오픈소스sw개론

인공지능공학과 12220632 이경민

- 프로젝트 간단한 소개

u.item 데이터

>> movie id | movie title | release date | video release date | IMDb URL | Genre..

u.data 데이터

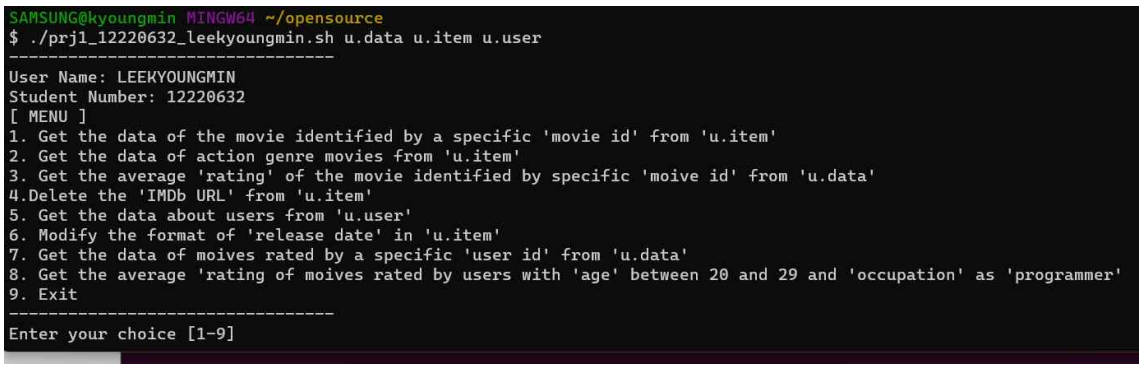
>> user id | movie id | rating | timestamp

u.user 데이터

>> user id | age | gender | occupation

u.item, u.data, u.user 파일의 데이터들을 이용해 아래 9가지 기능을 구현

- 1. 'u.item'에서 특정 'movie id'로 식별된 영화의 데이터를 가져오기
- 2. 'u.item'에서 '액션' 장르 영화 데이터 가져오기
- 3. 'u.data'에서 특정 'movie id'로 식별된 영화의 평균 '등급' 획득
- 4. 'u.item'에서 'IMDb URL' 삭제
- 5. 'u.user'에서 사용자에 대한 데이터 가져오기
- 6. 'u.item'의 '출시일' 형식 수정
- 7. 'u.data'에서 특정 'user id'로 평가된 영화의 데이터 가져오기
- 8. '연령'이 20~29세, '직업'이 '프로그래머'인 이용자가 평가한 영화의 평균 '등급'을 구하기
- 9. script 종료



>> ./prj1_12220632_leekyoungmin.sh u.data u.item u.user을 입력해 프로그램을 실행시키면 위에 사진과 같이 출력되고 사용자가 1~9의 숫자를 입력해 원하는 기능을 출력 가능하다.

- 자세한 기능 설명과 코드 설명

1>> 기능설명

2>> 코드설명

#! /bin/bash을 첫 줄에 작성하고 시작!!

```
echo "--------------------------------"
echo "User Name: LEEKYOUNGMIN"
echo "Student Number: 12220632"
echo "[ MENU ]"
echo "1. Get the data of the movie identified by a specific 'movie id' from 'u.item'"
echo "2. Get the data of action genre movies from 'u.item'"
echo "3. Get the average 'rating' of the movie identified by specific 'moive 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 moives rated by a specific 'user id' from 'u.data'"
echo "8. Get the average 'rating of moives rated by users with 'age' between 20 and 29 and 'occupation' as 'programmer'"
echo "9. Exit"
echo "--------------------------------"
```

