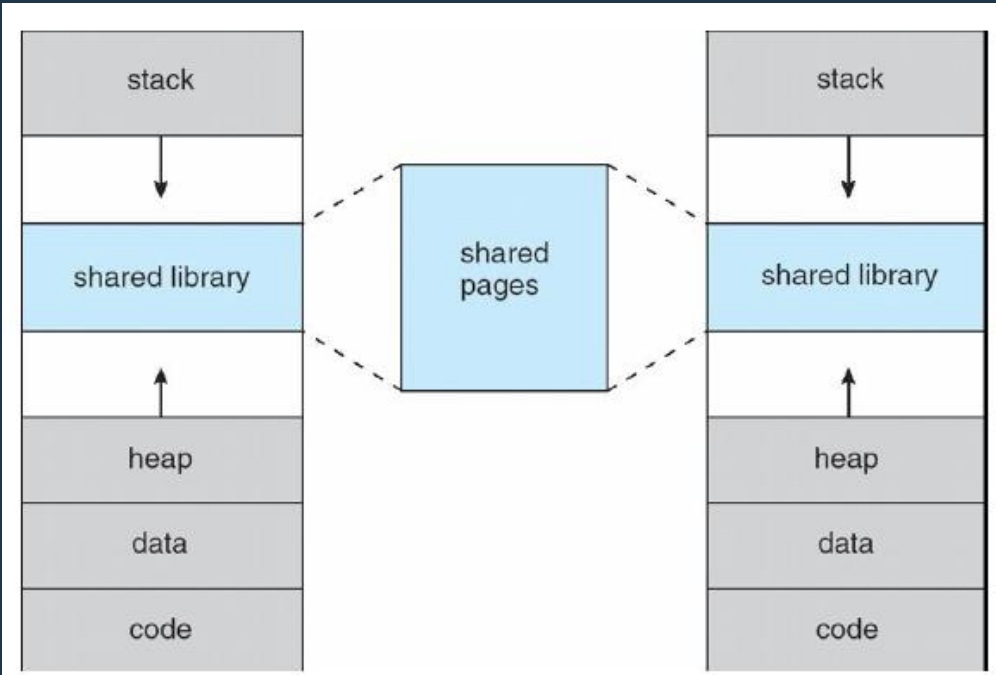


Virtual Memory(가상 메모리)

1. Virtual Memory

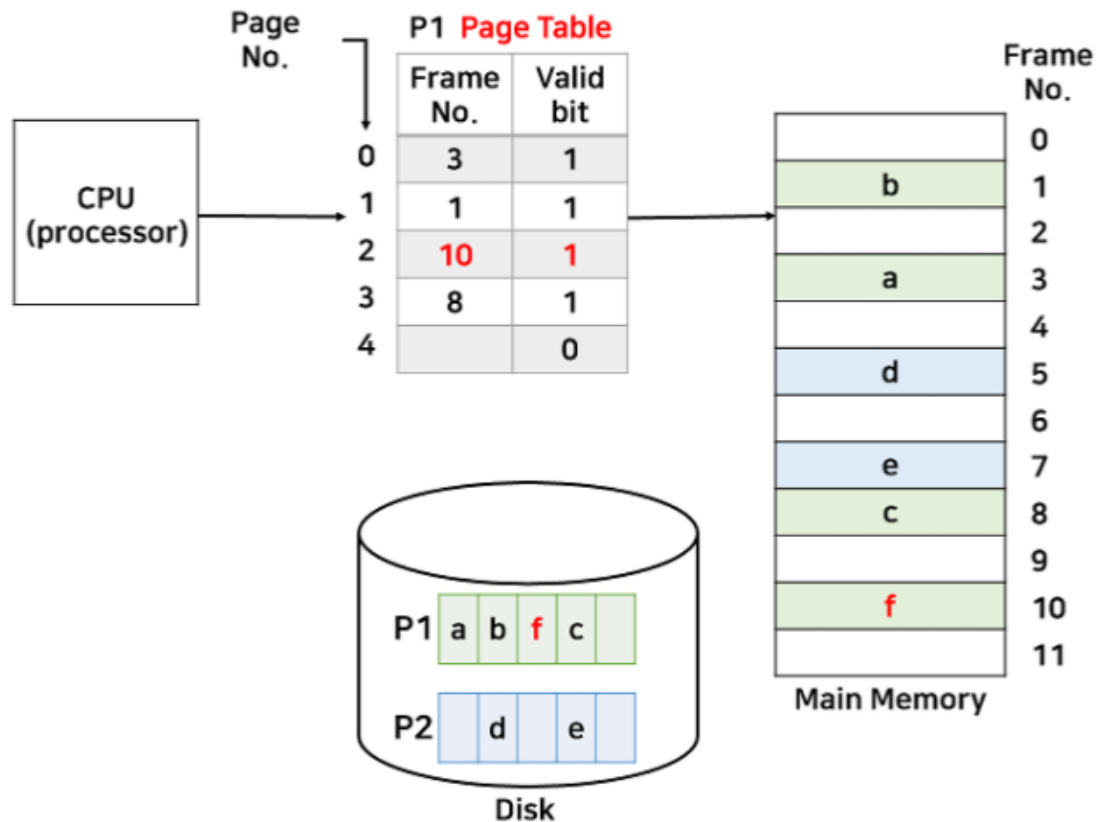
- Logical memory와 physical memory 구분
- Logical을 훨씬 더 많이 사용 가능.
- Programmer는 Memory 고려를 덜 할 수 있음.
- CPU utilization 증가

