CNG334 INTRODUCTION TO OPERATING SYSTEMS

ASSIGNMENT 3 – REPORT

Fatma Erem Aksoy

06/06/23

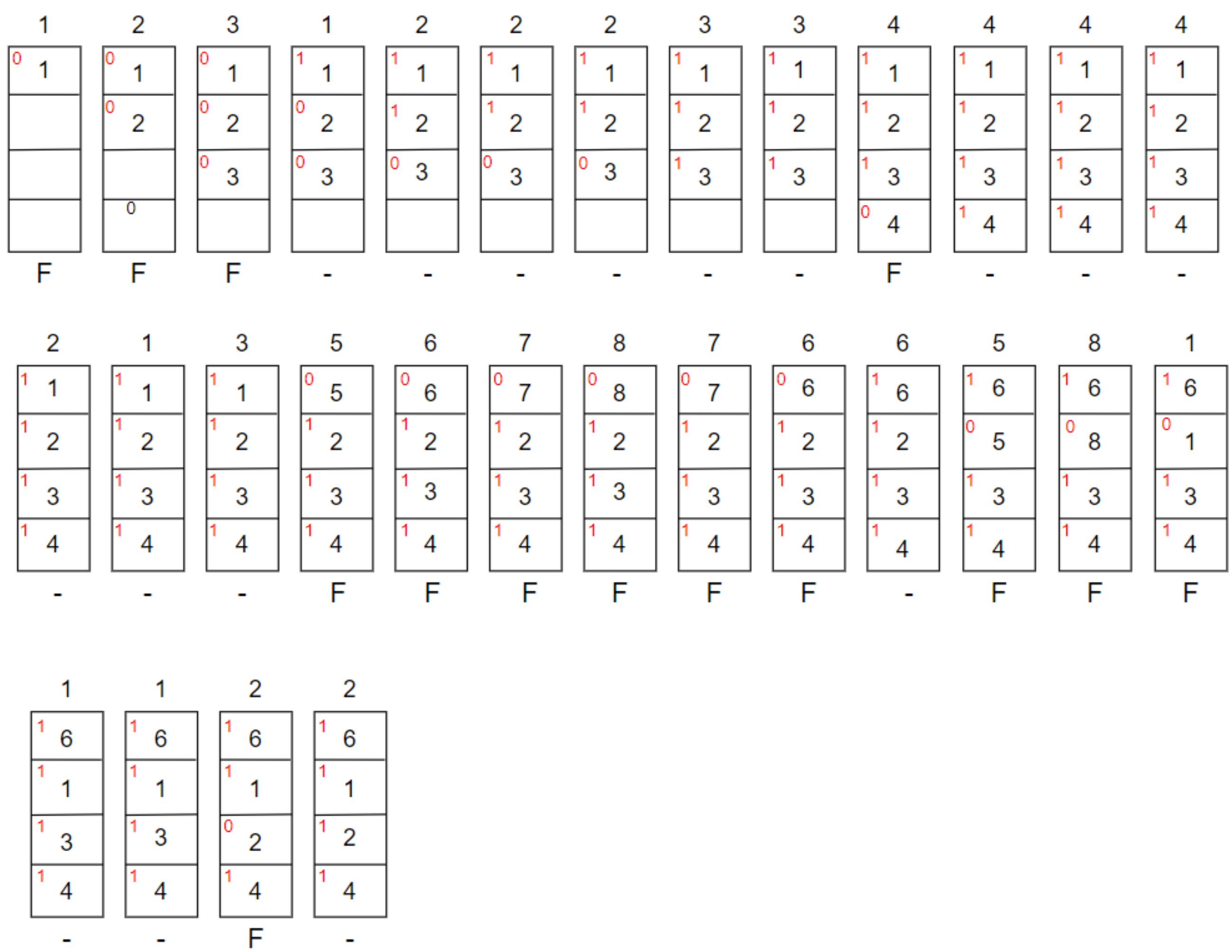
Task 1:

1,2,3,1,2,2,2,3,3,4,4,4,4,2,1,3,5,6,7,8,7,6,6,5,8,1,1,1,2,2

Compute the number of page faults resulting from the stated reference string with 4 frames.

