2018603020_김희찬

```bash
#!/bin/bash

while :
do
    echo "----------------------------------------"
    read -p "<<이름과 점수를 입력하세요>> : " name_score
    echo "----------------------------------------"
    if [ "$name_score" = "exit" ] || [ "$name_socre" = "EXIT" ] || [ "$name_score" = "Exit" ]
    then
        echo -e "*** 프로그램을 종료합니다.\n"
        break
    fi

    name=$(echo $name_score | cut -d ' ' -f 1)
    score=$(echo $name_score | cut -d ' ' -f 2)


    if [ "$score" -ge 90 ] && [ "$score" -le 100 ]
    then
        grade="A"
    elif [ "$score" -ge 80 ] && [ "$score" -lt 90 ]
    then
        grade="B"
    elif [ "$score" -ge 70 ] && [ "$score" -lt 80 ]
    then
        grade="C"
    elif [ "$score" -ge 60 ] && [ "$score" -lt 70 ]
    then
        grade="D"
    else
        grade="F"
    fi

    echo "*** 학생 $name은(는) $score점을 맞아 $grade학점을 취득하였습니다."
done
```

```
mjswindells@ubuntu:~$ chmod 755 assignment5.sh
mjswindells@ubuntu:~$ ./assignment5.sh
----------------------------------------
<<이름과 점수를 입력하세요>> : 김희찬 86
----------------------------------------
*** 학생 김희찬은(는) 86점을 맞아 B학점을 취득하였습니다.
----------------------------------------
<<이름과 점수를 입력하세요>> : 강민호 66
----------------------------------------
*** 학생 강민호은(는) 66점을 맞아 D학점을 취득하였습니다.
----------------------------------------
<<이름과 점수를 입력하세요>> : exit
----------------------------------------
*** 프로그램을 종료합니다.

mjswindells@ubuntu:~$
```

Step 1.

Line 6 : name_score이란 변수에 read 함수를 이용하여 이름과 점수를 한꺼번에 입력 받았다.

Line 8~12 :

If [조건] ; then ; 내용 ; fi 형식으로 조건문을 만들 수 있다.

name_score이 exit, EXIT, Exit 셋 중 하나라면 그 즉시 whille문을 빠져나올 수 있게 설정하였다. 즉, 프로그램을 종료한다.

Step 2.

Line 14~15 :

name_score에 입력받은 이름과 점수를 나누는 작업이 필요하였다.

cut -d '구분자' -f 1 : 구분자에 공백이 하나 있으므로 띄어쓰기를 기준으로 자른다.

-d 의 의미는 구분자를 의미한다.

-f 1로 첫번째 필드 즉, 공백을 기준으로 처음 나온 문자열을 name에 저장하고

두번째 나온 문자열을 score에 저장한다.

Line 18~32 :

score에 정수가 저장되었으므로 비교연산을 통해 grade를 초기화해준다.

Step 3.

Line 34 : $name , $score , $grade 변수를 통해 설정한 값을 출력하도록 하였다

이때 앞에 $(달러사인)을 붙이는 이유는 저 변수의 값을 사용하기 위함이다

만약 $을 붙이지 않았을 경우 name이 그대로 출력됨을 알 수 있다.

Step 4.

Line 3,4,35 :

무한루프를 돌리기 위해 while : 를 사용하였다.

while [조건] ; do ; 실행문장 ; done 형식으로 반복문을 설정하면 된다.

따라서 Step1에서 작성한 조건문을 만나기 전까지는 무한루프로 프로그램이 돌아감을 확인할 수 있다.

Step 5.

C언어로 작성된 프로그램은 컴파일하여 기계어로 변환된 목적파일로 만들어준 뒤 링크과정을 거치고 실행 퍼미션을 주어야만 실행이 가능하지만 쉘프로그래밍으로 작성된 파일은 이러한 과정이 필요 없고 실행 퍼미션만 주면 된다.

쉘프로그래밍은 간단한 프로그램을 빠른 시간 안에 개발해야 할 때 편리하다.

쉘프로그래밍을 배우면 전체 동작 상태를 점검해 볼 수 있기 때문에 전체 구조상의 중요한 결함을 발견하기 용이하다.

따라서 전체 프로젝트의 완성도가 높아질 뿐 아니라 리소스 낭비를 대폭 줄일 수 있다.