

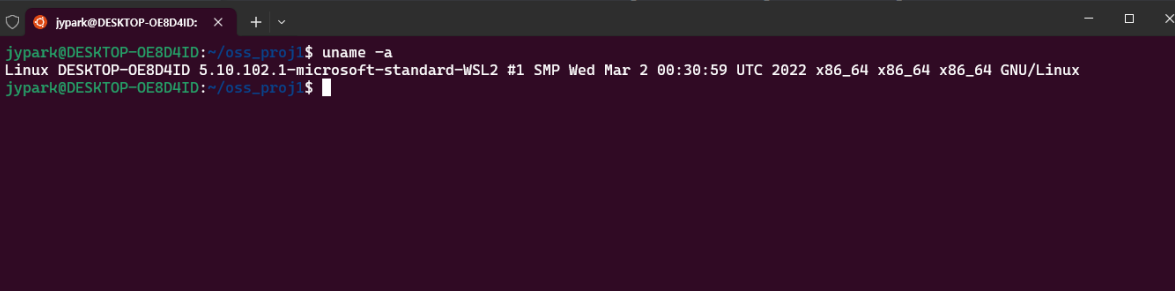
오픈소스개론 과제 보고서

12211618 박준용

구현

0. 실행 환경

- WSL 2 기반 Ubuntu 20.04 환경에서 진행하였습니다.



```
jypark@DESKTOP-OE8D4ID: ~/oss_proj1$ uname -a
Linux DESKTOP-OE8D4ID 5.10.102.1-microsoft-standard-WSL2 #1 SMP Wed Mar 2 00:30:59 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
jypark@DESKTOP-OE8D4ID: ~/oss_proj1$
```

1. Get the data of the movie identified by a specific 'movie id' from 'u.item'

1. `awk`의 `-F` 옵션을 통해 구분자를 `|`로 설정하였습니다.
2. `awk`의 `-v` 옵션을 통해 입력받은 영화의 ID를 저장하고, `$1`와의 일치여부를 검사하여 출력하도록 설계했습니다.

[illegible]

2. Get the data of 'action' genre movies from 'u.item'

1. `awk`의 `-F` 옵션을 통해 구분자를 `|`로 설정하였습니다.
2. 이후 7번째 필드인 `$7`이 `1`인지 확인하고, `awk` 내에서 `C` 스타일의 `printf`을 이용하여 출력하였습니다.
3. 2번의 출력을 파이프를 통해 `sort` 명령어로 넘겨주었고 정렬을 진행하였습니다.
4. 3번의 출력을 파이프를 통해 `head` 명령어로 넘겨주어 출력하도록 설계했습니다.

```
jypark@DESKTOP-OE8D4ID: x + v
jypark@DESKTOP-OE8D4ID:~/oss_proj1$ ./proj1_12211618_junyongpark.sh u.item u.data u.user
-----
User Name: Jun-Yong Park
Student Number: 12211618
[ MENU ]
1) Get the data of the movie identified by a specific 'movie id' from 'u.item'
2) Get the data of action genre movies from 'u.item'
3) Get the average 'rating' of the movie identified by specific 'movie id' from 'u.data'
4) Delete the 'IMDb URL' from 'u.item'
5) Get the data about users from 'u.user'
6) Modify the format of 'release date' in 'u.item'
7) Get the data of movies rated by a specific 'user id' from 'u.data'
8) Get the average 'rating' of movies rated by users with 'age' between 20 and 29 and 'occupation' as 'programmer'
9) Exit
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. Get the average 'rating' of the movie identified by specific 'movie id' from 'u.data'

1. `awk`의 `-F` 옵션을 통해 구분자를 공백 문자로 설정하였습니다.
2. `awk`의 `-v` 옵션을 통해 입력받은 영화의 ID를 저장하였습니다.
3. `awk`의 `BEGIN`을 통해 `count` 변수와 `sum`를 선언했습니다.
4. 두번째 필드 `$2`와 2번에서 입력받은 영화의 ID를 비교하였고, 조건에 따라 `sum` 변수에 `$3`을 누적하였습니다.
5. `awk`의 `END`와 `printf`를 통해 평균을 출력하도록 구현하였습니다.

- 이후 `printf`를 통해 `$1`, `$2`, `gender`, `$4` 순으로 출력을 진행하였습니다.
- 이후 파이프를 통해 `head` 명령어로 넘겨주어 출력을 진행하였습니다.

