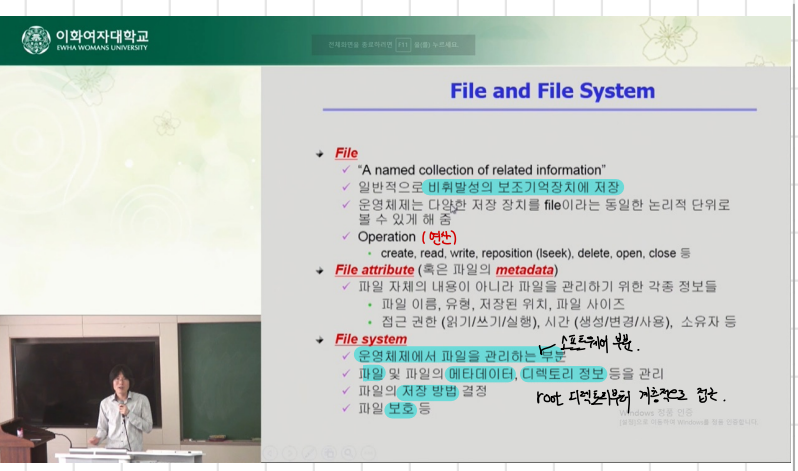


File: 이를 통해서 접근하는 단위.

저장 장치 관리하기 위해 File이란 단위로 관리

(ex:) Disk1, Disk2 ...)

devices special file

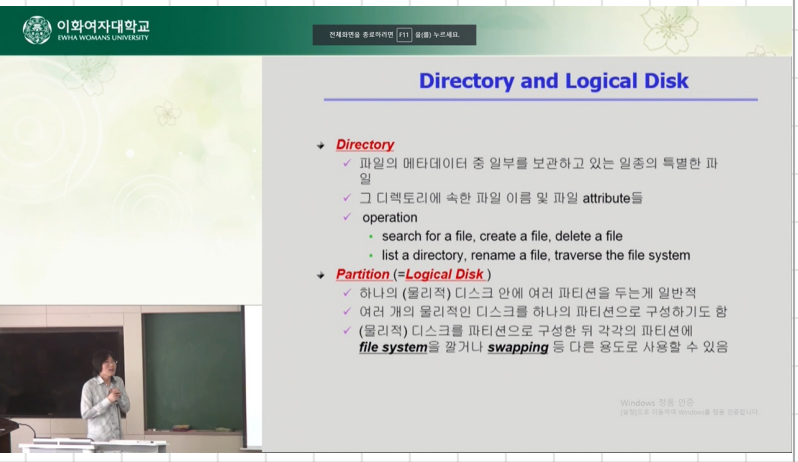
Operator

reposition. : 다른 부분부터 참조하고 싶을 때 쓰는 연산.
(보통 순차탐색)

open, close : 읽고 쓰기 위해 open, close 필요.

왜 open, close 연산 따로 있지?

open: file의 메타데이터를 올리는 것
(파일 관리에 필요한 데이터)

