

운영체제 (062 분반)

ch. 03 Process Assignment

학과: 정보컴퓨터공학부

학번/이름: 201924437 김윤하

제출일: 2023. 04. 04

2.1.

System call 의 목적은, 사용자 프로그램이 운영체제의 기능을 활용할 수 있도록 하며, 하드웨어 자원을 효율적으로 관리하고 사용자 프로그램 간의 충돌을 방지하며, 시스템의 안정성을 유지하는 것입니다.

2.3.

· Unix 시스템에서, 새 process 를 시작하기 위해 command interpreter / shell 을 실행해야 하는 system call ;

- fork() : 현재 process 복제 후 새 process 생성.
- exit() : 현재 process 종료 후 return 값 부모 process 에 반환.
- wait() : 자식 process 가 종료될 때까지, 부모 process 대기.

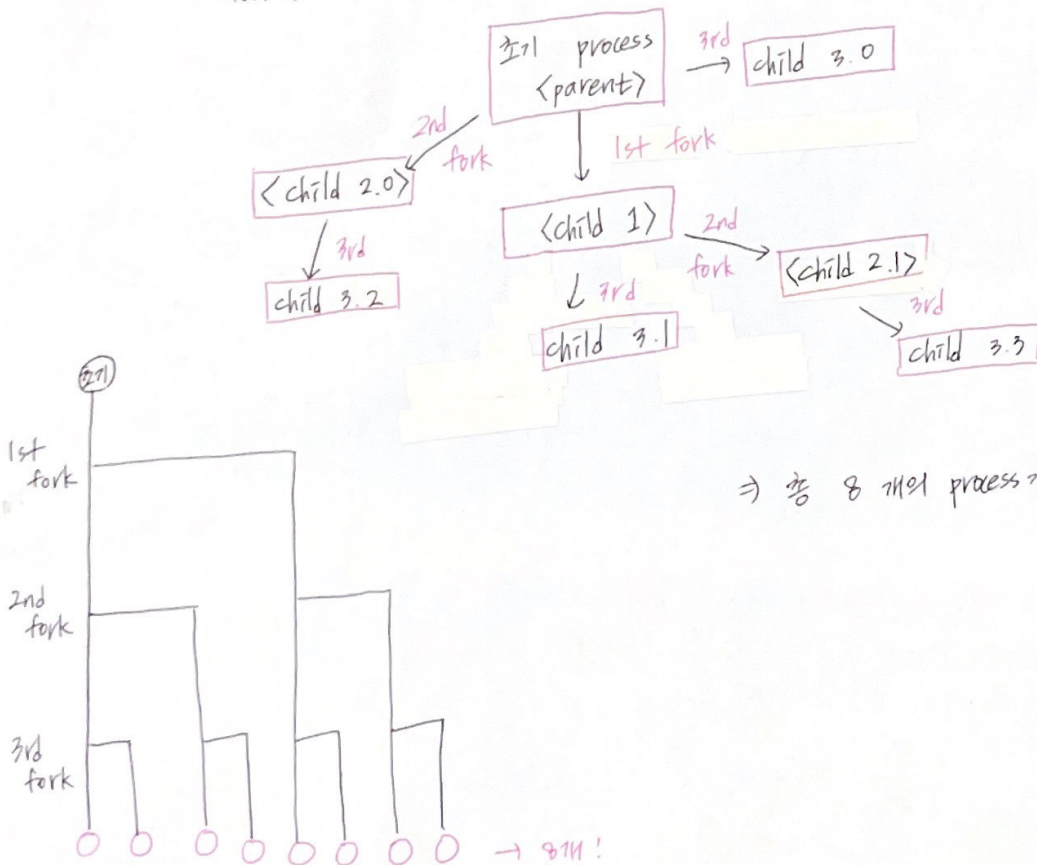
3.1.

실행 결과 \Rightarrow PARENT : value = 5

이 유 \Rightarrow 부모 process 와 자식 process 를 생성한 후,
 자식 process 의 value 값은 15 다해준다.
 부모는 자식 process 가 끝날 때까지 기다린 후에 value 출력한다.
 하지만, 자식 process 에서 value 를 변경해도, 부모 process 의
 value 는 변경되지 않으므로 value = 5 이다.
 \hookrightarrow 부모, 자식 process 가 독립적인 memory 공간을 갖기 때문이다.

3.2.

fork() \rightarrow 3 번 실행.



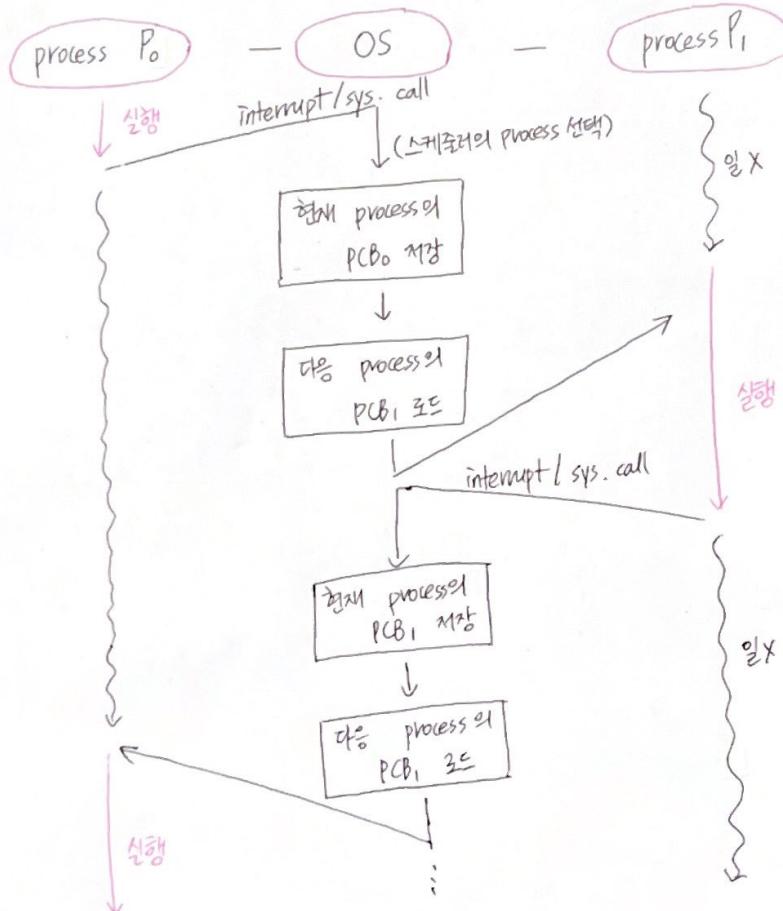
\Rightarrow 총 8 개의 process 가 생성된다.

3.5.

fork() 로 자식 process 를 생성할 때,
createProcess() 과는 다르게 부모 process로부터 주소 공간을 상속받는다.
반면, stack 과 heap은 부모-자식 process 간 공유되지 않는다.

∴ c. Shared memory segments.

3.8.



위 그림과 같이 Context - Switching 을 수행한다.

- ① interrupt / system call 발생 및 처리 → ② Scheduler에 의한 process 선택 →
- ③ 현재 process의 PCB 저장 (PCB: Process Control Block) → ④ 다음 process의 PCB 로드 →
- ⑤ 새로운 process 실행.