Tutorial 4: Memory Management

Q1. Differentiate between the followings:

(a) Fixed partition and dynamic partition

Fixed Partition	Dynamic Partition
partition sizes are critical	improves memory user over fixed partition
entire program is stored contiguously in memory during entire execution	available memory kept in contiguous blocks and jobs are given only as much memory as they request when loaded.
occur internal fragmentation when there are unused memory spaces in partition	fragments of free memory are created between blocks of allocated memory. (external fragmentation)

(b) Internal fragmentation and external fragmentation

Internal Fragmentation

Internal fragmentation occurs when there are unused memory spaces within the partition itself.

External Fragmentation

Fragments of free memory are created between blocks of allocated memory

(c) Best Fit algorithm and First Fit algorithm.

Best Fir	First Fit
Faster to implement but not may not be making efficient use of memory space.	Uses memory efficiently but slower to implement because the entire free list table needs to be searched before allocation can be made.
Algorithms are less complex.	Algorithms are more complex because they need to find the smallest block of memory into which the job can fit.
Memory list organized according to memory locations, low-order	Memory list organized according to memory size, smallest to largest.

Q2. Consider a system is using fixed *memory partition* techniques and 4 processes waiting in a queue as show in the figure 1.

Free Space List	
Partition Number	**Size (KB)**
A	350
B	550
C	750
D	500

Queuing Processes	
Process	**Size (KB)**
1	225
2	500
3	540
4	360

Figure 1

Show how the four processes, namely P1, P2, P3 and P4, are allocated memory partitions by using the following memory allocation algorithms.

(i) First-fit

Partition	**Job**
A	P1
B	P2
C	P3
D	P4

Remaining space = 125 + 50 + 210 + 140 = 525 KB (wasted)

(ii) Best-fit

Partition	**Job**
A	P1
B	P4
C	P3
D	P2

Remaining space = 125 + 210 + 190 = 525KB (wasted)

(iii) Worst-fit

*P3 need to wait

Partition	**Job**
A	
B	P2
C	P1
D	P4

Remaining space = 350 + 525 + 50 + 140 = 1065KB (wasted)

Q3. A system is using variable-size partitions (*dynamic memory partition* techniques are used); partitions can be allocated on the basis of first-fit, best-fit and worst-fit. In a contiguous memory allocation system, the free space list contains 6 entries in the following order: 190KB, 550KB, 220KB, 420KB, 650KB, 110KB.

Given the following requests in the input queue: A=210KB, B=430KB, C=100KB, D=430KB, determine how these requests can be satisfied based on each of the allocation schemes listed below.

- First-fit
- Best-fit
- Worst-fit

For the scenario mentioned above, which allocation scheme uses the memory more efficiently? Explain why.

First Fit (Process D waiting)

Memory	Process
190KB	C (90KB wasted)
550KB	A (340KB wasted)
220KB	
420KB	
650KB	B (220KB wasted)
110KB	

Total fragmentation = 90 + 340+ 220 = 650 KB

Best Fit

Memory	**Process**
190KB	
550KB	B (120KB wasted)
220KB	A (10KB wasted)
420KB	
650KB	D (220KB wasted)
110KB	C (10KB wasted)

Total fragmentation = 120 + 10+ 220+10 = 360 KB

Worst Fit (Process D waiting)

Memory	**Process**
190KB	C (90KB wasted)
550KB	A (340KB wasted)
220KB	
420KB	
650KB	B (220KB wasted)
110KB	

Total fragmentation = 90 + 340+ 220 = 650 KB

Best fit allocation scheme, this is because the total wasted space is the smallest KB and all process can be allocated.

Q4. Page replacement is fundamental in demand paging. Some popular page replacement algorithms are First-In-First-Out (FIFO), Optimal and Least-Recently-Used (LRU).

(i) Illustrate how each of the above page replacement algorithm works, by using the page reference string of a process as given below. You may assume that 3 frames are allocated to the process and they are initially empty. For each algorithm, determine also the number of page fault.

2, 3, 2, 1, 5, 2, 4, 5, 3, 2

2	**3**	**2**	**1**	**5**	**2**	**4**	**5**	**3**	**2**

FIFO

2	**2**		**2**	**5**	**5**	**5**		**3**	
	3		**3**	**3**	**2**	**2**			
			1	**1**	**1**	**4**			

Page Fault = 7

Optimal

2	**2**		**2**	**2**		**4**			2
	3		**3**	**3**		**3**			3
			1	**5**		**5**			5

Page Fault = 6

Least-Recently-Used (LRU).

