**OSS Project Readme**

**12223833 하수영**

기초구조)

기초적인 구조는 스크립트를 처음 실행했을 때, 사용자에게 어떠한 동작을 수행할 수 있는지 echo를 이용해 표기한다.

이후 사용자에게 어떠한 동작을 실행할 것인지 read를 이용해서 읽어온 후, 해당 번호에 맞는 동작을 실행하면 된다.

여기에서 사용자가 9를 입력하지 않으면 특정 동작 수행 후 다시 사용자에게 수행할 동작을 입력 받는 상태로 돌아가야 하므로 while문을 통해 userInput이 9인지 아닌지 검사하는 구조로 설계한다.

만약 userInput이 9라면, while문을 빠져나가며 Bye!를 출력하고 프로그램을 종료한다. (9번 함수 구현부분)

만약 userInput이 9가 아니라면, while문으로 진입해 userInput 값에 맞는 동작을 수행한다.

userInput의 값에 따라 동작을 다르게 하는 구현은 case 문을 이용해 구현한다.

각 동작[1-8]의 구현)

1. 1번 동작은 사용자에게 movie\_id를 입력 받고, u.item 파일에서 입력 받은 movie\_id와 동일한 id를 가지는 영화정보를 출력하는 함수이다

Awk를 이용해 구분자를 |로 설정하고, 사용자에게 입력 받은 변수를 사용하기 위해 -v 옵션을 사용한다

이후 u.item의 첫번째 필드가 id정보이므로 $1 == movie\_id 조건식을 통해 동일한 id를 가지는 Line을 탐색하고 print를 이용해 Line 전체를 출력한다.

1. 2번 동작은 action 장르에 속하는 영화의 id와 이름을 출력하는 함수이다

마찬가지로 awk를 이용해 구분자를 |로 설정하고 7번째 필드가 1인 경우에만 action 장르이므로 $7 == 1 조건식을 이용한다.

이후 {print $1, $2}를 이용해 id정보를 담는 첫번째 필드와 영화명을 담는 두번째 필드를 공백으로 구분해 출력한다

10줄만 출력하기 위해 pipe를 이용해 awk의 결과를 head -n 10으로 넘겨주었다.

1. 사용자에게 movie\_id를 입력 받고 해당 movie\_id의 평점을 계산하는 함수이다

Read를 이용해 사용자에게 movie\_id를 입력 받는다.

평점 정보는 u.data에 있으므로 해당파일($2)을 cat해서 그 결과를 pipe를 통해 awk로 전달한다.

awk에서 외부변수를 사용하기 위해 -v 옵션을 주었고, u.data에서 두번째 필드가 id 정보였으므로 $2 == movie\_id 조건을 이용한다.

해당 조건에 걸린 모든 Line의 rating 정보의 합, Line의 수를 계산해야 하므로 해당 조건에 걸릴 때마다 sum, count의 정보를 update 한다.

(세번째 필드가 rating 정보이므로 sum += $3) (걸릴 때마다 count++)

이후 awk 내부에서 END를 사용해 awk 동작이 종료되면 average = sum / count 를 통해 평균을 구한다.

이후 printf를 사용해 정해진 양식(라운딩처리 등)에 맞게 결과를 출력한다.

1. u.item 파일에서 url 부분만 지우고 10줄을 출력하는 함수이다.

사용자가 y를 입력했을 때만 작동해야 하므로 read를 통해 indicator에 사용자 입력 값을 저장 후 if 문을 이용해 조건에 맞게끔 설계한다.

(if [ ${indicator} == "y" ]; then)

cat $1 를 통해 u.item을 읽어오고 pipe를 통해 결과를 sed에 넘겨준다.

sed에서 http로 시작하고, |로 끝나는 부분을 빈 공백으로 바꿔준다 (전체 Line에 대해)

이 결과를 pipe를 통해서 head에 넘겨주고 head -n 10을 통해 10줄을 출력한다.

1. u.user 파일에서 상위 10줄의 user data정보를 읽어와, 새로운 형식에 맞게 출력하는 기능이다.

사용자가 y를 입력했을 때만 작동해야 하므로 read를 통해 indicator에 사용자 입력 값을 저장 후 if 문을 이용해 조건에 맞게끔 설계한다.

cat $3를 이용해 u.user 파일을 읽어오고 | 를 이용해 sed에 넘긴다.

