데이터 구조

성적 출력 프로그램 프로그래밍

컴퓨터 공학

2015211373 권정인

목차

1. 개요 및 작업 환경
2. 전체 코드 및 코드의 요약
3. 컴파일 및 결과
4. 평가 및 개선 방안
5. 개요 및 작업 환경

저는 Windows 10 내에 있는 Ubuntu 16.04 기반 Linux 서브 시스템 내에서, 코드 작성은 VI로, 코드 빌드는 GCC로 실행 하였습니다.

언어 코드 작성 시작 전, 우선 eClass에서 scr2016.txt 파일을 제 PC로 다운로드 한 후, Windows에 내장된 파일 탐색기로 C:\ 디렉토리로 파일을 옮겼습니다. 그 후, Bash 환경에서 C:\에 있는 scr2016.txt 파일을 Linux 파일시스템의 제 홈 디렉토리로 복사하였습니다. 그 명령어는 다음과 같습니다 : *cp /mnt/c/scr2016.txt ~/scr2016.txt*

Windows 10 내의 Linux 서브시스템에서 실제 C드라이브는 /mnt/c/ 디렉토리로 마운트 됨을 알 수 있었습니다.

만일, 파일 작성을 제 개인 PC에서가 아닌 학과 유닉스 서버에서 작성하고 실행한다면, scr2016파일을 홈 디렉토리로 복사한 후 scp명령어를 이용해, 학과 서버로 복사하면 됩니다. 이때의 명령어는 다음과 같습니다. *scp -P 11800 scr2016.txt [학과서버 ID]@dalma.dongguk.ac.kr:~/scr2016.txt*

1. 전체 코드 및 코드의 요약

#include "stdio.h"

struct student{ //구조체를 미리 정의해줍니다.

int identify;

int subs[5];

int sum;

float avg;

};

int main(void)

{

system("clear"); //실행 이전, 가독성 향상을 위해 UNIX 명령어인 ‘clear’를 실행하여 화면을 청소합니다.

FILE\* file=fopen("scr2016.txt", "rt"); //홈 디렉토리에 있는 scr2016.txt 텍스트 읽기 모드로 엽니다.

if(file==NULL) {

printf(" scr2016.txt File not found. Aborting..");

return 1; //파일이 없어서 열리지 않을 경우, 오류 메시지를 출력하고, return 1을 반환하여 프로그램을 종료합니다.

}

int i, score, class;

int id=0;

int grade=1;

struct student s;

s.sum = 0;

s.avg = 0;

s.identify = 0;

fscanf(file, "%d", &id);

for (class = 1; class<7; class++)

{

int check\_endoffile;

s.identify = 0;

for (i=0; i<6; i++) s.subs[i] = 0; //모든 반을 처리한 후 해당 반의 과목 값을 0으로 초기화

printf("성적 열람 : %d 학년 %d반\n", grade, class);

printf("학번 국어 산수 사회 과학 체육 총점 평균\n");

if(id ==0) fscanf(file, "%d", &id);

do //각 반의 성적 출력을 반복하기위해 do-while문을 사용했습니다

{

printf("%-7d\t ", id); //각 학생의 학번(변수 id)를 출력

for(i=0; i<5; i++) //for문 아래 대괄호로 묶인 내용은 각 과목의 성적을 출력 및 처리합니다

{

fscanf(file, "%d", &score);

printf("%-4d\t", score);

s.subs[i]+=score; //과목별 성적을 누적하여 더합니다.

s.sum+=score; //학생의 모든 과목 성적을 누적하여 더합니다.

}

s.identify++; //다음 학생의 성적을 처리하기 위해 s.identify를 1씩 증가합니다.

s.avg = (float) s.sum/5; //소숫점 아래수까지 계산을 하기 위해 강제 형변환(float로)을 하였습니다

printf(" %-7d %-7.1lf\n", s.sum, s.avg); //개인별 총 합과 총 과목의 평균을 표시합니다

s.sum = 0; //다음 학생 처리 이전, 개인의 모든 과목 합계를 0으로 초기화

s.avg = 0; //다음 학생 처리 이전, 개인의 평균 성적을 0으로 초기화

check\_endoffile = fscanf(file, "%d", &id); //파일에서 더이상 처리할 데이터가 없을 경우 -1을 지정하는 check\_endoffile을 정의합니다

