

네트워크 프로그래밍 과제 #06

1. fork() 및 sigaction() 함수를 활용한 다중 타이머 기능 구현 (총 15점)

- 제출파일: hw06.c

- 총 4개의 sigaction(signal, &handler) 사용

■ 자식 프로세스 기능 (5점)

- 자식 프로세스는 5초 간격으로 SIGALRM 시그널 생성
- sigaction() 함수를 이용하여 SIGALRM 을 등록 후 시그널 처리
- 타임아웃 시간 및 경과된 시간(elapsed time)을 함께 출력
- 5회 반복(25초 동작) 후 종료, 리턴값: 5

■ 부모 프로세스 기능 (5점)

- 부모 프로세스는 2초 간격으로 SIGALRM 시그널을 생성
- sigaction() 함수를 이용하여 SIGALRM 을 등록 후 시그널 처리
- 타임아웃 시간 및 경과된 시간(elapsed time)을 함께 출력
- 무한 반복: sleep(1) 적용
- 자식 프로세스의 종료 SIGCHLD 처리 및 종료 후 자식 프로세스의 리턴 값 출력

■ SIGINT(Ctrl + C) 시그널 처리 기능 (5점)

- 부모 프로세스에서 SIGINT를 등록 후 시그널 처리
- 부모 프로세스가 동작 중에 SIGINT(Ctrl + C) 시그널을 입력 받으면 종료를 선택
- “Do you want to exit (y or Y to exit)? “ 출력 후 ‘y’ 또는 ‘Y’를 입력 받으면 프로그램 종료

■ 실행 결과

```
$ gcc hw06_sigaction.c -o hw06
$ ./hw06
Parent process created.
Child process created.
<Parent> time out: 2, elapsed time: 2 seconds
<Parent> time out: 2, elapsed time: 4 seconds
[Child] time out: 5, elapsed time: 5 seconds(1)
<Parent> time out: 2, elapsed time: 6 seconds
<Parent> time out: 2, elapsed time: 8 seconds
[Child] time out: 5, elapsed time: 10 seconds(2)
<Parent> time out: 2, elapsed time: 10 seconds
<Parent> time out: 2, elapsed time: 12 seconds
<Parent> time out: 2, elapsed time: 14 seconds
[Child] time out: 5, elapsed time: 15 seconds(3)
<Parent> time out: 2, elapsed time: 16 seconds
<Parent> time out: 2, elapsed time: 18 seconds
```

```
[Child] time out: 5, elapsed time: 20 seconds(4)
<Parent> time out: 2, elapsed time: 20 seconds
<Parent> time out: 2, elapsed time: 22 seconds
<Parent> time out: 2, elapsed time: 24 seconds
[Child] time out: 5, elapsed time: 25 seconds(5)
Child id: 190535, sent: 5           // 자식 프로세스 종료 상황
<Parent> time out: 2, elapsed time: 26 seconds
<Parent> time out: 2, elapsed time: 28 seconds
<Parent> time out: 2, elapsed time: 30 seconds
^CSIGINT: Do you want to exit (y or Y to exit)? y // Ctrl + C 입력 후 y 입력 -> 종료
(base) $
```