

Assignment 1

2018년 1학기	
제출일	2018. 04. 06.
과목명	오픈소스 소프트웨어
담당교수	이기훈

학과	컴퓨터공학과
학번	2016722015
이름	오지현

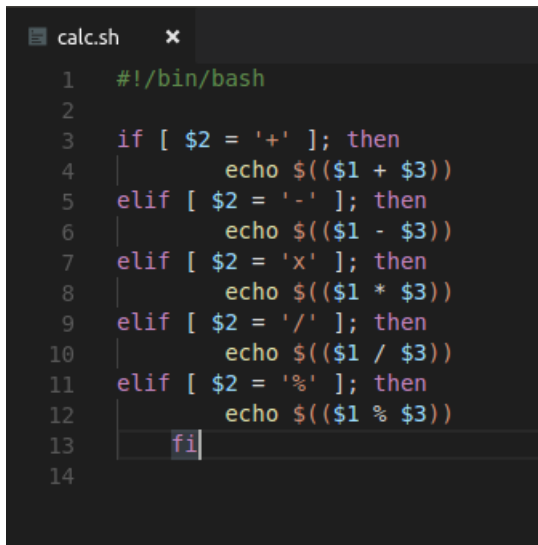
KWANGWOON UNIVERSITY

1. Introduction

Assignment1에서는 두가지 program을 구현한다. 첫번째는 bash로 실행가능한 사칙연산 계산기를 구현하여 쉘 스크립트에 대해 이해한다. 두번째 문제는 n개의 정수로 이루어진 임의의 수열을 사용자가 입력하면 연속된 숫자의 합 중에서 가장 큰 값을 출력한다.

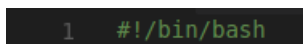
2. 화면 캡처 및 설명

(1)



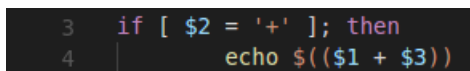
```
1  #!/bin/bash
2
3  if [ $2 = '+' ]; then
4      echo $(( $1 + $3 ))
5  elif [ $2 = '-' ]; then
6      echo $(( $1 - $3 ))
7  elif [ $2 = 'x' ]; then
8      echo $(( $1 * $3 ))
9  elif [ $2 = '/' ]; then
10     echo $(( $1 / $3 ))
11 elif [ $2 = '%' ]; then
12     echo $(( $1 % $3 ))
13 fi
14
```

- 전체 코드



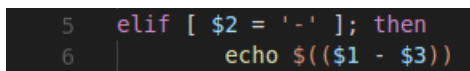
```
1  #!/bin/bash
```

- 쉘 스크립트의 첫번째 줄에 사용하는 구문으로 스크립트 해석기가 bash셸임을 알려준다.



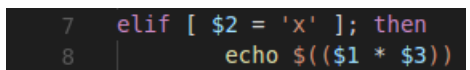
```
3  if [ $2 = '+' ]; then
4      echo $(( $1 + $3 ))
```

- 입력 받은 두번째 인자가 '+'라면 첫번째 인자와 세번째 인자를 더한 값을 출력하라는 명령이다.



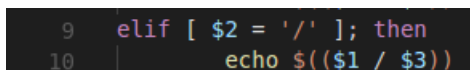
```
5  elif [ $2 = '-' ]; then
6      echo $(( $1 - $3 ))
```

- 입력 받은 두번째 인자가 '-'라면 첫번째 인자에서 세번째 인자를 뺀 값을 출력하라는 명령이다.



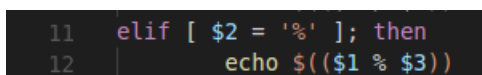
```
7  elif [ $2 = 'x' ]; then
8      echo $(( $1 * $3 ))
```

- 입력 받은 두번째 인자가 'x'라면 첫번째 인자와 세번째 인자를 곱한 값을 출력하라는 명령이다.



```
9  elif [ $2 = '/' ]; then
10     echo $(( $1 / $3 ))
```

- 입력 받은 두번째 인자가 '/'라면 첫번째 인자에서 세번째 인자 나눈 값을 출력하라는 명령이다.



```
11 elif [ $2 = '%' ]; then
12     echo $(( $1 % $3 ))
```