}

while(((id/100)%10 == class) && check\_endoffile != EOF); //같은 반에 더이상 처리할 데이터가 없을 경우 EndOfFile을 실행하여 반복문을 종료합니다

printf("--------------------------------------------------------------\n\n");

printf("반평균 ");

for(i=0;i<5; i++) printf("%-8d",s.subs[i]/s.identify); //총 5개의 과목의 전체 평균을 구합니다.

if (class!=6){

printf("\n\n[Enter]를 눌러 %d반의 성적을 확인하세요", class+1);}

else {

printf("\n\n모든 반의 성적을 확인하셨습니다. [Enter]를 눌러 프로그램 종료");}

//현재 표시하고있는 학급이 6반일 경우 프로그램 종료를 유도하고, 6반이 아닐 경우, 다음 분반의 표시를 유도하세요.

getchar();

system("clear"); //[Enter]를 누르면 UNIX 명령어인 ‘clear’에 접근하며, 반복문에서 지시한 것들을 반복합니다

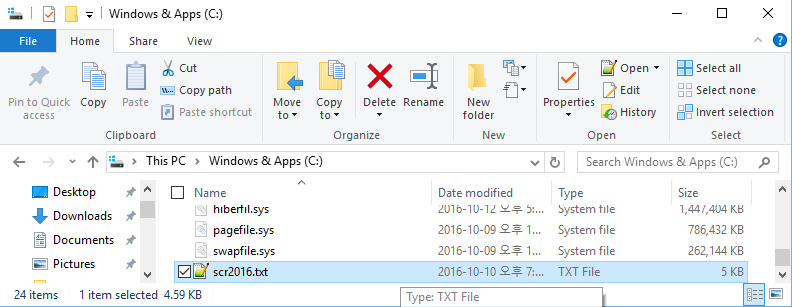
}

fclose(file);

return 0; //반복문이 끝나면 fclose를 통해 파일을 닫고 0을 리턴하여 프로그램을 종료하세요.

