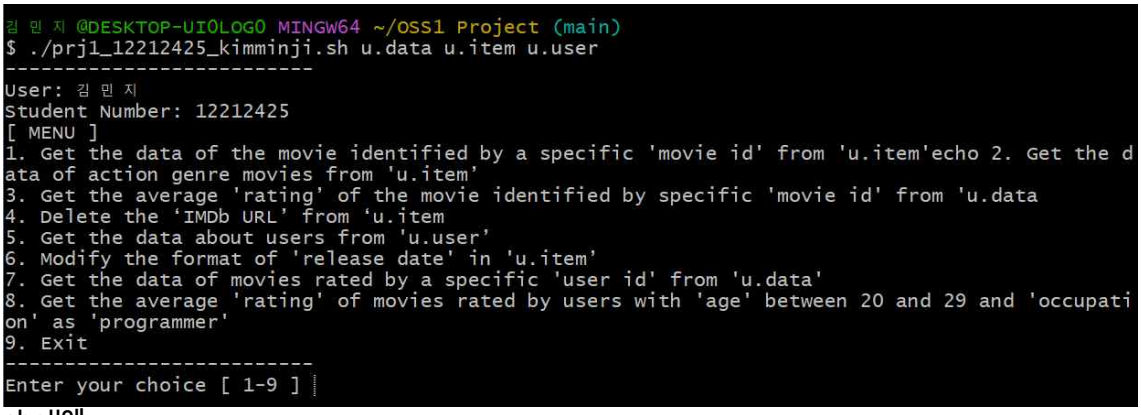
OSS Project1 과제 README 보고서

12212425 김민지

▼ 스크립트 사용법



shell에

$ ./prj1_12212425_kimminji.sh u.data u.item u.user

명령어를 입력하면, 다음과 같이 User의 이름과 학번과 함께 MENU가 출력된다.

이와 함께 MENU의 번호(1~9)를 선택하라는 명령이 나온다.

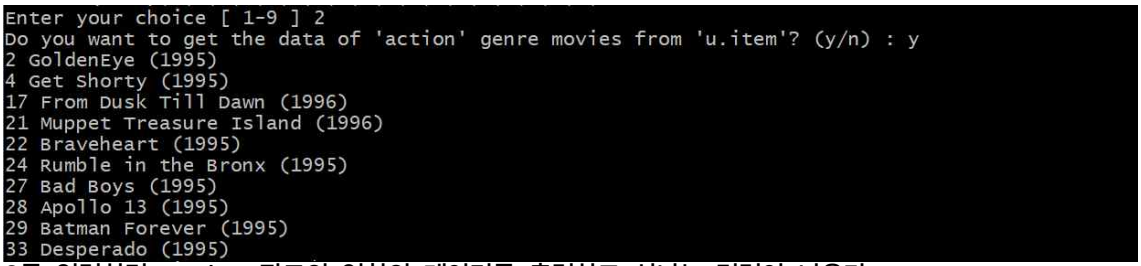
이제 각 번호를 선택함에 따라 어떻게 되는지 살펴보자.

1)

1을 입력하면, movie id를 입력하라는 명령이 나온다.

movie id를 입력하면, 해당 id를 갖는 movie의 정보가 출력된다.

2)



2를 입력하면, action 장르의 영화의 데이터를 출력하고 싶냐는 명령이 나온다.

여기서, y를 입력하면

movie id를 기준 오름차순 10개의 action 장르의 영화들이 출력된다.

이때, 출력 형태는 '영화 아이디' '영화 제목' 이다.

3)

```
Enter your choice [ 1-9 ] 3
Please enter the 'movie id' (1~1682): 1
average rating of 1: 3.87832
```

3을 입력하면,
movie id를 입력하라는 명령이 나온다.
movie id를 입력하면 해당 movie id의 평균 평점이 출력된다.
이때, 평균 평점은 소수점 다섯자리까지 반올림되어 출력된다.

