오픈소스개론 2분반

- 프로젝트 1 -

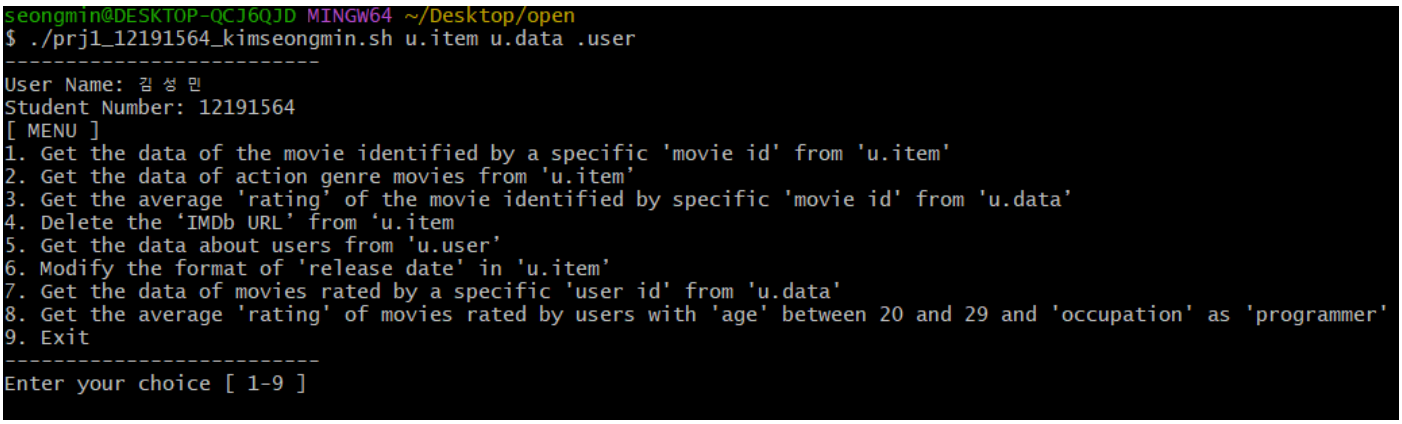
12191564
김성민

1. 개요

이 보고서는 오픈소스SW개론 강의의 중간 대체 과제인 OSS Project 1의 제출 shell script 파일 prj1_12191564_kimseongmin.sh의 사용 방법과 구현 내용을 설명하기 위한 보고서입니다.

2. 기본 사용법

아래 이미지와 같이 ./prj1_12191564_kimseongmin.sh으로 실행하는데 3개의 파일 경로들을 뒤에 파라미터로 적어 줘야 합니다. 순서는 영화목록(여기선 u.item), 평가목록(여기선 u.data), 유저목록(여기선 u.user) 순으로 입력하면 됩니다. (현재 아래 이미지는 3개의 데이터 파일이 모두 실행 파일과 같은 위치에 있습니다)



위와 같이 실행을 하게 되면 메뉴가 나온 후 "Enter your choice [1-9]" 라는 메시지가 나오게 되는데 1~9 사이에 번호를 입력하고 엔터를 치면 메뉴에서 해당 번호에 일치하는 명령을 수행합니다.

해당 번호와 일치하는 명령을 모두 수행 완료한 뒤엔 다시 "Enter your choice [1-9]" 라는 메시지를 출력하고 9번 명령인 Exit를 실행하기 전까지 반복합니다.

※ 각 명령들은 각각의 추가적인 입력을 필요로 합니다. 그러나 여기에 모두 적기 어려워 그 부분에 대한 설명은 더 아래 각 명령들에 대한 구현 방법과 실행 결과 부분에서 함께 설명하도록 하겠습니다.