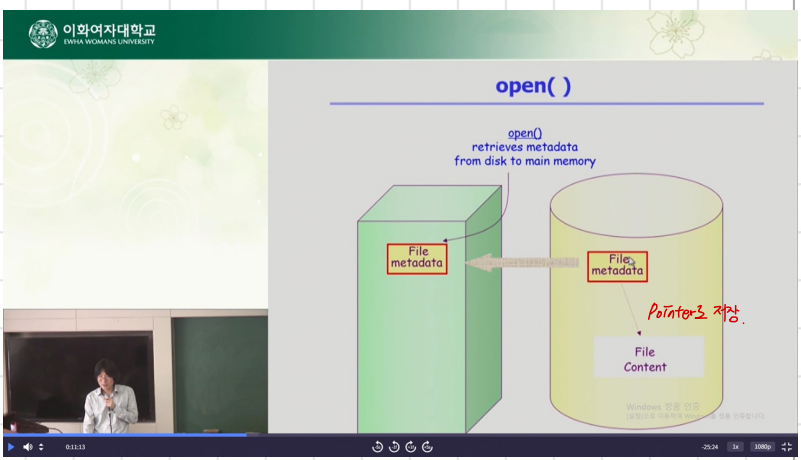


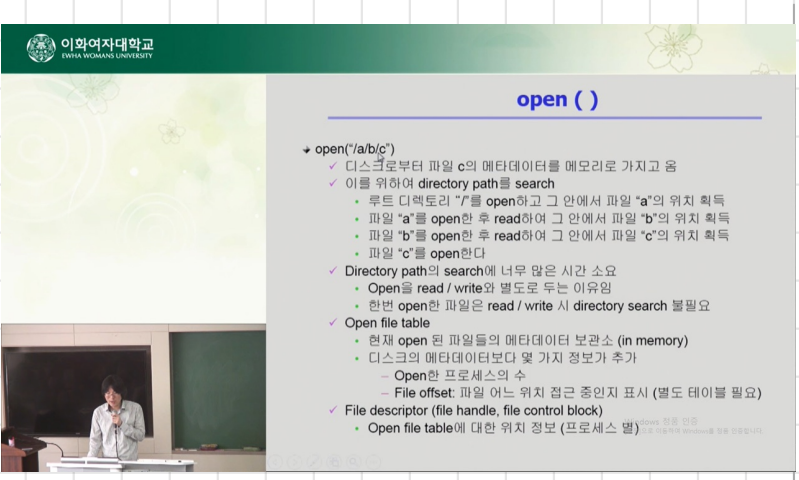
Directory file의 내용.

디렉토리 밑에 있는 파일이 어떤 건지. 메타데이터를 내용으로 하는 파일.

운영체제가 보는 디스크 = 논리적 디스크 := Partition.

- file system 용.
- swapping 용.

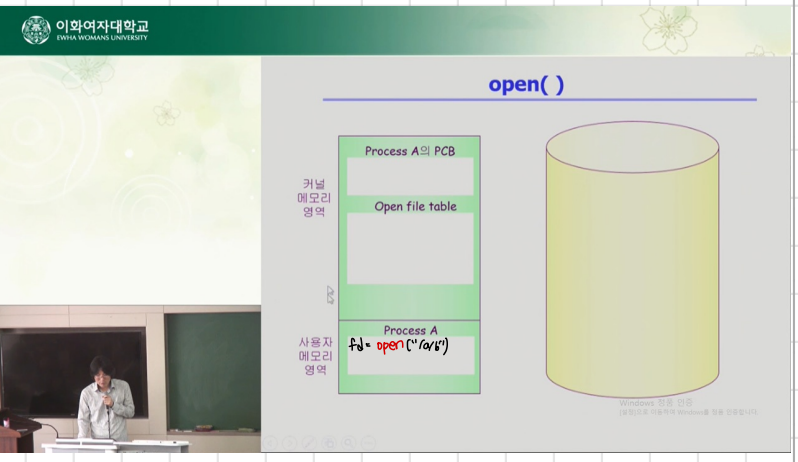




C라는 file open. ⇒ C의 메타데이터가 메모리로 올라온다.

C라는 directory 경로가 계층적으로 구성되어 있을 때 open하면 어디 저장되어 있는지 어떻게 찾을까.

root directory 위치는 알려져 있기 때문에 거기서 내려가며 C 위치 찾음.



- open 시스템 콜 (I/O)

open 과정.

1. open

2. CPU가 운영체제로 넘어감.

3. root의 메타데이터 open.

4. root의 실제 위치 찾음. (meta데이터에 저장되어 있음)

5. root의 content. (root는 디렉토리 파일이기 때문에 디렉토리 및 파일들의 메타데이터 있음)

