

# Project

---

시스템소프트웨어실습

## □ Shell

- 운영체제 상에서 다양한 운영체제 기능과 서비스를 구현하는 인터페이스를 제공하는 프로그램
  - 사용자로부터 입력받은 커맨드라인을 해석하는 프로세스
- 표준 입력으로부터 커맨드라인을 읽고, 입력된 라인을 해석하여 해당하는 명령을 실행

## □ Shell 기본 기능 및 ls 함수 구현하기

- 사용자로부터 입력받은 명령어 실행하기 (총 3점)
  - 사용자로부터 명령어 입력받기 (1점)
  - 입력받은 명령어 실행하기 (1.5점)
    - 입력받은 명령어 실행 후 정상적으로 Shell로 복귀
  - 'q' 명령어를 입력받으면 Shell 종료하기 (0.5점)
- ls 함수 구현하기 (총 1점)
  - 현재 경로의 파일 및 디렉토리 목록 출력 (1점)

```
User Shell >> ps
  PID TTY          TIME CMD
 2534 pts/11        00:00:00 bash
 3147 pts/11        00:00:00 sh
 3148 pts/11        00:00:00 ps
User Shell >> cat file1
Hello World
User Shell >>
```

명령어 실행 화면

```
User Shell >> ls
A C Dir1 file1 sh
User Shell >> q
rilyaus@ubuntu:~/Project/test$
```

ls 함수 및 Shell 종료 화면

## □ 자동완성(Tab 키) 기능 구현하기

### ■ 탭(Tab)키를 이용한 자동완성 기능 구현하기 (총 2점)

#### □ 파일명 일부 입력 후 탭키를 한 번 눌렀을 경우 파일의 이름 자동완성하기 (2점)

- 비슷한 이름의 파일이 여러 개일 경우 공통된 부분까지만 출력
- Tip : 명령 프롬프트 내용 및 탭 키 입력은 getch() 함수를 이용

### ■ 탭(Tab)키를 이용한 파일 목록 출력 기능 구현하기 (총 2점)

#### □ 지정한 경로에서 탭키를 두 번 눌렀을 경우 파일의 목록 출력하기 (2점)

```
User Shell >> ls  
A C Dir1 sh  
User Shell >> ls ./D
```



```
User Shell >> ls  
A C Dir1 sh  
User Shell >> ls ./Dir1
```

```
User Shell >> ls  
A C Dir1 sh  
User Shell >> ls ./
```



```
User Shell >> ls  
C Dir1 sh  
User Shell >> ls ./  
C Dir1 sh  
User Shell >> ls ./
```

탭 1번을 이용한 자동완성

탭 2번을 이용한 목록 출력

## □ 리다이렉션 기능 구현하기

### ■ 표준 입력(<) 리다이렉션 구현하기 (총 1.5점)

□ '<' 기호와 명령어 분리하기 (0.5점)

□ '<' 기호 뒤에 입력받은 파일로 표준 입력 대체 하기 (1점)

### ■ 표준 출력(>) 리다이렉션 구현하기 (총 1.5점)

□ '>' 기호와 명령어 분리하기 (0.5점)

□ '>' 기호 뒤에 입력받은 파일로 표준 출력 대체 하기 (1점)

```
User Shell >> cat file1
C
D
A
B
User Shell >> sort < file1
A
B
C
D
User Shell >>
```

표준 입력 리다이렉션

```
User Shell >> cat file2
Hello World
User Shell >> ls -l > file2
User Shell >> cat file2
total 24
-rw----- 1 rilyaus rilyaus    8 Nov 22 00:52 file1
-rw----- 1 rilyaus rilyaus    0 Nov 22 00:54 file2
-rwxrwxr-x 1 rilyaus rilyaus 18984 Nov 22 00:12 sh
User Shell >>
```