1>> 단순 9개의 메뉴 출력

2>> echo: 화면에 출력하는 명령어

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

        case $number in
                1)
                        first_function;;
                2)
                        second_function;;
                3)
                        third_function;;
                4)
                        fourth_function;;
                5)
                        fifth_function;;
                6)
                        sixth_function;;
                7)
                        seventh_function;;
                8)
                        eighth_function;;
                9)
                        echo "Bye!"
                        break;;
        esac
done
```

1>> 사용자가 9를 입력하기 전까지 기능을 수행하기 위함

2>> while 루프 이용해 9가 입력될 때까지 반복 read로 사용자에게 입력을 받고 number에 저장. case문을 이용해 각 함수를 호출. 9가 입력되면 Bye!를 출력하고 break(종료)

read: 입력하면 변수에 저장하는 명령어

-p: 해당 문자열 출력

case: 해당 값에 따라 작업을 나누는 조건문과 비슷

아래는 8개의 함수에 대한 설명 (중복되는 명령어나 옵션은 한번만 자세히 설명했음)

1) first_fuction() 설명 -- [사용자가 1 입력]

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

Please enter 'movie id'(1~1682) : 121212

Please enter 'movie id'(1~1682) : 1

1|Toy Story (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)|0|0|0|1|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0

Enter your choice [1-9]
```

1 >> 입력받은 movie id를 u.item 파일에서 해당 영화 데이터 출력. 1682까지의 영화 id가 있고 이를 벗어나면 재입력받게 구현함.

```
first_function() {
        while true
        do
                read -p "Please enter 'movie id'(1~1682) : " id_number
                echo

                if [ $id_number -ge 1 ] && [ $id_number -le 1682 ]
                then
                        break
                fi
        done

        awk -F '|' -v id_num="$id_number" '$1 == id_num {print $0}' u.item
        echo
}
```

2>> while문 사용 read로 사용자에게 입력받아 id_number에 저장 영화 아이디가 1~1682사이가 아니라면 계속 반복하게 구현. u.item는 awk에 입력되는 파일

awk: 데이터 처리와 출력,
-F: 데이터 구분, 이 코드에서는 | 로 구분함
-v: 외부 변수를 awk내에서 사용하기 위한 옵션 (id_num에 $id_number할당)
'$1 == id_num {print $0}' : read로 사용자에게 입력받은 id_num이 |로 구분한 첫 번째 부분과 같으면 $0(라인 전체) 출력
-ge: >= , -le: <=

2) second_function() 설명 -- [사용자가 2 입력]

```
--------------------------------
Enter your choice [1-9] 2

Do you want to get the data of 'action' genre movies from 'u.item'?(y/n) : n

Enter your choice [1-9] 2

Do you want to get the data of 'action' genre movies from 'u.item'?(y/n) : y

2 GoldenEye (1995)
4 Get Shorty (1995)
17 From Dusk Till Dawn (1996)
21 Muppet Treasure Island (1996)
22 Braveheart (1995)
24 Rumble in the Bronx (1995)
27 Bad Boys (1995)
28 Apollo 13 (1995)
29 Batman Forever (1995)
33 Desperado (1995)

Enter your choice [1-9]
```

1 >> 사용자에게 액션 장르의 영화 데이터를 가져올까?라고 질문을 하고 만약 n이라고 입력하면 다시 메뉴를 선택하게 하였고 y라면 u.item 데이터 파일에서 액션 장르 영화를 '영화 id' '영화제목'으로 오름차순에 따라 10개 출력하는 기능

```
second_function() {
        while true
        do
                read -p "Do you want to get the data of 'action' genre movies from 'u.item'?(y/n) : " yes_no
                echo

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

2>> while 사용 read로 사용자에게 입력받아 yes_no에 저장. 사용자의 답이 y거나 n이면 break하고 다른 것이 입력되면 while 계속 작동되게 구현. 만약 사용자의 입력이 y면 출력

이것도 1번과 똑같이 데이터 처리와 출력을 위해 awk 이용하고 -F로 | 로 구분한 뒤 7번째 부분이 1이면 첫 번째 부분과 두 번째 부분 출력. u.item은 awk에 입력하기 위한 파일

| : | 기준 왼쪽에서의 출력을 오른쪽으로 전달
head -10: 상위 10개 출력

3) third_function() 설명 -- [사용자가 3 입력]

