

## UNIX 프로그래밍 실습

(2019/12/05)

1. Parent process는 표준 입력으로 정수를 하나 입력 받아, "data" 파일에 쓴 후, 세 개의 child process들을 만듭니다. 각 child process는 "data" 파일의 정수 값을 읽고 5초간 기다렸다 +10 한 값을 씁니다. 세 child의 덧셈이 정확히 되도록 file locking을 써서 동기화를 합니다.

```
void do_child(int fd) {
    int in;
    struct flock lock;

    // file locking

    lseek(fd, 0, SEEK_SET);
    read(fd, &in, sizeof(int));
    sleep(5);

    in=in+10;
    lseek(fd, 0, SEEK_SET);
    write(fd, &in, sizeof(int));

    // file unlocking

    exit(0);
}

int main(void){
    int i, in, fd;
    pid_t pid;

    fd=open("data1", O_RDWR|O_CREAT, 0600);
    scanf("%d", &in);
    write(fd, &in, sizeof(int));

    for (i=0; i<3; i++){
        pid=fork();
        if (pid==0)
            do_child(fd);
    }

    for (i=0; i<3; i++)
        wait(0);

    lseek(fd, 0, SEEK_SET);
    read(fd, &in, sizeof(int));
    printf("%dWn", in);

    exit(0);
}
```

2. 네 개의 프로세스가 동기화를 하며 자신의 프로세스 id를 5회 출력하는 프로그램을 작성 합니다. 이 프로그램은 "turn1" file을 이용하여 동기화에 참여하는 전체 프로세스 중 자신의 출력 순서를 결정 합니다. 둘 이상의 프로세스가 동시에 자신의 id를 출력하지 않도록 하는 동기화 작업은 file locking을 사용합니다. 네 프로세스의 출력 순서가 정해져 있지 않습니다. 동시에 출력을 하지 않도록만 하면 됩니다.