Sed를 이용해 M은 male로 F는 female로 변경한다. (s/M/male/;s/F/female/)

이 결과를 다시 sed에 |를 이용해 넘기고 grouping을 이용해 |로 구분된 단어들을 필드에 할당하고 \1,\2등을 이용해 새로운 형식으로 출력한다.

(s/\(.\*\)|\(.\*\)|\(.\*\)|\(.\*\)|\(.\*\)/user \1 is \2 years old \3 \4/)

이 결과를 상위 10줄만 출력하기 위해 다시 pipe를 이용해 head로 넘기고 head -n 10을 통해 10줄만 출력한다.

1. u.item에서 release date 부분의 양식을 변경하여 하위 10줄만 출력하는 코드이다.

사용자가 y를 입력했을 때만 작동해야 하므로 read를 통해 indicator에 사용자 입력 값을 저장 후 if 문을 이용해 조건에 맞게끔 설계한다.

cat $1을 이용해 u.item을 읽어오고 |로 sed에 넘긴다.

sed에서 Jan,Feb,…,Dec와 같이 영어로 되어있는 부분을 01,02,…,12로 변경한다.

(sed -E 's/Feb/02/')

각각은 |로 연결되며 Dec까지 변경하고 나서 다시 sed로 연결해 19951202와 같은 형식으로 다시 출력한다 (Grouping 이용)

(sed 's/\([0-9]\{2\}\)-\([0-9]\{2\}\)-\([0-9]\{4\}\)/\3\2\1/')

(0-9사이 숫자가 두개 이상[date], - , 0-9사이 숫자가 두개 이상[month], - , 0-9사이 숫자가 네 개 이상[year])

이후 sed 결과를 |로 tail에 전달하여 하위 10개 줄만 출력하도록 한다 (tail -n 10)

1. userId를 입력 받고 해당 userId가 평가한 영화의 id를 |로 구분해 오름차순으로 출력, 이후 평가한 영화의 id와 제목을 |로 구분해 10줄만 출력하는 코드이다.

Read를 이용해 userId를 입력 받는다.

7-1. Movie Id를 |로 구분해 한 줄로 출력하는 부분

이후 cat $2를 이용해 u.data 파일을 읽어와 userId가 평가한 영화의 정보들을 가져올 준비를 한다.

이를 |로 awk와 연결한 후, 입력 받은 userId를 사용하기 위해 -v 옵션을 이용한다.

u.data 파일의 첫번째 필드가 userid 정보이므로 $1 == userId 조건식을 이용한다.

이후 해당 Line의 movieId만 출력하면 되므로 movieId 정보를 담는 두번째 필드를 출력한다 (print $2)

그냥 출력하면 형식에 맞지 않게 여러 줄이 출력되고 |도 존재하지 않으므로 이를 수정해야 한다.

우선 sort -n을 이용해 숫자를 기준으로 오름차순 정렬 시킨다.

Tr을 이용해 개행 문자들을 |로 변경해 한 줄에 출력되도록 만든다.

마지막 movieId 정보 끝에는 |가 들어가면 안되므로 sed를 이용해 |$인 문자를 \n으로 변경해준다.

여기까지가 movieId를 |로 구분해 출력하는 부분이다.

7-2. 평가한 영화의 id와 제목을 |로 구분해 출력하는 부분

Movie Id 기준 오름차순으로 출력해야 하므로 7-1에서와 유사하게 cat, sed, sort 를 이용해 Movie Id를 오름차순으로 정렬한다.

이후 > temp.sorted를 이용해 정렬된 movie id로 구성된 파일을 하나 만든다.

10번만 출력해야 하므로 for문을 이용해 10번만 돌아가게 설계한다.

이후 for문 안에서 temp.sorted 파일을 불러와 read -r movie\_id를 이용해 movie\_id에 한 Line을 읽어오도록 설계한다.

이후 영화제목을 출력하기 위해서 영화제목이 포함된 파일인 u.item 파일을 cat $1을 통해 읽어오고 이를 awk에 pipe를 사용해 넘긴다.

awk에서 외부변수 movie\_id를 사용하기 위해 -v 옵션을 선언한다.

u.item 파일의 구분자를 |로 설정하기 위해 -F 옵션을 선언한다.