4)

```
Enter your choice [ 1-9 ] 4
Do you want to delete the 'IMDb URL' from 'u.item'?(y/n): y
1|Toy Story (1995)|01-Jan-1995|||0|0|0|1|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0
2|GoldenEye (1995)|01-Jan-1995|||0|1|1|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
3|Four Rooms (1995)|01-Jan-1995|||0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
4|Get Shorty (1995)|01-Jan-1995|||0|1|0|0|0|1|0|0|1|0|0|0|0|0|0|0|0|0
5|Copycat (1995)|01-Jan-1995|||0|0|0|0|0|0|1|0|1|0|0|0|0|0|0|1|0|0
6|Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)|01-Jan-1995|||0|0|0|0|0|0
|0|0|1|0|0|0|0|0|0|0|0|0
7|Twelve Monkeys (1995)|01-Jan-1995|||0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|1|0|0|0
8|Babe (1995)|01-Jan-1995|||0|0|0|0|1|1|0|0|1|0|0|0|0|0|0|0|0|0|0
9|Dead Man Walking (1995)|01-Jan-1995|||0|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0
10|Richard III (1995)|22-Jan-1996|||0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|1|0
```

4를 입력하면,
'u.item'에서 'IMDb URL' 을 삭제하고 싶냐는 명령이 나온다.
y를 입력하면, u.item에서 각 줄마다 IMDb URL이 삭제되어 출력된다.
이때, 총 10개의 데이터만 출력된다.

5)

```
Enter your choice [ 1-9 ] 5
Do you want to get the data about users from 'u.user'?(y/n): y
user 1 is 24 years old male technician
user 2 is 53 years old female other
user 3 is 23 years old male writer
user 4 is 24 years old male technician
user 5 is 33 years old female other
user 6 is 42 years old male executive
user 7 is 57 years old male administrator
user 8 is 36 years old male administrator
user 9 is 29 years old male student
user 10 is 53 years old male lawyer
```

5를 입력하면,
'u.user'에서 user들에 대한 데이터를 가지고 오고 싶냐는 명령이 나온다.
y를 입력하면,
user 'user id' is 'age' years old 'gender' 'occupation' 의 형태로 총 10개의 데이터가 출력된다.

6)

6을 입력하면,

'u.item'에서 'release data'의 포맷을 변경하고 싶냐는 명령이 나온다.

y를 입력하면,

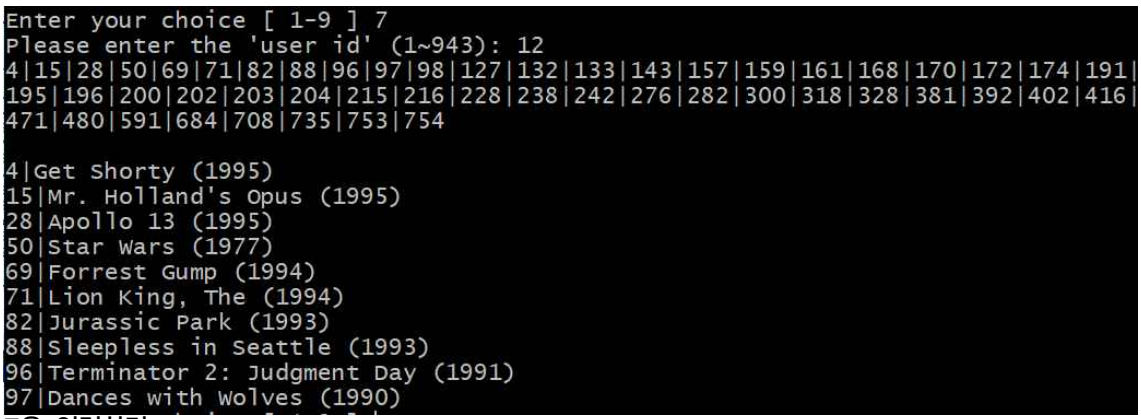
movie id가 1673 ~ 1682인 데이터드리이 release data의 포맷을 다음과 같이 변경하여 출력한다.

01-Jan-1995 -> 19950101 로 변경한다.

한 가지 예를 더 들자면,

02-Sep-1992 -> 19920902 로 변경한다.

7)



7을 입력하면,

user id를 입력하라는 명령이 출력된다.

user id를 입력하면, 해당 user가 평가를 매긴 movie들의 정보들을 |를 구분자로 하여 모두 출력한다.

또한 movie id를 기준으로 오름차순 정렬하여 해당 user가 평가를 매긴 movie들의 movie id와 movie title을 상위 10개를 출력한다.

8)

```
Do you want to get the average 'rating' of movies rated by users with 'age' betw
een 20 and 29 and 'occupation' as 'programmer'?(y/n): y
1 4.2941
2 3
3 3.5
4 3.7
5 3.25
7 4.2222
8 3.5
9 4.1
10 4
11 4.3125
12 4.6923
13 3.375
14 4
15 3.8571
16 3
17 3.5
19 4
20 1
21 2.8
22 4.1429
23 4.5714
24 3.7692
25 3.3846
26 3.25
27 3
28 4
```

8을 입력하면,
영화마다 20대(20~29)이면서 programmer인 user들이 매긴 평균 평점을 출력하고 싶냐는 명령이 나온다.
y를 입력하면,
영화별로 20대(20~29)이면서 programmer인 user들이 매긴 평균 평점을 소수점 다섯자리까지 반올림하여 출력한다. 이때, 불필요한 0은 제거되어 출력되고 조건에 부합하는 영화가 없는 경우는 출력되지 않는다.