```
--------------------------------
Enter your choice [1-9] 3

Please enter the 'movie id'(1~1682) : 231313

Please enter the 'movie id'(1~1682) : 1

average rating of 1: 3.87832

Enter your choice [1-9]
```

1 >> 입력받은 movie id를 u.data 파일에서 해당 영화 데이터 출력. 1682까지의 영화 id가 있고 이를 벗어나면 재입력받게 구현함. 입력받은 영화 id의 평점을 출력하는데 6번째에서 반올림해서 소수점 5번째까지 출력하는 기능.

```
third_function() {
        while true
        do
                read -p "Please enter the 'movie id'(1~1682) : " id_number
                echo

                if [ $id_number -ge 1 ] && [ $id_number -le 1682 ]
                then
                        break
                fi
        done
        total=$(awk -F'\t' -v id_num="$id_number" '$2 == id_num { sum+=$3 } END { print sum }' u.data)
        count=$(awk -F'\t' -v id_num="$id_number" '$2 == id_num' u.data | wc -l)
        if [ $count == 0 ]
        then
                echo "can't get rating"
                echo
        else
                average=$(echo "$total $count" | awk '{ print $1 / $2 }')
                average_float=$(echo "$average" | awk '{ print int($1 * 1000000 + 0.5) / 1000000 }')
                average_rating=$(echo "$average_float" | awk '{printf "%.5f", $1 }')
                echo "average rating of $id_number: $average_rating"
                echo
        fi

}
```

2>> while문 사용 read로 사용자에게 입력받아 id_number에 저장. 영화 아이디가 1~1682사이가 아니라면 계속 반복하게 구현. 만약 사이의 값을 입력했다면 출력. u.data는 awk에 입력되는 파일

이것도 앞에 코드와 같이 데이터 처리와 출력을 위해 awk 이용, -F를 이용 이번에는 \t로 구분하고 awk 스크립트 내부에서 값을 사용하기 위해 -v 이용. 두 번째 부분이 id_num과 같으면 세 번째 부분(rating)을 sum에 더하기.
END: 데이터의 모든 줄에 대해 작업하고 실행됨 (sum값 출력)
이를 total에 저장. 똑같이 이 기능을 한번 더 수행하는데
wc: 문자나 줄이나 개수를 세는 명령어, -l: line (출력의 줄 세기)
즉 해당 영화를 평가한 사람 수를 뜻함. 이때 if를 사용해 count가 0이면 can't get rating 출력하고 0이 아니라면 평균 구함. echo로 total과 count 값을 출력하고 출력된 값을 나눔. 6번째 자리에서 반올림 해야하기 때문에 1000000을 곱하고 0.5를 더하고 다시 1000000으로 나눠줬다. 이를 %.5f" 즉, 5번째까지 출력되게 구현하였다.

4) fourth_function() 설명 -- [사용자가 4 입력]

```
Enter your choice [1-9] 4

Do you want to delete the 'IMDb URL' from 'u.item'?(y/n) : n

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
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|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|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

Enter your choice [1-9]
```

1>> 사용자에게 IMDb URL을 삭제하고 싶나요? 라고 물어본다. 만약 n이라고 입력하면 다시

메뉴를 선택하게 하였고 y라면 u.item 데이터 파일에서 IMDb URL을 삭제하고 10줄만 출력한다.

