COMP2432

Tutorial 8 – Memory Management

Question 1

1. **Contiguous Memory Allocation.**
Consider a memory of total size of 5000K that is managed using IBM MVS ***MVT*** approach. After a sequence of memory requests, there is a list of 4 holes of size 200, 400, 600, and 300K, available in that order. Indicate how the requests of 280, 180, 320, 400, 120, and 160K *arriving in that order* are served using (*a*) **first-fit**, (*b*) **best-fit**, and (*c*) **worst-fit** memory allocation algorithms. Are there any unsatisfied requests using the three algorithms? What is the *utilization* of each of the three allocation algorithms? Utilization is defined to be amount of memory *used* divided by *total* amount of memory. Which is the best and which is the worst?
Using *compaction*, could you satisfy all the requests in the three algorithms?

First Fit

	200	400	600	300	
FF	200	400	600	300	UNSATISFIED
280					
180					
320					
400					
120					
160					

Best Fit

	200		400		600		300		

BF	200	400	600	300	UNSATISFIED
280					
180					
320					
400					
120					
160					

Worst Fit

	200	400	600	300	
WF	200	400	600	300	UNSATISFIED
280					
180					
320					
400					
120					
160					

Question 2

Try this question!

2. More Contiguous Memory Allocation.

Consider a memory of total size of 1000K managed using IBM MVS ***MVT*** approach. After a sequence of memory requests, there is a list of 3 holes of size 100, 180, and 120K. You are given these 10 requests: 31, 79, 24, 75, 34, 65, 46, 11, 27, and 18K, arriving in that order. Indicate how the requests are satisfied with (*a*) **first-fit** algorithm, (*b*) **best-fit** algorithm, and (*c*) **worst-fit** algorithm. What are the *utilizations* for the three algorithms respectively? Which is the best and which is the worst?

Using *compaction*, could you satisfy all the requests in the three algorithms?

Question 3

3. Paging.

Consider logical address translation in a paged system, based on byte-addressable memory. The size of the logical address space is 32MB and the size of a page is 512 bytes. The data is stored in a physical main memory of size 256KB.

(*a*) What is the length of the logical or virtual address?

(*b*) What is the length of the physical address?

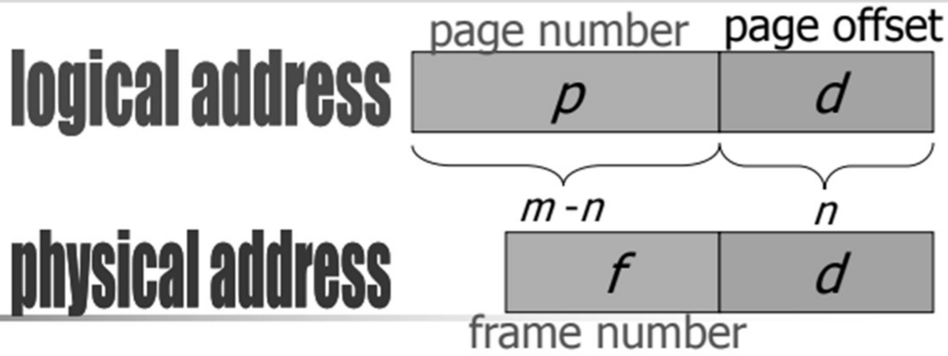
(*c*) What is the length of the displacement or offset field?

(*d*) What is the length of the page number field?

(*e*) How many frames are there in the main memory?

Review lecture notes!

Paging



- Since each logical address generated by CPU needs to be mapped or translated into a physical address in memory, this translation must be very efficient.
- This should be done with hardware support.
- The logical address is divided into two parts.
  - Page number (p): it is used as an index into a page table to look up for base address of each frame in physical memory.
  - Page offset (d): it is added (or appended) to the base address to form the physical memory address.
- Assume that the logical address space is of size 2^m and each page (and frame) has a size of 2^n.
  - Logical address has a size of m bits.
  - Page number p contains $m–n$ bits.
  - Page offset d contains n bits.
- Paging suffers from internal fragmentation.

Question 3

We would simply show you how we do the part a). For the rest of the question, please try them by yourself first before you read the solution.

What is the length of the logical or virtual address?

- First of all, could you tell me what the size of the virtual memory?
- 32MB
- How to convert that into bit(s)?

Question 3

- 32 MB could be rewritten as 2^5 MB.
- Then, it is $2^5 \times 2^{10}$ KB
- Expand it to $2^5 \times 2^{10} \times 2^{10}$ B
- Finally, we have 2^{25} B
- Check the lecture notes again. We have the following.

