

Spring 2019 Operating Systems

Lab 08: SIGNAL

0. 과제 수행 전 유의사항

프로그래밍은 자기 손으로 작성해야 한다. 아이디어를 서로 공유하고 도움을 받는 것은 좋으나, 다른 사람의 코드를 그대로 가져다 쓰는 것은 자신에게도 손해이고 다른 사람에게 피해를 끼치는 결과를 낳게되므로 허용하지 않는다. 과제 체크 시 두 사람의 코드가 유사하면, 두 사람 모두에게 과제 점수를 부여할 수 없으니 유의바란다.

1. 목표

블록화된 시그널을 가져오는 방법에 대하여 예제를 통해 학습한다.

2. 과제

[lab8_a.c]

```
#include <stdio.h>
#include <signal.h>
#include <unistd.h>

void signal_handler(int signo);

main()
{
    sigset_t set, oldset;
    struct sigaction act;

    act.sa_handler = signal_handler;
    sigemptyset(&act.sa_mask);
    act.sa_flags = 0;

    sigaction(SIGINT, &act, NULL);

    sigemptyset(&set);
    sigaddset(&set, SIGINT);

    sigprocmask(SIG_BLOCK, &set, &oldset);
    sleep(10);

    sigprocmask(SIG_SETMASK, &oldset, NULL);

    pause();
    exit(0);
}

void signal_handler(int signo)
{
    printf("I'm your signal %d\n", signo);
}
```

위 코드의 동작은 3가지의 경우로 나눌 수 있다. SIGINT(<Ctrl+C>)의 호출이 언제되느냐에 따라서 그 동작이 바뀐다. 3가지가 무엇인지 찾아보시오.

힌트: sigprocmask() 함수의 위치

[lab8_b.c]

```
#include <stdio.h>
#include <signal.h>
#include <unistd.h>

void signal_handler(int signo);

main()
{
    sigset_t set, oldset;
    struct sigaction act;

    act.sa_handler = signal_handler;
    sigemptyset(&act.sa_mask);
    act.sa_flags = 0;

    sigaction(SIGINT, &act, NULL);

    sigemptyset(&set);
    sigaddset(&set, SIGINT);

    sigprocmask(SIG_BLOCK, &set, &oldset);
    sleep(10);

    sigsuspend(&oldset);
    exit(0);
}

void signal_handler(int signo)
{
    printf("I'm your signal %d\n", signo);
}
```

이 코드는 앞서 작성한 코드의 문제를 개선한 것이다. 즉, 한 번의 SIGINT (<Ctrl+C>)로 프로그램이 종료될 수 있는데, 그 이유는 무엇이었는가?

3. 제출

2번에서 언급하는 2가지의 문제에 대해 서술하고 코드에 주석을 붙여 제출한다.