```
fourth_function() {
        while true
        do
                read -p "Do you want to delete the 'IMDb URL' from 'u.item'?(y/n) : " yes_no
                echo

                if [ "$yes_no" == "y" ] || [ "$yes_no" == "n" ]
                then
                        break
                fi
        done
        if [ "$yes_no" == y ]
        then
                sed 's/http:[^)]*)//' u.item | head -10
                echo
        fi
}
```

2<< while 사용 read로 사용자에게 입력받아 yes_no에 저장. 사용자의 답이 y거나 n이면 break하고 다른 것이 입력되면 while 계속 작동되게 구현. 만약 사용자의 입력이 y면 출력

sed: 문자열을 삭제, 수정등을 할 수 있음.
's/http:[^)]*)//' 이 부분을 자세히 설명하면 수정하는 명령어로 http:로 시작하고)로 끝나는 문자열을 찾아 // (//사이에 아무것도 입력되어 있지 않아서) 빈 문자열 출력 head -10으로 상위 10개만 출력
u.item은 파일

5) fifth_function() 설명 -- [사용자가 5 입력]

```
--------------------------------
Enter your choice [1-9] 5

Do you want to get the data about users from 'u.user'?(y/n) : n

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

Enter your choice [1-9]
```

1>> 사용자에 대한 데이터를 가져올까요? 라고 물어본다. 만약 n이라고 입력하면 다시 메뉴를 선택하게 하였고 y라면 u.user 데이터 파일에서 user 'user id' is '나이' years old '성별' '직업' 형태로 10줄만 인쇄한다.

```
fifth_function() {
        while true
        do
                read -p "Do you want to get the data about users from 'u.user'?(y/n) : " yes_no
                echo

                if [ "$yes_no" == "y" ] || [ "$yes_no" == "n" ]
                then
                        break
                fi
        done
        if [ "$yes_no" == "y" ]
        then
                awk -F '|' '{ printf("user %s is %s years old %s %s\n", $1, $2, ($3 == "M" ? "male" : "female"), $4) }' u.user | head -10
                echo
        fi
}
```

2>> while 사용 read로 사용자에게 입력받아 yes_no에 저장. 사용자의 답이 y거나 n이면 break하고 다른 것이 입력되면 while 계속 작동되게 구현. 만약 사용자의 입력이 y면 출력

이것도 앞에 코드와 같이 데이터 처리와 출력을 위해 awk 아용 -F를 이용 이번에도 | 로 구분하고
printf: 주어진 format에 따라 출력
| 로 구분된 첫 번째 부분, 2번째 부분, 3번째 부분은 M이면 male 아니라면 female을 차례로 반환한다. head를 이용해 상위 10줄만 출력한다.
u.user는 파일이름

6) sixth_function() 설명 -- [사용자가 6 입력]

```
--------------------------------
Enter your choice [1-9] 6

Do you want to Modify the format of 'release data' in 'u.item'?(y/n) : n

Enter your choice [1-9] 6

Do you want to Modify the format of 'release data' in 'u.item'?(y/n) : y

1673|Mirage (1995)|19950101||http://us.imdb.com/M/title-exact?Mirage%20(1995)|0|1|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
1674|Mamma Roma (1962)|19620101||http://us.imdb.com/M/title-exact?Mamma%20Roma%20(1962)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0
1675|Sunchaser, The (1996)|19961025||http://us.imdb.com/M/title-exact?Sunchaser,%20The%20(1996)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0
1676|War at Home, The (1996)|19960101||http://us.imdb.com/M/title-exact?War%20at%20Home%2C%20The%20%281996%29|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0
1677|Sweet Nothing (1995)|19960920||http://us.imdb.com/M/title-exact?Sweet%20Nothing%20(1995)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0
1678|Mat' i syn (1997)|19980206||http://us.imdb.com/M/title-exact?Mat%27+i+syn+(1997)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0
1679|B. Monkey (1998)|19980206||http://us.imdb.com/M/title-exact?B%2E+Monkey+(1998)|0|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|1|0|0
1680|Sliding Doors (1998)|19980101||http://us.imdb.com/Title?Sliding+Doors+(1998)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|1|0|0|0|0
1681|You So Crazy (1994)|19940101||http://us.imdb.com/M/title-exact?You%20So%20Crazy%20(1994)|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0|0|0
1682|Scream of Stone (Schrei aus Stein) (1991)|19960308||http://us.imdb.com/M/title-exact?Schrei%20aus%20Stein%20(1991)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|
0

