Tutorial 4

Pre-Requisite:

https://www.geeksforgeeks.org/partition-allocation-methods-in-memory-management/

Pg 407 -> Operating System Dhananjay Book

1. Select the correct alternative in each of the following questions:

a. A worst-fit allocator always splits the largest free memory area while making an allocation. A free list contains three memory areas of sizes 6 KB, 15 KB and 12 KB. The next four memory requests are for 10 KB, 2 KB, 5 KB, and 14 KB of memory. The only placement strategy that would be able to accommodate all four processes is
i. First-fit,
ii. best-fit,
iii. worst-fit,
iv. next-fit.

b. Three processes requiring 150 KB, 100 KB, and 300 KB of memory are in operation in an OS employing a paging system with a page size of 2 KB. The maximum internal memory fragmentation due to memory allocation to the three processes is
i. Approximately 2 KB
ii. Approximately 6 KB
iii. 275 KB
iv. None of (i)–(iii)

c. A reentrant program is one that
i. Calls itself recursively
ii. Can have several copies in memory that can be used by different users
iii. Can have a single copy in memory that is executed by many users Concurrently

2. Consider a logical address space of 64 pages of 1,024 words each, mapped onto a physical memory of 32 frames.
a. How many bits are there in the logical address?
b. How many bits are there in the physical address?

NOTE : PAGES ARE CHANGED TO **64**

ANS 1

Explanation: Logical address space has 8 pages, size of each page, offset = 1024 words

Number if bits in page# field of the logical address = $\log_2 8$ bits = 3 bits

Offset bits = $\log_2 1024$ = 10 bits

So, **Logical address = 3 + 10 = 13 bits**

Physical memory has 32 frames, offset = 1024 words

Number of bits in frame# field of the Physical address = $\log_2 32$ bits = 5 bits

Physical address = 5 + 10 = 15 bits

Option (C) is correct.

ANS 2

Problem 9.8

Consider a logical-address space of eight pages of 1,024 words each, mapped onto a physical memory of 32 frames.

a) How many bits are in the logical address?

The logical address is split into two parts: the page address and then the offset.

The page address must be able to reference 8 (or 2^3) distinct pages. This requires 3 address bits.

The offset must be able to reference 1,024 (or 2^{10}) distinct words on each page. This requires 10 address bits.

Therefore, **13 bits** are required for the logical address.

b) How many bits are in the physical address?

The physical address is also split into two parts: the frame address and then the offset.

The number of frames is 32 (or 2^5), so that will require 5 bits.

The size of the frame is the same as the size of the page, therefore this will require 10 bits, as described above.

Therefore, **15 bits** are required for the physical address.

3. Given five memory partitions of 100 KB, 500 KB, 200 KB, 300 KB, and 600 KB (ill order), how would the first-fit, best-fit, and worst-fit algorithms place processes of 212 KB, 417 KB, 112 KB, and 426 KB (in order)? Which algorithm makes the most efficient use of memory?
https://www.williamt.com/noctrl/archive/c2002_03_csc420/homework/hw05.htm

Problem 9.5

Given memory partitions of 100 KB, 500 KB, 200 KB, 300 KB, and 600 KB (in order), how would each algorithm place processes

a) First-fit

The first-fit algorithm selects the first free partition that is large enough to accommodate the request.

First-fit would allocate in the following manner:

- 212 KB => 500 KB partition, leaves a 288 KB partition
- 417 KB => KB partition, leaves a 183 KB partition
- 112 KB => 288 KB partition, leaves a 176 KB partition
- 426 KB would not be able to allocate, no partition large enough!

b) Best-fit

The best-fit algorithm selects the partition whose size is closest in size (and large enough) to the requested size.

Best-fit would allocate in the following manner:

- 212 KB => 300 KB, leaving a 88 KB partition
- 417 KB => 500 KB, leaving a 83 KB partition
- 112 KB => 200 KB, leaving a 88 KB partition
- 426 KB => 600 KB, leaving a 174 KB partition

c) Worst-fit

The worst-fit algorithm effectively selects the largest partition for each request.

Worst-fit would allocate in the following manner:

- 212 KB => 600 KB, leaving a 388 KB partition
- 417 KB => 500 KB, leaving a 83 KB partition
- 112 KB => 388 KB, leaving a 276 KB partition
- 426 KB would not be allowed to allocate as no partition is large enough!

d) Which algorithm makes the most efficient use of memory?

The **best-fit** algorithm performed the best of the three algorithms, as it was the only algorithm to meet all the memory requests.

First fit

212 K is put in 500 K partition.

417 K is put in 600 K partition.

112 K is put in 288 K partition. (New partition 288 K = 500 K - 212 K)

426 K must wait.

Best-fit

212 K is put in 300 K partition.

417 K is put in 500 K partition.

112 K is put in 200 K partition.

426 K is put in 600 K partition.

Worst-fit

212 K is put in 600 K partition.

417 K is put in 500 K partition.

112 K is put in 388 K partition. (600 K - 212 K)

426 K must wait.

In this example Best-fit is the best solution.

