

OPERATING SYSTEMS

Professor: Hungwen Li

HOMEWORK 6

Name: Jeyanthh Venkatachari Ravikumar

SJSU ID: 010720528

8.11 Memory Partitions are as follows,

- M1=300KB
- M2=600KB
- M3=350KB
- M4=200KB
- M5=750KB
- M6=125KB

The Algorithm process size are as follows,

- P1=115KB
- P2=500KB
- P3=358KB
- P4=200KB
- P5=375KB

First Fit: In this method, The OS scans through the available memory partitions and chooses whichever is big enough for the process to execute

Process/Memory	300KB	600KB	350KB	200KB	750KB	125KB
P1 115KB	✓(Rem:185)					
P2 500KB		✓(Rem:100)				
P3 358KB					✓(Rem:392)	
P4 200KB			✓(Rem:150)			
P5 375KB					✓(Rem:17)	
Memory Wasted	185	100	150		17	

Best Fit: In this method, The OS scans through the available memory partitions and chooses whichever is small enough for the process to execute

Process/Memory	300KB	600KB	350KB	200KB	750KB	125KB
P1 115KB						✓(Rem:10)
P2 500KB		✓(Rem:100)				
P3 358KB					✓(Rem:392)	
P4 200KB				✓(Rem:0)		
P5 375KB					✓(Rem:17)	
Memory Wasted		100			17	10

Worst Fit: In this method, The OS scans through the available memory partitions and chooses whichever is large enough for the process to execute

Process/Memory	300KB	600KB	350KB	200KB	750KB	125KB
P1 115KB					✓(Rem:635)	
P2 500KB					✓(Rem:135)	
P3 358KB		✓(Rem:242)				
P4 200KB			✓(Rem:150)			
P5 375KB						
Memory Wasted		242	150		135	

The P5 Process will wait until sufficient memory is free again.

Based on efficiency, In this scenario,

BestFit >First Fit>Worsts Fit

Generally,

Best Fit algorithm gives the best performance,

However it is difficult to compare worst for and first fit as it depends on the processes and sizes of memory partitions

8.12

a. **Contiguous memory allocation:** Contiguous memory allocation requires relocation of the entire program as it does not have sufficient space for the program to grow its allocated memory space.

b. **Pure Segmentation:** Pure Segmentation also requires relocation of the entire program as it does not have sufficient space for the segment to grow its allocated memory space.

c.**Pure Paging:** In Pure Paging incremental allocation of the new pages is possible without relocation of the entire program.

8.13

Contiguous memory allocation:

- **External Fragmentation:** This scheme suffers from external fragmentation as address spaces are allocated contiguously and holes develop as old processes dies and new processes are initiated.
- **Internal Fragmentation :** This scheme does not suffer from internal Fragmentation
- **Sharing Code across Processes:** It also does not allow processes to share code, since a process's virtual memory segment is not broken into noncontiguous fine grained segments.

Pure Segmentation:

- External Fragmentation : This scheme suffers from external fragmentation as the segments are laid out continuously and when segments of old processes are replaced by new ones, it may lead to holes
- Internal Fragmentation : This scheme does not suffer from internal Fragmentation
- Sharing Code across Processes: It can share code as two processes can share code segment having different data segments

Pure Paging:

- External Fragmentation :It does not suffer from external fragmentation
- Internal Fragmentation: If pages are not utilized properly, it may lead to internal fragmentation
- Sharing Code across Processes: It enables processes to share code at the granular level.

8.20

Assuming a 1-KB page size, what are the page numbers and offsets for the following address references (provided as decimal numbers):

Given 1KB= 1024 bytes

a. 3085

Page number= $3085/1024=3.01=3$

Offset= $3085\% 1024=13$

b. 42095

Page Number= $42095/1024=41.1=41$

Offset= $42095\% 1024=111$

c. 215201

Page Number= $215201/1024=210.15=210$

Offset= $215201\% 1024=161$

d. 650000

Page Number= $650000/1024=634.76=634$

Offset= $650000\% 1024=784$

e. 200000

Page Number= $200000 /1024=210.15=1953.12=953$

Offset= $200000 \% 1024=129$

8.28

Consider the following segment table:

Segment Base Length

0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

What are the physical addresses for the following logical addresses?

a. 0,430

This Lies in Segment 0

Therefore Base=219

Physical Address= $219+430=649$

b. 1,10

This Lies in Segment 1

Therefore Base=2300

Physical Address= $2300+10=2310$

c. 2,500

This Lies in Segment 2

Therefore Base=90

But it exceeds the limit, leading to an Trap

d. 3,400

This Lies in Segment 3

Therefore Base=1327

Physical Address= $1327+400=1727$

e.

4,112

This Lies in Segment 4

Therefore Base=1952

But it exceeds the limit, leading to an Trap