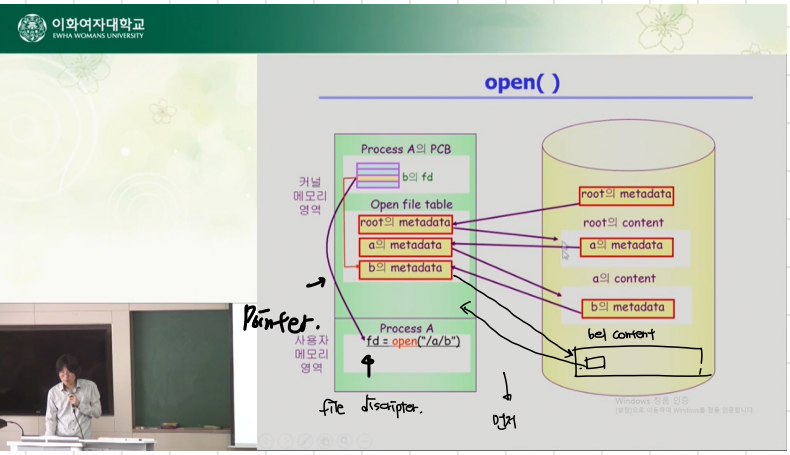
6. root 아래 a 파일의 메타데이터 얻음. (open)

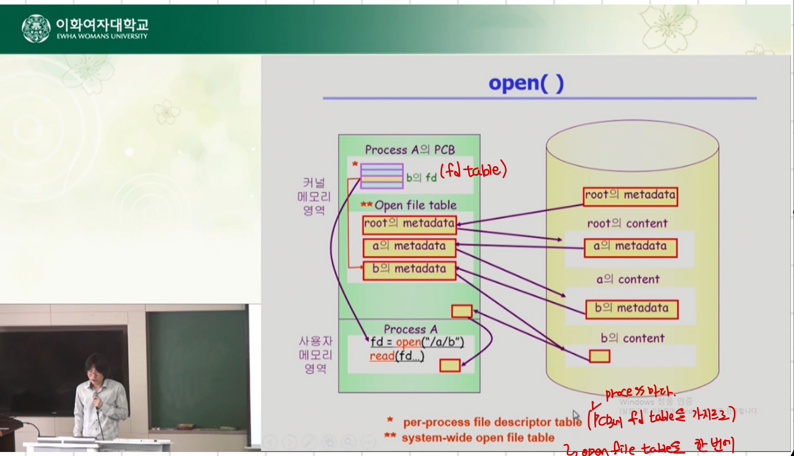
7. a의 메타데이터 중 file system 위치정보.
a도 디렉토리 파일임. → b라는 메타데이터가 있다.

8. b open.

open file table
전체 open한 프로그램이 우리가 관리해야 할 글로벌 테이블 위치.

open system call의 리턴값.
: 각 프로세스마다 open한 file들의 대한 메타데이터 포인터를 가지고 있는 일종의 배열이 정의되어 있음.
지금 open한 pointer가 배열어딘가에 만들어지고 그 배열에서 몇 번째 index냐. → file descriptor가 됨. 그 값을 사용자 프로세스에게 return하게 됨.





구현에 따라 table이 3종류 있을 수도 있음.

why? metadata가 disk에 없을 경우... 이전 내용간.

memory에 있으면 추가적으로 메타데이터 필요. 어느 위치 읽고 있는냐 알 수 있는 offset table 필요.

그래서 open file table 내 2개 두는것이 일반적

9. read(fd...) file discripter를 통해 read, write 요청가능.

10. read는 system call이니 다시 운영체제로 cpu가 넘어감.

11. 해당 fd에 대응하는 파일의 메타데이터 부분을 open file table에서 따라 간 다음, 위치 정보를 알고 있으므로 시작위치부터 읽는다. (아우 요청사항이 read.)

12. b의 내용을 사용자 프로그램에게 직접주는게 아닌, 운영체제가 자신의 메모리 공간 일부에 먼저 읽어 놓는다.

