Operating Systems CO-562

Abumansur Sabyrrakhim
Assignment 6

Problem 6.1

(a)

Best Fit: we are allocating the smallest free partition that meets all the requirements.

	12KiB	15KiB	19KiB	13KiB	7KiB	8KiB	16KiB
14KiB	12KiB	1KiB	19KiB	13KiB	7KiB	8KiB	16KiB
9KiB	3KiB	1KiB	19KiB	13KiB	7KiB	8KiB	16KiB
7KiB	3KiB	1KiB	19KiB	13KiB	0KiB	8KiB	16KiB
10KiB	3KiB	1KiB	19KiB	3KiB	9KiB	8KiB	16KiB

(b)

Worst Fit: we are allocating the largest free partition that meets all the requirements.

	12KiB	15KiB	19KiB	13KiB	7KiB	8KiB	16KiB
14KiB	12KiB	15KiB	5KiB	13KiB	7KiB	8KiB	16KiB
9KiB	12KiB	15KiB	5KiB	13KiB	7KiB	8KiB	7KiB
7KiB	12KiB	8KiB	5KiB	13KiB	7KiB	8KiB	7KiB
10KiB	12KiB	8KiB	5KiB	3KiB	7KiB	8KiB	7KiB

(c)

First Fit: we are allocating the first free partition that meets all the requirements.

	12KiB	15KiB	19KiB	13KiB	7KiB	8KiB	16KiB
14KiB	12KiB	1KiB	19KiB	13KiB	7KiB	8KiB	16KiB
9KiB	3KiB	1KiB	19KiB	13KiB	7KiB	8KiB	16KiB
7KiB	3KiB	1KiB	12KiB	13KiB	7KiB	8KiB	16KiB
10KiB	3KiB	1KiB	2KiB	13KiB	7KiB	8KiB	16KiB

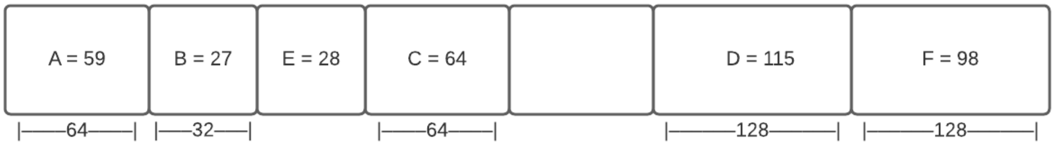
(d)

Next Fit: we are allocating the next free partition that meets all the requirements.

	12KiB	15KiB	19KiB	13KiB	7KiB	8KiB	16KiB
14KiB	12KiB	1KiB	19KiB	13KiB	7KiB	8KiB	16KiB
9KiB	12KiB	1KiB	10KiB	13KiB	7KiB	8KiB	16KiB
7KiB	12KiB	1KiB	10KiB	6KiB	7KiB	8KiB	16KiB
10KiB	12KiB	1KiB	10KiB	6KiB	7KiB	8KiB	6KiB

Problem 6.2

(a)



(b)

Internal Fragmentation:

A = 64 - 59 = 5
B = 32 - 27 = 5
C = 64 - 44 = 20
D = 128 - 115 = 13
E = 32 - 28 = 4
F = 128 - 98 = 30

Overall Internal Fragmentation:

A + B + C + D + E + F = 5 + 5 + 20 + 13 + 4 + 30 = 77 KiB

(c)

132 KiB cannot be accommodated, since partition can be in form 2^k, so here D is 128 KiB which is not able to accommodate 132 KiB. Hence, the answer is NO.

Problem 6.3

(a)

First-In-First-Out (FIFO) page replacement algorithm:

(i)

2 Frames:

reference string	1	2	3	4	1	1	4	2	1	2
Frame 0	1	1	3	3	1	1	1	1	1	1
Frame 1	-	2	2	4	4	4	4	2	2	2

First two references: frames are empty, so we insert them directly.

We swap out 1, 2 and 3, since they were first in the queue when 3, 4 and 1 arrived respectively.

Next 1 and 4 were already there, so we don't swap them.

4 is swapped out, since it was first in queue when 2 arrived.

Next 1 and 2 were already there, so we don't swap them.

NUMBER OF PAGE FAULTS: 6.

(ii)

3 Frames:

reference string	1	2	3	4	1	1	4	2	1	2
Frame 0	1	1	1	4	4	4	4	4	4	4
Frame 1	-	2	2	2	1	1	1	1	1	1
Frame 2	-	-	3	3	3	3	3	2	2	2

First three references: frames are empty, so we insert them directly.

We swap out 1 and 2, since they were first in the queue when 4 and 1 arrived respectively.

Next 1 and 4 were already there, so we don't swap them.

3 is swapped out, since it was first in queue when 2 arrived.

Next 1 and 2 were already there, so we don't swap them.

NUMBER OF PAGE FAULTS: 6.

(b)

Belady's Optimal(BO) page replacement algorithm:

(i)

2 Frames:

reference string	1	2	3	4	1	1	4	2	1	2
Frame 0	1	1	1	1	1	1	1	1	1	1
Frame 1	-	2	3	4	4	4	4	2	2	2

First two references: frames are empty, so we insert them directly.

We swap out 2 and 3, since their reference are the farthest when 3 and 4 arrived respectively.

Next 1, 1 and 4 were already there, so we don't swap them.

4 is swapped out, since it had no further reference when 2 arrived.

NUMBER OF PAGE FAULTS: 5.

(ii)

3 Frames:

reference string	1	2	3	4	1	1	4	2	1	2
Frame 0	1	1	1	1	1	1	1	1	1	1
Frame 1	-	2	2	2	2	2	2	2	2	2
Frame 2	-	-	3	4	4	4	4	4	4	4

First three references: frames are empty, so we insert them directly.

We swap out 3, since it had no further references when 4 arrived.

Next 1, 1, 4, 2, 1 and 2 were already there, so we don't change anything.

NUMBER OF PAGE FAULTS: 4.

(c)

Least Recently Used (LRU) page replacement algorithm.

(i)

2 Frames:

reference string	1	2	3	4	1	1	4	2	1	2
Frame 0	1	1	3	3	1	1	1	2	2	2
Frame 1	-	2	2	4	4	4	4	4	1	1

First two references: frames are empty, so we insert them directly.

We swap out 3, 4 and 1, since 2, 3 and 4 were used recently when 3, 4 and 1 arrived respectively.

Next 1, 1 and 4 were already there, so we don't swap them.

1 and 4 swapped out, since 4 and 2 were last referenced when 2 and 1 arrived respectively.

Next 2 was already there, so we don't swap them.

NUMBER OF PAGE FAULTS: 7.

(ii)

3 Frames:

reference string	1	2	3	4	1	1	4	2	1	2
Frame 0	1	1	1	4	4	4	4	4	4	4
Frame 1	-	2	2	2	1	1	1	1	1	1
Frame 2	-	-	3	3	3	3	3	2	2	2

First three references: at least one frame is empty, so we insert it directly.

We swap out 1, since 3 and 2 were recently used when 4 arrived.

We swap out 2, since 4 and 3 were recently used when 1 arrived.

Next 1, 1 and 4 were already there, so we don't swap them.

3 is swapped out, since 4 nd 1 were last referenced when 2 arrived.

Next 1 and 2 were already there, so we don't swap them.

NUMBER OF PAGE FAULTS: 6.