AACS2284 Operating Systems

Tutorial 4: Memory Management

Q1. Differentiate between the followings:

(a) Fixed partition and dynamic partition

Fixed partition	Dynamic partition
Fixed partition allows multiprogramming which is one partition for each job. The size of each partition remains static once the system is in operation	Available memory is kept in contiguous blocks and jobs are given only as much memory as they request when loaded

(b) Internal fragmentation and external fragmentation

Internal fragmentation	External fragmentation
Internal fragmentation occurs when there are unused memory spaces within partition itself	External fragmentation is fragments of free memory that are created between blocks of allocated memory

(c) Best Fit algorithm and First Fit algorithm.

Best Fit algorithm	First Fit algorithm
Allocate the first partition that is big enough	Allocate the smallest partition that is big enough
Keep free/busy lists organized by memory location (low-order to high-order)	Keep free/busy lists ordered by size (smallest to largest)
Faster in making the allocation	Produces the smallest leftover partition and make best use of memory

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

(ii) Best-fit

(iii) Worst-fit

(i) First Fit

Partition	Process	Fragmentation
A	P1	125
B	P2	50
C	P3	210
D	P4	140
Total		525

- No job needs to wait. Total fragmentation is 525kb

(ii) Best Fit

Partition	Process	Fragmentation
A	P1	125
B	P3	10
C	P4	390
D	P2	0
Total		525

- No job needs to wait. Total fragmentation is 525kb

(iii) Worst Fit

Partition	Process	Fragmentation
A		
B	P2	50
C	P1	525
D	P4	140
Total		715

- P3 will need to wait as there is no available block large enough (540kb) to store. Total fragmentation is 715kb

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	Best-fit	Worst-fit
A 550KB	A 220KB	A 650KB
B 650KB	B 550KB	B 550KB
C 110KB	C 110KB	C 420KB
D not allocated	D 650KB	D not allocated

- Best-fit uses the memory most efficiently as all the requests can be satisfied with the least amount of unused memory due to external fragmentation

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

FIFO:

2	3	2	1	5	2	4	5	3	2
2	2	2	2	5	5	5	5	3	3
	3	3	3	3	2	2	2	2	2
			1	1	1	4	4	4	4
F	F	H	F	F	F	F	H	F	H

Total of Page fault = 7, fault ratio = 7/10 = 0.7
Total of Page Hit = 3, hit ratio = 3/10 = 0.3

Optimal:

2	3	2	1	5	2	4	5	3	2
2	2	2	2	2	2	4	4	4	4
	3	3	3	3	3	3	3	3	2
			1	5	5	5	5	5	5
F	F	H	F	F	H	F	H	H	F

Total of Page fault = 6, fault ratio = 6/10 = 0.6
Total of Page Hit = 4, hit ratio = 4/10 = 0.4

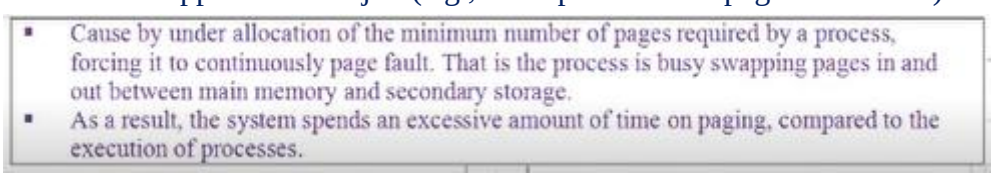
LRU:

2	3	2	1	5	2	4	5	3	2
2	2	2	2	2	2	2	2	3	3
	3	3	3	5	5	5	5	5	5
			1	1	1	4	4	4	2
F	F	H	F	F	H	F	H	F	F

Total of Page fault = 7, fault ratio = 7/10 = 0.7
Total of Page Hit = 3, hit ratio = 3/10 = 0.3

(ii) Highlight **ONE (1)** advantage and **ONE (1)** disadvantage of the optimal page replacement algorithm.
Advantage – Lowest page fault rate
Disadvantage – Difficult to implement

(iii) Explain the concept of thrashing in systems which support virtual memory via demand paging.
Thrashing –An excessive amount of page swapping back and forth between main memory and secondary storage.
– when a page is removed from memory but is called back shortly thereafter.
– Can occur across jobs, when a large number of jobs are fight for a relatively few numbers of free pages.
– Can happen within a job (e.g., in loops that cross page boundaries).


- Cause by under allocation of the minimum number of pages required by a process, forcing it to continuously page fault. That is the process is busy swapping pages in and out between main memory and secondary storage.
- As a result, the system spends an excessive amount of time on paging, compared to the execution of processes.