3. 기본 구현 (함수 부분 제외)

아래 사진은 파라미터로 받아온 파일들의 경로를 변수에 저장한 코드입니다. shell script를 실행시킬 때 같이 적은 첫번째 파일 경로를 moviesFile에 담고 두번째 파일 경로를 ratingsFile에 넣으며 세번째 파일 경로를 usersFile에 넣습니다

```
#!/bin/sh

moviesFile=$1
ratingsFile=$2
usersFile=$3
```

아래 사진은 프로그램이 실행되고 최초의 "Enter your choice [1-9]" 메시지가 나오긴 전 출력되는 제 이름과 학번 그리고 메뉴가 출력되는 코드입니다. echo를 이용하여 각 문자열을 출력했습니다.

```
echo "--------------------------"
echo "User Name: 김성민"
echo "Student Number: 12191564"
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 '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 'occupation' as 'programmer'"
echo "9. Exit"
echo "--------------------------"
```

아래 사진은 메뉴 출력 후 "Enter your choice [1-9]" 메시지를 출력하고 해당 번호에 맞는 함수들을 실행하는 코드입니다. "while true"를 통해 무한 반복하도록 하였고, case문을 통해 해당 번호에 맞는 함수들을 실행하도록 했습니다. 또한 '9'를 입력 받을 시에는 "Bye!"를 출력한 뒤 break로 while문을 멈추고 종료하도록 하고 1~9가 아닌 값을 입력 받으면 "wrong order"를 출력한 뒤 다시 while문을 반복하도록 했습니다.

```
while true
do
  read -p "Enter your choice [ 1-9 ] " order
  echo ""
  case $order in
  1)
    getMovieInfo;;
  2)
    getActionMoives;;
  3)
    getAverageRating;;
  4)
    getMovieInfoDelURL;;
  5)
    getUserInfo;;
  6)
    getModifiedMovieInfo;;
  7)
    getRatedMoviesByUser;;
  8)
    getAverageRatingByProgrammer;;
  9)
    echo "Bye!"
    break;;
  *)
    echo "wrong order";;
  esac
  echo ""
done
```

4. 각 함수 부분 구현 및 실행 화면 (1~8 순서대로)

0) 공통설명

코드에서 echo "" 만 적힌 부분은 모두 개행을 위한 코드입니다

"y" or "n"를 대답으로 입력해야 하는 모든 함수는 answer가 "n"이면 함수를 바로 리턴하며, "y" 혹은 "n"이 아닌 다른 값을 입력하면 "wrong answer" 메세지와 함께 함수를 리턴합니다.

1) getMovieInfo

- 코드

```
7   getMovieInfo() {
8     read -p "Please enter 'movie id'(1~1682):" movieId
9     echo ""
10
11    cat $moviesFile | awk -F\| -v _movieId=$movieId '$1 == _movieId {print $0}'
12  }
```

- 실행결과

입력된 영화의 id와 일치하는 영화의 정보를 출력합니다.

- 사용법

이 함수가 실행되면 영화의 id값인 1~1682 사이에 값을 입력해주어야 합니다.

- 구현방법

read에서 입력 받은 값을 movieId에 저장합니다.

cat $moviesFile로 영화 정보가 담긴 파일을 열고 읽은 뒤에 pipe("|")를 통해 awk 쪽으로 넘겨줍니다.

awk는 v 옵션으로 입력 받은 movieId 값을 담는 _movieId를 만들고 F 옵션으로 영화 정보를 담고 있는 각각의 줄을 "|"을 기준으로 필드를 나눕니다. 나눈 필드에서 첫번째 필드($1)가 _movieId와 같은 값을 가진 줄(해당 영화의 모두 정보)의 내용을 모두($0) 출력합니다.

- 실행결과

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

2) getActionMoives

- 코드

```
14  getActionMoives() {
15    read -p "Do you want to get the data of 'action' genre movies from 'u.item'?(y/n):" answer
16    echo ""
17
18    if [ $answer = "y" ]
19    then
20      cat $moviesFile | awk -F\| '$7 == 1 {print $1, $2}' | head -n 10
21    elif [ $answer = "n" ]
22    then
23      return
24    else
25      echo "wrong answer"
26    fi
27  }
```

- 실행결과

  액션 장르 영화의 id와 title을 앞에서 10개까지 출력합니다.

- 사용법

  이 함수가 실행되면 정말 실행할 것이지 "y"(실행) 혹은 "n"(취소)을 입력해주어야 합니다.

