

OSS prj-1

- 메뉴 출력

```
#!/bin/bash

echo "-----"
echo "User Name: yougwangjun"
echo "Student Number: 12201511"
echo "[ MENU ]"
echo "1. Get the data of the movie identified by a specific 'movie id' from 'u.item'
2. Get the data of action genre movies from 'u.item'
3. Get the average 'rating' of the movie identified by specific 'movie id' from 'u.data'
4. Delete the 'IMDb URL' from 'u.item'
5. Get the data about users from 'u.user'
6. Modify the format of 'release date' in 'u.item'
7. Get the data of movies rated by a specific 'user id' from 'u.data'
8. Get the average 'rating' of movies rated by users with 'age' between 20 and 29 and 'occupation' as 'programmer'
9. Exit"
echo "-----"
```

- Bash shell 에서 실행되어야 함을 지정하는 shebang을 지정하고 메뉴 포맷에 맞게 echo

- 반복문, 선택문 설정

```
while True
do
    read -p "Enter your choice [ 1-9 ] " choice;
    case $choice in
        1)~~~ ::
        2)~~~ ::
        .
        .
        .
        9) echo "Bye!"; break ;;
        *) echo "bad choice" ;;
    esac
done
```

- while문으로 계속해서 실행 - 9번이 선택되면 break으로 탈출(종료)
- choice를 입력받고 case문으로 choice가 1~9일때의 케이스를 나눔
- *) 으로 범위 이외의 값이 입력될 경우 예외처리

1. movie id에 해당하는 행 불러오기

```
echo
read -p "Please enter 'movie id' (1~1682): " movie_id; echo
cat $1 | awk -F\| -v id=$movie_id '$1==id {print}'; echo
```

- 한 행을 띄어주기 위해서 echo 사용
- \$1로 u.item 사용 (Positional Parameters)
- 파이프라인을 이용해 값 넘기기
- -F 옵션을 통해 | 를 구분자로 사용
- -v 옵션을 통해 id에 입력받은 movie_id를 대입
- u.item의 1열과 id값이 같은 행 전체 출력

2. 'action' 장르 영화 출력

```
echo
read -p "Do you want to get the data of 'action' genre movies from 'u.item'? (y/n): " is_on; echo
if [ $is_on == 'y' ]; then
    cat $1 | awk -F\| '$7==1 {print $1, $2}' | head -n 10
fi; echo
```

- if문으로 입력받은 is_on이 'y'일때 코드 실행
- \$7(장르 필드)가 1이면 해당 1열(movie id)과 2열(name) 출력
- head 명령어로 상위 10개 출력

3. movie id에 해당하는 영화의 점수 내기

```
echo
read -p "Please enter 'movie id' (1~1682): " movie_id; echo
echo -n "average rating of $movie_id: "
cat $2 | awk -v id=$movie_id '$2==id {sum+=$3; count+=1} END{print sum/count}'; echo
```

- \$2로 u.data 사용 (Positional Parameters)
- u.data의 \$2(movie id)가 입력값과 일치하는 행의 \$3(score)를 계산
- {sum+=\$3; count+=1} 으로 누적합과 몫수 계산
- END{print sum/count}를 통해 마지막일때 최종적으로 평균계산
- 자동으로 6자리 round 처리

4. 'IMDb URL' 필드 지우기

```
echo
read -p "Do you want to delete the 'IMDb URL' from 'u.item'? (y/n): " is_on; echo
if [ $is_on == 'y' ]; then
    cat $1 | head -n 10 | sed -E 's/http[^\]]*\)/g'; echo
fi; echo
```

- u.item 불러오고 head로 상위 10개 레코드 먼저 자르기 (메모리 절약)
- sed와 정규표현식을 사용
- s/http[^\]]*\): 'http'로 시작하여 1개이상 나오고 ')'로 끝나는 형식을
- // : 공백으로 만들고 이를
- g : 모든 영역에서 실행