- **Assume that the logical address space is of size 2^m and each page (and frame) has a size of 2^n.**
  - Logical address has a size of *m* bits.

- Length of logical address = **25** bits.

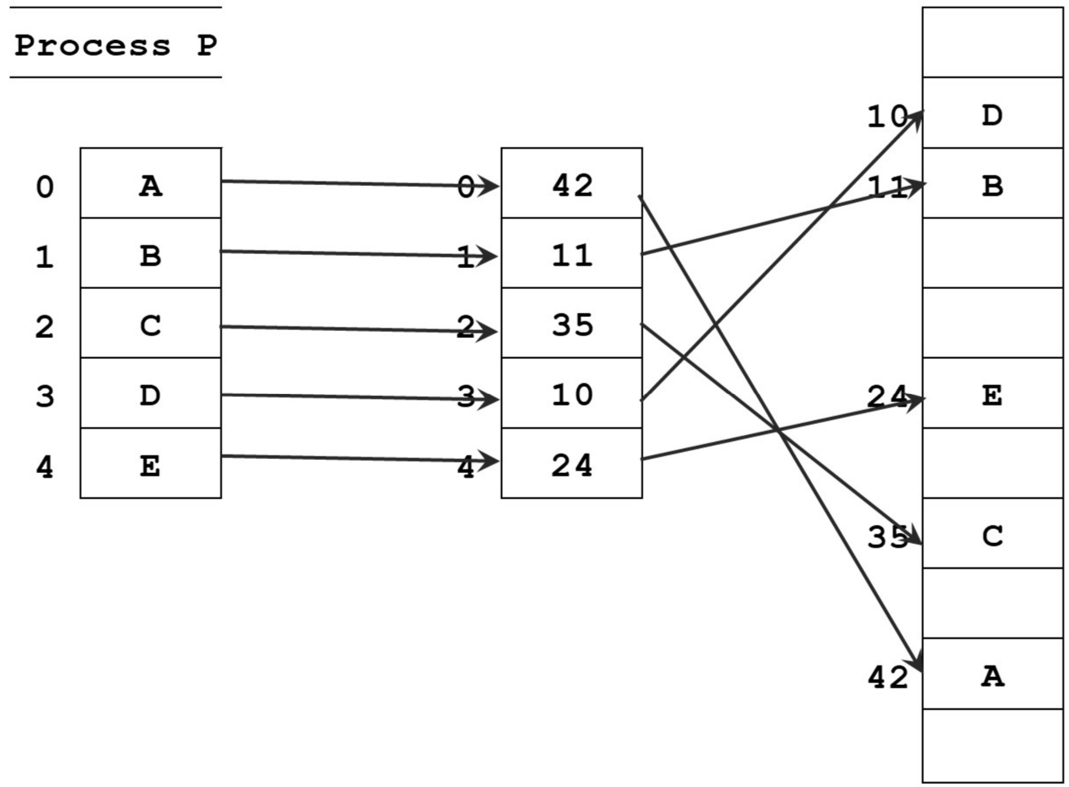
Question 4

4. Page Table and Memory Frame.

Consider a process *P* that occupies 5 pages (page 0 to page 4), storing contents *A*, *B*, *C*, *D* and *E* respectively. There are 7 free frames available in the memory, maintained by the operating system in the frame number order 42, 11, 35, 10, 24, 16 and 40. Draw a *diagram* to indicate the content of the frames and the page table after all the 5 pages of *P* are loaded into the physical memory. Assuming that each page is of size 1KB and 6 bits are used to indicate frame number, determine the *physical address* that the logical address 001001100011 is mapped to. What would be the size of the main memory in this case?

Question 4

Question 4



Question 5

5. Segmentation.

Consider the segment table for process P_1. The segment table contains the following entries.

P_1

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

Determine the physical addresses for the following logical addresses of P_1.

(*a*) (0, 430) (*b*) (1, 200) (*c*) (2, 30) (*d*) (3, 400) (*e*) (4, 112)

Question 5

P1		
Segment	Base	Length
0	1357	544
1	123	333
2	2011	331

P2		
Segment	Base	Length
0	2468	742
1	2011	331
2	789	445

Recall this slide.