}

1. 컴파일 및 결과

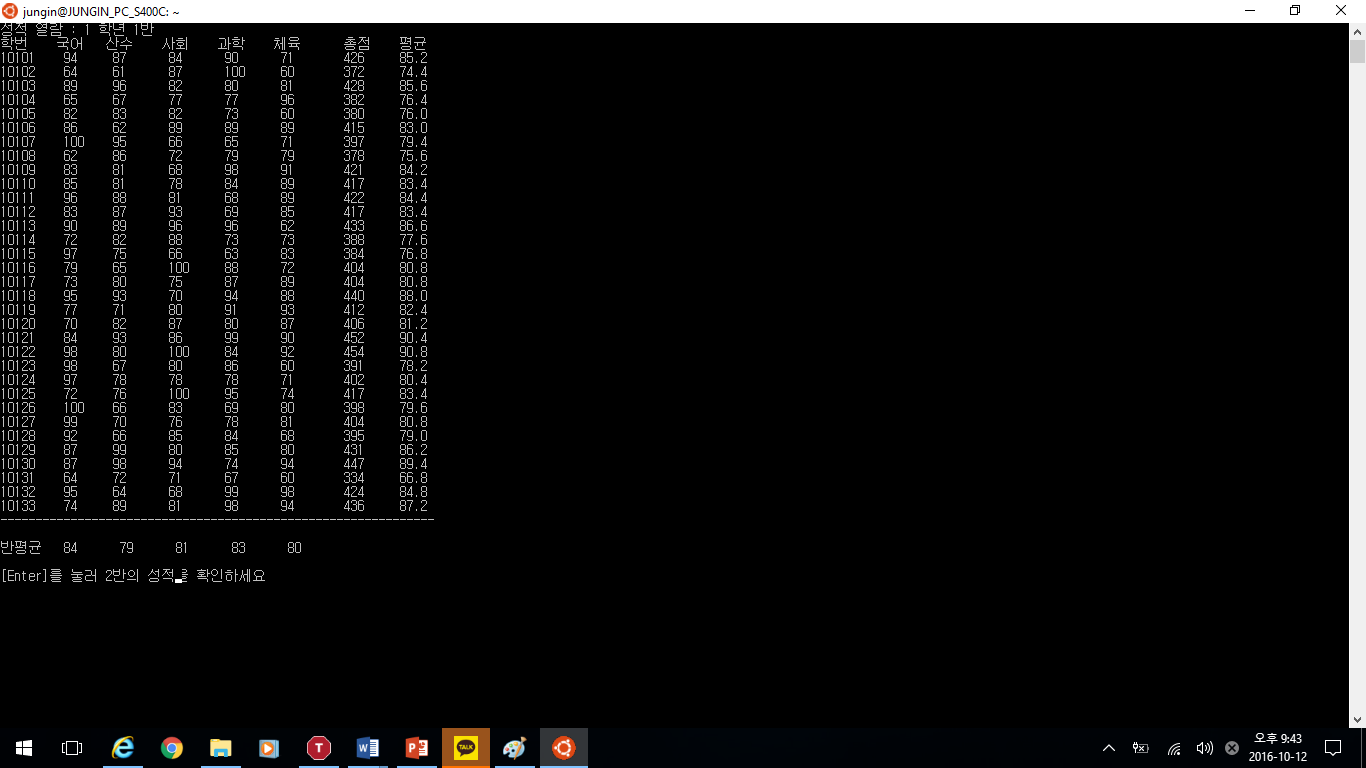


Windows 내의 파일 탐색기를 통해 scr2016.txt 파일을 C:\로 옮깁니다

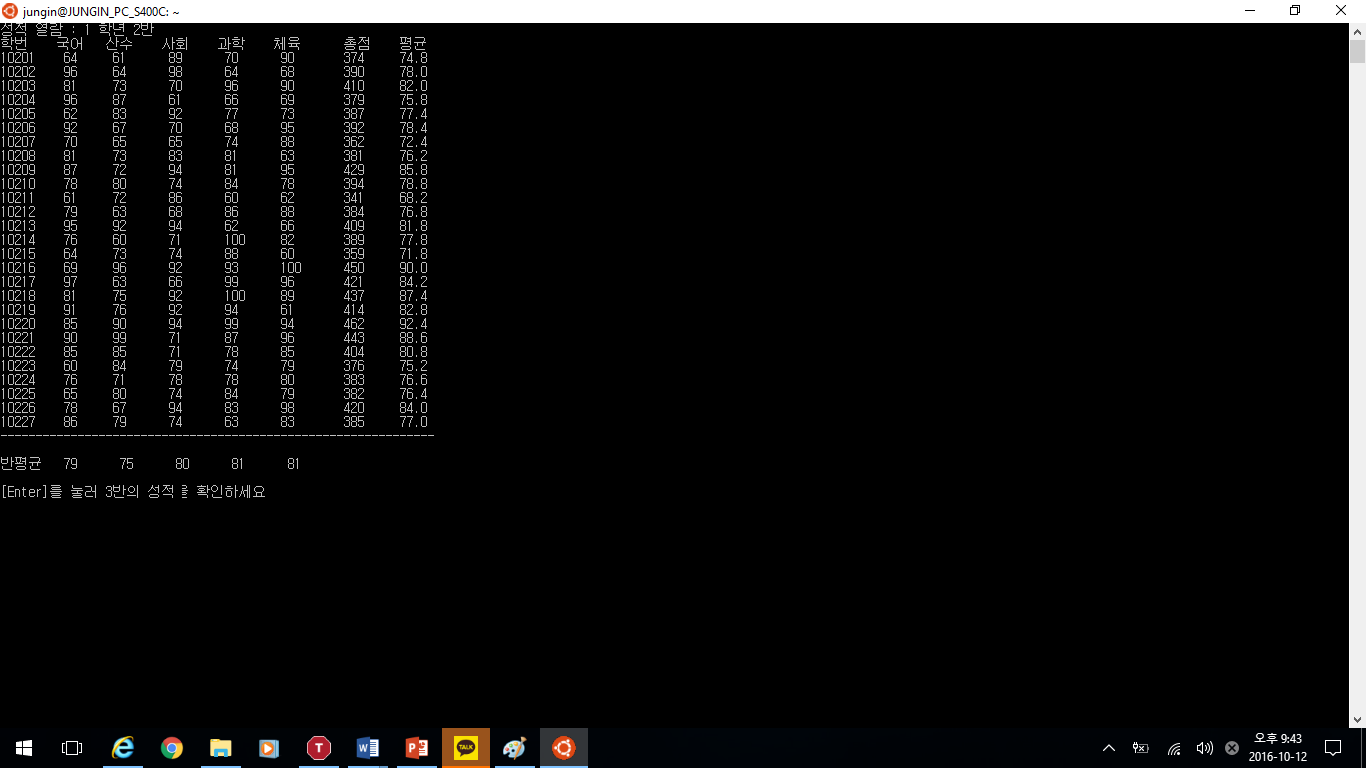