- 구현방법

  read에서 입력 받은 값을 answer에 저장합니다.

  answer가 "y"인 경우, cat $moviesFile로 영화 정보가 담긴 파일을 열고 읽은 뒤에 pipe("|")를 통해 awk 쪽으로 넘겨줍니다.

  awk는 F 옵션으로 영화 정보를 담고 있는 각각의 줄을 "|"을 기준으로 필드를 나눕니다. 나눈 필드에서 일곱번째 필드($7)가 1(action 장르가 맞다)이면, 해당 줄의 첫번째 필드($1 = movie id)와 두번째 필드($2 = movie title)을 뽑고 pipe("|")를 통해 head 쪽으로 넘겨줍니다.

  head에서는 n 옵션에 10으로 인해 이전 결과(movie id와 movie title 모음)에서 앞에서 10개까지 출력합니다.

- 실행결과

```
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 ]
```

3) getAverageRating

- 코드

```
29  getAverageRating() {
30    read -p "Please enter the 'movie id'(1~1682):" movieId
31    echo ""
32
33    cat $ratingsFile \
34    | awk -v _movieId=$movieId '$2 == _movieId {sum+=$3; amount++}
35    END {if (amount != 0) average = sum / amount; printf("average rating of %s: %.5f\n", _movieId, average)}'
36  }
```

- 실행결과

  입력된 영화의 id와 일치하는 영화의 평균 rating을 출력합니다.

  소수점 5자리까지 출력됩니다.

- 사용법

  이 함수가 실행되면 영화의 id값인 1~1682 사이에 값을 입력해주어야 합니다.

- 구현방법

  read에서 입력 받은 값을 movieId에 저장합니다.

  cat $ratingFile로 평점 정보가 담긴 파일을 열고 읽은 뒤에 pipe("|")를 통해 awk 쪽으로 넘겨줍니다.

  awk는 v 옵션으로 아까 입력 받은 movieId 값을 담는 _movieId를 만들고 평점 정보를 담고 있는 각각의 줄을 공백을 기준으로 필드를 나눕니다. 나눈 필드에서 두번째 필드($2)가 _movieId와 같은 값을 가진 줄에 세번째 필드($3 = 평점)을 sum에 더하고 amount 값을 1증가시킵니다. 모든 줄을 확인했다면 END로 가서 if문을 확인해 amount가 0이 아니라면 sum/amount로 average(평균)을 계산해줍니다. 이후 printf에서 "average rating of {_movieId(영화 아이디)}: {average(계산한 평점, 소수점 5자리까지)}"를 출력해 줍니다.

- 실행결과

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

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

average rating of 1: 3.87832

Enter your choice [ 1-9 ]
```

4) getMovieInfoDelURL

- 코드

```
38  getMovieInfoDelURL() {
39    read -p "Do you want to delete the 'IMDb URL' from 'u.item'?(y/n):" answer
40    echo ""
41
42    if [ $answer = "y" ]
43    then
44      cat $moviesFile | head -n 10 | sed 's/[^|]*|/|/5'
45    elif [ $answer = "n" ]
46    then
47      return
48    else
49      echo "wrong answer"
50    fi
51  }
```

- 실행결과

영화의 정보에서 IMDb URL 부분을 제외한 내용을 앞에서 10개까지 출력합니다.

- 사용법

이 함수가 실행되면 정말 실행할 것이지 "y"(실행) 혹은 "n"(취소)을 입력해주어야 합니다.

- 구현방법

read에서 입력 받은 값을 answer에 저장합니다.

answer가 "y"인 경우, cat $moviesFile로 영화 정보가 담긴 파일을 열고 읽은 뒤에 pipe("|")를 통해 head 쪽으로 넘겨줍니다.

head는 n 옵션 10으로 인해 영화 정보를 앞에서 10개 까지만 뽑아 pipe("|")를 통해 sed쪽으로 넘겨줍니다

sed는 맨 앞 s명령을 통해 문장을 교체합니다. 교체 대상은 "|"로 시작하지 않고 "|"로 끝나는("[^|]*|") 5번째 요소(마지막 /5)로 "|" 앞에 내용은 모두 없어지도록 "|"만 있도록 교체한 뒤 출력합니다.

- 실행결과