9)

```
Enter your choice [ 1-9 ] 9
Bye!
```

9를 입력하면,
Bye! 가 출력되면서 프로그램이 끝나게 된다.

▼ Script 구현 내용

이제 작성한 script의 코드를 자세히 분석해 보자.

코드를 설명하면서 반복되는 명령어들에 대한 설명은 처음 한 번씩만 작성하였다.

```
#!/bin/bash

echo "--------------------------"
echo "User: 김 민 지 "
echo "Student Number: 12212425"
echo "[ MENU ]"
echo "1. Get the data of the movie identified by a specific 'movie id' from 'u.item'"echo "2. G
et the data of action genre movies from 'u.item'"
echo "3. Get the average 'rating' of the movie identified by specific 'movie id' from 'u.data"
echo "4. Delete the 'IMDb URL' from 'u.item"
echo "5. Get the data about users from 'u.user'"
echo "6. Modify the format of 'release date' in 'u.item'"
echo "7. Get the data of movies rated by a specific 'user id' from 'u.data'"
echo "8. Get the average 'rating' of movies rated by users with 'age' between 20 and 29 and 'oc
cupation' as 'programmer'"
echo "9. Exit"
echo "--------------------------"
```

가장 첫 줄에 #!/bin/bash 라는 문구를 작성했다. 이 명령어의 의미는 bash shell을 사용하겠다는 의미이다.

그리고 echo 명령어를 사용하여 [MENU]를 출력하기 위한 문구를 작성한다.

```
while true
do
        echo -n "Enter your choice [ 1-9 ] "
        read choice

        case $choice in
```

그 후, while문을 활용하여 9가 입력될 때까지 "Enter your choice [1-9]"를 출력하고 Menu에 해당하는 숫자를 입력받는 과정을 반복한다.

choice라는 변수를 두어 case문을 사용하여 1~9번까지의 Menu들에 맞는 각각의 명령어들이 출력된다.

1)

```
 1)
         echo -n "Please enter 'movie id' (1~1682): "
         read movieId

         info=$(cat u.item | awk -F"|" -v movieId="$movieId" '$1==movieId')
         echo $info
         ;;
```

먼저, movieId를 입력 받는다.

awk 명령어는 텍스트를 처리하는 데 사용되고 -F 옵션은 필드 구분자를 설정하는 데에 사용된다.
-v 옵션은 awk 스크립트에 변수를 전달하는 데에 사용된다.

여기서는 | 를 구분자로 설정하고 u.item에서 movieId와 일치하는 첫번째 필드를 가진 라인을 찾아서 info 변수에 저장한다.

그 후, echo 명령어를 사용하여 info 변수의 값을 출력한다.

2)

```
2)
        echo -n "Do you want to get the data of 'action' genre movies from 'u.item'? (y/n) : "
        read respond

        if [ "$respond" == "y" ]
        then
                awk -F"|" '$7=="1" {print $1, $2}' u.item | sort -n | head -10
        fi
        ;;
```

if문을 활용하여 입력 받은 respond가 y인지 확인한다.
|를 구분자로 하여 u.item에서 7번째 필드가 1이면 첫번째와 두 번째 필드를 출력한다.
이때, sort 명령어를 활용해 오름차순으로 정렬하고 head 명령어를 활용하여 앞에서부터 10개 라인만을 출력하였다.
여기서 -n 옵션을 사용하면 숫자를 기준으로 정렬한다.
fi는 if문의 끝을 나타낸다. fi가 없으면 bash는 if문의 끝을 찾지 못해서 script가 제대로 실행되지 않는다.

3)

```
3)
        echo -n "Please enter the 'movie id' (1~1682): "
        read movieId

        awk -F" " -v movieId="$movieId" '
        $2==movieId {
                total += $3
                count++
        }
        END {
                printf "average rating of %s: %.5f\n", movieId, total/count
        }' u.data
        ;;
```

" "(공백)을 구분자로 설정하여 movieId라는 변수를 사용자가 입력한 movieId값으로 설정하였다.
두 번째 필드($2)와 movieId가 동일한 라인만 선택하였고,
선택된 라인의 세 번째 필드 값을 total 변수에 더하고, count 변수를 증가시키도록 하여 선택된 라인의 세 번째 필드 값의 합과 개수를 계산합니다.
END { printf "average rating of %s: %.5f₩n", movieId, total/count }
파일 처리가 끝나면 (END), 선택된 라인들의 평균(total/count)을 계산해 출력한다.
5f를 사용해 소수점 다섯번째 자리까지만 출력하도록 하였다.

