

## 2018 시스템 프로그래밍

- lab06 -

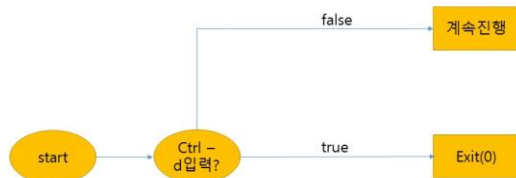
제출일자	2018.11.06
분 반	01
이 름	함지희
학 번	201702087

## Trace 00

```
b201702087@2018-sp: ~/shlab-handout
b201702087@2018-sp:~/shlab-handout$ ./sdriver -V -t 00 -s ./tshref
Running trace00.txt...
Success: The test and reference outputs for trace00.txt matched!
Test output:
#
# trace00.txt - Properly terminate on EOF.
#
Reference output:
#
# trace00.txt - Properly terminate on EOF.
#
```

```
b201702087@2018-sp: ~/shlab-handout
b201702087@2018-sp:~/shlab-handout$ ./sdriver -V -t 00 -s ./tsh
Running trace00.txt...
Success: The test and reference outputs for trace00.txt matched!
Test output:
#
# trace00.txt - Properly terminate on EOF.
#
Reference output:
#
# trace00.txt - Properly terminate on EOF.
#
```

## 각 trace 별 프로우 차트



## Trace 해결 방법 설명

```
if (feof(stdin)) { /* End of file (ctrl-d) */
    fflush(stdout);
    fflush(stderr);
    exit(0);
}
```

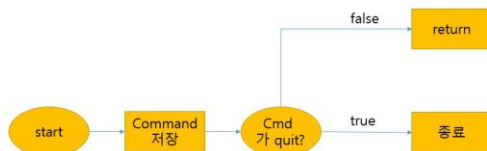
ctrl - d를 입력받으면 if 안에 들어가서 프로그램을 종료한다.

## Trace 01

```
b201702087@2018-sp: ~/shlab-handout
b201702087@2018-sp:~/shlab-handout$ ./sdriver -V -t 01 -s ./tshref
Running trace01.txt...
Success: The test and reference outputs for trace01.txt matched!
Test output:
#
# trace01.txt - Process builtin quit command.
#
Reference output:
#
# trace01.txt - Process builtin quit command.
#
```

```
b201702087@2018-sp: ~/shlab-handout
b201702087@2018-sp:~/shlab-handout$ ./sdriver -V -t 01 -s ./tsh
Running trace01.txt...
Success: The test and reference outputs for trace01.txt matched!
Test output:
#
# trace01.txt - Process builtin quit command.
#
Reference output:
#
# trace01.txt - Process builtin quit command.
#
```

## 각 trace 별 프로우 차트



## Trace 해결 방법 설명

```
void eval(char *cmdline)
{
    char *argv[MAXARGS];
    pid_t pid;

    parseline(cmdline, argv);

    builtin_cmd(argv);
}

int builtin_cmd(char **argv)
{
    char *cmd = argv[0];

    if(!strcmp(cmd, "quit")){
        exit(0);
    }

    return 0;
}
```

입력받은 command를 parseline으로 분리하고 그렇게 파싱된 명령어를 Builtin\_cmd()에 전달한다.  
builtin\_cmd함수에서 전달받은 명령어를 quit와 비교하여 같으면 종료하게 한다

## Trace 02

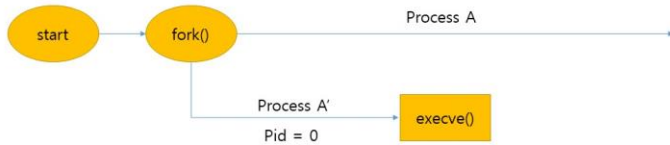
```
b201702087@2018-sp: ~/shlab-handout
b201702087@2018-sp:~/shlab-handout$ ./sdriver -V -t 02 -s ./tshref
Running trace02.txt...
Success: The test and reference outputs for trace02.txt matched!
Test output:
#
# trace02.txt - Run a foreground job that prints an environment variable
#
# IMPORTANT: You must pass this trace before attempting any later
# traces. In order to synchronize with your child jobs, the driver
# relies on your shell properly setting the environment.
OSTYPE=/home/sys01/b201702087/bin:/home/sys01/b201702087/.local/bin:/usr/local/s
bin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games

Reference output:
#
# trace02.txt - Run a foreground job that prints an environment variable
#
# IMPORTANT: You must pass this trace before attempting any later
# traces. In order to synchronize with your child jobs, the driver
# relies on your shell properly setting the environment.
OSTYPE=/home/sys01/b201702087/bin:/home/sys01/b201702087/.local/bin:/usr/local/s
bin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
```

```
b201702087@2018-sp: ~/shlab-handout
b201702087@2018-sp:~/shlab-handout$ ./sdriver -V -t 02 -s ./tsh
Running trace02.txt...
Success: The test and reference outputs for trace02.txt matched!
Test output:
#
# trace02.txt - Run a foreground job that prints an environment variable
#
# IMPORTANT: You must pass this trace before attempting any later
# traces. In order to synchronize with your child jobs, the driver
# relies on your shell properly setting the environment.
OSTYPE=/home/sys01/b201702087/bin:/home/sys01/b201702087/.local/bin:/usr/local/s
bin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games

Reference output:
#
# trace02.txt - Run a foreground job that prints an environment variable
#
# IMPORTANT: You must pass this trace before attempting any later
# traces. In order to synchronize with your child jobs, the driver
# relies on your shell properly setting the environment.
OSTYPE=/home/sys01/b201702087/bin:/home/sys01/b201702087/.local/bin:/usr/local/s
bin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
```

## 각 trace 별 프로우 차트



## Trace 해결 방법 설명

```
if(!builtin_cmd(argv)) {
    if((pid=fork()) == 0) {
        if((execve(argv[0],argv,envirom)<0)) {
            printf("%s : Command not found\n",argv);
            exit(0);
        }
    }
}
```

fork함수로 자식프로세스를 생성하면 그 자식프로세스의 pid는 0이 된다. 이 코드는 pid가 0인지 확인해서 0이면(자식 프로세스임) execve함수를 실행하는 코드이다.

execve함수는 현재 프로세스를 execve함수로 호출한 프로세스로 교체해준다.