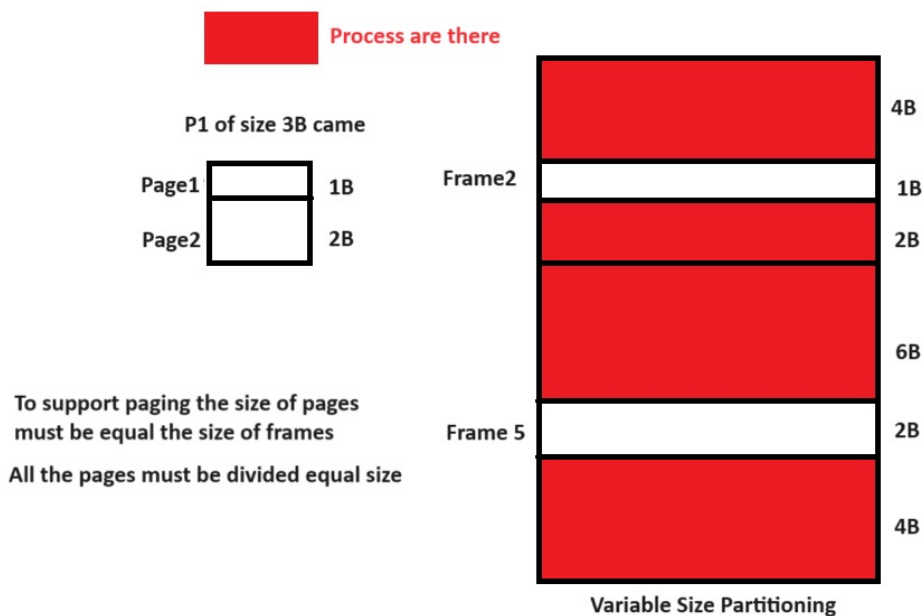# OS Notes Day-4 Date: 30-08-2024

Paging – What is paging?

- Paging
    - Dividing the process into fixed sizes pages is know as paging.
    - Why Paging: Instead of loading the whole process which can't be loaded into main memory OS loads few pages of the process into main memory accoring to frames available and other pages loaded from storage device on demand of CPU.

- Paging table: It is a special table maintained by MMU (Memory Management Unit) to map the logical address of the demanded page with the physical address of the page in main memory where actualy the page is placed.
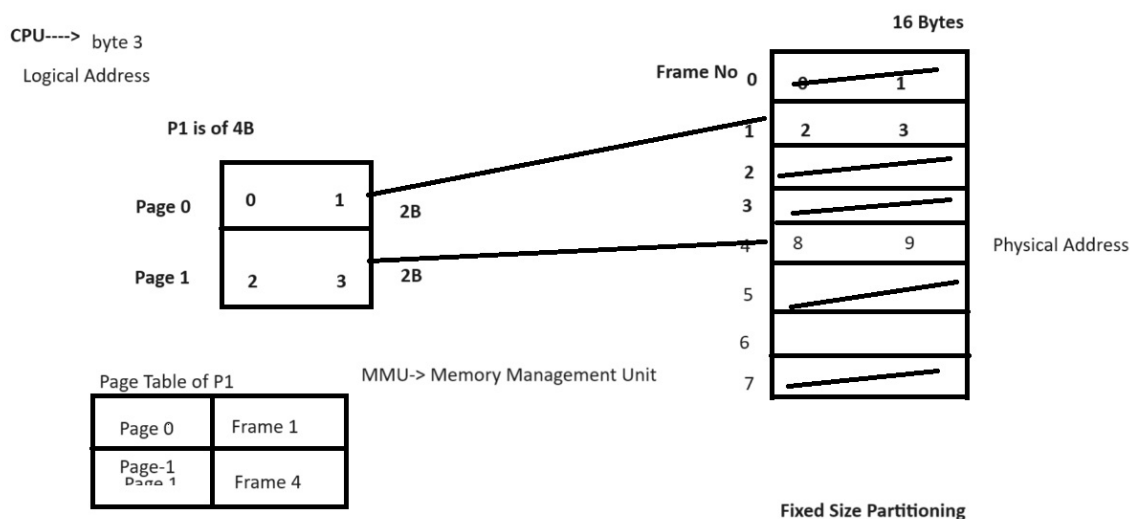
- Demand paging
    - If the page is loaded in memory frame as per the demand of CPU i.e. known as demand paging.
- Page faults: If CPU demands a page and its not there in main memory. That time an interrupt generated to load the page from virtual memory / hard-drive.
- Page replacement algorithms

1. FIFO (First in First Out)

P1---->	0	2	1	2	0	3	5	3	2	1	0
FIFO											
Frame-3			1	1	1	1	1	1	2	2	2
Frame-2		2	2	2	2	2	5	5	5	5	0
Frame-1	0	0	0	0	0	3	3	3	3	1	1
	MISS	MISS	MISS	HIT	HIT	MISS	MISS	HIT	MISS	MISS	MISS

- Belady's Anomly: It state that if we increase the number of frames, the HIT Ratio decreased.