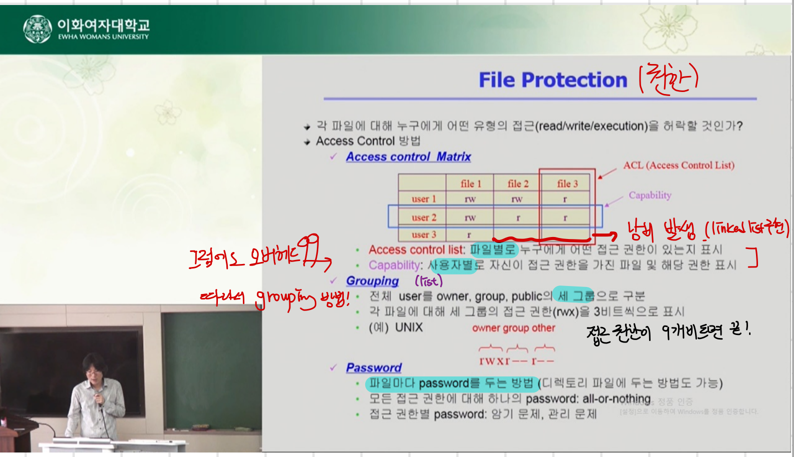
13. 그 내용을 사용자 프로그램에게 copy하여 전달. (read 작업 완료)

↳ 만약 다른 프로그램이 동일 파일의 내용을 요청하면 다시 읽을 필요없이 OS의 메모리에 있는 내용 전달.

file system의 buffer cashe.

→ 시스템콜이기에 요청 내용이 있든 없든 OS로 넘어감.

즉, OS가 모든 내용을 아므로 LRU, LFU 알고리즘 사용가능



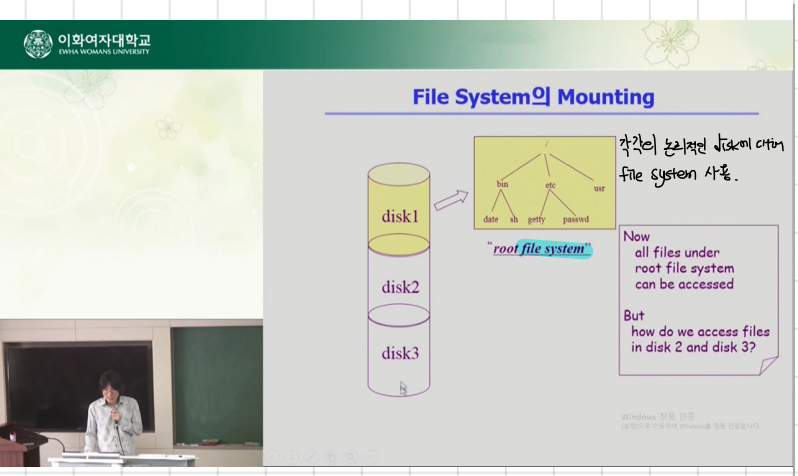
memory protection = 프로세스마다 메모리를 별도로 둔다므로 자기만 참조가능

file은 여러 사용자, 여러 프로그램이 사용가능.

따라서 1. 접근 권한이 누구에게 있는가.

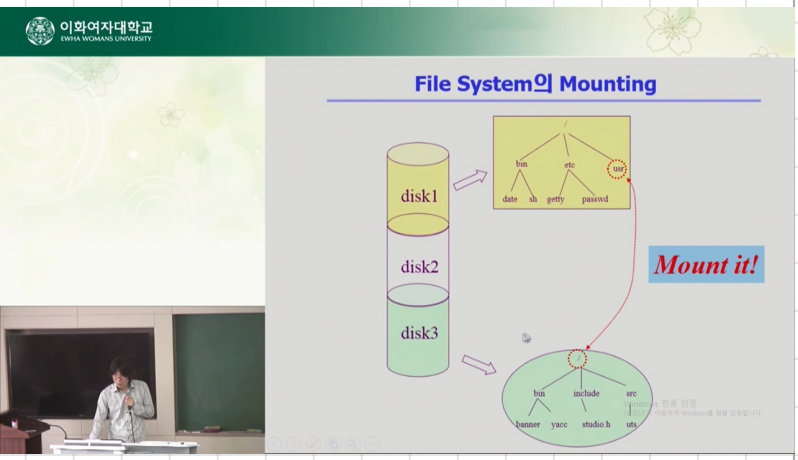
2. 접근 연산이 어떤것이 가능한가? (r,w,e)

고려.



만약 다른 partition에 설치되있는 file system에 접근해야 한다면?

Mounting 연산 제공.



Mount된 directory에 접근하게되면 또다른 file system의 root directory를 접근하는 것임.