Windows 10 내의 Linux 서브시스템에 접근하여 스크린샷 속 명령을 실행합니다.

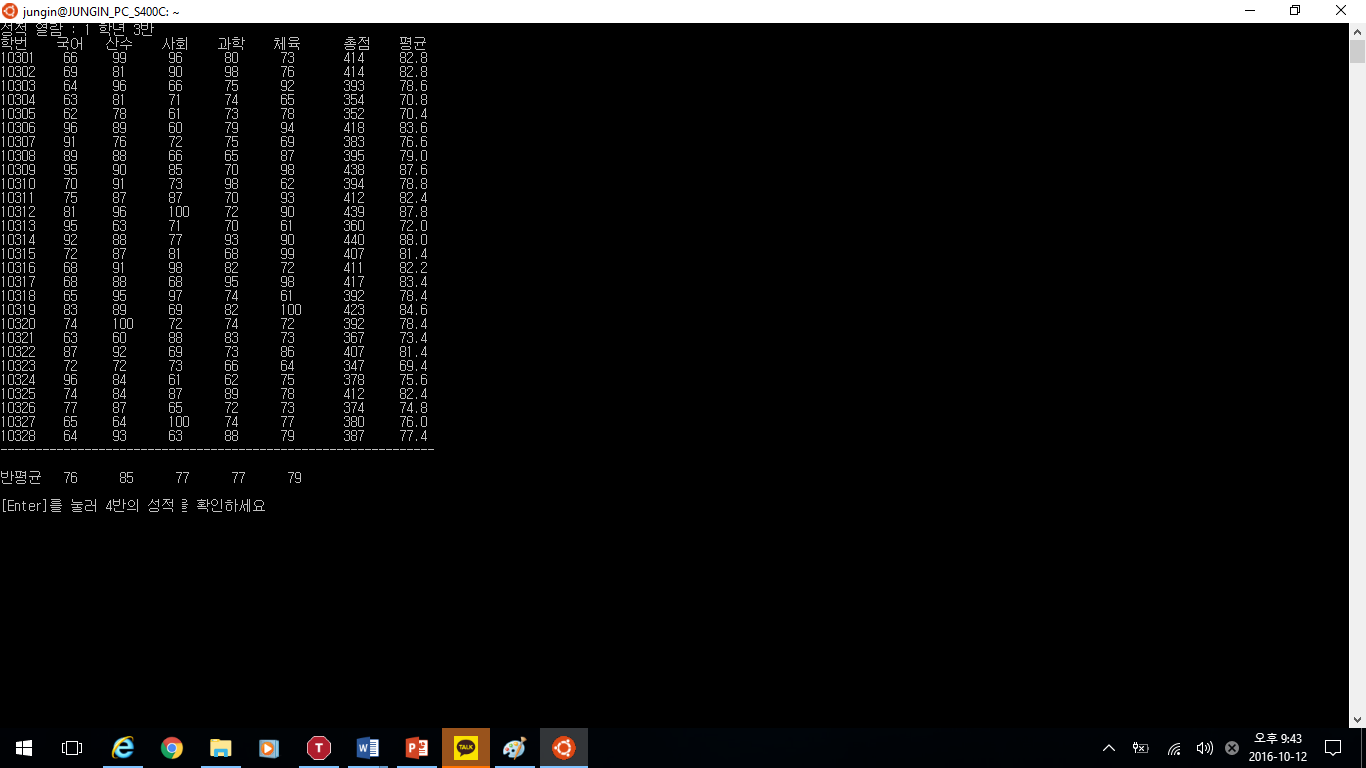
위에부터, C:\의 scr2016.txt 파일을 홈 디렉토리로 복사, GCC로 코드 빌드(컴파일), 컴파일하여 생성된 a.out 실행



