OSS Project 1

12191673 정재홍

개요

"prj1_12191673_jungjaehong.sh"은 BASH를 사용하여 작성된 스크립트이다. u.item, u.data, u.user 파일에 대한 9가지 기능을 지원하며 **"./prj1_12191673_jungjaehong.sh u.item u.data u.user"** 명령어를 통해 실행이 가능하다.

기능 설명

1. 'movie id'를 입력 받아 u.item 파일에서 해당하는 영화의 정보를 출력
   - read 명령어로 입력을 받고 movie_id에 저장한다.
   - u.item 파일 출력에서 awk 명령어로 한 줄씩 읽으면서 'movie id'가 입력받은 movie_id 와 같다면 해당 줄을 전체 출력한다.

2. u.item 파일에서 action 장르에 해당하는 영화의 정보를 ['movie id' 'title'] 포맷으로 상위 10개만 출력
   - u.item 파일 출력에서 awk 명령어로 한 줄씩 읽으면서 'action' 장르에 해당하는 $7이 '1'과 같다면 'movie id'와 'title'을 출력한다.
   - 상위 10개만 출력하기 위해 위의 출력을 head 명령어의 표준 입력으로 받아 처리한다

3. ‘movie id’를 입력 받아 u.data 파일에서 해당하는 영화의 평균 평점을 소수점 6번째 자리에서 반올림하여 출력

   - read 명령어로 입력을 받고 movie_id에 저장한다.
   - 변수 sum과 cnt를 0으로 초기화한다.
   - u.data 파일 출력에서 awk 명령어로 ‘movie id’와 입력 받은 movie_id가 같다면 ‘rating’을 한 줄씩 출력한다.
   - 위 출력을 for 문의 리스트에 넣는다.
   - 리스트에 한 줄씩 i 변수를 통해 접근하면서 ‘rating’ 정보를 sum에 더하고 cnt 변수를 1씩 증가시킨다.
   - 마지막에 awk 명령어의 printf를 사용하여 평균 평점인 sum / cnt 을 소수점 6번째 자리에서 반올림하여 출력한다.

4. u.item 파일에서 URL 부분을 삭제하여 상위 10개만 출력

   - u.item 파일 출력에서 sed 명령어를 사용하여 [‘http’로 시작하고 ‘|’를 포함하지 않는 문자열]을 찾아서 빈 문자열로 교체한다.
   - 상위 10개만 출력하기 위해 위의 출력을 head 명령어의 표준 입력으로 받아 처리한다.

5. u.user 파일에서 데이터를 [user ‘user id’ is ‘age’ years old ‘gender’ ‘occupation’] 과 같은 포맷으로 상위 10개만 출력

   - u.user 파일 출력을 awk 명령어로 ‘zip code’ 에 해당하는 $5를 제외하고 출력한다.
   - 위의 출력을 sed명령어의 입력으로 받는다.
   - sed 명령어의 -e 옵션을 통해 ‘M’과 ‘F’ 문자를 각각 ‘male’과 ‘female’으로 바꾸고 [([0-9]+) ([0-9]+) (.*) (.*)] 와 같은 포맷의 문자열을 [user ₩1 is ₩2 years old ₩3 ₩4] 포맷으로 바꿔 출력한다.
   - 상위 10개만 출력하기 위해 위의 출력을 head 명령어의 표준 입력으로 받아 처리한다.

6. u.item 파일의 'release date'의 포맷을 [01-Jan-1995] -> [19950101] 와 같이 바꿔서 상위 10개만 출력

   - u.item 파일 출력을 sed 명령어의 -e 옵션을 사용하여 [-Jan-] 과 같이 month에 해당하는 문자열 포맷을 [-01-] 과 같이 바꾼다. 그리고 [([0-9]+)-([0-9]+)-([0-9]+)] 포맷의 문자열('release date')을 [₩3₩2₩1] 포맷으로 바꿔 출력한다.

7. 특정 'user id'를 가진 유저가 평가한 영화의 'movie id'들을 ['movie id'|'movie id'| ... |'movie id'] 포맷으로 출력하고 출력한 'movie id' 들의 정보를 ['movie id' 'title'] 포맷으로 상위 10개만 출력

   - read 명령어로 입력을 받아 user_id에 저장한다.
   - touch 명령어로 임시 파일 tmp.txt 를 생성한다.
   - u.data 파일 출력에서 awk 명령어로 'user id'가 입력 받은 user_id와 같다면 'movie id'에 해당하는 $2를 출력한다.
   - 위 출력을 sort 명령어의 입력으로 받아서 숫자에 대하여 정렬 후 tmp.txt 파일에 출력한다.

   - cat 명령어로 tmp.txt 파일을 출력하여 tr 명령어의 입력으로 받는다.
   - tr 명령어로 '₩n' 문자를 '|' 문자로 모두 교체하여 출력한다.
   - 위 출력을 sed 명령어의 입력으로 받아서 맨 뒤의 '|' 문자를 지워주고 출력한다.

   - tmp.txt 파일을 상위 10줄만 출력하여 for 문의 리스트로 넣는다.
   - 리스트에 변수 i로 한 줄씩 접근하여 u.item 파일에서 'movie id'와 i 가 같을 때 ['movie id' 'title'] 포맷으로 출력한다.
   - tmp.txt 파일을 지운다.

8. 20대 프로그래머의 영화평가만을 기반으로 하여 20대 프로그래머가 평가한 영화들의 평점 평균을 모두 출력

   - 임시 파일인 tmp.txt를 생성한다.
   - awk 명령어로 u.user 파일에서 나이가 20대이고 직업이 프로그래머인 사람의 'user id'를 for문의 리스트로 넣는다.
   - 그리고 리스트의 'user id'를 하나씩 접근하면서 awk 명령어로 u.data 파일의 해당 유저가 평가한 영화의 'movie id'와 'rating'을 찾아 tmp.txt. 파일에 추가한다.
   - 영화의 'movie id'는 1682번까지 존재한다. 1번부터 하나씩 tmp.txt 파일에 접근하여 평점 평균을 구한다. 이 과정에서 awk 명령어의 END 기능과 printf 를 사용한다.
   - tmp.txt 파일을 삭제한다.

결론

과제에서 구현한 BASH 스크립트는 원본 파일을 수정하지 않고 데이터를 외부에서 가공해서 보여준다. 이러한 원본 파일을 수정하지 않는다는 점 덕분에 라이선스 문제로 프로그램을 수정하지 못하거나 코드에 접근하지 못할 때 프로그래머의 의사대로 출력을 바꿀 수 있다는 장점이 존재한다.

하지만 BASH 스크립트는 느리다. 더구나 이번 과제에서는 시간과 메모리에 대한 제한이 없어서 스크립트의 복잡도를 생각하지 않고 구현하였지만, BASH 스크립트의 성능을 고려하여 최적화할 필요가 있어보인다.