Enter your choice [1-9]
```

1>> u.item의 release 데이터 형식을 수정하기를 원하나요? 라고 물어본다. 만약 n이라고 입력하면 다시 메뉴를 선택하게 하였고 y라면 u.item 데이터 파일에서 01-Jan-1995의 release 데이터를 19950101 형태로 바꿔서 인쇄한다. 이 기능은 앞에 것과 다르게 뒤에 10줄을 출력한다.

```
sixth_function() {
        while true
        do
                read -p "Do you want to Modify the format of 'release data' in 'u.item'?(y/n) : " yes_no
                echo

                if [ "$yes_no" == "y" ] || [ "$yes_no" == "n" ]
                then
                        break
                fi
        done

        if [ "$yes_no" == "y" ]
        then
                IFS=$'\n'
                for movie_info in $(cat u.item | tail -10)
                do
                        original_date=$(echo $movie_info | awk -F'|' '{print $3}')
                        change_date=$(date -d "$original_date" +"%Y%m%d")
                        print_date=$(echo "$movie_info" | sed -E "s/([^|]*\|[^|]*\|)[0-9]{2}-[A-Za-z]{3}-[0-9]{4}/\1$change_date/")
                        echo "$print_date"
                done
        echo
        fi
}
```

2>> while 사용 read로 사용자에게 입력받아 yes_no에 저장. 사용자의 답이 y거나 n이면 break하고 다른 것이 입력되면 while 계속 작동되게 구현. 만약 사용자의 입력이 y면 출력

IFS: 구분하기 위함. 이 코드에서는 \n(줄)로 구분하기 위함이다. 이는 for문 항목들이 줄로 구분하기 위해 작성하였다.
cat: 파일 출력하는 명령어
tail: 마지막 출력하는 명령어 코드에서는 -10으로 마지막 10줄을 출력한다.
echo를 이용해 movie_info 값 출력한 뒤 이것도 앞에 코드와 같이 데이터 처리와 출력을 위해 awk 이용 -F를 이용 이번에도 | 로 구분하고 3번째 부분 release data를 original_date에 저장
date: 날짜와 관련된 명령어
original_date를 yyyymmdd 형식으로 변환하기 위해 +"%Y%m%d"를 이용. 변경된 값을 change_date에 저장.
echo를 이용해 movie_info 값 출력한 뒤 sed를 이용.
-E: 더 많은 정규 표현식을 이용한다는 옵션이다. ([^|]*\|[^|]*\|)는 |로 구분된 첫 번째 부분, 두 번째 부분 뒤에 | 문자를 찾는다. [0-9]{2}-[A-Za-z]{3}-[0-9]{4}는 그 뒤에 **-***-****을 찾는다. 첫 번째 부분, 두 번째 부분, | 을 그대로 사용하고 chage_date값으로 대체함. 이를 print_date에 저장후 인쇄

7) seventh_function() 설명 -- [사용자가 7 입력]

```
----------------------------------
Enter your choice [1-9] 7

Please enter the 'user id'(1~943) : 123123

Please enter the 'user id'(1~943) : 12

4|15|28|50|69|71|82|88|96|97|98|127|132|133|143|157|159|161|168|170|172|174|191|195|196|200|202|203|204|215|216|228|238|242|276|282|300|318|328|381|392|402|
416|471|480|591|684|708|735|753|754

4|Get Shorty (1995)
15|Mr. Holland's Opus (1995)
28|Apollo 13 (1995)
50|Star Wars (1977)
69|Forrest Gump (1994)
71|Lion King, The (1994)
82|Jurassic Park (1993)
88|Sleepless in Seattle (1993)
96|Terminator 2: Judgment Day (1991)
97|Dances with Wolves (1990)

Enter your choice [1-9]
```

1>> 943개의 user id가 있고 이를 벗어나면 재입력받게 구현함. 입력받은 user id가 rating한 영화 id를 | 로 구분하여 모두 인쇄, 또한 rating된 영화 id 10개를 'movie id' | 'movie title'로 인쇄하는 기능.