```
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 ]
```

5) getUserInfo

- 코드

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

  if [ $answer = "y" ]
  then
    cat $usersFile | head -n 10 \
    | sed -e 's/|M|/|male|/' -e 's/|F|/|female|/' \
    -Ee 's/([^|]*)\|([^|]*)\|([^|]*)\|([^|]*)\|([^|]*)/user \1 is \2 years old \3 \4/'
  elif [ $answer = "n" ]
  then
    return
  else echo "wrong answer"
  fi
}
```

- 실행결과

유저 정보를 "1|24|M|technician|85711" -> "user 1 is 24 years old male technician"처럼 형식을 바꿔서 앞에서 10명까지 출력합니다.

- 사용법

이 함수가 실행되면 정말 실행할 것이지 "y"(실행) 혹은 "n"(취소)을 입력해주어야 합니다.

- 구현방법

read에서 입력 받은 값을 answer에 저장합니다.

answer가 "y"인 경우, cat $usersFile로 유저 정보가 담긴 파일을 열고 읽은 뒤에 pipe("|")를 통해 head 쪽으로 넘겨줍니다.

head는 n 옵션 10으로 인해 유저 정보를 앞에서 10개 까지만 뽑아 pipe("|")를 통해 sed쪽으로 넘겨줍니다

sed는 e 옵션으로 여러 번 시행됩니다(여기서는 3번 시행합니다). E 옵션은 기본보다 더 확장된 regular expression을 제공합니다.

첫번째는 맨 앞 s명령을 통해 문장을 교체합니다. 교체 대상은 "|M|"으로 "|male|"로 교체합니다.

두번째도 맨 앞 s명령을 통해 문장을 교체합니다. 교체 대상은 "|F|"로 "|female|"로 교체합니다.

마지막 세번째도 맨 앞 s명령을 통해 문장을 교체합니다. 문장 전체를 "|"로 시작하지 않는 녀석들("([^|]*)")을 "|"("₩|")을 구분으로 5개의 그룹을 나눕니다. 이후 "₩숫자"를 통해 그룹 번호에 맞는 내용을 이용할 수 있습니다(현재 함수에서 5개의 그룹이 있으므로 그룹 번호는 1~5까지 있습니다). 이를 통해 최종적으로 문장을 "user {첫번째 그룹(유저의 id)} is {두번째 그룹(유저의 나이)} years old {세번째 그룹(유저 성별, 아까 바꾼 male, female)} {네번째 그룹(유저의 직업군)}"으로 바꿔서 출력합니다.