2. LRU

P1----->	2	3	2	4	3	5	4	3	2	7
LRU										
		2	3	2	4	3	5	4	3	2	7
Frame-1	8	8	8	8	4	4	4	4	4	4
Frame-2	3	3	3	3	3	3	3	3	3	3
Frame-3	2	2	2	2	2	2	2	2	2	2
Frame-4	6	6	6	6	6	6	5	5	5	7
	Init	HIT	HIT	HIT	MISS	HIT	MISS	HIT	HIT	HIT	MISS

Mis %	30
Hit %	70

3. MRU

P1---->	0	2	4	6	8	7	3	5	3	2
MRU										
		0	2	4	6	8	7	3	5	3	2
Frame-1	0	0	2	4	6	8	7	7	7	7	7
Frame-2	3	3	3	3	3	3	3	3	3	3	2
Frame-3	5	5	5	5	5	5	5	5	5	5	5
	Initial	HIT	MISS	MISS	MISS	MISS	MISS	HIT	HIT	HIT	MISS

MISS Ration=60%	
HIT=	40%

If the page is not in the frame, its known as miss of the mage and it's a page fault

- Hardware required for paging (Virtual + Cache)
- What is virtual memory
    - Its a memory space in hard-drive, which work like physical memory to entertain large processes whose size is bigger than the RAM. It is an illusion memory.
    - In Virtual Memory process is divided into fixed sized partions know as pages.
    - OS loads the process's pages from virtual memory to physical memory on demand of CPU i.e. known as Demand Paging.
    - While working with virtual memory pages are saved as per their logical address.
    - While loading the pages from virtual memory to physical memory the pages get physical address.
    - If a page loaded from virtual memory to physical memory i.e. know as swap-in process.

- If the page is shifted from physical memory to logical memory by the replacement process i.e. know swap out.

Virtual Memory — Page Size: 2B — CPU — LA — Page No — Page Offset — 3 P — 1 D — MMU — Frame-1 — Memory — P1 — P2 — VAS
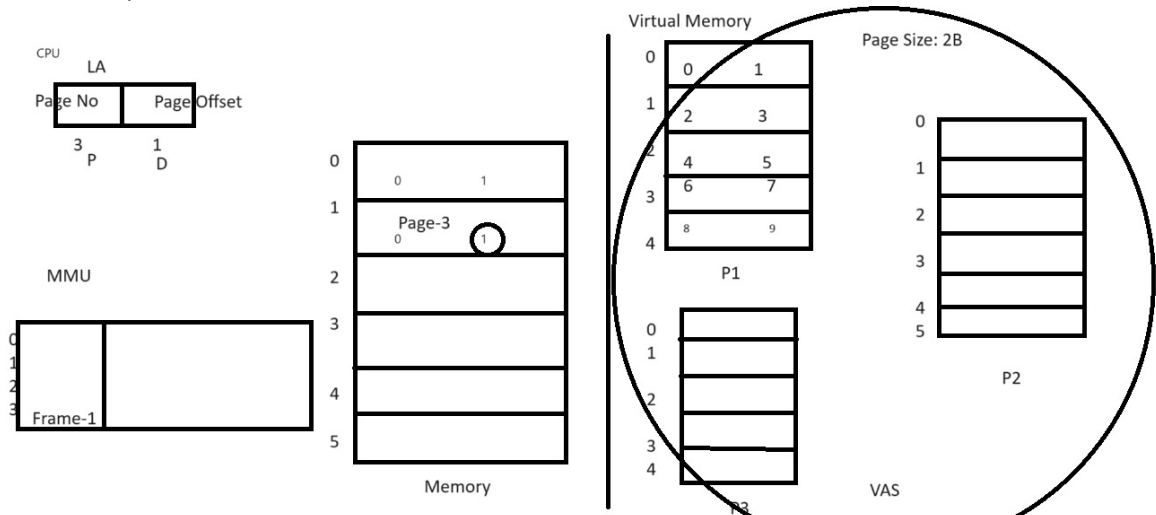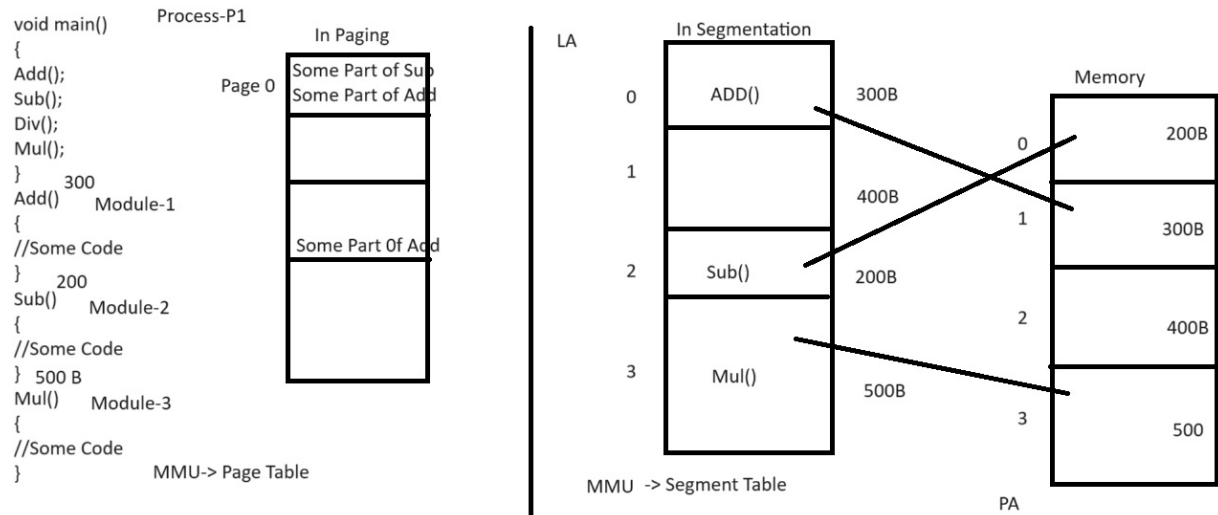
- Translation look aside buffer
    - It is a page table represented in cache memory to fasten the process of finding physical address for a given logical address.
    - In paging we load only few pages in main memory of a particuler process. Entries of that pages is made in MMU Page Table which is actually present at physical memory. To find a frame MMU first look into page table and then according page table entry it accress the given frame no. This process is actually access the RAM twice. So to reduce this access OS use Translation Look Aside Buffer in cache memory.

