오픈소스SW개론 OSS Project1 README

12183475 김지환

쉘 스크립트 파일 명: prj1_12183475_kimjihwan.sh

-메뉴 번호 입력 ,루프 방식 : stop변수선언, until문을 이용해 stop이 N이 될 경우 until문이 탈출되도록 구현

```
echo "Student Number: 12183475"
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'"
echo "9. Exit"
echo "--------------------------"

stop="N"
until [ $stop = "Y" ]
do
    read -p "Enter your choice [ 1-9 ] " choice
    case $choice in
```

```
        *)  echo "Error: Invalid option..."
            ;;
    esac
done
```

각 메뉴 설명

1. 특정 movie id를 가진 영화의 정보 출력

read문을 이용해 id변수에 값을 입력->

이후 u.item의 내용을 awk문에 입력해 -F옵션으로 |(버티컬바)를 기준으로 각 칼럼을 나눈이후 영화의 id에 해당하는 $1이 일치하는 행을 출력하도록 구현($0은 해당 행 전체 출력)

```
1)  echo""
    read -p "Please enter 'movie id' (1~1682):" id
    echo""
    cat u.item | awk -v id=$id -F '|' '$1==id {print $0}'
    echo""
    ;;
```

2. action 장르의 영화의 목록 출력 (id순 상위 10개만)

ans변수에 대한 입력이 y일 경우 진행,

u.item의 내용을 awk문에 입력해 -F옵션으로 |(버티컬바)를 기준으로 각 칼럼 분할, 이후 액션 장르에 해당하는 $7이 1일 경우 해당 행의 $1(movie id),$2(movie title) 출력(head를 이용해 상위 10개만)

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

3. 특정 id의 영화의 평점 출력

영화의 id 입력-> awk문을 이용해 u.data에서 해당 movie id의 영화에 대한 평점만 출력($3은 점수에 해당하는 열)->awk문을 이용해 sum변수에 각 행의 수 합산->END에서 NR(행 수)로 나눠줌

```
3)  echo""
    sum=0
    read -p "Please enter 'movie id' (1~1682):" id
    echo""
    cat u.data | awk -v id=$id '$2==id {print $3}' | awk -v sum=$sum '{sum+=$1} END {print sum/NR}'
    echo""
    ;;
```

4. Movie id 순으로 상위 10개의 영화 정보 출력, IMDb URL은 제외하고 출력

u.item의 상위 10개 행에대해서 sed문을 s커맨드를 이용해 http로 시작하고, 이후 |가 아닌 문자열까지 선택, 이후 치환할 문자열은 비워놓아서 삭제시킴.

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

5. 유저 id순으로 상위 10명의 유저에 대한 정보 출력

u.user의 상위 10개행에 대해 |를 기준으로 나눠, {"user " $1 " is " $2 " years old " $3 " " $4} 형식으로 출력($1:id $2:나이 $3:성별 $4:직업)->이후 sed를 이용해 M의 경우 male, F의 경우 female로 치환해 출력

```
5)  echo""
    read -p "Do you want to get the data about users from 'u.user'?(y/n):" ans
    echo""
    if [ "$ans" = "y" ]
        then
            cat u.user | awk -F '|' '{print "user " $1 " is " $2 " years old " $3 " " $4 }' | head | sed -E 's/M/male/g' | sed -E 's/F/female/g'
            echo""
    fi
    ;;
```

6. Movie id순으로 하위 10개의 영화 정보 출력(release date를 dd-mmm-yyyy 에서 yyyymmdd로 변환해 출력, 달의 경우 영어에서 수로 변환)

u.item에서 하위 10개의 행 출력->이후 sed를 이용해 각 행의 달을 수로 치환(-e옵션으로 여러 개의 변환기준 설정) -> 이후 sed의 s커맨드를 이용, 길이2의 숫자(₩1)-길이2의 숫자(₩2)-길이4의 숫자(₩3) 형태로 이루어진 문자열을 타겟으로 설정한 이후, ₩3₩2₩1의 형태로 순서 바꾸기 및 출력

```
if [ "$ans" = "y" ]
    then
        cat u.item | tail | sed -e 's/Jan/01/' -e 's/Feb/02/' -e 's/Mar/03/' -e 's/Apr/04/'
```

```
-e 's/May/05/' -e 's/Jun/06/' -e 's/Jul/07/' -e 's/Aug/08/' -e 's/Sep/09/'
```

```
-e 's/Oct/10/' -e 's/Nov/11/' -e 's/Dec/12/' | sed 's/\([0-9]\{2\}\)-\([0-9]\{2\}\)-\([0-9]\{4\}\)/\3\2\1/g'
```

7. 특정 유저id의 유저가 평가한 영화들 movie id 및 title 출력

유저 id 입력->awk를 이용해 u.data에서 입력받은 id와 $1(유저 id)가 같은 행의 $2(movie id)만 출력->이후 sort -n을 이용해 각 행을 오름차순으로 정렬(-n옵션을 주지 않을 경우 첫글자 기준으로만 정렬되어 정렬이 돼지 않음.)

위 작업에 대해 하나의 경우 출력, 하나의 경우 파일로 저장.

출력의 경우 id|id|id|...형태로 출력을 해야 하기에 tr을 이용해 개행문자를 |로 치환해 줄바꿈을 없엔 이후에 출력(마지막 버티컬 바를 sed를 이용해 삭제)

파일에 저장하는 경우 상위 10개의 행만을 head를 이용해 자른 후 tmp파일(임시)에 redirection 해 저장해줌.