- 실행결과

```
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 ]
```

6) getModifiedMovieInfo

- 코드

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

  if [ $answer = "y" ]
  then
    cat $moviesFile | tail -n 10 \
    | sed -Ee 's/\|([0-9]{2})-Jan-([0-9]{4})\|/|\201\1|/' \
    -Ee 's/\|([0-9]{2})-Feb-([0-9]{4})\|/|\202\1|/' \
    -Ee 's/\|([0-9]{2})-Mar-([0-9]{4})\|/|\203\1|/' \
    -Ee 's/\|([0-9]{2})-Apr-([0-9]{4})\|/|\204\1|/' \
    -Ee 's/\|([0-9]{2})-May-([0-9]{4})\|/|\205\1|/' \
    -Ee 's/\|([0-9]{2})-Jun-([0-9]{4})\|/|\206\1|/' \
    -Ee 's/\|([0-9]{2})-Jul-([0-9]{4})\|/|\207\1|/' \
    -Ee 's/\|([0-9]{2})-Aug-([0-9]{4})\|/|\208\1|/' \
    -Ee 's/\|([0-9]{2})-Sep-([0-9]{4})\|/|\209\1|/' \
    -Ee 's/\|([0-9]{2})-Oct-([0-9]{4})\|/|\210\1|/' \
    -Ee 's/\|([0-9]{2})-Nov-([0-9]{4})\|/|\211\1|/' \
    -Ee 's/\|([0-9]{2})-Dec-([0-9]{4})\|/|\212\1|/'
  elif [ $answer = "n" ]
  then
    return
  else
    echo "wrong answer"
  fi
}
```

- 실행결과

  영화 정보의 날짜를 "01-Jan-1995" -> "19950101"처럼 형식을 바꿔서 뒤에서 10개까지 출력합니다.

- 사용법

  이 함수가 실행되면 정말 실행할 것이지 "y"(실행) 혹은 "n"(취소)을 입력해주어야 합니다.

- 구현방법

  read에서 입력 받은 값을 answer에 저장합니다.

  answer가 "y"인 경우, cat $moviesFile로 영화 정보가 담긴 파일을 열고 읽은 뒤에 pipe("|")를 통해 tail 쪽으로 넘겨줍니다.

  tail은 n 옵션 10으로 인해 영화 정보를 뒤에서 10개 까지만 뽑아 pipe("|")를 통해 sed쪽으로 넘겨줍니다

  sed는 e 옵션으로 월의 정보를 각 월의 맞는 숫자로 바꾸기 위해 12번 시행합니다.

  각각의 sed 실행은 s명령을 통해 문장을 교체합니다. 교체 대상은 "|{두자리 숫자(일정보)}-{월정보 문자}-{네자리 숫자(년정보)}"으로 일정보와 년정보는 ()를 통해 그룹으로 만들어줍니다. 이를 통해 최종적으로 목표 문장을 "|{두번째 그룹(년정보)}{각 월에 맞는 월 숫자}{첫번째 그룹(일정보)}|"으로 바꾼 뒤에 전체 문장을 출력합니다.

- 실행결과

```
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|1|0|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 ]
```

7) getRatedMoviesByUser

- 코드

```
96  getRatedMoviesByUser() {
97    read -p "Please enter the 'user id'(1~943):" userId
98    echo ""
99
100   sortedRatedMovies=$(cat $ratingsFile | awk -v _userId=$userId '$1 == _userId {print $2}' | sort -n)
101   echo $sortedRatedMovies | tr ' ' '|' | sed 's/|$/\n/'
102   echo ""
103
104   n=1
105   while [ $n -le 10 ]
106   do
107     movieId=$(echo $sortedRatedMovies | awk -v _n=$n '{print $_n}')
108     cat $moviesFile | awk -F\| -v _movieId=$movieId '$1 == _movieId {print $1 "|" $2}'
109     let n=n+1
110   done
111 }
```

- 실행결과

입력된 유저의 id와 일치하는 유저가 평가한 영화들의 id를 출력하고 해당 영화들의 id와 title을 앞에서 10개까지 출력합니다.

- 사용법

이 함수가 실행되면 유저의 id값인 1~943 사이에 값을 입력해주어야 합니다.

- 구현방법

read에서 입력 받은 값을 userId에 저장합니다.

cat $ratingsFile로 평점 정보가 담긴 파일을 열고 읽은 뒤에 pipe("|")를 통해 awk 쪽으로 넘겨줍니다.

awk는 v 옵션으로 입력 받은 userId 값을 담는 _userId를 만들고 띄어쓰기로 나눠진 평점 정보의 첫번째 필드($1 평점 준 유저 id)가 _userId와 같은 평점 정보의 두번째 필드(평점을 준 영화의 id)를 뽑아서 pipe("|")를 통해 sort쪽으로 넘겨줍니다.

sort는 n 옵션으로 인해 숫자 순서대로 정렬을 해주고 그 값들은 sortedRatedMovies에 저장됩니다.

① 첫번째 출력

echo $sortedRatedMovies로 값을 꺼낸 뒤 pipe("|")를 통해 tr쪽으로 넘겨줍니다.

tr은 영화 id들이 띄어쓰기로 구분 되어있는 것을 "|"으로 구분되도록 바꾼뒤 pipe("|")를 통해 sed쪽으로 넘겨줍니다. 이후 sed는 s명령으로 문장을 바꾸는데 "|$"으로 마지막 "|"을 개행("₩n")으로 바꾸고 나서 출력합니다.

② 두번째 출력

while문을 통해 n이 1부터 10("[$n -le 10]", 아래 let n=n+1로 반복할 때마다 1씩 증가)이 될 때까지 수행합니다.

echo $sortedRatedMovies로 값을 꺼낸 뒤 pipe("|")를 통해 awk쪽으로 넘겨줍니다.

awk는 띄어쓰기로 구분하여 평가한 영화의 n번째 id를 꺼내고 movieId에 저장합니다.

이후 cat $moviesFile로 영화 정보가 담긴 파일을 열고 읽은 뒤에 뒤에 pipe("|")를 통해 awk 쪽으로 넘겨줍니다.

awk는 v 옵션으로 저장한 movieId 값을 담는 _movieId를 만들고 F 옵션으로 영화 정보를 담고 있는 각각의 줄을 "|"을 기준으로 필드를 나눕니다. 나눈 필드에서 첫번째 필드($1)가 _movieId와 같은 값을 가진 줄의 정보를 이용해 "{첫번째 필드(영화의 id)}|{두번째 필드(영화의 title)}" 형식으로 출력합니다. (while문으로 인해 10번째까지만 출력됩니다.)

- 실행결과

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

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|19
5|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 ]
```