4. An OS has 110 MB available for user processes. The maximum memory requirement of a process for its own code and data is 20 MB, while the average memory requirement of a process is 10 MB. If the OS uses contiguous memory allocation and does not know sizes of individual processes, what is the average internal and external fragmentation?

https://www.chegg.com/homework-help/os-110-mb-available-user-processes-maximum-memory-requiremen-chapter-11-problem-12-solution-9780077460228-exc

5. A page-replacement algorithm should minimize the number of page faults. We can achieve this minimization by distributing heavily used pages evenly over all of memory, rather than having them compete for a small number of page frames. We can associate with each page frame a counter of the number of pages associated with that frame.
Then, to replace a page, we can search for the page frame with the smallest counter.

Step 1 1 of 6

The Algorithm

The algorithm minimizes number of faults by distributing heavily used pages evenly over all of memory, rather than having them compete for a small number of page frames

a. Define a page-replacement algorithm using this basic idea. Specifically address these problems:
i. What is the initial value of the counters?
ii. When are counters increased?
iii. When are counters decreased?
iv. How is the page to be replaced selected?

Step 2 2 of 6

part (a)

(i) The initial values of the counters is 0

(ii) When a new page is associated with the the frame, its counter is increased by one

(ii) When a page associated with the the frame is no longer required, its counter is decreased by one

(iv) The page frame with the smallest counter is replaced. However, when tie occurs, the FIFO algorithm will be used

b. How many page faults occur for your algorithm for the following reference string with four page frames?
1, 2, 3, 4, 5, 3, 4, 1, 6, 7, 8, 7, 8, 9, 7, 8, 9, 5, 4, 5, 4, 2.

Step 3

Part (b)

Consider the figure shown below that gives the Tabulation

(1) Page, 1,2,3 and 4 are filled to all 4 frames counters are 1

(2) Page 5 moves to Frame-1, based on FIFO algorithm, its counter is 2 now

(3) Page 3 and 4 are available, so no fault occurs and counter remains same

(4) Page 1 is moved to Frame-2 based on FIFO algorithm and its counter is changed to 2

(5) Page 6 is moved to Frame-3 based on FIFO algorithm and its counter is changed to 2. But since 3 is not required later on, its counter is again decreased by 1. So its counter is now 1 again

(6) Page 7 is moved to Frame-4 based on FIFO algorithm and its counter is changed to 2

ELEMENT	1	2	3	4	5	3	4	1	6	7	8	7	8	9	7	8	9	5	4	5	4	2
FRAME-1	1 (1)				5 (2)											8 (3)						
FRAME-2		2 (1)						1 (2)									5 (3-1=2)			NF		
FRAME-3			3 (1)			NF			6 (2-1 =1)		8 (2-1=1)		NF	9 (2)			NF					
FRAME-4				4 (1)			NF			7 (2)		NF			NF				4 (2-1=1)		NF	2 (1)
FAULTS=	22-8	14																				

Step 4

(7) Page 8 is moved to Frame-3 based on new algorithm (its counter is lowest) and its counter is changed to 2. But since 6 is not required later on, its counter is again decreased by 1. So its counter is now 1 again

(8) Page 7 and 8 are available in frames 3 and 4 so no fault occurs and counter remains same

(9) Page 9 is moved to Frame-3 based on new algorithm (its counter is lowest) and its counter is changed to 2

(10) All the counters are 2 now, so Page 8 is moved to Frame-1 with FIFO algorithm and its counter is 3 now

(11) Page 9 is available in frames 3 so no fault occurs and counter remains same

(12) Page 5 is moved to frame-2 based on FIFO algorithm and counter is changed to 2. However, 1 will not be required again, so it is changed to 1 back

(13) Page 4 is moved to frame-4 based on FIFO algorithm and counter is changed to 2. However, 7 will not be required again, so it is changed to 1 back

(14) Page 5 and 4 are available in frames 2 and 3 so no fault occurs and counter remains same

(15) Page 2 is moved to frame-4 based on new algorithm

c. What is the minimum number of page faults for an optimal page replacement strategy for the reference string in part b with four page frames?

Step 5 5 of 6

Part (c)

The optimal algorithm is shown below

ELEMENT	1	2	3	4	5	3	4	1	6	7	8	7	8	9	7	8	9	5	4	5	4	2
FRAME-1	1							NF	6		8		NF			NF						
FRAME-2		2			5													NF		NF		2
FRAME-3			3			NF				7		NF			NF				4		NF	
FRAME-4				4			NF							9			NF					
FAULTS=	22-11	11																				

Result 6 of 6

(a) Works based on Counter

(b) 14 Page faults

(c) 11 Page faults

6. 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.
How many page faults would occur for the following replacement algorithms, assuming one, two, three, four, five, six, and seven frames? Remember that all frames are initially empty, so your first unique pages will cost one fault each.