```
seventh_function() {
        while true
        do
                read -p "Please enter the 'user id'(1~943) : " user_number
                echo

                if [ $user_number -ge 1 ] && [ $user_number -le 943 ]
                then
                        break
                fi
        done

        user_movie_id=$(awk -F'\t' -v user_num="$user_number" '$1 == user_num {print $2}' u.data | sort -n | tr '\n' '|' | sed 's/|$//' )
        echo $user_movie_id
        echo

        top10_user_movie_id=$(awk -F'\t' -v user_num="$user_number" '$1 == user_num {print $2}' u.data | sort -n | head -10)

        for topten_user_movie_id in $top10_user_movie_id
        do
                movie_title=$(awk -F'|' -v mvid="$topten_user_movie_id" '$1 == mvid {print $2}' u.item)
                echo "$topten_user_movie_id|$movie_title"
        done
        echo
}
```

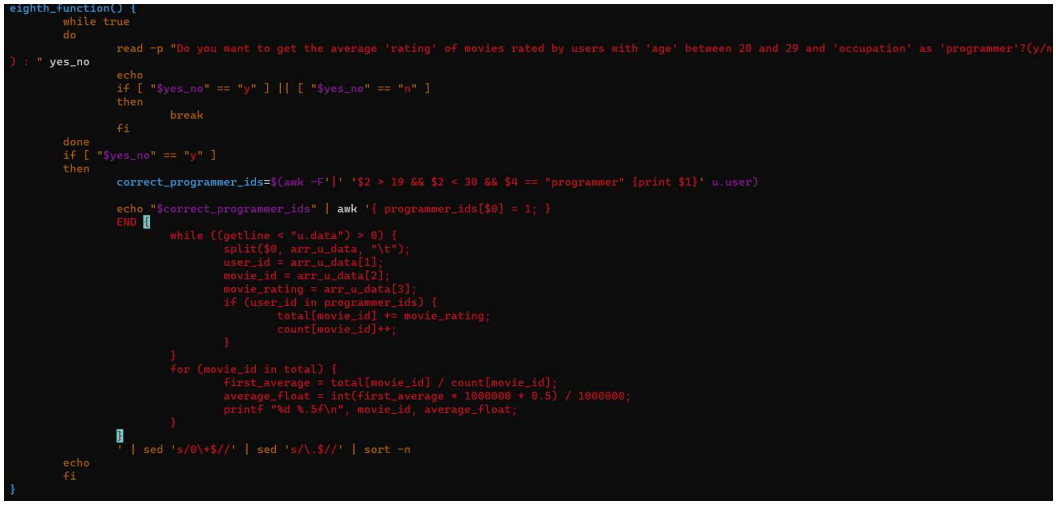
2>> while문 사용 read로 사용자에게 입력받아 user_number에 저장. user id가 1~943사이가 아니라면 계속 반복하게 구현. 만약 사이의 값을 입력했다면 출력.

입력받은 user id가 rating한 영화 id 출력을 먼저 한다. 이것도 앞에 코드와 같이 데이터 처리와 출력을 위해 awk 이용. -F를 이용해 이번에는 \t 로 구분하고 외부 변수를 사용하기 위해 -v를 이용. 입력받은 user_number을 user_num에 할당 |로 구분한 첫 번째 부분이 user_num과 같다면 영화 아이디를 출력.
sort: 정렬을 위한 명령어 -n: 오름차순으로 정렬
tr: 다른 문자로 변환하는 명령어. 이 코드에서는 \n을 | 로 변환. 그 뒤 sed를 이용해 마지막 | 제거. user_movie_id에 저장 후 인쇄

rating된 영화 id 10개를 'movie id' | 'movie title'로 출력. 이것은 위에 과정과 똑같이 진행하는데 sort로 정렬하고 head로 상위 10개만 top10_user_movie_id에 저장. 이것을 for문을 돌리면서 awk 이용 -F를 이용해 이번에는 |로 구분하고 외부 변수를 사용하기 위해 -v를 이용해 mvid에 할당. 이때 mvid가 첫 번째 부분과 같으면 영화 제목을 movie_title에 저장. 그 뒤에 "영화 id"|"영화 제목"형태로 echo로 인쇄

8) eigthth_function() 설명 -- [사용자가 8 입력]

1>> 20대 프로그래머가 rating한 영화 평균 등급을 원하나요? 라고 물어본다. 만약 n이라고 입력하면 다시 메뉴를 선택하게 하였고 y라면 출력 '영화id' '영화평균' 형태로 인쇄하는 기능



