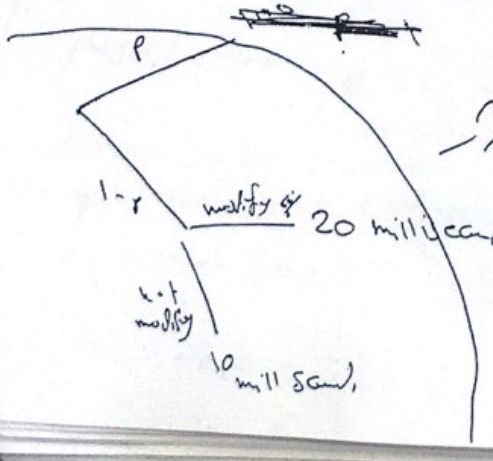
$2*10^9 = p(100) + (1-p)((.6*20*10^9)+(.4*10*10^9)+100)$

(1-p) + p

$2*10^9 = 100p + 1-p(12*10^9 + 4*10^9 + 100)$

$= 100p + 1-p(16*10^9 + 100)$

$= 100p + 16*10^9 + 100 - 16*10^9 p - 100p$



$\Rightarrow 2*10^9 = 16*10^9 + 100 - 16*10^9 p$

$16*10^9 p = 16*10^9 - 2*10^9 + 100$

$16*10^9 p = 14*10^9 + 100$

$p = \frac{14*10^9 + 100}{16*10^9} \approx 0.88$

∴ p ≈