**[2024 OSS Project 1]**

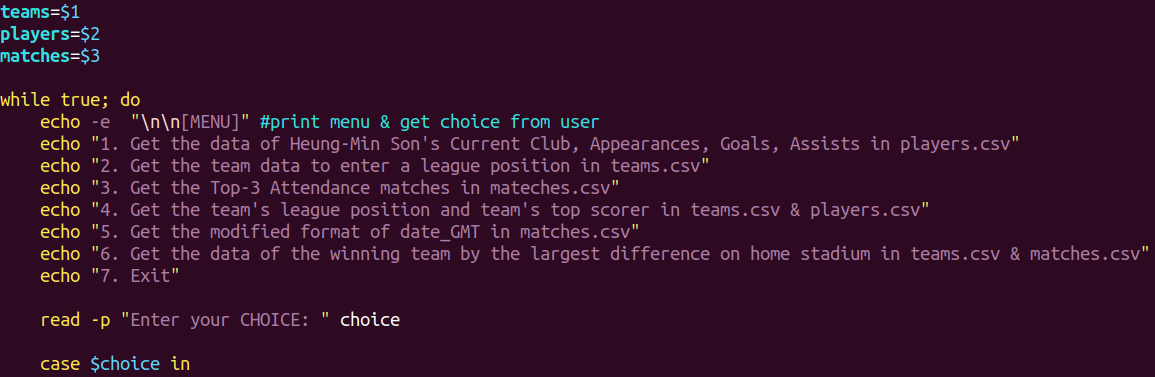
– 제출 보고서

과목명: 오픈소스SW개론(001분반)

학번: 12201702

이름: 김민지

* 전체 구조

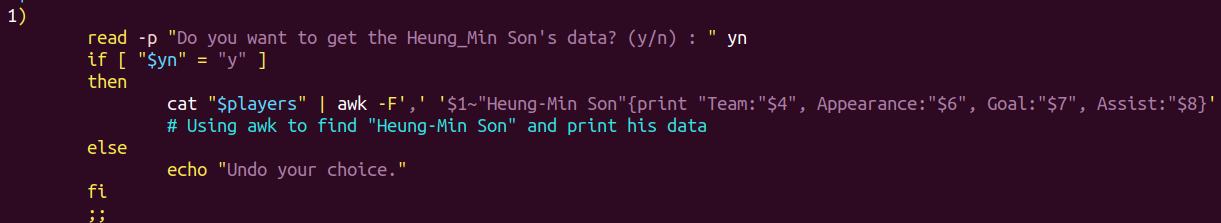


* Choice에 대한 입력은 while문으로 받았으며, choice가 7인 경우 break를 통해 반복문을 종료하도록 하였다.
* 각 choice에 대한 기능들은 case문을 통해 숫자를 변수로 주어 처리하도록 하였다.
* 구현

1. **오류 처리**

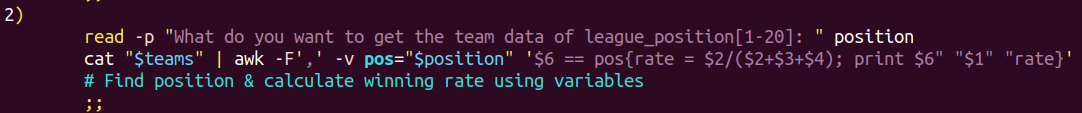
* 입력 받은 파일의 개수가 3개가 아닌 경우 오류 메시지를 출력하도록 하였다.
* Choice를 입력 받을 때, 1-7 외의 입력이 들어올 경우 case문에서 default로 처리하여 다음 메뉴를 입력하도록 하였다.

1. **손흥민 선수의 정보 출력**



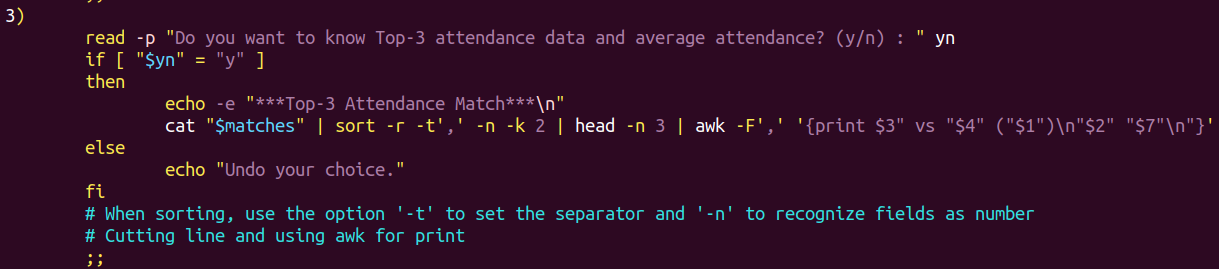
* 손흥민 선수에 대한 정보는 players.csv 파일을 읽은 후 awk를 통해 이름으로 조건에 맞는 행을 찾고, 필요한 field만을 출력하였다.