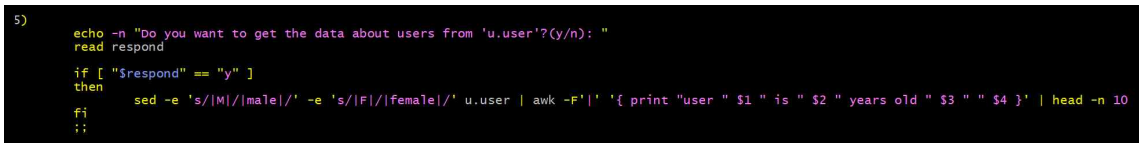
4)

```
4)
        echo -n "Do you want to delete the 'IMDb URL' from 'u.item'?(y/n): "
        read respond

        if [ "$respond" == "y" ]
        then
                sed 's/|http[^|]*|/||/g' u.item | sort -n | head -n 10
        fi
        ;;
```

if문을 활용하여 입력 받은 respond가 y인지 확인하고 y라면,
여기서 사용한 sed는 스트림 편집기로 텍스트 파일을 처리하는 데에 사용되는 명령어이다.

's/|http[^|]*|/||/g' 이 부분을 통해 u.item 파일에서 |http로 시작하여 |로 끝나는 모든 문자열을 찾고, 그것들을 ||로 치환하도록 하였다.
sort -n을 사용해 숫자 기준으로 정렬하였고, head -n 10 을 통해 앞 10개만을 출력하도록 하였다.

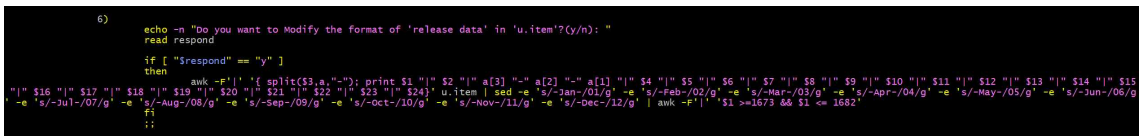
5)



```
5)
        echo -n "Do you want to get the data about users from 'u.user'?(y/n): "
        read respond

        if [ "$respond" == "y" ]
        then
                sed -e 's/|M|/|male|/' -e 's/|F|/|female|/' u.user | awk -F'|' '{ print "user " $1 " is " $2 " years old " $3 " " $4 }' | head -n 10
        fi
        ;;
```

if문을 활용하여 입력 받은 respond가 y인지 확인하고 y라면,
-e 옵션을 통해서 2개의 sed 치환 명령을 동시에 실행하도록 하였다.
|M|을 |Male|로 , |F|를 |Female|로 치환하였다.
-F 옵션을 사용해 | 를 구분자로 하여 필드를 나누어 “user” $1 + “is” 등과 같이 원하는 문장을 출력하도록 하였다.
또한, head 명령어를 사용하여 앞 10개 라인만 출력하도록 하였다.

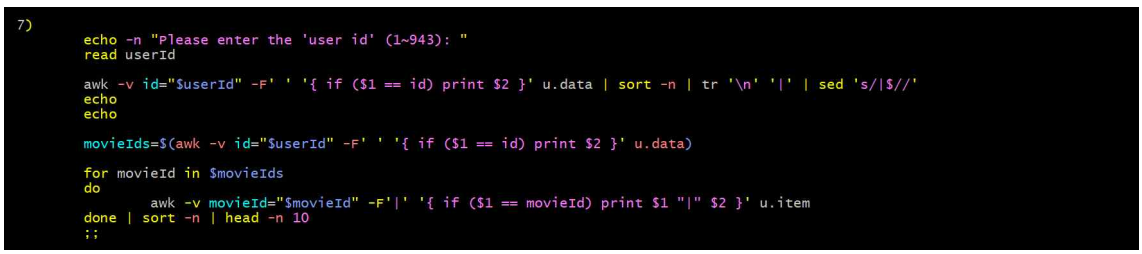
6)



```
6)
        echo -n "Do you want to Modify the format of 'release data' in 'u.item'?(y/n): "
        read respond

        if [ "$respond" == "y" ]
        then
                awk -F'|' '{ split($3,a,"-"); print $1 "|" $2 "|" a[3] "-" a[2] "-" a[1] "|" $4 "|" $5 "|" $6 "|" $7 "|" $8 "|" $9 "|" $10 "|" $11 "|" $12 "|" $13 "|" $14 "|" $15 "|" $16 "|" $17 "|" $18 "|" $19 "|" $20 "|" $21 "|" $22 "|" $23 "|" $24}' u.item | sed -e 's/-Jan-/01/g' -e 's/-Feb-/02/g' -e 's/-Mar-/03/g' -e 's/-Apr-/04/g' -e 's/-May-/05/g' -e 's/-Jun-/06/g' -e 's/-Jul-/07/g' -e 's/-Aug-/08/g' -e 's/-Sep-/09/g' -e 's/-Oct-/10/g' -e 's/-Nov-/11/g' -e 's/-Dec-/12/g' | awk -F'|' '$1 >=1673 && $1 <= 1682'
        fi
        ;;
```

