University Center of Mila
2nd year bachelor's degree in computer science
Course: Operating Systems 1

Tutorial Series No. 4

Exercise 1:

1) Consider a segmented virtual memory system with the following segment table:

Segment Number	Presence in Memory	Memory Address	Size
0	1	230	150
1	1	25	200
2	0	1200	1500
3	1	533	70

Give the corresponding physical addresses for the following virtual addresses:
(3,20), (1,150), (0,0), (1,250), (2,1000)

2) Now consider a paged virtual memory system where the page size is 2 KB.
What are the corresponding physical addresses for the same virtual addresses: (3,20), (1,150), (0,0), (1,250), (2,1000)
Assume that physical memory is composed of 4 frames, containing the following pages:

Page 0	
Page 4	
Page 3	
Page 1	Adresse 0

3) A program makes references to the following virtual addresses:
(0, 11), (3, 400), (5, 150), (2, 45), (0, 1200), (4, 900), (5, 0), (6, 5), (2, 230), (3, 50), (0, 70)
Calculate the page fault rate under the following replacement strategies:

- **a)** LRU (Least Recently Used)
- **b)** Second-Chance Algorithm
- **c)** FIFO (First In, First Out)

Assume the physical memory size is 8 KB and the page size is 2 KB.

4) Repeat question 3 if the physical memory size is 6 KB.

5) If the size of a virtual address (p,d) is 20 bits, the physical memory size is 8 KB, and the physical page size is 4 KB, what is the bit size of p?

Exercise 2:

A program has a code section occupying 1024 bytes, and it uses a vector of 1000 characters (1 character = 1 byte). This program is executed in a system using paging, with the following characteristics: Real memory size: 1 MB, Page size: 512 bytes, Memory reference instructions use a 24-bit address field

1) Provide:

- **a)** The size of the logical address space
- **b)** The number of bits for the offset
- **c)** The number of bits for the virtual page number
- **d)** The number of bits for a physical address
- **e)** The number of bits for the physical page number (frame)
- **f)** The number of entries in the page table

2) Does loading this program into memory cause internal fragmentation? Justify your answer.

Exercise 3:

Consider a segmented and paged virtual memory system, where: Page size: 4 KB, Physical memory: 64 KB, Process P has an address space composed of three segments: S1: 16 KB, S2: 8 KB and S3: 4 KB.

At a certain point in time, the following pages are loaded into physical memory:

Pages 2 and 3 of S1 → frames 0 and 2, Page 2 of S2 → frame 9, Page 1 of S3 → frame 12

1) For a data item located at virtual address 8212, indicate:

- **a)** The segment
- **b)** The page number in the segment
- **c)** The offset within the page
- **d)** The frame number
- **e)** The offset within the frame
- **f)** The physical address

Exercise 4: (Supplement)

A machine is connected to a text terminal managed by a controller. The controller has a buffer (called video RAM) through which data travels from the computer (CPU + main memory) to the terminal.

1) Given that the terminal can display 25 lines × 80 columns, how much video RAM is needed at the controller?

2) If the terminal is replaced by a graphical screen with a 24-bit color palette and a resolution of 1024 × 768 pixels, how much video RAM is now required?

Exercise 5: (Supplement)

You want to manage text printing (a stream of characters) on a printer. The printer is controlled by a controller made up of three registers:

- RegD: An 8-bit data register
- RegC: A 1-bit command register, written by the processor to indicate that it is ready (i.e., a new character is available to print)
- RegE: A 2-bit status register, First bit: Read by the processor to indicate the printer is ready to process a character or that the previous character has been printed, Second bit: Signals a paper out error.

1) What I/O mode is used by this device?

2) Now we want to connect the controller to an interrupt circuit to enable interrupt-driven I/O. Is it worthwhile to use this mode, given the following:

- Printing speed: 6 pages/min, with each page containing 50 lines of 80 characters
- Writing a character into RegD takes negligible time
- Handling an interrupt for each printed character takes 50 microseconds