Ch9. Virtual Memory (완)

Created	@Jun 26, 2020 5:15 PM
Tags	운영체제

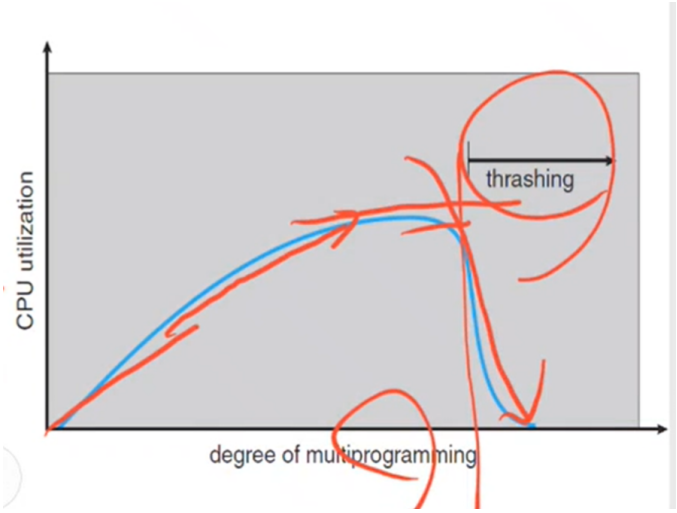
https://s3-us-west-2.amazonaws.com/secure.notion-static.com/d2ed7b91-8a25-4cf7-a986-eb2181132302/CH9._VirtMem.pdf

Copy-On-Write (COW)

- fork() + exec() : 새로운 프로세스를 만드는 메커니즘.
  - **fork()시에 메모리 copy가 일어나는데, 오버헤드가 큰 작업이다.** 어차피 exec() 해서 덮어씌울건데 굳이?
- **Copy-On-Write**
  - **child가 만들어지고** parent의 메모리를 복사해오는 게 아니고, **parent가 가리키는 페이지들을 모두 가리키게 한다.**
  - 그리고 **child에서 페이지 수정이 일어날 때만, 새 페이지에 복사가 일어나고 수정된다.**
  - **복사를 child에서의 수정 시점까지 최대한 미루겠다는 것.**
- **vfork()** : 페이지 복사가 일어나지 않음. 바로 exec()를 호출해야 한다. (애초에 child가 exec하기 전까지 parent가 suspend됨 ㅋㅋ)

Thrashing

- 프로세스마다 한 순간에 필요한 최소한의 프레임 수가 정해져 있다. → **Minimum number of frames.**
- **Thrashing : 프로세스에게 페이지가 충분하지 않아서 page fault가 계속 발생하는 것.**



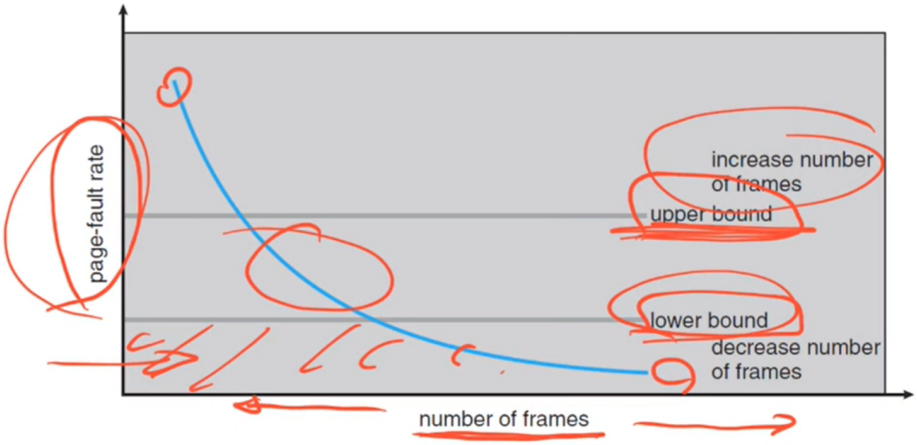
- 여러 프로그램들을 올리는 건 결국 **Multiprogramming의 degree를 올리는 것**이다.
- 이 때, 새 프로그램을 올릴 때마다 프레임을 할당해줘야 하는데, 당연히 프레임 빼 오니까 page fault 일어나는 일이 는다. (다른 프로그램의 minimum number of frames에 해당하는 프레임을 빼 오는 경우도 있을 수 있다.)
- page fault가 증가하면, 디스크에서 읽어서 메모리에 올려야 하는 건데, 이건 IO작업이다. CPU Utilization이 떨어진다.
- **그래서, Thrashing이 일어나는 시점을 기준으로 CPU Utilization이 급격히 감소.**
- Thrashing 방지
  - **Working-Set Model**
  - **Page-Fault Frequency**

Working-Set Model

Thrashing 해결책 : **minimum number of frames만큼 딱 맞게 들고 있는 프로세스들에게는 프레임을 빼 오면 안 된다!**

- **Working Set** : 특정 프로세스의 minimum number of frames. **프레임이 최소 몇 개 있어야 하는가?** → 이것만 알아내서 잘 맞추면 된다.
- OS는 **모든 프로세스들의 WS의 합, D(컴퓨터에서 필요로 하는 전체 프레임 개수)**를 모니터링한다.
  - D > (사용 가능한 프레임)이면, 프로세스 몇 개를 멈춰야 한다.
  - D < (사용 가능한 프레임)이면, 프로세스를 더 늘려도 된다.

Page-Fault Frequency (PFF)



기본적으로, 프레임 개수를 많이 주면 page-fault rate가 내려가고, 프레임 적게 주면 fault rate가 올라간다.

lower bound, upper bound를 놔서, **upper bound 이상으로 page fault가 올라가면 프레임을 더 투입**하고, **lower bound 이하로 page fault가 내려가면 너무 넉넉한 거라 프레임을 좀 뺏어간다.**