if문을 활용하여 입력 받은 respond가 y인지 확인하고 y라면,
|를 구분자로 하여 세 번째 필드를 “-”로 split 하여 원하는 문장을 출력하도록 하였다.
이때, sed 명령어를 통해 영어로 된 각 월의 name을 숫자로 치환하였다.
또한 |를 구분자로 하여 첫 번째 필드가 1673~1682인 데이터들만을 추출하여 출력하도록 하였다.

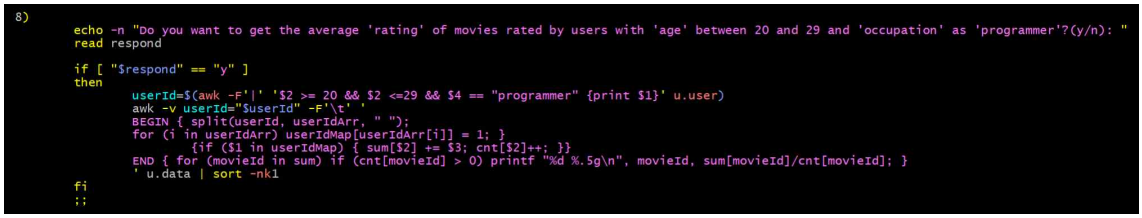
7)



```
7)
        echo -n "Please enter the 'user id' (1~943): "
        read userId

        awk -v id="$userId" -F' ' '{ if ($1 == id) print $2 }' u.data | sort -n | tr '\n' '|' | sed 's/|$//'
        echo
        echo

        movieIds=$(awk -v id="$userId" -F' ' '{ if ($1 == id) print $2 }' u.data)

        for movieId in $movieIds
        do
                awk -v movieId="$movieId" -F'|' '{ if ($1 == movieId) print $1 "|" $2 }' u.item
        done | sort -n | head -n 10
        ;;
```

u.data 파일에서 입력 받은 userId와 첫 번째 필드가 같으면 두 번째 필드를 id 변수에 저장한다.
sort -n 을 사용하여 숫자를 기준으로 정렬하였다.
tr은 문자 변환을 수행하는 auddfudjd이다. ₩n을 |로 변환하도록 하였다.
sed (치환 명령어)를 사용하여 맨 끝에 있는 |를 삭제하도록 하였다.

8)



if문을 활용하여 입력 받은 respond가 y인지 확인하고 y라면,
awk 명령어를 사용하여 u.user 파일에서 age가 20~29 이고, occupation가 programmer인 userId를 추출하여 userId라는 변수에 저장한다.
BEGIN 블럭에서는 userId를 공백을 기준으로 하여 분리해 userIdArr라는 배열에 저장하고,
userId를 key로 하는 userIdMap을 생성하도록 하였다.

{if ($1 in userIdMap) { sum[$2] += $3; cnt[$2]++; }} : u.data 파일을 읽으면서 userId가 userIdMap에 포함되어 있으면, 해당 movieId의 rate를 sum이라는 변수에 더하고 count를 증가시키도록 하였다.

END { for (movieId in sum) if (cnt[movieId] > 0) printf "%d %.5g\n", movieId, sum[movieId]/cnt[movieId]; } : 이렇게 모든 라인을 처리하면, 각 movieId에 대해 count가 0보다 크면 rate의 평균을 계산하여 movieId와 함께 출력하도록 하였다.

그리고 sort -nk1을 통해 movieId를 통해서 오름차순 정렬하였다.

9)

echo 명령어를 사용하여 “Bye!”를 출력하고,
break를 사용하여 while문을 빠져나오도록 하였다.