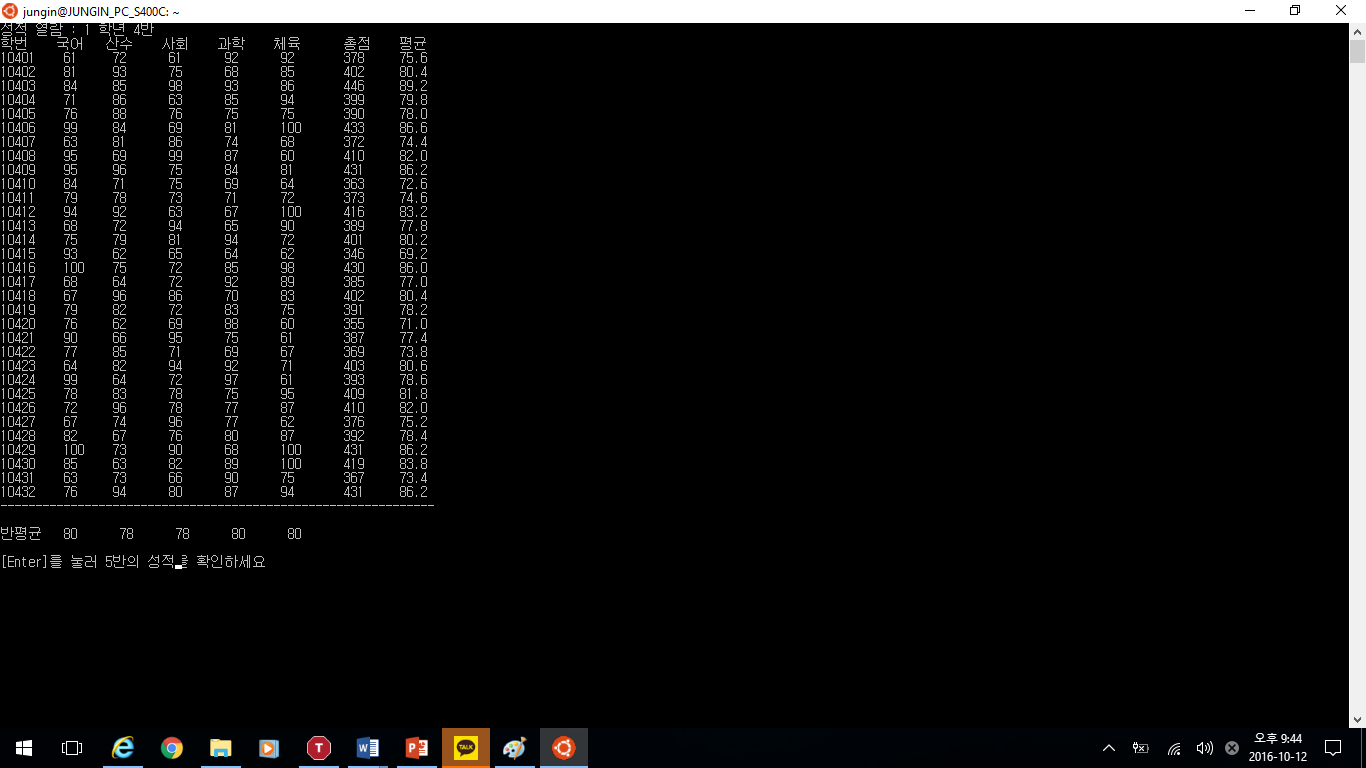
1학년 1반 출력 스크린샷



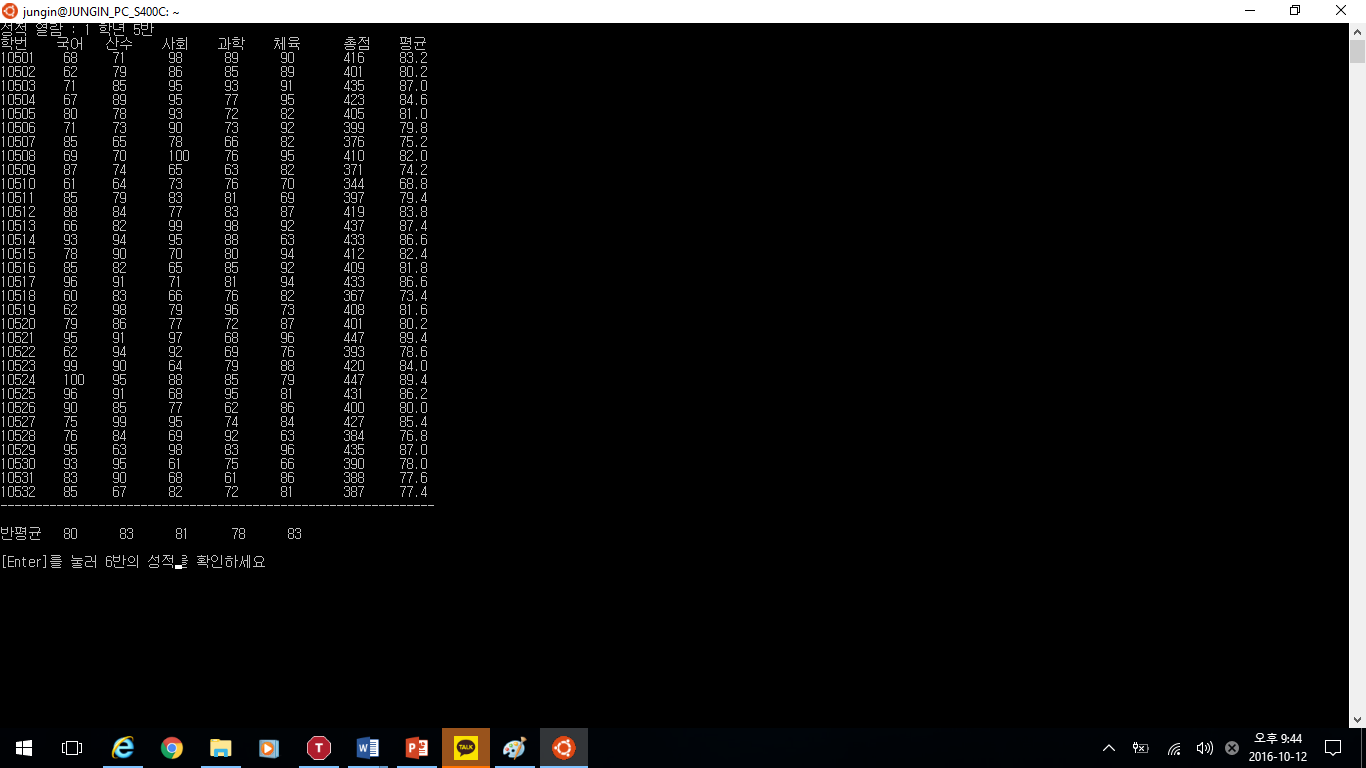
1학년 2반 출력 스크린샷



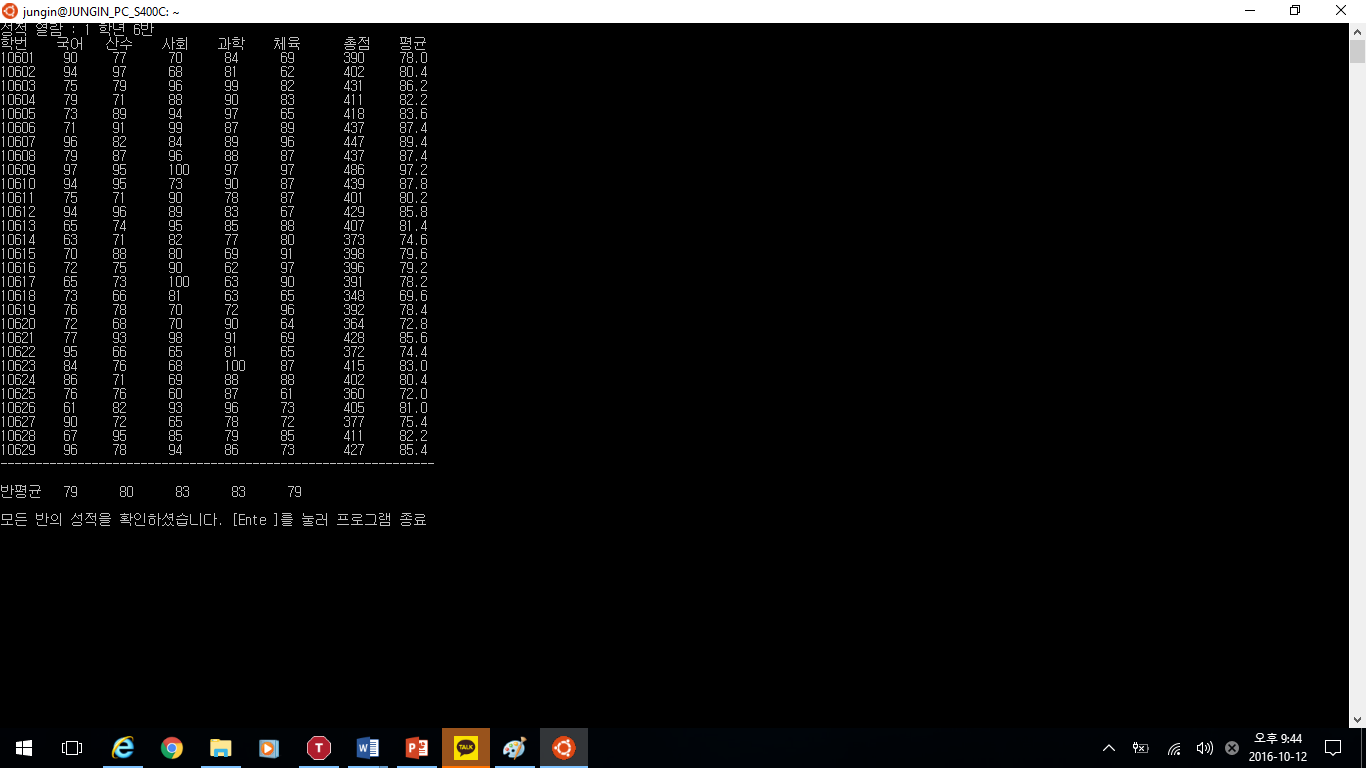
1학년 3반 출력 스크린샷



1학년 4반 출력 스크린샷



1학년 5반 출력 스크린샷



1학년 6반 출력 스크린샷 (마지막 분반이므로 프로그램 종료를 유도하고 있습니다.)

1. 평가 및 개선 방안
2. 코드 작성을 시작하기 전에, scr2016.txt 파일을 리눅스 서브 시스템 내 홈 디렉토리로 옮기는데 어려움이 있었습니다. 