표준 출력 리다이렉션

## □ 파이프라인( | ) 기능 구현하기

- 파이프를 이용하여 둘 이상의 명령어 함께 실행하기 (총 4점)
  - '|' 기호 앞의 명령 결과를 '|' 기호 뒤의 명령의 입력으로 전달하기 (1점)
  - '|' 기호와 리다이렉션(<, >) 기호 함께 사용하기 (1점)
  - 파이프 사용 개수를 임의로 할 수 있도록 구현하기 (2점)
    - Tip : 재귀 함수 사용

```
User Shell >> ps
  PID TTY          TIME CMD
 2490 pts/1        00:00:00 bash
 4698 pts/1        00:00:00 sh
 4701 pts/1        00:00:00 ps
User Shell >> ps | sort
 2490 pts/1        00:00:00 bash
 4698 pts/1        00:00:00 sh
 4702 pts/1        00:00:00 ps
 4703 pts/1        00:00:00 sort
  PID TTY          TIME CMD
User Shell >>
```

파이프라인 사용

```
User Shell >> ps | sort
 2490 pts/1        00:00:00 bash
 4722 pts/1        00:00:00 sh
 4724 pts/1        00:00:00 ps
 4725 pts/1        00:00:00 sort
  PID TTY          TIME CMD
User Shell >> ps | sort > file1
User Shell >> cat file1
 2490 pts/1        00:00:00 bash
 4722 pts/1        00:00:00 sh
 4726 pts/1        00:00:00 ps
 4727 pts/1        00:00:00 sort
  PID TTY          TIME CMD
User Shell >>
```

파이프라인, 리다이렉션 사용

- 백그라운드 실행 및 시그널 처리 기능 구현하기
  - ‘&’ 기호를 이용하여 백그라운드로 명령 실행하기 (총 3점)
    - ‘&’ 기호와 명령어 분리하기 (0.5점)
    - 입력받은 명령어 백그라운드로 실행하기 (1점)
    - 파이프를 이용한 명령을 백그라운드로 실행하기 (1.5점)
  - ‘SIGINT’, ‘SIGQUIT’ 시그널 무시하도록 구현하기 (총 2점)
    - 시그널 함수를 ‘SIGINT’, ‘SIGQUIT’ 동작 재정의하기 (2점)
      - ex) SIGINT(Ctrl + C) 시그널로 Shell 종료를 방지하기

```
User Shell >> sleep 5 &  
User Shell >> ps  
  PID TTY          TIME CMD  
 2490 pts/1    00:00:00 bash  
 4734 pts/1    00:00:00 sh  
 4779 pts/1    00:00:00 sleep  
 4780 pts/1    00:00:00 ps  
User Shell >>
```

명령어 백그라운드 실행

```
User Shell >> ^C  
User Shell >> ^C  
User Shell >> ^C  
User Shell >>
```

시그널 재정의로 Shell 종료 방지

- ❑ System("clear") 를 제외한 시스템 함수 사용 금지
- ❑ Tip : getch( ) 함수
  - 해당 내용을 작성하여 함수로 사용하거나 "getch.h" 파일로 추가해 사용

```
#include <stdio.h>
#include <termio.h>
int getch(void){
    int ch;
    struct termios buf, save;
    tcgetattr(0,&save);
    buf = save;
    buf.c_lflag &= ~(ICANON|ECHO);
    buf.c_cc[VMIN] = 1;
    buf.c_cc[VTIME] = 0;
    tcsetattr(0, TCSAFLUSH, &buf);
    ch = getchar();
    tcsetattr(0, TCSAFLUSH, &save);
    return ch;
}
```



## □ 과제 제출 기한

■ 17. 12. 20(수) 23:59

## □ 제출 양식

■ 프로젝트 전체 파일과 보고서가 포함된 압축 파일

■ 보고서에 들어가야 하는 것

□ 코드에 관한 설명

■ 본인이 받을 수 있는 점수(구현한 내용)를 정확히 명시할 것

■ 정확히 명시하지 않았을 경우 점수가 누락 될 수 있음

□ 고찰

■ 프로젝트를 진행하며 느낀점, 의견 등을 작성

## □ 제출 시 유의사항

■ 압축 파일 명

□ 예시 ) [프로젝트]2015726000\_홍길동.zip