```
jypark@DESKTOP-OE8D4ID: ~/oss_proj1$ ./proj1_12211618_junyongpark.sh u.item u.data u.user
-----
User Name: Jun-Yong Park
Student Number: 12211618

[ MENU ]
1) Get the data of the movie identified by a specific 'movie id' from 'u.item'
2) Get the data of action genre movies from 'u.item'
3) Get the average 'rating' of the movie identified by specific 'movie id' from 'u.data'
4) Delete the 'IMDb URL' from 'u.item'
5) Get the data about users from 'u.user'
6) Modify the format of 'release date' in 'u.item'
7) Get the data of movies rated by a specific 'user id' from 'u.data'
8) Get the average 'rating' of movies rated by users with 'age' between 20 and 29 and 'occupation' as 'programmer'
9) Exit
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. Modify the format of 'release date' in 'u.item'

1. `sed` 를 사용하여 날짜의 월에 해당하는 문자열을 숫자로 대체하였습니다.
 - `s/JAN/01/g` 와 같은 형식과 `;` 를 사용하여 12개의 형식을 만들었습니다.
2. 1에서 `sed` 를 수행한 결과를 `redirection` 을 통해 임시 파일로 저장하였습니다.
3. 저장한 임시파일을 `cat` 으로 불러와 파이프를 통해 다시 한번 `sed` 를 수행했습니다.
 - 정규 표현식 `'s/([0-9]+)-([0-9]+)-([0-9]+)/\3\2\1/g'` 를 사용하여 `YYYYMMDD` 형식으로 변환했습니다.
4. 3번의 출력 결과를 파이프로 `tail` 명령어에 넘겨주었습니다.
5. 이후 임시 파일을 삭제하였습니다.

[illegible]

7. Get the data of movies rated by a specific 'user id' from 'u.data'

1. `awk` 의 `-F` 옵션을 통해 구분자를 공백 문자로 설정하였습니다.
2. `awk` 의 `-v` 옵션을 통해 입력받은 유저의 ID를 저장했습니다.
3. `u.data` 에서의 `awk` 를 통해 `$1` 과 ID의 일치 여부를 검사하고, `print` 를 이용하여 `$2` 를 출력하도록 구현했습니다.
4. 3번의 출력을 파이프로 `sort` 명령어에게 넘겨주었으며, `redirection` 을 통해 임시 파일을 생성했습니다.
5. 이후 임시 파일에서의 `awk` 를 통해 영화의 ID를 출력하였습니다.
6. 이후 `while` 문으로 임시 파일을 구분자 `|` 로 구분하여 데이터를 불러왔습니다.
7. `u.item` 에서의 `awk` 를 통해 6번에서 불러온 ID에 해당하는 영화를 출력하도록 구현했습니다.
8. 이후 임시 파일을 삭제하였습니다.

```
jypark@DESKTOP-0E8D4ID: x + v
jypark@DESKTOP-0E8D4ID: ~/oss_proj1$ ./proj1_12211618_junyongpark.sh u.item u.data u.user
-----
User Name: Jun-Yong Park
Student Number: 12211618
[ MENU ]
1) Get the data of the movie identified by a specific 'movie id' from 'u.item'
2) Get the data of action genre movies from 'u.item'
3) Get the average 'rating' of the movie identified by specific 'movie id' from 'u.data'
4) Delete the 'IMDb URL' from 'u.item'
5) Get the data about users from 'u.user'
6) Modify the format of 'release date' in 'u.item'
7) Get the data of movies rated by a specific 'user id' from 'u.data'
8) Get the average 'rating' of movies rated by users with 'age' between 20 and 29 and 'occupation' as 'programmer'
9) Exit
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|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 ]: █
```