8) getAverageRatingByProgrammer

- 코드

```
getAverageRatingByProgrammer() {
  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):" answer
  echo ""

  if [ $answer = "y" ]
  then
    programmerUserIds=$(cat $usersFile | awk -F\| '$2 >= 20 && $2 < 30 && $4 == "programmer" {print $1}')

    ratings=""
    for userId in $programmerUserIds
    do
      ratings+=$(cat $ratingsFile | awk -v _userId=$userId '$1 == _userId {print $2 "|" $3}')
      ratings+=' '
    done
    movieId=1
    movieAmount=$(cat $moviesFile | wc -l)
    while [ $movieId -le $movieAmount ]
    do
      echo $ratings | tr ' ' '\n' \
      | awk -v _movieId=$movieId -F\| '$1 == _movieId {sum+=$2; amount++}
      END {if (amount != 0) printf("%s %.5f\n", _movieId, sum / amount)}' \
      | sed -e 's/[0]*$//' -e 's/\.$//'

      let movieId=movieId+1
    done
  elif [ $answer = "n" ]
  then
    return
  else echo "wrong answer"
  fi
}
```

- 실행결과

  20~29세의 프로그래머들이 평가한 영화들의 id와 평균 rating을 출력합니다.

  소수점 5자리까지 출력됩니다.

  평가되지 않은 영화들은 출력되지 않습니다.

- 사용법

  이 함수가 실행되면 정말 실행할 것이지 "y"(실행) 혹은 "n"(취소)을 입력해주어야 합니다.

- 구현방법

  read에서 입력 받은 값을 answer에 저장합니다.

  answer가 "y"인 경우, cat $usersFile로 유저 정보가 담긴 파일을 열고 읽은 뒤에 pipe("|")를 통해 awk쪽으로 넘겨줍니다.

  awk은 F 옵션으로 유저 정보를 담고 있는 각각의 줄을 "|"을 기준으로 필드를 나눕니다. 나눈 필드에서 두번째 필드($2) 나이가 20이상 29미만이면, 해당 줄의 첫번째 필드($1) 유저 id를 뽑아 그것들을 programmerUserIds에 저장합니다.

  ratings이라는 빈 문자열을 만들어 줍니다. 이후 "for userId in $programmerUserIds"을 통해 programmerUserIds에 담겨있는 유저 id를 하나씩 꺼내 userId에 담고 반복문을 실행합니다.

  cat $ratingsFile로 평점 정보가 담긴 파일을 열고 읽은 뒤 pipe("|")를 통해 awk쪽으로 넘겨줍니다.