- 입력 받은 두번째 인자가 '%'라면 첫번째 인자에서 세번째 인자를 나눈 나머지를 출력하라

는 명령이다.

```
13 | fi
```

- 조건문을 마친다는 명령어이다.

```
jenny@ubuntu:~/OSS$ bash ./calc.sh 2 + 5
7
```

- Bash shell을 이용하여 calc.sh 프로그램을 실행하였다. 2 + 5를 인자로 대입하였으므로 2+5의 덧셈 연산이 되어, 7이 출력되었다.

```
jenny@ubuntu:~/OSS$ bash ./calc.sh 8 - 2
6
```

- Bash shell을 이용하여 calc.sh 프로그램을 실행하였다. 8 - 2를 인자로 대입하였으므로 8-2의 뺄셈 연산이 되어, 6이 출력되었다.

```
jenny@ubuntu:~/OSS$ bash ./calc.sh 10 / 2
5
```

- Bash shell을 이용하여 calc.sh 프로그램을 실행하였다. 10 / 2를 인자로 대입하였으므로 10/2의 나눗셈 연산이 되어, 5가 출력되었다.

```
jenny@ubuntu:~/OSS$ bash ./calc.sh 2 x 5
10
```

- Bash shell을 이용하여 calc.sh 프로그램을 실행하였다. 2 x 5를 인자로 대입하였으므로 2x5의 곱셈 연산이 되어, 10이 출력되었다.

```
jenny@ubuntu:~/OSS$ bash ./calc.sh 10 % 3
1
```

- Bash shell을 이용하여 calc.sh 프로그램을 실행하였다. 10 % 3를 인자로 대입하였으므로 10을 3으로 나눈 나머지 값인 1이 출력되었다.

(2)

```
int size;
int* input;
int* temp;
int max;
int i;

scanf("%d", &size);

input = (int*)malloc(size*4);
temp = (int*)malloc(size*4);

for(i=0;i<size;i++)
{
    scanf("%d", &input[i]);
    temp[i] = input[i];
}

max = input[0];

for(i=1;i<size;i++)
{
    if(input[i]<=temp[i-1]+input[i])
    {
        temp[i] = temp[i-1]+input[i];
    }

    if(max < temp[i])
    {
        max = temp[i];
    }
}
printf("%d\n",max);
return 0;
```

- 전체 코드

```
int size;
int* input;
int* temp;
int max;
int i;
```

- 배열의 크기를 저장할 size변수, 사용자로부터 입력 받은 배열을 저장할 input 배열(동적 할당), 최대값을 계산하기 위해 각각 덧셈을 임시로 저장할 temp배열, 최대값을 저장할 max변수를 선언해 주었다.

```
scanf("%d", &size);

input = (int*)malloc(size*4);
temp = (int*)malloc(size*4);
```

- 사용자로부터 수열의 크기를 입력 받고, int형 배열을 동적할당 해주었다.

```
for(i=0;i<size;i++)
{
    scanf("%d", &input[i]);
    temp[i] = input[i];
}
```

- 사용자로부터 수열을 input배열에 입력 받고 이를 temp배열에 똑같이 복사해준다.

```
max = input[0];
```

- 최대값의 초기값을 사용자 input의 맨 처음 값으로 대입한다.

```

for(i=1;i<size;i++)
{
    if(input[i]<=temp[i-1]+input[i])
    {
        temp[i] = temp[i-1]+input[i];
    }

    if(max < temp[i])
    {
        max = temp[i];
    }
}

```

- 배열의 두번째 값부터 비교하며 최대값을 만든다. 만약 해당 배열의 값이 (이전 temp의 값 + 현재 배열 값) 보다 작다면 (이전 temp의 값 + 현재 배열 값)에 현재 배열 값을 더 해준다. 그리고 현재 temp배열에 덧셈 값을 저장한다. 만약 현재 배열 값이 더 크다면 그 값이 max값이 되므로 해당 temp값은 변하지 않는다.

temp값을 수정한 후 최대값을 업데이트해주어야 한다. 만약 max값 보다 현재 temp 값이 더 크다면 그 값이 max값이 된다.

```

jenny@ubuntu:~/OSS/Assigntment1$ ./run
6
4 -3 2 10 -5 1
13

```

- 배열 4 -3 2 10 -5 1에서 합의 최대값은 $4+(-3)+2+(10) = 13$

```

jenny@ubuntu:~/OSS/Assigntment1$ ./run
10
10 -4 3 1 5 6 -35 12 21 -1
33

```

- 배열 10 -4 3 1 5 6 -35 12 21 -1에서 합의 최대값은 $12+21 = 33$

```

jenny@ubuntu:~/OSS/Assigntment1$ ./run
4
1 2 3 -1
6

```

- 배열 1 2 3 -1에서 합의 최대값은 $1+2+3 = 6$

```

jenny@ubuntu:~/OSS/Assigntment1$ ./run
4
1 2 -1 3
5

```

- 배열 1 2 -1 3에서 합의 최대값은 $1+2+(-1)+(3) = 5$

```

jenny@ubuntu:~/OSS/Assigntment1$ ./run
6
4 -3 2 10 -5 1
13

```

- 배열 4 -3 2 10 -5 1에서 합의 최대값은 $4+(-3)+2+10 = 13$

[디버깅]

- 배열값을 1 2 3 -1을 주었다.