2. Sharing Library



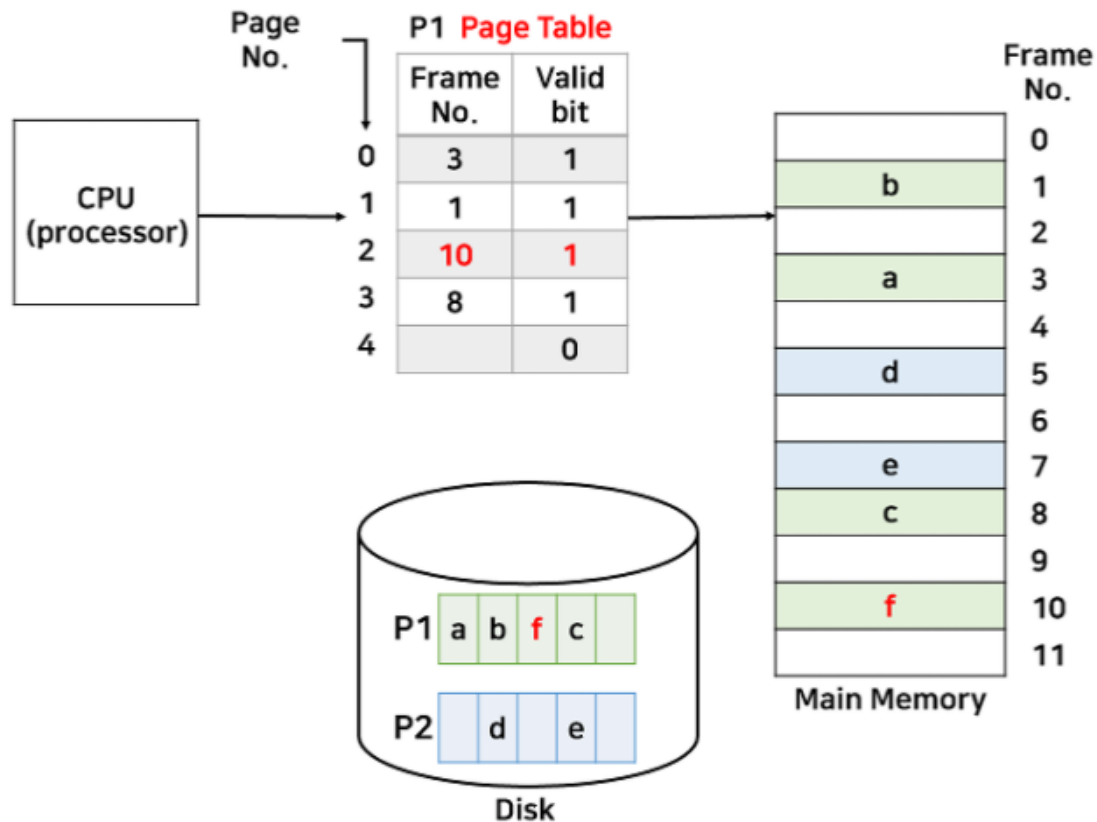
- 시스템 라이브러리 공유
- 각 프로세스는 자신의 가상 주소 공간에 두고 사용하는 것처럼 인식