Solution:

- **For Second Chance Algorithm:** In this algorithm, we look at the value of reference bit of each page to replace. The reference bit (shown in red in the figure below) is set to 0 when the page appears for the first time in the memory (when page fault), and it is changed to 1 if it comes again (no page fault). The replacement takes place with the one that has 0 as a reference bit. If all the reference bits are 1 or 0 when a replacement is needed, then FIFO is applied and the one that is in the memory for longest is replaced. (F represents page fault and - represents that there is no page fault, it is a hit)



- **For LRU:** In this algorithm, we look at the past allocations. We replace the page that has not been used for the longest period of time. (F represents page fault and - represents that there is no page fault, it is a hit)

1	2	3	1	2	2	2	3	3	4	4	4	4
1	1	1	1	1	1	1	1	1	1	1	1	1
	2	2	2	2	2	2	2	2	2	2	2	2
		3	3	3	3	3	3	3	3	3	3	3
									4	4	4	4
F	F	F	-	-	-	-	-	-	F	-	-	-

2	1	3	5	6	7	8	7	6	6	5	8	1
1	1	1	1	1	7	7	7	7	7	7	7	1
2	2	2	2	6	6	6	6	6	6	6	6	6
3	3	3	3	3	3	8	8	8	8	8	8	8
4	4	4	5	5	5	5	5	5	5	5	5	5
-	-	-	F	F	F	F	-	-	-	-	-	F

1	1	2	2
1	1	1	1
6	6	2	2
8	8	8	8
5	5	5	5
-	-	F	-

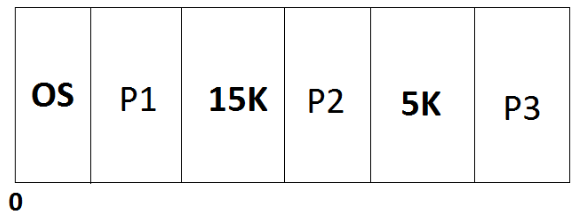
- **For Optimal algorithm**: In this algorithm, we look ahead, next pages to be allocated. We replace the page that will not be used for the longest period of time in future (F represents page fault and - represents that there is no page fault, it is a hit). When there is more than one option to replace as they do not occur in the rest of the string, it is assumed that we replace the one that has been changed the longest amount of time ago (like in the last part of the string -1,1,1,2,2- as none of the numbers in the current memory, we first replaced 5 and then 6 as they are the ones that came to the memory before other ones, basically FIFO is applied in this situation)

1	2	3	1	2	2	2	3	3	4	4	4	4
1	1	1	1	1	1	1	1	1	1	1	1	1
	2	2	2	2	2	2	2	2	2	2	2	2
		3	3	3	3	3	3	3	3	3	3	3
									4	4	4	4
F	F	F	-	-	-	-	-	-	F	-	-	-

2	1	3	5	6	7	8	7	6	6	5	8	1
1	1	1	1	1	1	8	8	8	8	8	8	8
2	2	2	2	2	7	7	7	7	7	7	7	7
3	3	3	3	6	6	6	6	6	6	6	6	6
4	4	4	5	5	5	5	5	5	5	5	5	1
-	-	-	F	F	F	F	-	-	-	-	-	F

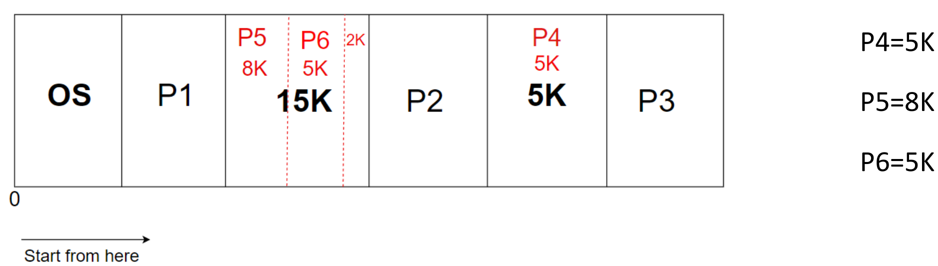
1	1	2	2
8	8	8	8
7	7	7	7
6	6	2	2
1	1	1	1
-	-	F	-

Task 2: As seen in the given variable-partition based memory layout, the processes P1, P2 and P3 are currently in memory. If the processes P4=5k, P5=8k and P6=5k arrive in this sequence, plot the memory layouts.

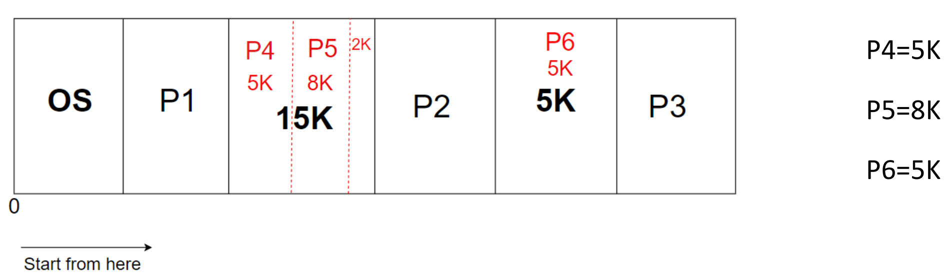


Solution: As it is variable-partition based, it means that the allocation is handled dynamically.