```
eighth_function() {
        while true
        do
                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'?(y/n
) : " yes_no
                echo
                if [ "$yes_no" == "y" ] || [ "$yes_no" == "n" ]
                then
                        break
                fi
        done
        if [ "$yes_no" == "y" ]
        then
                correct_programmer_ids=$(awk -F'|' '$2 > 19 && $2 < 30 && $4 == "programmer" {print $1}' u.user)

                echo "$correct_programmer_ids" | awk '{ programmer_ids[$0] = 1; }
                END {
                        while ((getline < "u.data") > 0) {
                                split($0, arr_u_data, "\t");
                                user_id = arr_u_data[1];
                                movie_id = arr_u_data[2];
                                movie_rating = arr_u_data[3];
                                if (user_id in programmer_ids) {
                                        total[movie_id] += movie_rating;
                                        count[movie_id]++;
                                }
                        }
                        for (movie_id in total) {
                                first_average = total[movie_id] / count[movie_id];
                                average_float = int(first_average * 1000000 + 0.5) / 1000000;
                                printf "%d %.5f\n", movie_id, average_float;
                        }
                }
                ' | sed 's/0\+$//' | sed 's/\.$//' | sort -n
        echo
        fi
}
```

2>> while 사용 read로 사용자에게 입력받아 yes_no에 저장. 사용자의 답이 y거나 n이면 break하고 다른 것이 입력되면 while 계속 작동되게 구현. 만약 사용자의 입력이 y면 출력
이 코드의 핵심은 20대 프로그래머를 u.user에서 뽑아내고 u.data에서는 앞서 뽑아낸 user id에 맞게 영화 rating을 뽑아내 영화 별로 평균 rating을 인쇄하게 해야한다.

데이터 처리와 출력을 위해 awk 이용. -F를 이용해 이번에는 | 로 구분하고 19보다 크고 30보다 작고 |로 구분한 4번째 부분이 programmer이면 user_id를 correct_programmer_ids에 저장. 다음 오는 코드는 영화 평점을 계산한다.
programmer_ids[$0]=1 이것은 앞에서 구한 correct_programmer_ids를 배열에 저장.
END의 블록 안에 있는 코드는 입력 파일의 모든 줄 실행한 뒤 실행하는 코드
getline을 이용해 u.data를 한줄 씩 읽어온다. split을 이용해 \t 기준으로 구분하고 arr_u_data 배열에 저장. 그리고 각 부분을 user_id, movie_id, movie_rating에 저장.
그리고 if를 이용해 programmer_ids 배열에 user_id가 존재하면 movie id 키마다 total에 movie_rating을 더한다. 또한 count에도 1을 반복한다.
for문으로 total 배열의 movie id를 movie_id(for에서 받는 변수)로 넘겨 받는다. 다음 나오는 코드는 3번 함수와 같으므로 생략. 마지막으로 sed를 이용해 불필요한 0과.을 삭제했다. 그 뒤에 sort로 정렬하였다.