하지만 인터넷 검색 결과, 절대 경로 C:\는 리눅스 내 가상 경로 /mnt/c/ 에 마운트 됨을 알고, CP 명령어를 통해 리눅스 홈 디렉토리로 복사하였습니다.
3. 파일을 가져오는 fopen 명령, 유닉스 명령어에 직접 접근하는 system 명령 등은 지난 첫주차 과제의 소스 코드를 약간 참고하여 이번 과제의 코드를 작성하였습니다.
4. 개인적으로, 반복문을 여러번 사용하는 코드를 작성하는데 약하여, 코드를 수십번 추가하고 고쳐 내려갔습니다.
5. 지난 과제에서 [개선 사항] 에 있던 if문을 사용하여 사용성을 향상시키려는 의도를 적용하였습니다.
6. 코드를 보다 더 간략화하고 효율적으로 작성하여, 프로그램 실행 시 시스템 리소스를 조금 더 적게 차지하도록 하면 좋을 것 같습니다.
7. 프로그램 실행 화면에서 드러나듯이, 과목별 전체 평균을 나타내는 printf문에서, 항목 위치가 약간 엇나갔습니다. 처음 코드를 작성할때는 ‘\t’를 사용하려 했으나, ‘\t’ 사용 시, 위치가 더 많이 엇나갔음을 확인 한 후 ‘ ‘(띄어 쓰기)를 여러번 작성하는 방법을 사용하였습니다. 다음에는, 이러한 문제가 해결되어야 할 것 같습니다.
8. 유닉스 및 리눅스에서 사용하는 텍스트 편집기인 VI(또는 VIM)에서 작업할 때, 다음부터는 자동 들여쓰기 설정을 하면, 조금 더 편하게 코드를 작성할 수 있을 것 같습니다.