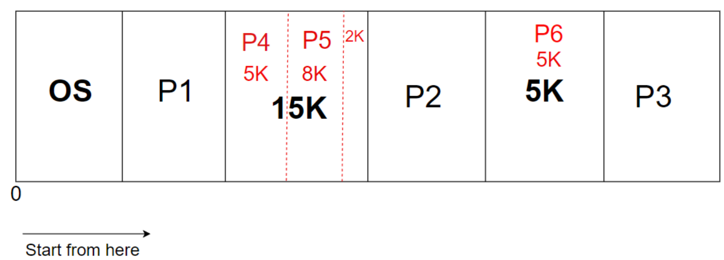
- **Best-fit:** All the three processes given were able to execute and allocated accordingly like shown in the figure below. As it is best-fit algorithm, it assigns the processes to the smallest location that they can fit into to minimize the internal fragmentation (allocation is done in P4, P5, P6 order).



- **First-fit**: All the three processes given were able to execute and allocated accordingly like shown in the figure below. As it is first-fit algorithm, it assigns the processes to the first location that they can fit into (allocation is done in P4, P5, P6 order).



- **Worst-fit:** All the three processes given were able to execute and allocated accordingly like shown in the figure below. As it is worst-fit algorithm, it assigns the processes to the largest memory location that they can fit into (allocation is done in P4, P5, P6 order). For process P5; the algorithm assigns it to the 15K as there will be 10K empty space after placing the P4 there and when we compare this amount of space with the other option which is 5K, the algorithm selects the larger one, 10K, and place P5 to 15Kresulting 2K of empty space as shown in the figure.



P4=5K

P5=8K

P6=5K

Task 3: What is the **largest possible value of X** so that DCBA is a safe sequence?

Solution:

Processes	Maximum					Allocated					Needed				
	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5	R1	R2	R3	R4	R5
A	1	0	2	1	1	1	1	2	1	3	0	1	0	0	2
B	2	0	1	1	1	2	2	X	1	1	0	2	(X-1)	0	0
C	1	1	0	1	0	2	1	3	1	0	1	0	3	0	0
D	1	1	1	1	0	1	1	2	2	1	0	0	1	1	1

Available sources:

0	0	2	1	1
R1	R2	R3	R4	R5

For the sequence DCBA to be a safe sequence, all the processes must execute in this order; so start executing from Process D:

Current available resources satisfy the need of D, so execute the process:

Process D:

  0 0 2 1 1
  1 1 1 1 0
+ _______
  1 1 3 2 1 -> new available resources; as it satisfy the need of Process C, execute C now

Process C:
 1 1 3 2 1
 1 1 0 1 0
+_______
 2 2 3 3 1 -> new available resources

For the new available resources to satisfy Process B, we check the need of B. For it to be satisfied, R3 must be less than or equal to 3 (X-1<=3). So using this equality, the largest value X can take is 4. (X=4)

When X=4, execute Process B:

Process B:
 2 2 3 3 1
 2 0 1 1 1
+_______
 4 2 4 4 2 -> new available resources; as it satisfy the need of Process A, execute A now

Process A:
 4 2 4 4 2
 1 0 2 1 1
+_______
 5 2 6 5 3 -> all the processes are executed successfully; so DCBA is a safe sequence with the value X=4

Task 4:

1) Swap-in: 300 MB
Swap-out: 150 MB & 200 MB
Transfer rate = 60 MB/s
Average latency time = 10 ms

1. The transfer time if the process to be swapped into hard disk:
   300/60 = 5 s = 5000 ms
   5000 + 10 = **5010 ms** (time spent **for swap-in**)

2. The transfer of the processes to be swapped out from memory:
   150/60 = 2.5 s = 2500 ms
   2500 + 10 = 2510 ms

   200/60 = 3.33 s = 3330 ms
   3330 + 10 = 3340 ms

   2510 + 3340 = **5850 ms** (time spent **for swap-out**)

⇨ So the total swap-time: 5010 ms + 5850 ms = 10860 ms = **10.86 s**

2) Code size = 700 K
Relocation register = 30010 K
Limit register address = 31000 K
Logical addresses: 990 & 1020

So knowing that **relocation register + logical address = physical address**,
We need to find the physical addresses and then check if it is in the requested range of register address.

For 990: physical address = 30010 + 990 = 31000 -> it is in the range, so this physical address is accepted.

For 1020: physical address = 30010 + 1020 = 31030 -> it is out of the range, so this physical address isn't accepted, so there will be an interrupt.