8. Get the average 'rating' of movies rated by users with 'age' between 20 and 29 and 'occupation' as 'programmer'

1. `declare` 를 통해 배열을 선언하였습니다.
2. 이후 `while` 문을 이용하여 `u.user` 파일을 불러와 사용자의 나이와 직업의 조건과 비교하여 배열에 저장했습니다.
3. `while` 문을 이용하여 `u.item` 파일을 불러와 아래와 같은 과정을 진행했습니다.
 1. `u.data` 에서 `awk` 를 진행했습니다.
 2. 구분자는 공백 문자로 설정했습니다.
 3. `-v` 옵션을 통해 1번에서 정의한 배열을 불러왔습니다.
 4. 이후 배열과 비교하여 조건에 맞는 사용자인지 확인하였습니다.
 5. 조건과 부합하는 사용자인 경우와 `while` 문에서 불러온 영화의 ID를 모두 검사하여 사용자가 평가한 영화의 평점을 계산했습니다.
 6. 이후 `END` 를 통해 영화의 평점을 출력했습니다.

```
jypark@DESKTOP-OE8D4ID: x + v
jypark@DESKTOP-OE8D4ID:~/oss_proj1$ ./proj1_12211618_junyongpark.sh u.item u.data u.user
-----
User Name: Jun-Yong Park
Student Number: 12211618
[ MENU ]
1) Get the data of the movie identified by a specific 'movie id' from 'u.item'
2) Get the data of action genre movies from 'u.item'
3) Get the average 'rating' of the movie identified by specific 'movie id' from 'u.data'
4) Delete the 'IMDb URL' from 'u.item'
5) Get the data about users from 'u.user'
6) Modify the format of 'release date' in 'u.item'
7) Get the data of movies rated by a specific 'user id' from 'u.data'
8) Get the average 'rating' of movies rated by users with 'age' between 20 and 29 and 'occupation' as 'programmer'
9) Exit
Enter your choice [ 1-9 ]: 8
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)
: 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
```

9. Exit

`break` 문을 이용하였습니다.

```
jypark@DESKTOP-OE8D4ID: x + v
jypark@DESKTOP-OE8D4ID:~/oss_proj1$ ./proj1_12211618_junyongpark.sh u.item u.data u.user
-----
User Name: Jun-Yong Park
Student Number: 12211618
[ MENU ]
1) Get the data of the movie identified by a specific 'movie id' from 'u.item'
2) Get the data of action genre movies from 'u.item'
3) Get the average 'rating' of the movie identified by specific 'movie id' from 'u.data'
4) Delete the 'IMDb URL' from 'u.item'
5) Get the data about users from 'u.user'
6) Modify the format of 'release date' in 'u.item'
7) Get the data of movies rated by a specific 'user id' from 'u.data'
8) Get the average 'rating' of movies rated by users with 'age' between 20 and 29 and 'occupation' as 'programmer'
9) Exit
Enter your choice [ 1-9 ]: 9
Bye!
jypark@DESKTOP-OE8D4ID:~/oss_proj1$
```

느낀 점

`awk` 를 사용하다보니 `awk` 에서 지원하는 C 스타일의 `printf` 가 정말 편리했습니다.

`printf` 를 사용하면 `sed` 사용 없이도 해결 가능한 문제들이 있었고, 문자열을 포매팅하는데 있어서 정말 편리했습니다.

8번 문제에서 `while` 문을 중첩해서 사용하다보니 수행 시간이 오래 걸린 점이 아쉬웠습니다.

현재 오픈소스개론에서 `Pandas` 를 배우고 있었는데, `Pandas` 가 데이터 처리에 있어서 엄청나게 좋은 도구임을 다시 느꼈습니다.

데이터를 추출하는 과정에서 많은 어려움을 겪었는데, `Pandas` 의 인덱싱은 이러한 과정을 쉽게 해줄 뿐만 아니라, 더 적은 비용으로 데이터를 추출한다는 것이 매우 편리하게 느껴졌습니다.