

## Operating Systems Assignment #7

Bilal Waraich

Problem 7.1:

**Free Blocks:**

17 KiB, 8 KiB, 10 KiB, 21 KiB, 12 KiB, 13 KiB

- a) **11KiB:** 17 KiB, 8 KiB, 10 KiB, 21 KiB, 1 KiB, 13 KiB  
**9 KiB:** 17 KiB, 8 KiB, 1 KiB, 21 KiB, 1 KiB, 13 KiB  
**7 KiB:** 17 KiB, 1 KiB, 1 KiB, 21 KiB, 1 KiB, 13 KiB  
**16 KiB:** 1 KiB, 1 KiB, 1 KiB, 21 KiB, 1 KiB, 13 KiB
- b) 17 KiB, 8 KiB, 10 KiB, 21 KiB, 12 KiB, 13 KiB  
**11 KiB:** 17 KiB, 8 KiB, 10 KiB, 10 KiB, 12 KiB, 13 KiB  
**9 KiB:** 8 KiB, 8 KiB, 10 KiB, 10 KiB, 12 KiB, 13 KiB  
**7 KiB:** 8 KiB, 8 KiB, 10 KiB, 10 KiB, 12 KiB, 6 KiB  
**16 KiB:** Cannot Allocate – Memory Allocation Failure
- c) 17 KiB, 8 KiB, 10 KiB, 21 KiB, 12 KiB, 13 KiB  
**11 KiB:** 6 KiB, 8 KiB, 10 KiB, 21 KiB, 12 KiB, 13 KiB  
**9 KiB:** 6 KiB, 8 KiB, 1 KiB, 21 KiB, 12 KiB, 13 KiB  
**7 KiB:** 6 KiB, 1 KiB, 1 KiB, 21 KiB, 12 KiB, 13 KiB  
**16 KiB:** 6 KiB, 1 KiB, 1 KiB, 5 KiB, 12 KiB, 13 KiB
- d) 17 KiB, 8 KiB, 10 KiB, 21 KiB, 12 KiB, 13 KiB  
**11 KiB:** 6 KiB, 8 KiB, 10 KiB, 21 KiB, 12 KiB, 13 KiB  
**9 KiB:** 6 KiB, 8 KiB, 1 KiB, 21 KiB, 12 KiB, 13 KiB  
**7 KiB:** 6 KiB, 8 KiB, 1 KiB, 14 KiB, 12 KiB, 13 KiB  
**16 KiB:** Cannot Allocate – Memory Allocation Failure

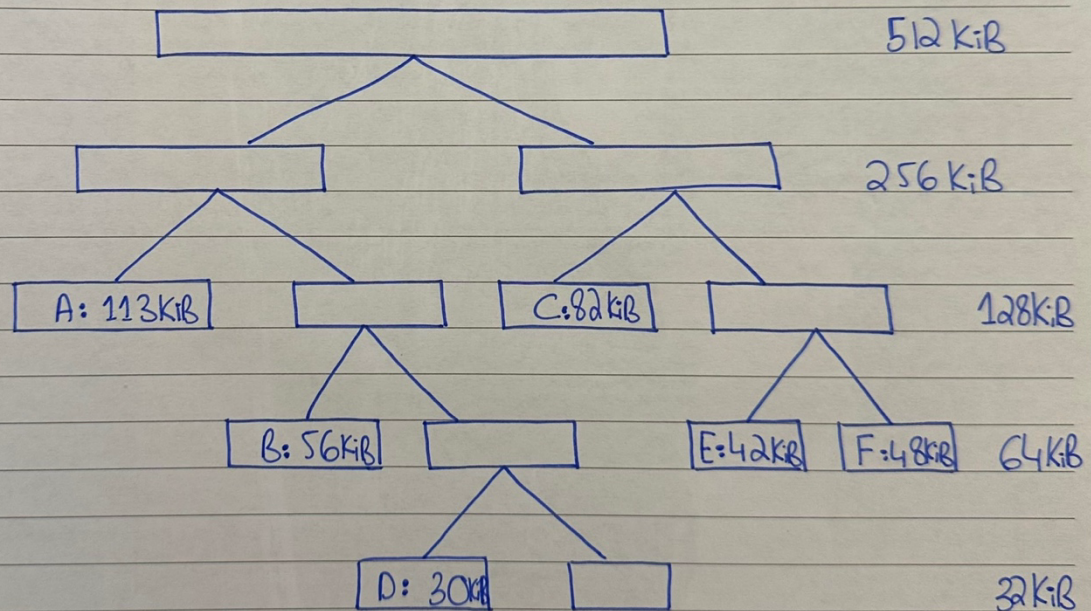
Problem 7.2:

a)

a)

512						
A	128		256			
A	B	64	256			
A	B	64	C	128		
A	B	D	32	C	<del>128</del>	128
A	B	D	32	C	E	64
A	B	D	32	C	E	F

Remaining memory = 32 KiB free



b)

$$A: 128 \text{ KiB} - 113 \text{ KiB} = 15 \text{ KiB}$$

$$B: 64 \text{ KiB} - 56 \text{ KiB} = 8 \text{ KiB}$$

$$C: 128 \text{ KiB} - 82 \text{ KiB} = 46 \text{ KiB}$$

$$D: 32 \text{ KiB} - 30 \text{ KiB} = 2 \text{ KiB}$$

$$E: 64 \text{ KiB} - 42 \text{ KiB} = 22 \text{ KiB}$$

$$F: 64 \text{ KiB} - 48 \text{ KiB} = 16 \text{ KiB}$$

$$15 + 8 + 46 + 2 + 22 + 16 = 109 \text{ KiB}$$

The largest chunk of memory is 32 KiB that can be allocated.

- c) If process C returns, 128 KiB is freed. We now have 128 KiB and 32 KiB free, but 132 KiB cannot be allocated since they aren't contiguous. Hence, allocation of 132 KiB is not possible.

Problem 7.3:

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

Reference string	1	4	2	3	4	4	1	3	2	1
Frame 0	1	1	2	2	4	4	4	3	3	1
Frame 1		4	4	3	3	3	1	1	2	2

Reference string	1	4	2	3	4	4	1	3	2	1
Frame 0	1	1	1	3	3	3	3	3	3	3
Frame 1		4	4	4	4	4	1	1	1	1
Frame 2			2	2	2	2	2	2	2	2

- b) Belady's Optimal (BO) page replacement algorithm:

Reference string	1	4	2	3	4	4	1	3	2	1
Frame 0	1	1	2	3	3	3	3	3	2	2
Frame 1		4	4	4	4	4	1	1	1	1

Reference string	1	4	2	3	4	4	1	3	2	1
Frame 0	1	1	1	1	1	1	1	1	1	1
Frame 1		4	4	4	4	4	4	4	4	4
Frame 2			2	3	3	3	3	3	2	2

- c) Least Recently Used Algorithm:

Reference string	1	4	2	3	4	4	1	3	2	1
Frame 0	1	1	2	2	4	4	4	3	3	1
Frame 1		4	4	3	3	3	1	1	2	2

Reference string	1	4	2	3	4	4	1	3	2	1
Frame 0	1	1	1	3	3	3	3	3	3	3
Frame 1		4	4	4	4	4	4	4	2	2
Frame 2			2	2	2	2	1	1	1	1