이후 tmp파일의 각 행을 while문으로 읽어옴(각 행은 mid변수에 read해옴.) 이후 awk를 이용해 u.item에서 mid와 일치하는 행만을 찾아 낸 이후 $1 "|" $2 형식으로 출력($1:movie id $2:title)

출력이 끝난 이후 tmp파일 삭제.

```
7)  echo""
    read -p "Please enter 'user id' (1~943):" id
    echo""
    cat u.data | awk -v id=$id '$1==id {print $2}' | sort -n | tr '\n' '|' | sed 's/|$//'
    cat u.data | awk -v id=$id '$1==id {print $2}' | sort -n  | head > tmp
    echo ""
    echo ""
    while read mid; do
        cat u.item | awk -F '|' -v mid=$mid '$1==mid {print $1 "|" $2}'
    done < tmp
    rm tmp
    echo""
    ;;
```

8. 20~29세의 프로그래머들의 영화 평가에 따른 영화의 평점 출력

우선, 크게 보면 20~29세의 프로그래머들의 user id를 추출 한 이후, 영화 평점이 저장된 파일에서 해당 user id가 평가한 점수를 가진 행만을 남긴다. 이렇게 하면 조건에 맞는 평점 데이터만 기록된다. 이후 해당 데이터에서 movie id만을 추출해서 중복을 제거하고 오름차순으로 정렬해준다. 여기까지 해서 얻은 데이터는 조건에 맞는 평점기록, 그리고 해당 평점기록들에 소속 된 movie id들의 리스트이다. 이제 리스트에 있는 movie id들의 각각의 평점을 조건에 맞는 평점기록을 참조해서 계산해낸다.

각 과정의 구현을 살펴보겠다. 우선 u.user로부터 awk문을 이용해 $2(나이)가 20~29, $4(직업)가 programmer인 행의 $1(user id)만을 출력해 tmp1임시파일에 저장한다.

이후 while문을 이용해 tmp1의 각 행(user id)을 id에 read, u.data에서 id가 평가 한 내용을 담은 행을 임시파일 tmp2에 append해준다. ->tmp2는 20~29세 프로그래머의 영화 평가만 저장

tmp2의 내용에서 awk를 이용해 각 행의 $2(영화 id) 만을 tmp3에 저장->tmp3에는 20~29세 프로그래머의 영화 평가에 포함 된 영화들의 movie id가 정렬되지 않은 채, 중복된 채로 저장 되어 있음.

Sort -u -n을 이용해 중복을 제거하고 정렬 된 movie id 리스트를 tmp4에 저장.(-u는 중복 제거, -n은 숫자 기준 정렬)

이후 while문으로 tmp4의 각 행을 mid변수에 read해줌->각 루프내부에서 0으로 초기화 된 sum 변수를 선언해 tmp2의 행들 중 $2(movie id)와 mid변수의 값이 일치 하는 경우의 행의 $3(점수) 출력->출력에 대해 awk를 이용해 sum에 각 행의 값(평점)을 sum에 더한 이후 END에서 NR로 나눈 값 출력(movie id는 루프의 시작에서 먼저 출력)

이후 tmp1,tmp2,tmp3,tmp4 임시파일 삭제

```
8)  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'?(y/n):" ans
    echo""
    if [ "$ans" = "y" ]
        then
            # save the people's id who's matched with condition
            cat u.user | awk -F '|' '$2>=20 && $2 <=29 && $4=="programmer" {print $1}' > tmp1
            # save the u.data's rows that has id on tmp1 file.
            while read id; do
                cat u.data | awk -v id=$id '$1==id {print $0}' >>tmp2
            done < tmp1
            #tmp2 file has a row of u.data which matched with condition
            #so we should get moive id list from tmp2, and delete the duplication
            cat tmp2 | awk '{print $2}' > tmp3
            sort -u -n tmp3 >tmp4
            #tmp4 has a sorted, not duplicated list of movies matched with condiiton,
            #so wh should read tmp4 line by line and get rating on tmp2
            while read mid; do
                sum=0
                echo -n "$mid "
                cat tmp2 | awk -v mid=$mid '$2==mid {print $3}' | awk -v sum=$sum '{sum+=$1} END {print " " sum/NR}'
            done < tmp4
            rm tmp1 tmp2 tmp3 tmp4
            echo""
    fi
    ;;
```

9. Exit

Bye를 출력 한 이후 stop변수에 Y 할당->until문의 탈출 조건 만족->프로그램 종료

느낀 점

주로 C,C++을 이용해 코딩을 하다가 이번과제에서 처음으로 쉘 스크립트를 이용해 프로그램을 만들어보게 되었습니다. 명령어도 익숙하지 않고, 특히나 배열 등을 활용 할 수 없다 보니 정말 힘들었습니다. 또한 변수를 사용할 때 $를 붙여야 하는 것도 자꾸 까먹게 되고, sed와 awk명령어는 아직도 숙달이 덜 된 것 같습니다. 하지만 이번 과제를 통해 파일 리디렉션등을 이용해 임시 파일을 만드는 것, 그 임시 파일을 읽어와서 데이터를 처리하는 것, 또 다양한 방법으로 로우칼럼 형 데이터를 다루는 방법을 접해볼 수 있어서 좋은 시간이었습니다.