- LRU replacement
- FIFO replacement
- Optimal replacement

Step 1

Given Data

We are given with following page reference string

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

Step 2

LRU: Frames = 1

Consider the Least recently used method with one frame:

As shown in the figure below, there will be 20 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6

FAULTS=	20

Step 3

LRU: Frames = 2

Consider the Least recently used method with two frames:

As shown in the figure below, there will be 18 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1		3		2		5		2		NF		7		3		1		3	
FRAME-2		2		4		1		6		1		3		6		2		NF		6

FAULTS=	20-2	18

Step 4

LRU: Frames = 3

Consider the Least recently used method with 3 frames:

As shown in the figure below, there will be 15 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1			4			5			1			7			2		NF		
FRAME-2		2			NF			6				3			NF				NF	
FRAME-3			3			1			2		NF			6			1			6

FAULTS=	20-5	15

Step 5

LRU: Frames = 4

Consider the Least recently used method with 4 frames:

As shown in the figure below, there will be 10 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1					NF				NF				6						NF
FRAME-2		2			NF				NF		NF					NF		NF		
FRAME-3			3				5					3			NF				NF	
FRAME-4				4				6					7				1			

FAULTS=	20-10	10

Step 6

LRU: Frames = 5

Consider the Least recently used method with 5 frames:

As shown in the figure below, there will be 8 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1					NF				NF							NF			
FRAME-2		2			NF				NF		NF					NF		NF		
FRAME-3			3					6						NF						NF
FRAME-4				4								3			NF				NF	
FRAME-5							5						7							
FAULTS=	20-12	8																		

Step 7

LRU: Frames = 6

Consider the Least recently used method with 6 frames:

As shown in the figure below, there will be 7 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1					NF				NF							NF			
FRAME-2		2			NF				NF		NF					NF		NF		
FRAME-3			3									NF			NF				NF	
FRAME-4				4																
FRAME-5							5						7							
FRAME-6								6						NF						NF
FAULTS=	20-13	7																		

Step 8

LRU: Frames = 7

Consider the Least recently used method with 7 frames:

As shown in the figure below, there will be 7 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1					NF				NF							NF			
FRAME-2		2			NF				NF		NF					NF		NF		
FRAME-3			3									NF			NF				NF	
FRAME-4				4																
FRAME-5							5													
FRAME-6								6						NF						NF
FRAME-7													7							
FAULTS=	20-13	7																		

Step 9

FIFO: Frames = 1

Consider the First in First Out method with 1 frames:

As shown in the figure below, there will be 20 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6

FAULTS=	20

Step 10

FIFO: Frames = 2

Consider the First in First Out method with 2 frames:

As shown in the figure below, there will be 18 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1		3		2		5		2		NF		7		3		1		3	
FRAME-2		2		4		1		6		1		3		6		2		NF		6

FAULTS=	20-2	18

Step 11

FIFO: Frames = 3

Consider the First in First Out method with 3 frames:

As shown in the figure below, there will be 16 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1			4				6				3			NF	2		NF		6
FRAME-2		2			NF	1			2		NF		7				1			
FRAME-3			3				5			1				6					3	

FAULTS=	20-4	16

Step 12

FIFO: Frames = 4

Consider the First in First Out method with 4 frames:

As shown in the figure below, there will be 14 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1					NF	5					3			NF		1			
FRAME-2		2			NF			6					7						3	
FRAME-3			3						2		NF			6						NF
FRAME-4				4						1						2		NF		
FAULTS=	20-6	14																		

Step 13

FIFO: Frames = 5

Consider the First in First Out method with 5 frames:

As shown in the figure below, there will be 10 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1					NF		6						NF						NF
FRAME-2		2			NF				NF	1							NF			
FRAME-3			3								2					NF		NF		
FRAME-4				4								3			NF				NF	
FRAME-5							5						7							
FAULTS=	20-10	10																		

Step 14

FIFO: Frames = 6

Consider the First in First Out method with 6 frames:

As shown in the figure below, there will be 10 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1					NF				NF			7							
FRAME-2		2			NF				NF		NF					NF	1			
FRAME-3			3									NF			NF			2		
FRAME-4				4															3	
FRAME-5							5													
FRAME-6								6						NF						NF
FAULTS=	20-10	10																		

Step 15

FIFO: Frames = 7

Consider the First in First Out method with 7 frames:

As shown in the figure below, there will be 7 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1					NF				NF							NF			
FRAME-2		2			NF				NF		NF					NF		NF		
FRAME-3			3									NF			NF				NF	
FRAME-4				4																
FRAME-5							5													
FRAME-6								6						NF						NF
FRAME-7													7							
FAULTS=	20-13	7																		

Step 16

Optimal: Frames = 1

Consider the Optimal method with one frame:

As shown in the figure below, there will be 20 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6

FAULTS=	20

Step 17

Optimal: Frames = 2

Consider the Least recently used method with 2 frames:

As shown in the figure below, there will be 15 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1		3	4		1	5	6		1		3	7	6	3		1		3	6
FRAME-2		2			NF				NF		NF					NF		NF		

FAULTS=	20-5	15

Step 18

Optimal: Frames = 3

Consider the Least recently used method with 3 frames:

As shown in the figure below, there will be 11 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1					NF				NF		3			NF				NF	
FRAME-2		2			NF				NF		NF		7			2		NF		
FRAME-3			3	4			5	6		1				NF			1		3	6

FAULTS=	20-9	11

Step 19

Optimal: Frames = 4

Consider the Least recently used method with 4 frames:

As shown in the figure below, there will be 8 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1					NF				NF			7				1			
FRAME-2		2			NF				NF		NF					NF		NF	3	
FRAME-3			3									NF			NF				NF	
FRAME-4				4			5	6						NF		2				NF
FAULTS=	20-12	8																		

Step 20

Optimal: Frames = 5

Consider the Least recently used method with 5 frames:

As shown in the figure below, there will be 7 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1					NF				NF							NF			
FRAME-2		2			NF				NF		NF					NF		NF		
FRAME-3			3									NF			NF				NF	
FRAME-4				4									7							
FRAME-5							5	6						NF						NF
FAULTS=	20-13	7																		

Step 21

Optimal: Frames = 6

Consider the Least recently used method with 6 frames:

As shown in the figure below, there will be 7 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1					NF				NF							NF			
FRAME-2		2			NF				NF		NF					NF		NF		
FRAME-3			3									NF			NF				NF	
FRAME-4				4																
FRAME-5							5						7							
FRAME-6								6						NF						NF
FAULTS=	20-13	7																		

Step 22

Optimal: Frames = 7

Consider the Least recently used method with 7 frames:

As shown in the figure below, there will be 7 page faults

ELEMENT	1	2	3	4	2	1	5	6	2	1	2	3	7	6	3	2	1	2	3	6
FRAME-1	1					NF				NF							NF			
FRAME-2		2			NF				NF		NF					NF		NF		
FRAME-3			3									NF			NF				NF	
FRAME-4				4																
FRAME-5							5													
FRAME-6								6						NF						NF
FRAME-7													7							
FAULTS=	20-13	7																		

Step 23

	No. of Faults		
NO OF FAULTS	LRU	FIFO	Optimal
FRAME-1	20	20	20
FRAME-2	18	18	15
FRAME-3	15	16	11
FRAME-4	10	14	8
FRAME-5	8	10	7
FRAME-6	7	10	7
FRAME-7	7	7	7

Result

	No. of Faults		
NO OF FAULTS	LRU	FIFO	Optimal
FRAME-1	20	20	20
FRAME-2	18	18	15
FRAME-3	15	16	11
FRAME-4	10	14	8
FRAME-5	8	10	7
FRAME-6	7	10	7
FRAME-7	7	7	7

7. Consider a demand-paged computer system where the degree of multiprogramming is currently fixed at four. The system was recently measured to determine utilization of the CPU and the paging disk. The results are one of the following alternatives. For each case, what is happening? Can the degree of multiprogramming be increased to increase the CPU utilization? Is the paging helping?

a. CPU utilization 13 percent; disk utilization 97 percent
b. CPU utilization 87 percent; disk utilization 3 percent
c. CPU utilization 13 percent; disk utilization 3 percent

Question 2 [Points 5] Consider a demand-paged computer system where the degree of multiprogramming is currently fixed at four. The system was recently measured to determine utilization of the CPU and the paging disk. Three alternative results are shown below. For each case, what is happening? Can the degree of multiprogramming be increased to increase the CPU utilization? Is the paging helping?

A. CPU utilization 13 percent; disk utilization 97 percent
Disk utilization at 97% is too high while the CPU at 13% is too low this causes thrashing. Processes that are being executed spend much time on the disk in attempts to be paged. The degree of multiprogramming cannot be increased because processes need to be suspended to increase CPU utilization.

B. CPU utilization 87 percent; disk utilization 3 percent
CPU at 87% and disk at 3% means that the CPU is going to cause the bottleneck eventually. Disk is being underused and increasing the degree of multiprogramming would inevitably cause thrashing.

C. CPU utilization 13 percent; disk utilization 3 percent
CPU and disk space are both very low. The obvious answer would be an increase the degree of multiprogramming would increase CPU utilization.