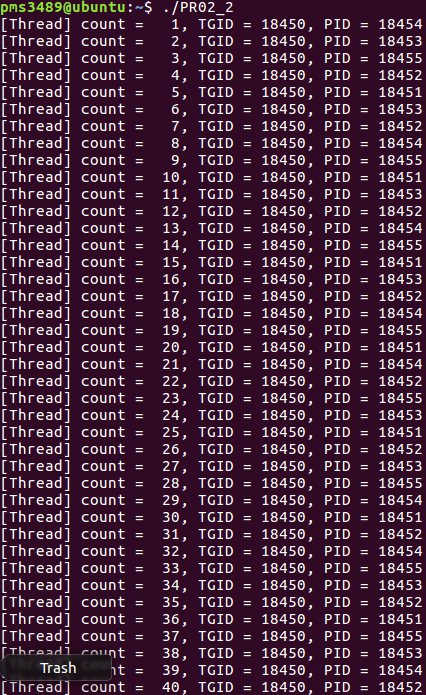
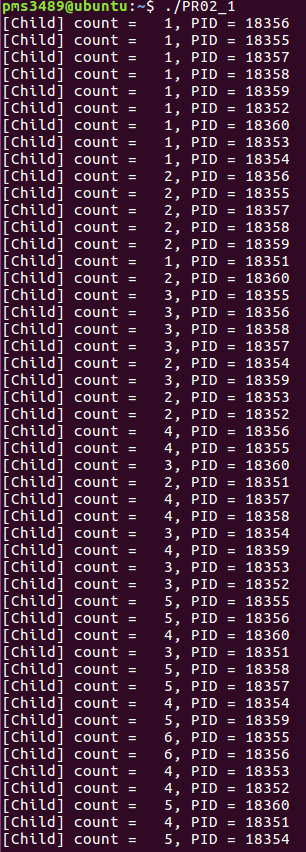
Operating System Report

2016115805 박미선



<PR02\_1 - 프로세스> <PR02\_2 - 쓰레드>

위는 프로그램을 실행시켰을 때의 모습입니다. 먼저 프로세스는 자식프로세스가 모두 동시에 실행되기 때문에 경쟁적으로 프로세스가 실행됩니다. 그래서 count가 무작위로 실행 됩니다. 하지만 쓰레드는 count를 공유하므로 차례대로 출력이 됩니다.

먼저 프로세스는 자식 프로세스를 생성하고 각 프로세스가 개별적으로 count와 PID를 출력합니다. 비록 count는 전역변수 일지라도 개별적으로 프로세스가 돌아가므로 서로 영향을 미치지 않습니다. 따라서 부모 프로세스에 있는 count 또한 영향을 받지 않으므로 초기화 된 0을 그대로 출력합니다.

그 다음 쓰레드는 다같이 data와 코드를 공유하므로 서로서로 영향을 받습니다. 코드에서 보는 바와 같이 쓰레드를 생성하고 전역 변수인 count를 증가시키면서 출력합니다. 이때 count는 계속해서 증가하고 다른 쓰레드가 count를 사용할 때도 전 쓰레드가 변경시킨 값이 그대로 들어있기 때문에 거기서 증가된 값을 출력합니다. 따라서 마지막으로 final 값을 출력할 때 500번 사용된 값이 그대로 출력이 된다.

이 때 PR02\_2 코드를 보면서 어째서 getpid()가 항상 같은지, 왜 PID값을 (int)syscall(\_\_NR\_gettid) 로 표현했는지 의문이 들었습니다. 조사해본 결과 (int)syscall(\_\_NR\_gettid)는 커널서비스를 통해서 threadID를 출력해주는 함수입니다. 이 코드는 쓰레드를 이용하므로 PID값이 쓰레드 ID여야해서 이 함수를 이용했다고 생각하였습니다. TGID는 메모리를 공유할 때 같은 값을 가지는 데 쓰레드는 이 메모리를 같이 공유하므로 같은 TGID를 가집니다. 하지만 fork는 다른 메모리를 가지므로 부모와 자식간에 TGID가 다릅니다.

프로세스와 쓰레드의 차이, 코드를 분석해보고 배웠지만 어떻게 프로그램에 적용되는지 모르고 있던 사실을 알게 되었습니다. 이렇게 실습을 통해 다양한 코드를 보고 적용시켜 보면서 수업에서 배운 것을 다시 한번 고민해 볼 수 있는 시간이 되었습니다.