12162114 류윤선 bash shell script 설명 보고서

bash shell을 이용한 프로그램이므로 맨 위쪽에 #!/bin/bash 을 적는다. 해당 프로그램은 1~9까지 총 9가지 기능이 있다. 이 프로그램을 실행시켰을 때 맨 먼저 보이는 화면은 User name과 Student Number이다. 여기에 나의 이름과 학번을 기입하였다. 그리고 9가지 기능을 보여주는 MENU가 나온다. 그러고 나면 “Enter your choice [ 1-9 ]” 에 원하는 기능의 번호를 입력 한다. 이 때, echo -n을 썼는데 그 이유는 질문에 대해 줄 바뀜이 되지 않은 채로 번호를 입력하기 위해서이다. 이 외에도 계속 echo -n을 쓰는데, 이와 같은 이유이므로 뒤에 나오는 echo -n에 대한 설명은 생략하겠다.

첫 번째 기능은 movie id를 입력 받으면 해당 영화의 정보를 알려주는 기능이다. read num1으로 입력을 받고 해당 번호와 일치하는 레코드 번호(NR)을 awk를 이용하여 찾아 해당 레코드를 출력하면 해당 영화정보가 출력되는 것이다. -> awk 'NR == '"$num1"' { print $0 }' u.item

두 번째 기능은 액션 장르의 영화 정보를 출력하는 것이다. 이 때 조건은 앞쪽 순서부터 10개만 출력하고, 영화 아이디와 영화 제목만 출력하는 것이다. 2번 째 기능을 입력해보면 맨 먼저 "Do you want to get the data of 'action' genre movies from 'u.item'? (y/n) : " 라는 질문이 나온다. read yesORno로 대답을 받는다. 이 때 y, yes, Y, YES를 치면 그 다음 과정으로 넘어갈 수 있게 설정하였다. 이 외에도 여러 기능에서 y/n을 물어보는데, 이와 중복되므로 뒤에 나오는 y/n에 대해서는 설명을 생략하겠다.

액션 영화를 찾기 위해서 awk -F를 사용하여 ‘|’ 를 구분자로 지정해주고, if문을 써서 액션영화가 맞는 지를 찾는다. u.item 파일의 일곱 번째 필드가 액션 장르 유무 정보가 담긴 필드이므로 일곱 번째 필드가 1인 지를 살핀다. -> if($7 == “1”) 맞다면 해당 레코드 중 첫 번째 필드와 두 번째 필드(영화 아이디와 영화 제목)만을 출력한다. 이 때, 10개만 출력을 해야 하므로 맨 앞에 if문을 하나 더 추가하여 -> if(i==10) { exit }; 영화 정보가 10개가 넘어가면 종료되도록 한다.

세 번째 기능은 movie id를 입력 받으면 해당 영화가 평가받은 모든 점수를 합쳐 평균 낸 결과 값을 소수점 다섯째자리까지 출력하는 것이다. u.data 파일의 두 번째 필드가 movie id 정보이니 두 번째 필드가 입력 받은 movie id와 일치하면 세 번째 필드 정보(점수)를 sum 변수에 넣는다. 그리고 나중에 평균을 내기 위해서는 총 개수를 따져줘야 하므로 count++ 도 같이 해준다. -> { if($2 == '"$num3"') { count++; sum+=$3 } } 모든 레코드에 대한 탐색이 끝나면 마지막에 sum과 count를 나누어 평균 점수를 구하고 이를 출력해주면 된다. -> END { print sum/count }

네 번째 기능은 u.item 파일에서 IMDb URL 기능을 삭제하는 기능이다. 이 때 조건은 10개만 출력하고, 삭제한 IMDb URL 부분은 비어 있어야 한다. IMDb URL은 u.item 파일의 다섯 번째 필드에 정보가 있다. 따라서 삭제를 하기 위해서는 다섯 번째 필드를 “”로 바꿔주어 아무것도 없게 한다. -> $5="" 그런데 이 때 $5="”를 해주고 해당 레코드를 출력하면 구분자 ‘|’가 없어진 채로 출력이 된다. 따라서 각각의 필드 뒤에 “ | “ 를 덧붙여주어 출력을 하면 필드 사이 사이에 구분자 ‘|’가 있는 채로 출력이 된다.

다섯 번째 기능은 user id와 나이, 성별, 직업을 출력하는 기능이다. 이 때 조건은 정해진 포맷에 맞게 10개만 출력하는 것이다. 해당 정보들은 u.user 파일에 있다. user id, 나이, 성별, 직업은 u.user 파일의 첫 번째, 두 번째, 세 번째, 네 번째 필드에 정보가 있다. if문을 통해 세 번째 필드가 M인지 F인지 확인한다. M이면 if($3=="M") { print "user " $1 " is " $2 " years old male " $4} 와 같이 출력을 해주고, F이면 if($3=="F") { print "user " $1 " is " $2 " years old female " $4 } 와 같이 출력을 해준다. 그리고 10개만 출력해야 하므로 NR < 11을 써준다.

여섯 번째 기능은 u.item 파일의 세 번째 필드인 release date의 표기 형식을 수정하는 기능이다. 이 때 조건은 뒤에서 10개의 영화 정보만 출력하는 것이다. 우선 awk -F로 구분자를 ‘|’와 ‘-‘로 두 개 지정하였다. -> awk -F '[|-]' 첫 번째 필드와 두 번째 필드 정보에는 ‘-‘가 없으므로 세 번째 필드부터 ‘-‘로 필드가 나누어진다. 즉, 세 번째 필드 내부의 서브필드를 구하기 위한 셈이다. 따라서 세 번째 필드는 며칠인지 알려주는 정보가 되고, 네 번째 필드는 몇 월인지 알려주는 정보가 되고, 다섯 번째 필드는 연도를 알려주는 정보가 된다. 그러나 이때 네 번째 필드는 숫자가 아닌 영문으로 적혀있다. 따라서 1월(Jan)이면 01로 바꿔주고, YYYYMMDD 순이어야 하니 다섯 번째 필드 네 번째 필드 세 번째 필드 순으로 세 개의 필드를 합쳐주고 출력한다. -> if($4=="Jan") { $4=($5 "01" $3); $3=$5=""; print $0 } 이와 마찬가지로 2월부터 12월까지 모든 월에 대해서도 똑같이 해준다. 그리고 뒤쪽 순서부터 10개만 출력이므로 영화는 1682개까지 있으니 NR>=1673 && NR<=1682를 써준다.

일곱 번째 기능은 user id를 입력 받고 해당 user가 평가한 모든 영화들을 정해진 포맷에 맞게 오름차순으로 출력하는 기능이다. 그리고나서 평가한 영화 아이디와 영화 제목들을 오름차순으로 열개만 출력하는 기능도 있다. awk를 이용하여 입력 받은 user를 찾고 해당 user가 평가한 모든 영화들을 u.data에서 찾아 배열에 저장한다.