  awk는 v 옵션으로 하나씩 꺼낸 userId 값을 담는 _userId를 만들고 띄어쓰기로 나눠진 평점 정보의 첫번째 필드($1 평점 준 유저 id)가 _userId와 같은 평점 정보를 이용해 "{두번째 필드(평점을 준 영화의 id)}|{세번째 필드(평점)}"형식으로 뽑아서 ratings 문자열에 추가해줍니다. (마지막 요소는 띄어쓰기가 들어가지 않아 반복문 마지막마다 rating 문자열에 " "을 추가해 줍니다)

  while문을 통해 movieId가 1부터 영화 정보 수("movieAmount=$(cat $moviesFile | wc -l)", wc -l을 통해 moviesFile이 몇 줄이진 알 수 있습니다)만큼 ("[$movieId -le $movieAmount]") 반복해 수행합니다. (여기선 1682줄이라 1682번 반복합니다)

  echo $ratings으로 영화 id와 평점 정보가 담긴 문자열을 읽은 뒤에 pipe("|")를 통해 tr쪽으로 넘겨줍니다.

  tr은 띄어쓰기(" ")로 구분된 정보를 개행으로("₩n") 구분하도록 바꾼 뒤 pipe("|")를 통해 awk쪽으로 넘겨줍니다.

awk는 v 옵션으로 1부터 영화 정보 수의 값을 가지는 movieId 값을 담는 _movieId를 만들고 영화id와 평점 정보를 담고 있는 각각의 줄을 "|"을 기준으로 필드를 나눕니다. 나눈 필드에서 첫번째 필드($1) 영화 id가 _movieId와 같은 값을 가진 줄에 두번째 필드($2) 평점을 sum에 더하고 amount 값을 1증가시킵니다. 모든 줄을 확인했다면 END로 가서 if문을 확인해 amount가 0이 아니라면(amount가 0이면 출력하지 않습니다) printf에서 "{_movieId(영화 아이디)} {sum / amount(평점 평균, 소수점 5자리까지)}"형식으로 만든 뒤 pipe("|")를 통해 sed쪽으로 넘겨줍니다.

sed는 s명령을 통해 문장을 수정합니다. 목표는 각 행 별로 소수점 아랫부분에 끝부분이 0인 부분을 전부("'s/[0]*$//'") 지우고나서 그 뒤에 소수점 아래가 모두 0이라서 지워진 뒤 마지막이 "."이라면("'s/₩.$//'") 그것도 지워주는 수정을 한 뒤에 출력합니다.

- 실행결과

```
Enter your choice [ 1-9 ] 8

Do you want to get the average 'rating' of movies rated by users with 'age' betwee
n 20 and 29 and 'occupation' as 'programmer'?(y/n):y

1 4.29412
2 3
3 3.5
4 3.7
5 3.25
7 4.22222
8 3.5
9 4.1
10 4
11 4.3125
12 4.69231
13 3.375
14 4
15 3.85714
16 3
17 3.5
19 4
20 1
21 2.8
```

(중간은 내용이 많아 생략했습니다)

```
1438 4
1439 4
1440 2.5
1443 5
1446 3
1456 4
1478 3
1480 1
1483 5
1485 3
1487 2
1491 1
1509 1
1512 3
1513 2
1518 4
1531 3
1552 2
1597 1
1600 4
1621 1
1655 2

Enter your choice [ 1-9 ]
```

9) exit (함수는 아니지만 설명을 위해 추가했습니다)

- 코드

```
160  while true
161  do
162    read -p "Enter your choice [ 1-9 ] " order
163    echo ""
164    case $order in
165    1)
166      getMovieInfo;;
167    2)
168      getActionMoives;;
169    3)
170      getAverageRating;;
171    4)
172      getMovieInfoDelURL;;
173    5)
174      getUserInfo;;
175    6)
176      getModifiedMovieInfo;;
177    7)
178      getRatedMoviesByUser;;
179    8)
180      getAverageRatingByProgrammer;;
181    9)
182      echo "Bye!"
183      break;;
184    *)
185      echo "wrong order";;
186    esac
187    echo ""
188  done
189
```

- 실행결과

  Bye!를 출력 후 while문을 탈출합니다.

- 사용법

  처음 메뉴를 선택할 때 9를 입력하면 실행합니다.

- 구현방법

  case문에 따라 order의 값이 9이면 "echo "Bye!""를 통해 "Bye!"를 출력하고 break를 실행 해 while문을 탈출합니다.

- 실행결과

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

Bye!

seongmin@DESKTOP-QCJ6QJD MINGW64 ~/Desktop/open
$
```