1. **승률 출력**



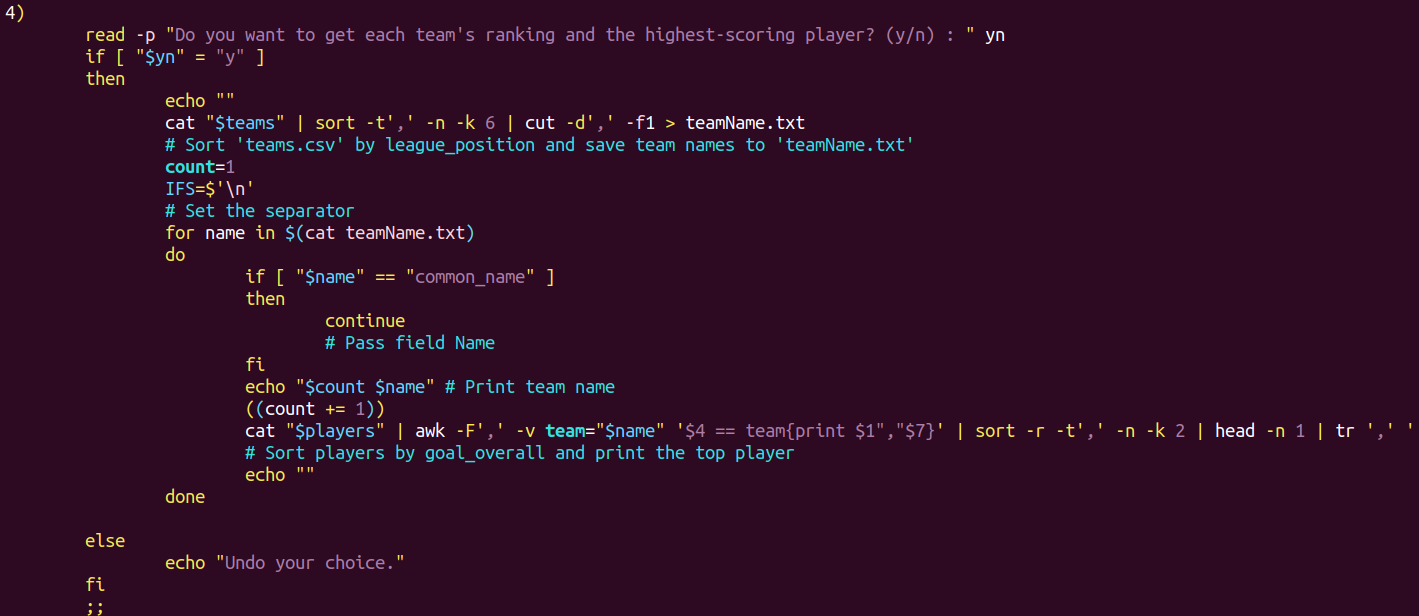
* 원하는 팀의 league position을 입력 받고, 해당 position을 사용해 ‘teams.csv’파일에서 awk를 통해 동일한 팀을 찾아 승률을 계산하고 출력하였다.

1. **Top-3 출력**



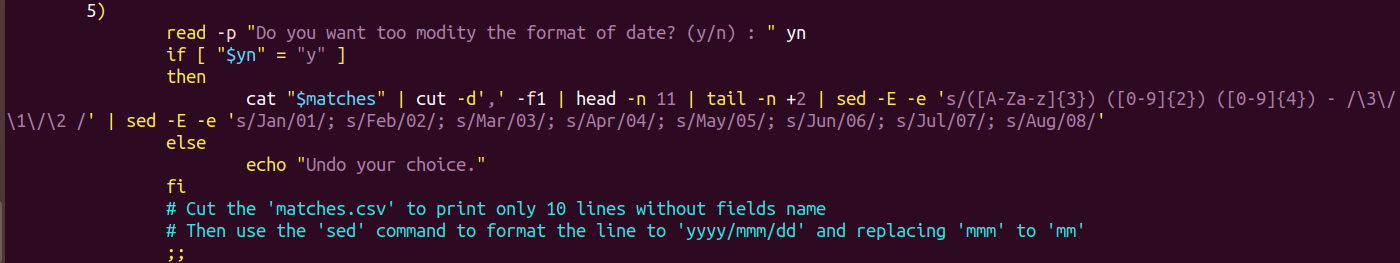
* ‘matches.csv’ 파일을 출력한 후, 2번째 field(attendance)를 기준으로 정렬한 뒤 head를 통해 상위 3개만 나타내도록 하였다.
* 이후 awk를 통해 적절한 형식으로 나타내도록 하였다.