이후 $1(movie\_id) == movie\_id를 조건식으로 만들어 temp.sorted에서 읽어온 영화id를 가지는 Line을 찾는다.

이후 해당 조건을 만족할 경우 printf를 이용해 movieId | movieName 형식으로 출력한다.

(For 문이 10번 돌기 때문에 Loop한번에 한 줄만 출력하기 위해 head -n 1을 사용했다.)

이후 사용한 임시파일인 temp.sorted를 rm을 이용해 삭제한다.

1. 20-29살인 Programmer 직업을 가진 사람들이 평가한 영화들의 평균 rating을 movie id와 함께 출력하는 코드이다

우선 사용자가 y를 입력했을 때만 코드가 실행되어야 하므로 read를 이용해 사용자 입력 값을 받아온다.

이후 if 문을 사용해서 받은 값이 y인지 확인하고 y이면 if 문 안으로 진입한다.

8.1

우선 20~29살 Programmer의 userId를 뽑아오기 위해서 u.user 파일을 cat로 읽어온다.

u.user 파일은 |로 구분되어 있으므로 awk에서 구분자를 |로 설정한다.

Awk의 조건식에서 나이를 나타내는 두번째 필드의 값이 범위 내에 오도록 맞추고, ‘직업을 나타내는 4번째 필드의 값이 programmer와 일치하는지’를 조건식으로 만든다.

해당 조건이 맞을 때, userId를 나타내는 첫번째 필드를 출력한다.

이를 파일로 관리하면 편하므로 userid.temp 파일로 내보낸다.

* Userid.temp 파일에는 20-29살 프로그래머인 사람들의 userid가 들어있을 것

8.2

while문에 userid.temp 파일을 넣고 read -p user\_id를 이용해 한 Loop를 돌 때 user\_id에 userid.temp Line 한 줄을 읽어오도록 만든다.

User\_id가 평가한 영화의 id와 rating 정보를 알아내기 위해 u.data 파일을 불러오고 awk 문에게 넘긴다.

Awk 문에서 외부 변수 user\_id를 사용하기 위해 -v 옵션을 사용한다.

조건식으로 $1 == user\_id를 사용해 조건이 맞을 경우 $2(movie\_id), $3(rating)을 공백으로 구분해 movie\_and\_rating.info 파일에 append한다.

(append 하는 이유는 overwrite하게 된다면 while이 한번 돌았을 때 이전 정보가 지워지므로 정상적으로 동작하지 않음)

(movie\_and\_rating.info는 movie\_id rating 형식의 정보들이 저장되었을 것)

Cat과 sort를 이용해 movie\_and\_rating.info를 첫번째 행인 movie\_id 기준으로 정렬하여 movie\_and\_rating\_sorted.info에 저장한다.

(movie\_and\_rating\_sorted.info 에는 movie\_id가 중복 되어있을 수 있음)

8.3

각 movie\_id에 해당하는 rating의 평균을 구하고 이를 movie\_id와 함께 출력하기 위한 부분이다

우선 movie\_id가 1682개이므로 for문을 이용해서 1682번 loop를 돌도록 설계한다.

이후 movie\_and\_rating\_sorted.info 파일을 cat로 불러와 |를 이용해 awk로 넘긴다.

awk에서 외부 변수 var를 사용하기 위해 -v 옵션을 설정한다.

(var는 1부터 1682의 값을 차례로 가지게 될 것)

movie\_and\_rating\_sorted.info 파일의 첫번째 필드는 movie\_id이고 두번째 필드는 rating이므로 awk의 조건식으로 첫번째 필드와 var의 값이 같을 때로 설정한다.

Awk의 동작문으로는 sum += $2를 설정해 조건이 맞을 때 rating 정보를 sum에 누적합 하도록 만든다, 또 조건이 걸렸을 때 count 변수에 +1을 하도록 만든다.

awk에서 END를 통해 끝까지 파일을 읽고, count가 0보다 크다면, 평균 계산이 가능하다는 의미이므로, average에 평균을 계산하고 printf를 이용해 형식에 맞게 출력한다

(%.5f는 float 자료형 소수점 5자리까지 표현하는 것을 의미함)

모든 동작이 종료되었다면 rm을 이용해서 생성했던 임시 파일들 (movie\_and\_rating.info, userid.temp, movie\_and\_rating\_sorted.info)을 삭제한다. (if문 안에서만 rm 키워드들이 있어야함)