Lecture-25
BDU

Date 02/04/18

Process creation

Virtual memory allows other benefits :-

- Copy-on-write
- Memory-mapped files

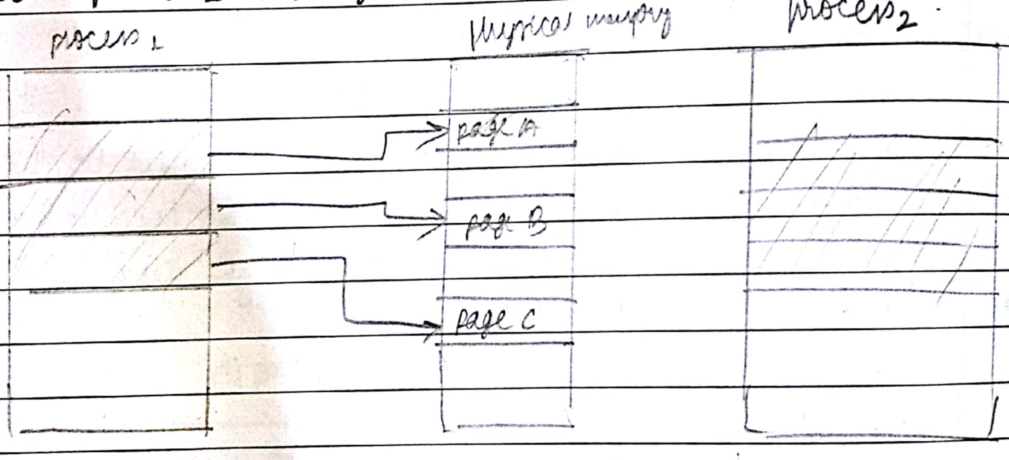
Copy-on-write (COW)

It allows both parent & child processes to initially share the same pages in memory.

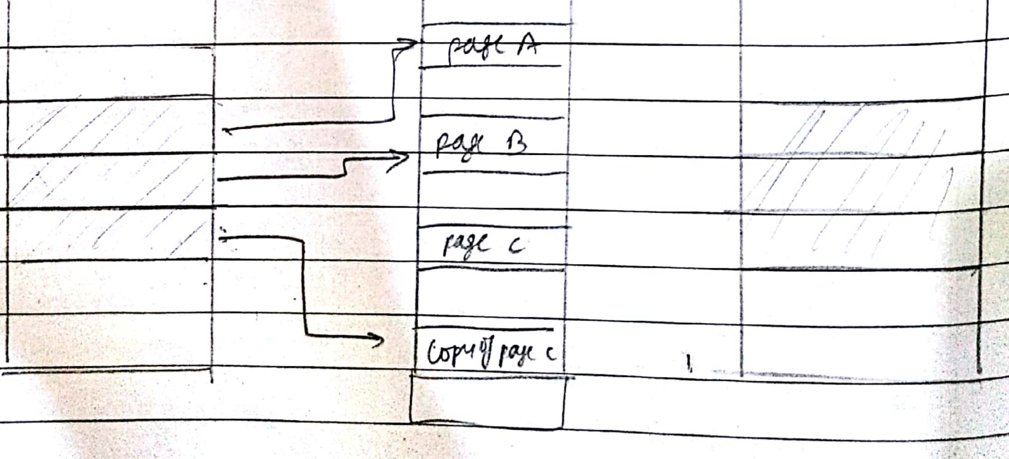
If either process modifies a shared page, ~~th~~ only then page is copied.

COW allows more efficient process creation as only modified pages are copied.

Before process1 modifies page C



After process 1 modifies page C.



What happens if there is no free frame?

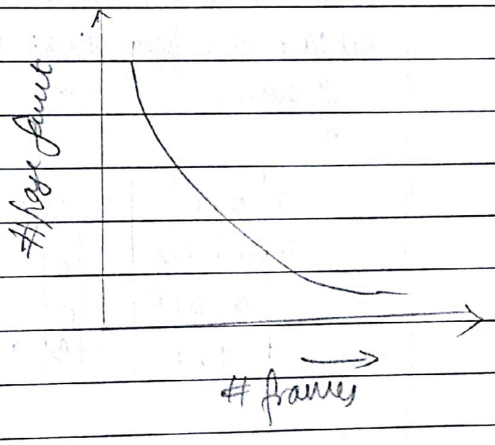
- **Page replacement** – find some page in memory, but not really in use swap it out.
  - algorithm
  - performance – want an algo which will result in min no. of page faults.
- Same page may be brought into memory several time.

Basic page replacement

- Find the location of the desired page on disk.
- Find a free frame –
  - If there is a free frame, use it
  - If there is no free frame, use a page replacement algo to select victim frame.
- Bring the desired page into the (newly) free frame update the page and frame tables.
- Restart the process.

Page replacement algorithms

- FIFO
- optimal
- LRU (least recently used)



FIFO page replacement

The page which comes first is replaced first.

F	F	F	F	H	F	F	F	F	F	F	H	H	F	F	H	H	F	F	F
7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2	·	2	2	4	4	4	0			0	0	8	[illegible]	7	7	7
	0	0	0		3	3	3	2	2	2			1	1		·	1	0	0
		1	1		1	0	0	0	3	3			3	2			2	2	1

20 total references

15 – Faults
5 – Hits

Optimal page replacement

- Replace the page that will not be used for the longest period of time.

Belady's anomaly (in FIFO)

If ~~fo~~ page fault increases even after increase in the # of frames.

F	F	F	F	H	F	H	F												
7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2		2	·	2			2		·	2				7		
	0	0	0		0		4			0			0				0		
		1	1		3		3			3			1				1		

9 - Faults   Better performance.

11 - Hits

Used for theoretically comparing other methods.

Least recently used (LRU)

Replace the page that has not been used for the longest period of time.

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2		2		4	4	4	0			1		1		1		
	0	0	0		0		0	0	3	3			3		0		0		
		1	1		3		3	2	2	2			2		2		7		