3. Demand Paging



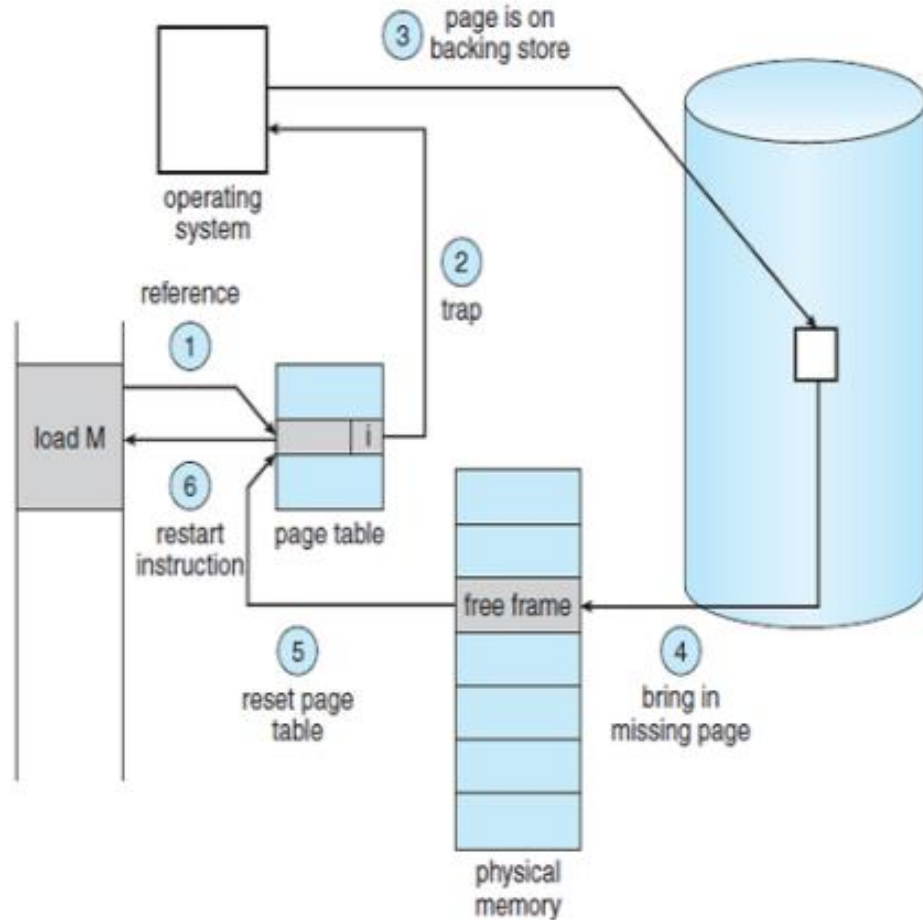
- 빈 page가 있을 때!
- Valid bit 사용
- 필요할 때만 Memory에 할당

3. Demand Paging



- 빈 page가 있을 때!
- Valid bit 사용
- 필요할 때만 Memory에 할당
- Valid = 0인 경우 => Page Fault

4. Page Fault



- Valid가 0이면 Interrupt 보냄.
- ISR에서 Disk 탐색하여 페이지 할당
- Valid = 1로 변경

5. Effective Access Time

=> Page Fault 고려해서 평균 소요시간 계산하는 것

- P : 페이지 부재 확률
- T_m : 메모리를 읽는 시간
- T_p : Page fault가 발생 했을 때 걸리는 시간

5. Effective Access Time

- $T_m = 200\text{nsec}$ (DRAM)
- $T_p = 8\text{msec}$ (seek time + rotational delay + transfer time)
- $T = (1-p) 200 + p 8,000,000 = 200 + 7,999,800 * p$
- $p = 1/1,000 \Rightarrow T = 8.2\text{usec}$ (40배 정도 느림)
- $p = 1/399,990 \Rightarrow T = 220\text{nsec}$ (10% 정도 느림)