2	**3**	**2**	**1**	**5**	**2**	**4**	**5**	**3**	**2**

2	**2**		**2**	**2**		**2**		**3**	**3**
	3		**3**	**5**		**5**		**5**	**5**
			1	**1**		**4**		**4**	**2**

Page Fault = 7

(ii) Highlight **ONE (1)** advantage and **ONE (1)** disadvantage of the optimal page replacement algorithm.

Advantage- does not suffer from Belady's anomaly

Disadvantage -difficult to implement, requires future knowledge of the referencing s tring

(iii) Explain the concept of thrashing in systems which support virtual memory via demand paging.

an excessive amount of page swapping back and forth between main memory and secondary storage.

Q5. Support a page size of 256 bytes is used in demand paging system. Given the following sequence of addresses:

321, 150, 700, 510, 1031, 400, 350, 150, 842, 910

(i) Translate the given virtual addresses inti a page reference string.
1,0,2,1,4,1,1,0,3,3

(ii) Prepare a page trace analysis and count the number of page faults by First-In-First-Out (FIFO), Optimal and Least-Recently-Used (LRU) page replacement algorithms, assuming 3 page frames to be allocated. Then, compute the hit ratio for each algorithm.

FIFO

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
321	321	321	510	510	510	350	350	350	910
	150	150	150	1031	1031	1031	150	150	150
		700	700	700	400	400	400	842	842

Page Fault = 10

ORU

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
321	321	321	510	510	510	350	350	350	910
	150	150	150	1031	1031	1031	150	150	150
		700	700	700	400	400	400	842	842

Page Fault = 10

LRU

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
321	321	321	510	510	510	350	350	350	910
	150	150	150	1031	1031	1031	150	150	150
		700	700	700	400	400	400	842	842

Page Fault = 10

Q6. CPU generates a logical address that is mapped to physical memory location. This is implemented by the operating system which maintains page and segment tables for the mapping.

(i) Why the page sizes always in the powers of 2?
Page number (p) is an index into a page table which contains base address of each page in physical memory.

(ii) Consider a logical address space of 32 pages of 1024 words each, mapped onto a physical memory of 64 frames. How many bits are there in the logical address and physical address respectively?
logical address space = total no.of bits needed + bit required to map
(total pages) (page offset)

64 pages + 1024 words = 5 + 10 = 15 bits

Physical address = frame no + frame offset
= 6 + 10
= 16 bits
LA - page size + number of page
PA - number of frame + page size (offset)

Q7.
(i) Memory management using paging is more common than segmentation. Highlight **TWO (2)** advantages that paging systems have over segmentation systems
1.Doesn't allow external fragmentation
2.Programs are divided into equal-sized pages

(ii) In a particular segmentation system, a process P has a segment table shown below:

Segment	Base	Length (Bytes)
0	1250	700
1	300	90
2	2400	550

Figure 2: Segment Table

Explain how the system establishes the corresponding physical address for a logical address of <1, 35> through the use of the segment table.

What will happen during address translation for a logical address of <2,600>?

<1,35> , 1 = segment number , 35 = displacement / offset in the segment, we need to retrieve the < base, length > . The offset 35 is checked against the length to ascertain its validity (must be less than length). As 35 < 90 , the offset is then added to the base ie 300 + 35 to obtain the physical address ie 335.

<2,600> = segment 2 requests 600 lines, this results in a trap because it is invalid ,segment 2 only has 550 length.

Self-Review

Q1. Given the following information in **Table 2**:

Free Space List	
Partition	Size (KB)
A	650
B	600
C	250
D	300
E	650

Queuing Processes	
Process	Size (KB)
1	600
2	650
3	250
4	300

Table 2

Show how the four processes are allocated into fixed memory partitions when the ***first-fit*** and ***best-fit*** file allocation algorithms are applied.
Note: Show your answers using the following table format.

Table format:

Partitio n	Partition Size	Proces s	Process Size	Internal Fragmentation

Q2. Consider the following page reference string:

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

Assume that 5 frames are initially empty in the memory. Perform the page faults trace to determine the number of page faults that will occur for the following page replacement algorithms.

(i) First-In-First-Out (FIFO)

(ii) Least Recently Used (LRU)