Translation Look Aside Buffer (Cache)

CPU->P1

| P | D |

LA

3   1

Page Table

Page-3	Frame-1
Page 9	Frame-7

y+x

x+x

MISS-

Page Table

Frame No	Valid/ InVld	Dirty Bit		
Frame-1	1	1		
Frame-7				

Page-8
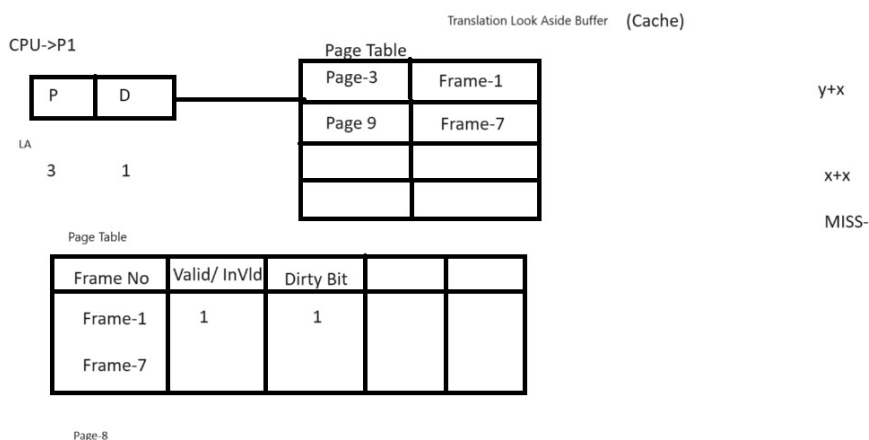
- Concept of dirty bit
    - It is a bit in page table which describe either the page content is updated during the execution or not. If for a specific frame the content is updated then its dirty is set to 1, other wise its always 0.

Segmentation – What is segmentation?

- It is the method of dividing a process into variable size partitions known as Segments on the basis of module / Functions of the process to be executed.

Process-P1

```
void main()
{
Add();
Sub();
Div();
Mul();
}       300
Add()   Module-1
{
//Some Code
}       200
Sub()   Module-2
{
//Some Code
} 500 B
Mul()   Module-3
{
//Some Code
}
```

In Paging

Page 0 | Some Part of Sub
Some Part of Add

Some Part Of Add

MMU-> Page Table

LA

In Segmentation

0 | ADD() | 300B
1 | | 400B
2 | Sub() | 200B
3 | Mul() | 500B

MMU -> Segment Table

Memory

0 | 200B
1 | 300B
2 | 400B
3 | 500

PA

Reading Assigment

- Shared pages and reentrant code
- Throttling

Shell Programming

- What is shell?
    - Shell is interface b/w user and kernal.
    - It take input from user and pass it on to the kernal.
    - An user can interact with kernal by using shell commands or shell script / program.
- What are different shells in Linux?
    - /bin/sh
    - /bin/bash
    - /usr/bin/bash
    - /bin/rbash
    - /usr/bin/rbash
    - /usr/bin/sh
    - /bin/dash
    - /usr/bin/dash
    - /usr/bin/tmux
    - /usr/bin/screen
- Shell variables
    - Shell varibale can be defined using any name without the type of the variable
- Example:

```
X=100;         //X will store value 100
Y="Malkeet"    //Y will store the value "Malkeet"
echo $X        //To access the variable you should use $ sign before the variable
name
```

- Read: Its is used to read input from keyboard