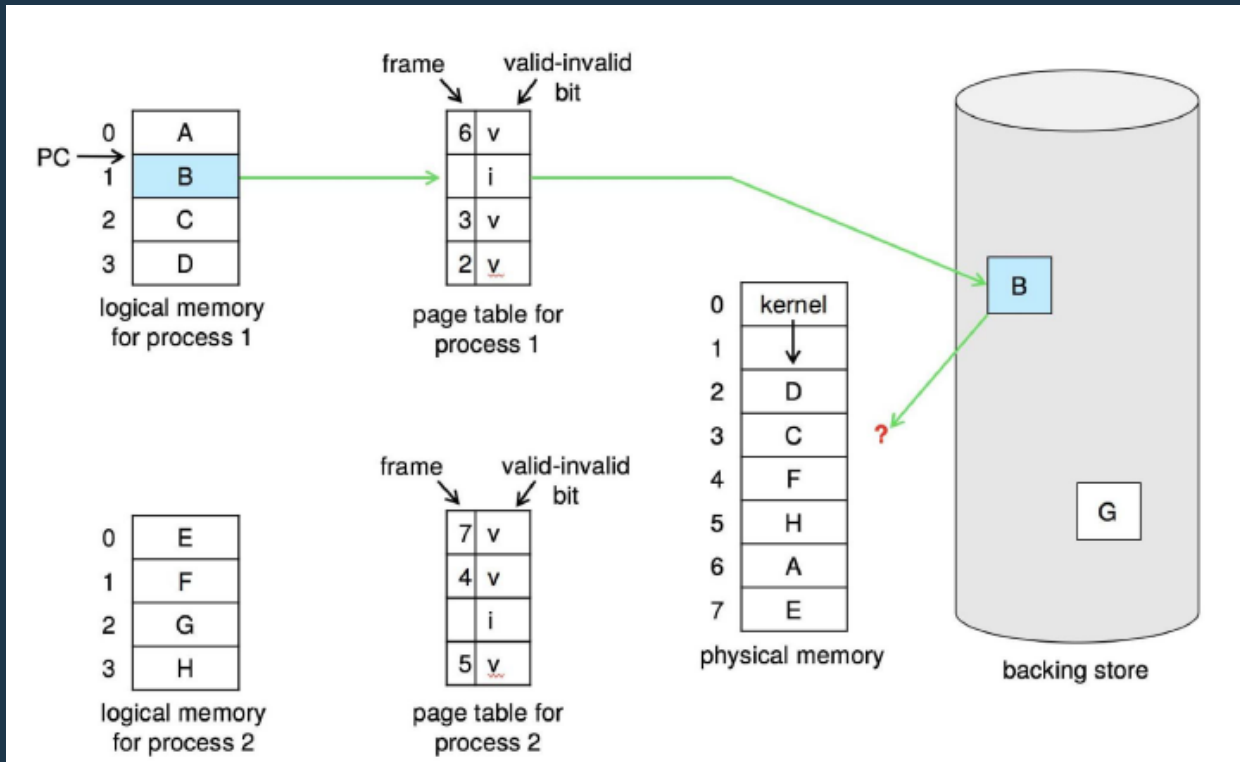
- 실제로는 Page Fault 발생 확률 매우 낮음

=> Locality(시간적, 공간적)

6. Swapping / Demand paging

- Swapping은 Process가 Disk로 In/Out
- Demand Paging은 Page가 Disk로 In/Out

7. Page Replacement



- 빈 page가 없을 때!

=> 누군가를 밀어내고 차지해야함

- Victim을 설정하여 필요없다고 생각되는 page
를 backing store로 보냄

- 필요한 page를 backing store에서 memory로
이동

8. Page Replacement Algorithm

⇒ Victim을 “선택”하는 방식

1) FIFO(First in First out)

장점: 쉽고 직관적이다. 구현하기 쉽다

단점: 오래된 페이지가 불필요한건 아니다.(초기 변수 등)

활발하게 사용되는 것이 제거될 수 있다.

Belady의 모순: Page를 저장할 수 있는 용량을 늘렸는데도

Page Fault가 더 많이 발생하는 상황

8. Page Replacement Algorithm

2) Optimal Page Replacement

=> 앞으로 가장 오랫동안 사용되지 않을 페이지를 교체

장점: 가장 좋은 성능

단점: 미래를 예측하는 것이므로 구현하기 어렵다.

현실적으로 만들기 어려움

8. Page Replacement Algorithm

3) LRU Page Replacement(Least Replacemently Used)

=> 가장 오랫동안 사용되지 “않았던” 페이지를 교체(OPT를 근사)

성능: FIFO << LRU << OPT

4) LFU Page Replacement(Least Frequently Used)

=> 참조 횟수가 가장 적었던 페이지를 교체

=> 특정 페이지를 계속 사용하다가 다른 기능 사용하면 성능 낮음.

8. Page Replacement Algorithm

5) MFU Page Replacement(Most Frequently Used)

=> 참조 횟수가 가장 많은 페이지를 Victim으로

9. Reference

- Operating System Concepts(9th Edition. International Student Version)
- 운영체제(아주대학교 소프트웨어학과 김재훈)
- <https://velog.io/@deannn/CS-%EA%B8%B0%EC%B4%88-%EC%9A%B4%EC%98%81%EC%B2%B4%EC%A0%9C-%EB%A9%94%EB%AA%A8%EB%A6%AC-%EA%B4%80%EB%A6%AC-%EC%A0%84%EB%9E%B5>
- <https://velog.io/@codemcd/%EC%9A%B4%EC%98%81%EC%B2%B4%EC%A0%9C%OS-15.-%EA%B0%80%EC%83%81%EB%A9%94%EB%AA%A8%EB%A6%AC>