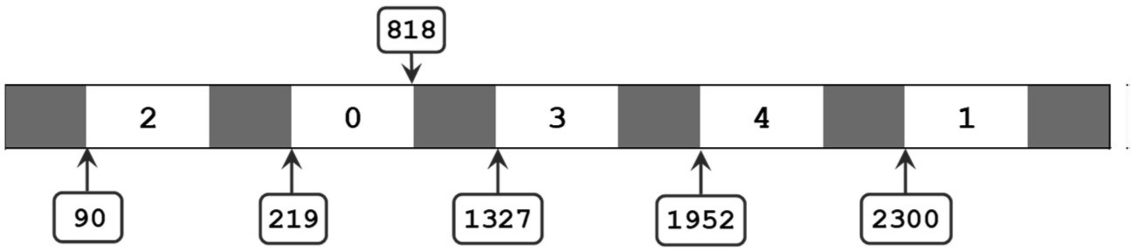
Shared Segments

- Use of segments also allows easy sharing of program code or shared data across processes.
- Page sharing is easier, since all pages are of the same size, at fixed boundary, but segment sharing needs more works.
- An example of two processes sharing data with 4K memory.
  - P_1 has three segments: S_0 to S_2.
  - P_2 also has three segments: S_0 to S_2.
  - P_1-S_2 and P_2-S_1 refer to the same shared segment.

- Are the following addresses valid?
  - P_1 tries to access <1,123>
  - P_2 tries to access <2,432>
  - P_1 tries to access <0,789>
  - P_2 tries to access <1,331>

Question 5

- Simply show you a) (0, 430).
- For Segment 0, its base address is [219] and has a length of [600].
- That means, the valid address for Segment 0 is from [219] to [818].
- For (0, 430), we simply check [219 + 430] whether is falls in the range.
- [219 + 430] = 649
- Obviously, [649] is a valid address.
- Try figure out the rest of the question.
- And, it would be better to prepare a drawing



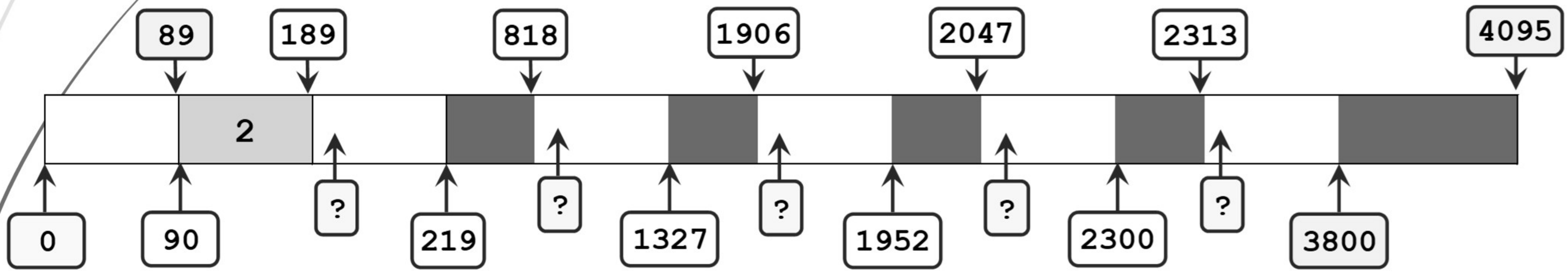
Question 6

6. Segmentation with Sharing.

Suppose that segment 2 of P_1 in **Question 5** is an array for grade information to be *shared* to another process P_2, which has a first segment of size 781 for its program, a second segment of size 483 for its global variables and a final fourth segment of size 425 for its stack. Assume that the computer has a small memory of 4KB, with physical address from 0 to 4095 and that the segment tables are stored at the high memory from 3800 to 4095. Segments for P_2 are to be allocated from the free memory using (*a*) *first-fit algorithm*, (*b*) *best-fit algorithm,* and (*c*) *worst-fit algorithm*. Show the three different segment tables as produced by the three algorithms for P_2. ***Determine*** the physical addresses by performing the five logical address translations as in **Question 5** for the three algorithms, assuming that they are now generated by P_2.

Question 6

- First, we need to find out holes which are available for P2. For example, hole (0 to 89) is one of the holes available.
- How about the others?



Question 6

- For Process 2,

Segment	Base	Length
0	?	781
1	?	483
2	90	100
3	?	425

- For Segment 2, it is shared with P1.

Question 6

- We can expand the table and it would look like the one below.
- Here, we simply show you the "base addresses" of Segment 0 of the three algorithms.

Segment P2	**Base Address**			**Length**
	First Fit	**Best Fit**	**Worst Fit**	
0	2314	2314	2314	781
1				483
2	90	90	90	100
3				425