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	0	2	1	4	1	1	0	3	3
1	1	1	1	4	4	4	4	3	3
	0	0	0	0	1	1	1	1	1
		2	2	2	2	2	0	0	0
F	F	F	H	F	F	H	F	F	H

Total of Page fault = 7, fault ratio = 7/10 = 0.7
Total of Page Hit = 3, hit ratio = 3/10 = 0.3

Optimal:

1	0	2	1	4	1	1	0	3	3
1	1	1	1	1	1	1	1	3	3
	0	0	0	0	0	0	0	0	0
		2	2	4	4	4	4	4	4
F	F	F	H	F	H	H	H	F	H

Total of Page fault = 5, fault ratio = 5/10 = 0.5
Total of Page Hit = 5, hit ratio = 5/10 = 0.5

LRU:

1	0	2	1	4	1	1	0	3	3
1	1	1	1	1	1	1	1	1	1
	0	0	0	4	4	4	4	3	3
		2	2	2	2	2	0	0	0
F	F	F	H	F	H	H	F	F	H

Total of Page fault = 6, fault ratio = 6/10 = 0.6
Total of Page Hit = 4, hit ratio = 4/10 = 0.4

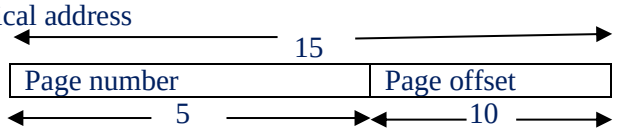
Q6. CPU generates a logical address that is mapped to physical memory location. This is implemented by the operating system which maintains page and segment tables for the mapping.

(i) Why the page sizes always in the powers of 2?
- The bytes within a page are addressed using the last N bits of a virtual address, for some value of N. Since the number of addresses that can be expressed with N bits is 2^N, the page size is a power of 2.

(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?

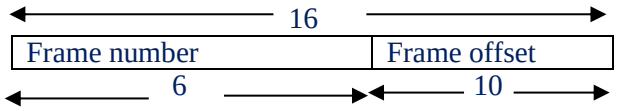
Addressing within a 1024 words page requires 10 bits because $1024 = 2^{10}$. Since the logical address of $32 = 2^5$ pages, the logical address must be 10 + 5 = 15 bites.

Logical address



Similarly, since there are $64 = 2^6$ frames, physical addresses are 6 + 10 = 16.

Physical address



Q7. (i) Memory management using paging is more common than segmentation. Highlight **TWO (2)** advantages that paging systems have over segmentation systems

Advantages of paging over segmentation

- Allocation of memory space for processes is easy as the size of the page and frame are the same
- Eliminate external fragmentation

(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>

Logical address *<1,35>* where *1* represents the segment number and *35* the displacement/offset in the segment. During address translation, the segment number is used as index to the segment table 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 i.e. 335. As for the logical address *<2, 600>*, since the offset is larger the segment length (*550*) thus a trap occurs due to addressing error.

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.

First-fit

Partition	Partition Size	Process	Process Size	Internal Fragmentation
A	650	1	600	50
B	600	3	250	350
C	250			
D	300	4	300	0
E	650	2	650	0

Best-fit

Partition	Partition Size	Process	Process Size	Internal Fragmentation
A	650	2	650	0
B	600	1	600	0
C	250	3	250	0
D	300	4	300	0
E	650			

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)

1	2	4	3	5	6	6	3	5	1	4	3	7	2	4
1	1	1	1	1	6	6	6	6	6	6	6	6	6	6
	2	2	2	2	2	2	2	2	1	1	1	1	1	1
		4	4	4	4	4	4	4	4	4	4	7	7	7
			3	3	3	3	3	3	3	3	3	3	2	2
				5	5	5	5	5	5	5	5	5	5	4
F	F	F	F	F	F	H	H	H	F	H	H	F	F	F

Total of Page fault = 10, fault ratio = 10/15 = 1.5
Total of Page Hit = 5, hit ratio = 5/15 = 1/3

(ii) Least Recently Used (LRU)

1	2	4	3	5	6	6	3	5	1	4	3	7	2	4
1	1	1	1	1	6	6	6	6	6	6	6	7	7	7
	2	2	2	2	2	2	2	2	1	1	1	1	1	1
		4	4	4	4	4	4	4	4	4	4	4	4	4
			3	3	3	3	3	3	3	3	3	3	3	3
				5	5	5	5	5	5	5	5	5	2	2
F	F	F	F	F	F	H	H	H	F	H	H	F	F	H

Total of Page fault = 9, fault ratio = 9/15 = 0.6
Total of Page Hit = 6, hit ratio = 6/15 = 0.4