1. **팀별 최고 득점 선수 출력**



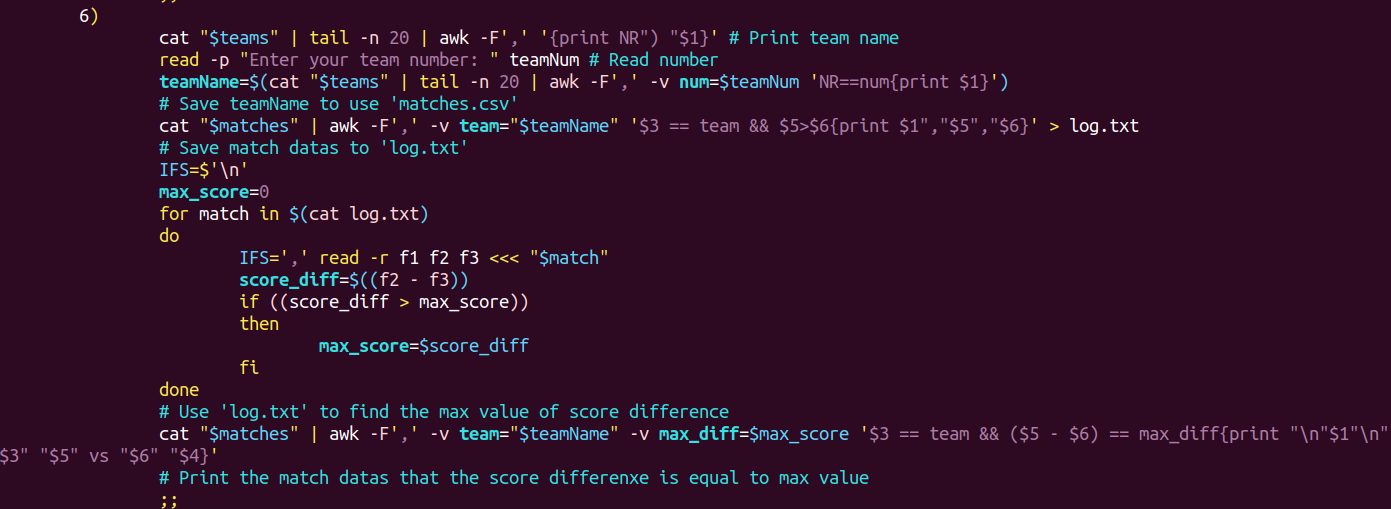
* 가장 먼저 ‘teams.csv’ 파일에서 team name만 불러와 줄바꿈을 구분자로 하여 ‘teamName.txt’에 저장하였다.
* IFS를 통해 for문에서 사용될 구분자를 줄바꿈으로 지정해 주고, for문 내에서 각 팀에서 최고 득점 선수를 출력하도록 하였다. 이때 최고 점수의 기준은 goal\_overall을 사용하였다.
* for문 내에서 ‘player.csv’ 파일을 사용하였고, awk를 통해 각 팀에 속해 있는 모든 선수들을 출력한 후, goal\_overall을 기준으로 내림차순 정렬하고, head를 사용해 제일 위의 한 선수만 출력하였다.

1. **Data\_GMT**



* ‘matches.csv’에서 cut을 통해 날짜와 시간 정보만 추출을 한 뒤 head와 tail을 사용해 field name을 제외한 10개의 행을 출력했다.
* 10개의 행에 대해 sed를 사용해 ‘mmm dd yyyy – ‘를 찾아 ‘yyyy/mmm/dd’로 변경하였다.
* Sed를 한 번 더 사용해 mmm을 mm으로 바꾸어 최종 형태를 만든 후 출력하였다.

1. **가장 크게 이긴 경기 정보 출력**



* ‘teams.csv’로 팀 명에 대해서만 우선 출력하고, 입력 받은 숫자를 다시 ‘teams.csv’와 awk를 사용해 teamName 변수에 팀 명을 저장하였다.
* ‘matches.csv’를 통해 해당 팀이 이긴 보든 경기에 대해 팀명, 팀 점수, 상대팀 점수를 ‘log.txt’에 저장한 뒤 for문에서 각 줄을 읽고 가장 큰 점수차를 ‘max\_score’ 변수에 저장하였다.
* ‘max\_score’, ‘teamName’변수와 ‘matches.csv’ 파일을 사용해 각 팀이 이긴 경기에서 점수차가 max\_score과 같은 경기를 출력하였다.