커널 오브젝트란?

- 윈도우 운영체제에서 리소스(프로세스, 쓰레드, 파이프 등)를 관리하기 위한 데이터 블록

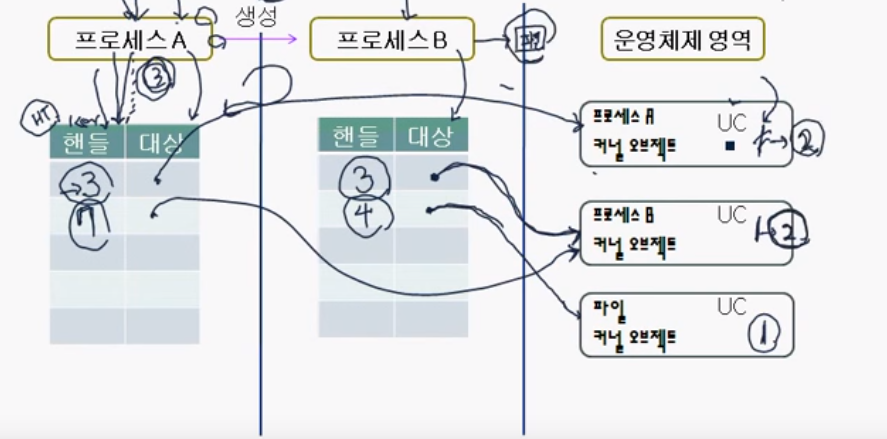
- 리소스가 생성됨과 동시에 운영체제에 의해 리소스의 정보를 담은 커널 오브젝트가 생성된다

- 또한 해당 리소스의 커널 오브젝트에 간접적으로 접근하게끔 해주는 ‘전용’ 핸들 테이블을 생성

- 리소스의 특성을 변경하기 위해서는 핸들을 이용해 해당 리소스의 커널 오브젝트에 간접적으로 접근해야만 한다.

- 커널 오브젝트는 전적으로 운영체제에 의해 생성되고 소멸한다.

- 커널 오브젝트를 참조하는 대상이 하나도 없을 때 커널 오브젝트는 운영체제에 의해 소멸



시나리오) 프로세스 A는 프로세스 B를 생성하고, 프로세스 B는 파일을 여는 작업을 수행한다

1) 프로세스 A와 프로세스 B는 ‘전용’ 핸들 테이블을 가지고 있으므로 핸들값이 같을 수도 있다.

-> 프로세스는 핸들값을 딱 하나만 가지고 있다고 볼 수 없다.

-> 프로세스의 핸들값(핸들 테이블)은 해당 ‘프로세스’에 종속적이라고 볼 수 있다.

2) 프로세스 A는 프로세스 B의 ‘부모’이므로 프로세스 B를 참조할 수 있다.

3) 모든 프로세스는 자기자신을 참조하는 ‘부모’ 프로세스가 있으므로 Usage Count = 2이다.

-> 프로세스를 생성할 때 아이콘을 더블클릭하거나 콘솔창으로 실행시키게 되므로 결과적으로 다른 프로세스에 의해서 생성이 된다고 봐야한다.(부모가 존재한다고 봐야한다)

Usage Count와의 관계

- 해당 리소스를 참조하는 대상은 Usage Count로 관리하며 이 값이 0이 되었을 때 소멸한다.

(ex. 프로세스 B를 종료한다고 해서 프로세스 B의 커널 오브젝트가 바로 사라지는 것이 아니라 프로세스 A가 참조하지 않게 됐을 때 비로소 B의 UC가 0이 되면서 커널 오브젝트가 소멸한다)

- 리소스의 생성과 동시에 자식 리소스는 핸들을 획득하므로 커널 오브젝트의 Usage Count=1

- 리소스의 생성이 완료되면 부모 또한 자식 리소스의 핸들을 획득하기 때문에 Usage Count=2

- 자식 리소스를 없애면 Usage Count=1, 부모가 자식 리소스의 소멸을 감지하면 Usage Count=0

- 파일 같은 리소스의 경우는 참조하는 프로세스가 하나밖에 없으므로 Usage Count = 1