Expression	Type	Value
input[i]	int	2
temp[i]	int	2
max	int	1
Add new expression		

- max값을 input의 맨 처음 값을 배열의 처음 값을 주었으므로 max = 1이된다.

Expression	Type	Value
input[i]	int	2
temp[i]	int	3
max	int	3
Add new expression		

- 이전 temp배열의 값과 input배열의 현재 값을 더한 것이 3, max=1보다 크므로 max=3, 현재 temp값 = 3이 된다.

Expression	Type	Value
input[i]	int	3
temp[i]	int	6
max	int	6
Add new expression		

- 이전 temp배열의 값과 input배열의 현재 값을 더한 것이 6, max=3보다 크므로 max=6, 현재 temp값 = 6이 된다.

Expression	Type	Value
input[i]	int	-1
temp[i]	int	-1
max	int	6
Add new expression		

- 현재 input배열의 값은 -1이다.

Expression	Type	Value
input[i]	int	-1
temp[i]	int	5
max	int	6
Add new expression		

- 이전 temp배열의 값과 input배열의 현재 값을 더한 것이 5, max=6보다 작으므로 max값은 변하지 않고 temp값은 5가 된다.

3. 고찰

Assignment1을 하면서 중요하거나 새롭게 배운 것이 3가지가 있다.

첫번째는 bash shell이 무엇이며 어떻게 사용하느냐 이다. shell은 사용자가 입력한 명령을 해석하여 커널에 전달하고 연산하는 것으로 Assignment1에서 사용한 shell은 bash shell이다. 여러가지 문법이 있지만 과제에서는 문자열 비교, 조건문에 대한 내용만 필요했다. 코드를 작성하기에 앞서 셸 스크립트의 첫 줄은 `#!/bin/bash`로 시작해야한다. 이는 스크립트 해석기가 bash shell임을 알려준다. 사용자가 입력한 인자에서 연산자를 구분하는 부분에서 숫자 비교 연산자와 같은 방법을 사용하면 오류가 발생한다. 필자는 일치하는 경우의 조건을 주었기에 '='를 사용하였다. 이외에도

제어문	설명
<code>-z string</code>	문자열의 길이가 0인 경우 true를 반환
<code>-n string</code>	문자열의 길이가 0이 아닌 경우 true를 반환
<code>string1 != string2</code>	두개의 문자열이 일치하지 않는 경우
<code>string</code>	문자열이 NULL이 아닌 경우

의 경우가 있다. 연산자를 확인했다면 연산한 결과를 프롬프트에 출력하기 위해 `echo $((연산식))`을 이용했다. 또한 프로그램 실행 시 입력한 인자를 사용하기 위해 '\$인자번호'을 이용하였다.

두번째는, 배열의 동적할당이다. 평소 사용했던 동적할당은 char형 배열에 대한 동적 할당이였다. 그래서 동적할당의 크기를 배열의 길이로 설정해 주었다. Assignment1-2를 디버깅하면서 10개의 인자를 받을 때 배열에 엉뚱한 값이 들어가 있음을 확인하였다. 처음에는 for문의 조건을 잘못 설정했다고 생각하여 직접 그려보며 차근차근 단계를 생각해보았다. 하지만 알고리즘은 문제가 없음을 확인했다. 그렇다면 배열의 문제라고 생각하여 동적할당의 크기를 (배열의 길이 * 변수의 크기)로 설정했더니, 정상 출력되었다. 이제까지는 char형 배열이어서 배열의 길이를 동적할당의 크기로 설정하여도 char형 변수 크기가 1byte여서 문제가 되지 않았다. 하지만 int형 배열은 변수가 4byte이므로 (배열의 길이 * 4byte)를 해준 값으로 동적할당을 해줘야 한다.

4. Reference

1. 동적할당 - <https://blog.naver.com/85271/220626342607>
2. 우분투 code 디버깅 - <http://webnautes.tistory.com/905>