5. 형식에 맞게 u.user 출력하기

```
echo
read -p "Do you want to get the data about users from 'u.user'? (y/n): " is_on; echo
if [ $is_on == 'y' ]; then
    cat $3 | head -n 10 | sed -E 's/^/user /
                                s/|/ is /
                                s/|/ years old /
                                s/M/male/; s/F/female/
                                s/|/ /
                                s/|.*//'; echo
fi; echo
```

```
s/^user /      : 첫번째 공간에 'user' 삽입
s/ / is /      : 다음에 나오는 ' ' 을 'is' 로 대체
s/ / years old / : 그 다음에 나오는 ' ' 을 'years old' 로 대체
s/M/male/; s/F/female/ : 'M'은 'male'로 'F'는 'female'로 대
s/ / /         : ' ' 을 ' ' 로 대체
s/./.*//       : '.'이후로 나오는 것들을 제거
```

6. u.item의 날짜 형식 바꾸기

```
echo
read -p "Do you want to Modify the format of 'release data' in 'u.item'? (y/n): " is_on; echo
if [ $is_on == 'y' ]; then
    cat $1 | tail -n 10 | sed -E 's/Jan/01/; s/Feb/02/; s/Mar/03/; s/Apr/04/; s/May/05/; s/Jun/06/; s/Jul/07/; s/Aug/08/; s/Sep/09/; s/
    s/([0-9]{2})-([0-9]{2})-([0-9]{4})/\3\2\1/'; echo
fi; echo
```

```
-tail 명령어를 통해 마지막 10개 레코드 먼저 자르기
s/Jan/01/ ~ s/Dec/12/ : 각 월에 해당하는 값 바꾸기
s/([0-9]{2})-([0-9]{2})-([0-9]{4}) : 숫자가 2개(일수)-2개(월수)-4개(연도)의 형식 지정
/3\2\1 : 순서를 연도-월수-일수 순으로 변경
```

7. 특정 유저가 평가한 영화들 출력

```
echo
read -p "Please enter the 'user id'(1-943): " user_id; echo
filter=$(cat $2 | awk -v id=$user_id ' $1==id {print $2}' | sort -n)
echo $filter | tr ' ' '\n'; echo

for movie_id in $(echo "$filter" | head -n 10); do
    cat $1 | awk -F\| -v id=$movie_id ' $1==id {print $1 "|" $2}'
done; echo
```

- \$2(u.data)에서 user id와 일치하는 행의 movie id를 찾고 sort를 통해 정렬한뒤 filter변수에 저장
- 행마다 적용하는 것이 아닌 전체적인 형식에서 ' ' 을 '|'으로 바꾸기 위해 tr명령어 사용
- filter의 상위 10개 movie id에 대해서 반복문 실행
- \$1(u.item)에서 각 movie id 와 일치하는 행의 1,2열 출력 (10번 반복)

8. 다수의 유저가 평가한 영화의 점수 내기

```
echo
read -p "Do you want to get the average 'rating' of movies rated by users with 'age' between 20 and 29 and 'occupation' as 'programmer"
if [ $is_on == 'y' ]; then
    users=$(cat $3 | awk -F\| ' $2>=20 && $2<=29 && $4=="programmer" {print $1} ')

    arr=()
    for user in $(echo "$users"); do
        arr+=$(cat $2 | awk -v id=$user ' $1==id {print $2 "|" $3} ')
        arr+=${'\n'}
    done;
    echo "$arr" | awk -F\| '{sum[$1]+=$2; count[$1]++;}END{for (ind in sum) if (ind > 0) {print ind, sum[ind]/count[ind]}}'
fi; echo
```

- \$3(u.user) 에서 특정 범위의 \$2(나이)와 \$4(직업)에 해당하는 \$1(유저)를 users변수에 저장
- for 문을 통해 각 user에 대해 movie id 와 score를 arr 리스트에 저장
- arr에 입력할 때 다음 입력은 다음 행에서 이루어져야 하므로 입력마다 개행문자 삽입
- arr저장된 movie id 와 score가 해당되는 sum, count 배열에 지속적으로 누적합, 몫수 계산
- END일 때 for문을 이용하여 인덱스별로 ind(movie id) 와 sum[ind]/count[ind] (평점) 계산
- 개행문자행을 제거하기 위해 if (ind > 0) 사용