Chapter 4: Memory management.

PAGE NO.:
DATE:

2075 Bhadra

Q. What are the differences between fixed partitioning and variable partitioning system of memory for multiprogramming?

Given reference to the following pages by a program
0, 9, 0, 1, 8, 1, 8, 7, 8, 7, 1, 2, 8, 2, 7, 8, 2, 2, 8, 3
How many page faults will occur if the program has 4 frames for Optimal Page Replacement algorithm?

Ans:

Fixed Partitioning	Variable Partitioning
1. In fixed partitioning, each partition is of fixed size and contains only one process.	In variable partitioning, the amount of memory allocated is exactly the amount of memory the process requires
2. No. of process depends upon no. of partitions.	No. of processes in memory doesn't depend upon no. of partitions.
3. Memory cannot be used efficiently for small size processes.	Memory can be used efficiently for small size processes.

Here,
Let the four frames be f1, f2, f3 and f4.

f1	0	0	-	0	0			0				2		-			-	-		2
f2		9		9	9			7		-		7			-					7
f3				1	1	-		1			-	1								3
f4					8		-	8	-			8	-			-			-	8

PAGE NO.:
DATE:

No. of page faults: (20-13) = 7 //

2074 Bhadra.

Q What is thrashing? Consider the following page-reference storing -
7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1. How many page faults would occur for the following page replacement algorithms, assuming 3 frames :-
a) FIFO b.) Optimal c.) LRU d.) LFU.

Ans: Thrashing is the condition when excessive paging operations are taking place. When multiple processes are competing for same resource, then excessive swapping takes place and page faults will occur frequently. It results in poor system performance.

a) FIFO.

f1	7	7	7	2		2	2	4	4	4	0			0	0	-		7	7	7
f2		0	0	0	-	3	3	3	2	2	2		-	1	1		-	1	0	0
f3			1	1		1	0	0	0	3	3	-		3	2			2	2	1

No. of page faults = 15

b.) Optimal.

f1	7	7	7	2		2		2	-		2		-	2	-			7		
f2		0	0	0	-	0	-	4			0			0		-		0	-	
f3			1	1		3		3		-	3	-		1			-	1		-

No. of page faults: 9

c) LRU

f1	7	7	7	2		2		4	4	4	0			1		1	–	1		–
f2		0	0	0	–	0	–	0	0	3	3	–		3		0		0	–	
f3			1	1		3		3	2	2	2		–	2	–	2		7		

No. of page fault : 12

d) LFU : 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

Frequency : ~~1~~ 1 ~~1~~ ~~1~~ 2 ~~1~~ 3 ~~1~~ 1 1 4 2 ~~2~~ ~~1~~ ~~1~~ 5 ~~1~~ ~~1~~ 5

f1	7	7	7	2		2		4	4	3		–		3	3		3	3		3
f2		0	0	0	–	0	–	0	0	0	–			0	0	–	0	0	–	0
f3			1	1		3		3	2	2			–	1	2		1	7		1

No. of page fault : 13.

2073 Bhadra.

Q. Differentiate compaction and coalescing technique. How logical address is mapped to physical address by Paging Technique? Explain with suitable example.

Ans.

Compaction	Coalescing
1. The process of combining all the holes into a big one by moving all the processes downward as far as possible is compaction.	The process of merging two adjacent holes to form a single larger hole is called coalescing.
2. It requires more CPU time.	It requires lesser CPU time.

PAGE NO.:
DATE:

Physical memory is divided into fixed size blocks known as page frames.
Logical address is divided into fixed size blocks known as pages. Size of page & page frames are same. During execution pages are loaded to frames.
For mapping of logical address to physical address.
Let, page number : p
frame number : f
page offset : d
Logical address (L) : p+d

- Page number is used to index into page number and fetch corresponding frame number (f).
- The physical address is obtained by combining f and d.
- For m bit processor, logical address will be m bits long

Page size : 2^n bytes
Then, n bits will specify offset and m-n bits specify page number.

Example: For logical address of 16 bits & page size 4KB.
Size of logical address space : 2^{16}
Page size : 4KB = 2^{12}
i.e. 12 bit offset
4 bit page number.

Page 3	
Page 2	
Page 1	
Page 0	

(Logical memory)

Page no.	Frame no.
0	3
1	5
2	1
3	6

Page table.

7	
6	Page 3
5	Page 1
4	.
3	Page 0
2	.
1	Page 2
0	.

(Physical memory)

Fig: Page Table.

PAGE NO.:
DATE:

Q. With an example, show that FIFO page replacement algorithm suffers from Belady's anomaly

Ans. Belady's anomaly is the phenomenon in which increasing the number of page frame results in an increase in the number of page faults.

Let us consider a page sequence:
3, 2, 1, 0, 3, 2, 4, 3, 2, 1, 0, 4, 2, 3, 2, 1, 0, 4

When no. of frames = 3

f1	3	3	3	0	0	0	4			4	4	4	2	2	–	2	0	0
f2		2	2	2	3	3	3	–		1	1	1	1	3		3	3	4
f3			1	1	1	2	2		–	2	0	0	0	0		1	1	1

No. of page faults = 14

When no. of frames = 4

f1	3	3	3	3	–		4	4	4	4	0	0		0	0	1	1	1
f2		2	2	2		–	2	3	3	3	3	4		4	4	4	0	0
f3			1	1			1	1	2	2	2	2	–	3	3	3	3	4
f4				0			0	0	0	1	1	1		1	2	2	2	2

No. of page faults = 15

So, as no. of frame was increased, faults were also increased.

