

오픈소스소프트웨어개발입문 REPORT(HW5)

목차	쪽수
표지	1쪽
사진1 : calculator.c	2쪽
calculator.c 설명	2-3쪽
사진2 : add.c	3쪽
add.c 설명	3쪽
사진3 : minus.c	3쪽
minus.c 설명	3쪽
사진4 : multi.c	3쪽
multi.c 설명	3쪽
사진5 : divide.c	3-4쪽
divide.c 설명	4쪽
사진6 : calculator.h	4쪽
calculator.h 설명	4쪽
사진7 : makefile	4쪽
makefile 설명	4쪽
사진8 : 실행 결과	5쪽
실행 결과 설명	5쪽

오픈소스소프트웨어개발입문

분반 : 062분반

학번 : 201924451

이름 : 김태훈

calculator.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include "calculator.h"
4  enum{Add=1, Minus, Multi, Divide};
5  int main(int argc, char **argv) {
6      double val1, val2;
7      int select;
8      double result;
9      if(argc>4){
10         printf("!!!Too many arguments.\n");
11         return 0;
12     }
13     else if(argc<4){
14         printf("!!!More arguments needed!\n");
15         return 0;
16     }
17     select=atoi(argv[1]);
18     val1=atof(argv[2]);
19     val2=atof(argv[3]);
20     switch(select)
21     {
22         case Add:
23             {result=Addfunc(val1, val2);break;}
24
25         case Minus:
26             {result=Minusfunc(val1, val2);break;}
27
28         case Multi:
29             {result = Multifunc(val1, val2);break;}
30
31         case Divide:
32         {
33             if(val2==0){
34                 printf("!!!Cannot divide by 0!\n");
35                 return 0;
36             }
37             result = Dividefunc(val1, val2);break;
38         }
39         default:
40             {printf("!!!No operator found!\n");return 0;}
41     }
42     printf("Result value: %lf\n",result);return 0;}
43 }
```

<사진1 : calculator.c : vi editor에서 한 화면으로 찍히지 않아 Dev-C++을 이용하여 캡처>
calculator.c는 들어온 파라미터 개수(argc)가 4개인지 확인하고 4개보다 많거나 적으면 오류 메시지를 출력하고 종료한다.

argc==4이면 각 변수(select, val1, val2)에 argv 문자열을 정수와 실수로 변환(atoi, atof)하여 대입하고, select가 1(==Add)이면 Addfunc를 호출하여 리턴 값을 result에 대입하고, select가 2(==Minus)이면 Minusfunc를 호출하여 리턴 값을 result에 대입하고 select가 3(==Multi)이면 Multifunc를 호출하여 리턴 값을 result에 대입한다.

select가 4(==Divide)이면 0으로 나누는지 확인하고, 0으로 나눈다면 오류 메시지를 출력하고 종료한다.

0으로 나누지 않는다면 Dividfunc를 호출하여 리턴 값을 result에 대입한다.

select가 그외 값인 경우에는 오류 메시지를 출력하고 종료한다.

오류없이 진행된 경우에는 result 값을 소수점 이하 6번째 자리까지 출력하고 종료한다.

```
minchocoin@ubuntu: ~/HW5
File Edit View Search Terminal Help
double Addfunc(double val1, double val2)
{
    return val1 + val2;
}
```

<사진2 : add.c>

add.c에 있는 함수 Addfunc는 double값으로 val1, val2가 들어오면 val1 + val2를 반환한다.

```
minchocoin@ubuntu: ~/HW5
File Edit View Search Terminal Help
double Minusfunc(double val1, double val2)
{
    return val1 - val2;
}
```

<사진3 : minus.c>

minus.c에 있는 함수 Minusfunc는 double 값으로 val1, val2가 들어오면 val1-val2를 반환한다.

```
minchocoin@ubuntu: ~/HW5
File Edit View Search Terminal Help
double Multifunc(double val1, double val2)
{
    return val1 * val2;
}
```

<사진4 : multi.c>

multi.c에 있는 함수 Multifunc는 double 값으로 val1, val2가 들어오면 val1*val2를 반환한다.

```
minchocoin@ubuntu: ~/HW5
File Edit View Search Terminal Help
double Dividfunc(double val1, double val2)
{
    return val1 / val2;
}
```

<사진5 : divide.c>

divide.c에 있는 함수 Dividfunc는 double 값으로 val1, val2가 들어오면 val1/val2를 반환한다.

```
minchocoin@ubuntu: ~/HW5
File Edit View Search Terminal Help
#ifndef _CALCULATOR_H_
#define _CALCULATOR_H_
double Addfunc(double val1, double val2);
double Minusfunc(double val1, double val2);
double Multifunc(double val1, double val2);
double Dividfunc(double val1, double val2);
#endif
```

<사진6 : calculator.h>

calculator.h에서 중복 정의를 방지하기 위해 #ifndef, #endif를 넣고, main함수에서 Addfunc, Minusfunc, Multifunc, Dividfunc 함수를 사용할 수 있도록 각 함수를 선언하였다.

```
minchocoin@ubuntu: ~/HW5
File Edit View Search Terminal Help
#calchw makefile
CC=gcc
CFLAGS= -I. -Wall -g -c
OBJS = calculator.o add.o minus.o multi.o divide.o
all:calchw

%.o:%.c
    $(CC) $(CFLAGS) $<
calchw:$(OBJS)
    $(CC) -o $@ $(OBJS)
clean:
    rm -f calchw $(OBJS)
```

<사진7 : makefile>

사진7의 makefile은 calculator.c, calculator.h, add.c, minus.c, multi.c, divide.c를 이용하여 calchw라는 하나의 실행파일을 만든다.

먼저 *.c파일을 오브젝트 파일로 만든다.(%.o:%.c) 이때 \$(CFLAGS)옵션을 통해 헤더파일 검색 위치에 current directory를 추가하고 debugging info를 오브젝트 파일에 넣고 -c 옵션을 이용해 실행파일로 만들지 않고 오브젝트 파일로 만든다.

그리고 오브젝트 파일을 Link하여 calchw라는 하나의 프로그램을 만들어 낸다.(calchw:\$(OBJS))

또 파일을 삭제하려면 make clean를 사용하여 calchw실행파일과 오브젝트 파일을 삭제할 수 있다.

```
minchocoin@ubuntu: ~/HW5
File Edit View Search Terminal Help
minchocoin@ubuntu:~$ cd HW5
minchocoin@ubuntu:~/HW5$ PATH=$PATH:.
minchocoin@ubuntu:~/HW5$ calchw 3 4 5
Result value: 20.000000
minchocoin@ubuntu:~/HW5$ calchw 3 1 4 5
!!!Too many arguments.
minchocoin@ubuntu:~/HW5$ calchw 1 1
!!!More arguments needed!
minchocoin@ubuntu:~/HW5$ calchw 7 8 9
!!!No operator found!
minchocoin@ubuntu:~/HW5$ calchw 4 5 0
!!!Cannot divide by 0!
minchocoin@ubuntu:~/HW5$
```

<사진8 : calchw 실행 결과>

calchw 실행 결과이다. 숫자를 3개보다 많이 입력하거나 적게 입력하면 오류 메시지를 출력하고, 첫번째 인수를 1,2,3,4외의 숫자를 입력하면 오류 메시지를 출력하고 종료한다. 0으로 나누려 할 때도 오류메시지가 출력되고 종료된다.