

개인 과제 10: signal + shared memory

- 내용: 프로그램 1개 작성
 - `fork()`를 수행하여 자식 프로세스를 생성하는 프로그램
 - `SIGUSR1`, `SIGUSR2`에 대해 핸들러 등록. 간단히 시그널 수신 결과를 출력함
 - 키로 사용할 파일은 임의의 문자열로 작성
 - Parent process
 - 4KB의 공유 메모리 공간을 할당하고, 0으로 초기화
 - Child에게 `SIGUSR1`을 보내 공간이 할당되었음을 알림
 - `SIGUSR2`를 대기. 다른 모든 시그널은 블록
 - 공유 메모리 공간에서 초반 80 B 영역을 문자열로 출력
 - Child process
 - `SIGUSR1`을 대기. 다른 모든 시그널은 블록
 - 공유 메모리 공간을 연결하고,
 - 공유 메모리를 할당할 때 사용한 키 파일의 내용을 공유 메모리 공간에 복사 (80 B)
 - Parent에게 `SIGUSR2`를 보내어 수정이 완료되었음을 알림
- 제출 기한
 - 11/30 (월) 23:59 (지각 감점: 5%p / 12H, 1주 이후 제출 불가)
 - 동작 설명과 결과가 포함된 간단한 보고서와 소스 파일을 압축하여 LMS 제출

`getppid()` returns the process ID of the parent of the calling process. This will be either the ID of the process that created this process using `fork()`, or, if that process has already terminated, the ID of the process to which this process has been reparented (either `init(1)` or a "subreaper" process defined via the `prctl(2)` `PR_SET_CHILD_SUBREAPER` operation).