PAGE NO.:
DATE:

2073 Magh

Q. Under what circumnstances do page fault occur?
Page reference string:
1, 2, 3, 4, 2, 1, 5, 6, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6
i) LRU
ii) FIFO
iii) Optimal replacement. , No. of frames = 3

i) LRU:

f1	1	1	1	4		4	5	5	5	3	3	3	–	3	3		–	3
f2		2	2	2	–	2	2	6	6	6	7	7		2	2	–		2
f3			3	3		1	1	1	2	2	2	6		6	1			6

No. of page fault = 14

ii) FIFO:

f1	1	1	1	4		4	4	6	6	6	7	7		7	1		1	1
f2		2	2	2	–	1	1	1	2	2	2	6		6	6		3	3
f3			3	3		3	5	5	5	3	3	3	–	2	2	–	2	6

No. of page fault = 15

iii) Optimal Replacement

f1	1	1	1	1		–	1	1		3	3		–	3	3		–	6
f2		2	2	2	–		2	2	–	2	7			2	2	–		2
f3			3	4			5	6		6	6	–		6	1			3

No. of Page fault = 11.

A page fault occurs when an access to

PAGE NO.:
DATE:

a page that has not been brought into main memory takes place. The OS verifies the memory access, aborting the program if it is invalid. If it is valid, a free frame is located and I/O is requested to read the needed page into the free frame. Upon completion of I/O, the process table and page table are updated and instruction is restarted.

2072 Ashwin.

Q. Define page fault and demand paging. Consider a paged memory system with eight pages of 8kB page size each and 16 pages frames in memory. Using the given page table, compute the physical address for logical address 18325.

7	10
6	4
5	0
4	7
3	13
2	11
1	14
0	5

Ans: A page fault is a type of exception raised by computer hardware when a running program accesses a memory page that is not currently mapped by the memory management unit (MMU) into the virtual address space of a process.

PAGE NO.:
DATE:

Demand paging is a method of virtual ~~man~~ memory management. In a system that uses demand paging, the OS copies a disk page into physical memory only if an attempt is made to access it and that page is not already in memory.

Here,

Logical address (m) = 16

Total pages each = 8 = 2^3

Page number (p) = 3

Size of page = 8kB = 2^{13}

$\therefore n = 13$

Offset bit = 13 bits

Page bit = 3 bit

No. of bits per frame = 4 bits

Logical address = 18325

$= \underbrace{0100}_{p}\ \underbrace{0111\ 1001\ 0101}_{d}$ [in Binary]

p = 010 = 2

From page table, the equivalent frame number for the given page is f = 11 = 1011

$\therefore$ Physical address (PA) = $\underbrace{1011}_{f}\ \underbrace{00111\ 1001\ 0101}_{d}$

= 92053

2072 Magh

Q. Consider the following page reference string
1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6.

⇒ 2073 Magh.

2071 Bhadra

Q. What is the role of TLB?

Ans: A translation lookaside buffer (TLB) is a memory cache that is used to reduce the time taken to access a user memory location. The TLB stores the recent translations of virtual memory to physical memory.

2071 Magh.

Q. What is residence monitor? Consider logical address spaces of eight pages of 1024 words, each mapped onto a physical memory of 32 frames then.

a.) How many bits are in logical address?

b.) How many bits are in physical address?

Ans: Residence monitor is a type of system software program that was used in many early computers from 1950s to 1970s. It can be considered a percursor to operating system. The name is derived from a program which is always present in the computer's memory thus being "resident". Because memory was very limited on these systems the resident monitor was often little more than a stub which would gain control at the end of a job and load a non-resident portion to perform job cleanup and setup task.

Here,

No. of page bits : 3 bits

No. of offset bits : 10 bits

No. of bits for frame : 5 bits

PAGE NO.:
DATE:

∴ Bits in physical address = (10+5) bits = 15 bits
Bits in logical address = (10+3) bits = 13 bits

2070 Bhadra

Q. Sequence.

2 3 5 4 2 5 7 3 8 7

i) FIFO
ii) LRU
iii) Optimal, No of frames = 3

i) FIFO:

f1	2	2	2	4	4		4	3	3	
f2		3	3	3	2		2	2	8	
f3			5	5	5	-	7	7	7	-

No. of page hit = 2
No. of page fault = 8.

ii) LRU.

f1	2	2	2	4	4		7	7	7	-
f2		3	3	3	2		2	3	3	
f3			5	5	5	-	5	5	8	

No. of page hit = 2
No of page fault = 8

PAGE NO.:
DATE:

iii) Optimal

f1	2	2	2	2	–		2	3	3		
f2		3	3	4			7	7	7	–	
f3			5	5		–	5	5	8		

No. of page hit = 8
No. of page fault = 7.