**C언어 및 실습**

**2015540001**

**수학과 강민성**

프로그램의 개요

고등학생 때 부터 농구를 취미생활로 해왔다. 대학까지 이어져 지금도 중앙동아리 및 과 내 소모임에서 주기적으로 농구를 하고 있을 정도로 농구에 흥미를 느끼고 있다. 프로 경기 관람 역시 좋아해 선수들의 인적사항도 많이 외우고 있을 정도이다. 평상시에는 메모장에 선수들의 기본적인 신상정보를 적곤 하였는데 이를 프로그램화 해서 관리하면 더 편할 것 같다는 생각이 들어 그 생각을 실천하게 되었다. 특히 그 중에서도 가장 좋아하는 SK night의 선수단을 관리하고자 하였다.

프로그램의 설계

파일로부터 선수들의 정보를 불러와 이를 상황에 맞게 처리하고 다시 파일에 기록하는 **파일 입출력** 방식으로 제작하였다. 선수들의 정보로는 이름, 탄생 년도, 포지션, 신장, 체중이 있으며 이를 메모리에 유지하기 위해서 **구조체로 선언**하였다. 프로그램 내에서 사용 가능한 기능은 총 4가지로, **선수 검색, 선수 등록, 선수 삭제, 선수 목록 출력**이 있다. 선수 검색은 이미 등록 되있는 선수의 이름을 검색함으로서 그 선수의 자세한 정보를 알 수 있게 해주는 기능이다. 두 번째로 선수 등록 기능은 팀에 새로운 선수가 스카우트 혹은 트레이드 되었을 경우, 선수 목록에 추가해주는 기능이다. 세 번째로 선수 삭제는 선수가 방출 혹은 트레이드 되서 나갈 경우, 선수의 데이터를 삭제해주는 기능이다. 마지막 네 번째로 선수 목록 출력은 현재 팀에 어떤 선수들이 있는데 목록을 불러주는 기능이다. 각 기능들에 대해서는 서브 함수로 만들었고 전체적인 구성을 살펴보면 main.c / sub\_func.c / player.h , 3가지로 **모듈화** 하였는데 main.c는 main함수가 있는 부분, sub\_func.c에는 각 기능을 하는 4가지 함수가 있는 부분, player.h에는 선수 정보 구조체를 포함시켰다.

프로그램의 동작 방법

우선 player.txt 에서 선수들의 정보를 불러온다. 이 때 선수가 몇 명인지 유동적으로 바뀔 수 있으니 calloc으로 구조체를 **동적할당** 해주었고 데이터 수가 증가하면 realloc으로 점층적으로 증가하게 하였다. 파일의 데이터를 구조체의 맴버들에 저장을 해주는데 이름 역시도 개인마다 길이 차이가 많이 나므로 malloc을 통해 동적할당해주었다. 그 후 switch문을 이용해서 선수 검색, 선수 등록, 선수 삭제, 선수 목록을 고를 수 있다. 각 함수들은 **함수 포인터**를 사용하여 실행하였다. 첫번째로 선수검색은 scanf로 선수의 이름을 입력받아 “end”를 입력하면 검색을 중지하고, 반복문을 순회하며 입력받은 값이 선수명단에 존재하면 그 선수의 정보들을 출력하게 된다. 두번째로 선수등록이다. 위와 마찬가지로 “end”를 입력받으면 끝나며, 각 맴버들을 하나씩 입력받게 된다. 입력을 다 받은 후에는 파일을 “a” 형태로 열어 파일에 작성해준다. 세번째로 선수삭제이다. 위와 마찬가지로 “end”면 종료를 하고 반복문을 순회하며 삭제하고자 하는 선수의 이름이 명단에 있으면 선수 명단을 한 칸씩 쉬프트 시켜준다. 쉬프트 시킨 후 파일을 “\”로 열어 처음부터 다시 써주는 방식으로 삭제프로그램이 동작한다. 네 번째로 선수 명단출력은 자기참조 구조체를 사용하여 다음 선수로 넘어가며 한 명씩 데이터를 출력하는 방식으로 작동한다.

프로그램 실행 결과

프로그램을 실행 한 후, 내 정보를 기입하고 검색, 목록 확인 후, 삭제한 후 다시 목록확인을 해보도록 하겠다. 원활한 확인을 위해 선수들의 목록을 6명으로 줄인 상태에서 시작했다.

프로그램의 개선 방향

여러 기능들을 중첩하고, 코드가 상대적으로 기존에 짜왔던 코드들에 비해 길어지다 보니 계속 가독성이 떨어지는 현상이 발생했다. 코드를 전부 다 짠 후, 보기 좋게 다듬는 작업도 많은 시간을 투자하였지만 내가 짠 코드를 내가 수정 하려다 보니 쉽지 않았다. 교수님 혹은 실력 있으신 분에게 코드 리뷰를 받아 전체적인 가독성을 높이고 싶다. 다음으로는 예외처리 사항이다. 사용자로 부터 값을 입력받을 때 엉뚱한 값을 입력하면 애러가 발생하는 케이스가 있는데 이런 부분을 예외처리를 통해 보완할 수 있다고 한다. 그 부분도 스스로 학습하여 추가할 계획이다.