2020년 1학기

프로그래밍언어 과제3 : Toy Language [Version 2]

컴퓨터학부

20162481 안승훈

1. 개요

- 프로그래밍을 이용해 Toy Language의 interpreter를 작성한다.

- Toy Language의 Interpreter 의 구현을 통해 Parser 및 interpretive 언어에 대한 이해를 높인다.

- Stack를 이용하여 interpreter의 역할을 수행한다.

2. 구현 방법

- Visual Studio 프로그램과 C++언어를 이용하여 설계하였다.

- 파일 로드와 스택을 이용하여 인터프리팅을 할 수 있도록 구현하였다.

- 사용자로부터 파일의 이름을 입력 받아 파일을 열어볼 수 있도록 설계하였다.

- 스택을 사용하여 결과 값을 계산할 수 있도록 하였다.

- DEFUN을 이용하여 사용자가 직접 정의한 명령어를 저장, 불러올 수 있도록 하였다.

3. 구현한 자료구조 및 함수 설명

- C++을 이용하여 프로그램을 작성하였다

* Main 함수를 비롯한 다른 함수

**< int main(void) >**

**int** main**(void)**

**{**

**int** choose**;**

**while** **(1)** **{**

choose **=** menu\_select**();**

**if** **(**choose **==** **1)** **{**

menu1**();**

**}**

**else** **if** **(**choose **==** **2)** **{**

menu2**();**

**}**

**else** **if** **(**choose **==** **3)** **{**

inperpreter**();**

**}**

**else** **if** **(**choose **==** **4)** **{**

printf**(**"종료중....\n"**);**

**break;**

**}**

**}**

**return** **0;**

**}**

- main함수에서는 menu\_select() 함수를 통해 사용자가 입력하는 번호에 따라 Define DEFUN, Load DEFUN, 인터렉티브 모드, 프로그램 종료의 기능을 분기할 수 있도록 한다.

- if - else if 문을 이용하여 각각의 선택에 따라 분기하도록 하였다.

**< int menu\_select(void), void print\_menu(void) >**

**int** menu\_select**(void)** **{**

**char** input\_c **=** **0;**

**int** input\_i **=** **0;**

print\_menu**();**

**while** **(1)** **{**

scanf**(**"%c"**,** **&**input\_c**);**

getchar**();**

**if** **(**input\_c **==** '1' **||** input\_c **==** '2' **||** input\_c **==** '3' **||** input\_c **==** '4'**)** **{**

input\_i **=** input\_c **-** '0'**;**

**return** input\_i**;**

**}**

**else** **{**

printf**(**"다시 입력하세요\n"**);**

printf**(**"메뉴를 선택하세요 >> "**);**

**}**

**}**

**}**

**void** print\_menu**(void)** **{**

printf**(**"=========================\n"**);**

printf**(**"1. Define DEFUN\n"**);**

printf**(**"2. Print DEFUN\n"**);**

printf**(**"3. Interpreter\n"**);**

printf**(**"4. Exit\n"**);**

printf**(**"=========================\n"**);**

printf**(**"메뉴를 선택하세요>> "**);**

**}**

- menu\_select() 함수는 사용자로부터 메뉴의 선택을 입력 받는 함수이다.

- print\_menu() 함수를 통해 사용자의 선택지를 출력한 후, 사용자로부터 번호를 입력 받는다.

- 사용자로부터 입력 받은 것이 올바른 선택지가 아닐 경우, 사용자로부터 올바른 입력이 이루어질 때까지 재입력을 받을 수 있도록 설계하였다.

- 입력이 제대로 된 경우, 사용자의 선택을 리턴한다.

**< int check( char buf[ ] ) >**

**int** check(**char** buf[]) {

**int** lcnt = 0, rcnt = 0;

**int** len = strlen(buf);

*//괄호 갯수 체크*

**for** (**int** i = 0; i < len; i++) {

**if** (buf[i] == '(')

lcnt++;

**else** **if** (buf[i] == ')')

rcnt++;

}

**if** (lcnt > rcnt) {

printf("')' 갯수가 부족합니다.\n");

**return** -1;

}

**else** **if** (lcnt < rcnt) {

printf("'(' 갯수가 부족합니다.\n");

**return** -1;

}

*// -가 여러개 쓰인 경우 체크*

**for** (**int** i = 0; i < len; i++) {

**if** (buf[i] == '-') {

**if** (buf[i + 1] == '-') {

printf("'-'사용이 잘못되었습니다.\n");

**return** -1;

}

}

}

**return** 1;

}

- check() 함수는 프로그램 파일에서 읽어온 문장이 올바른 형식인지 확인하는 함수이다.

- 괄호 개수를 비교하여, 괄호가 부족할 경우 에러 메시지를 출력하고, -1을 리턴한다.

- ( – ) 기호가 2개 이상 사용된 경우, 사용 오류가 있음을 출력하고 -1을 리턴한다.

- 위의 오류가 없는 경우 1을 리턴한다.

**<void menu1(void) >**

**void** menu1**(void)** **{** *//1번 메뉴용 함수*

**FILE** **\***fp**;**

**char** input**[**BUF\_SIZE**];**

memset**(**input**,** **0,** **sizeof(**input**));**

**if** **((**fp **=** fopen**(**"defun.txt"**,** "a+"**))** **==** **0)** **{**

fprintf**(**stderr**,** "fopen error for defun.txt\n"**);**

**return;**

**}**

fseek**(**fp**,** **0,** SEEK\_END**);**

fgets**(**input**,** BUF\_SIZE**,** stdin**);**

fputs**(**input **+** **6,** fp**);**

**return;**

**}**

- 사용자가 1번을 선택했을 때 호출 되는 함수이다.

- defun.txt를 오픈하고, 파일에 사용자가 정의한 명령어를 추가한다.

**<void menu2(void) >**

**void** menu2**(void)** **{**

**FILE** **\***fp**;**

**char** buf**[**BUF\_SIZE**];**

**if** **((**fp **=** fopen**(**"defun.txt"**,** "r+"**))** **==** **0)** **{**

fprintf**(**stderr**,** "fopen error for defun.txt\n"**);**

**return;**

**}**

printf**(**"\n"**);**

**while** **(**fgets**(**buf**,** BUF\_SIZE**,** fp**)** **!=** NULL**)** **{**

fputs**(**buf**,** stdout**);**

**}**

printf**(**"\n"**);**

**return;**

**}**

- 사용자가 2번을 선택했을 때, 호출되는 함수이다.

- defun.txt를 읽어온 후, defun.txt에 저장되어있는 정의한 명령어를 출력한다.

**<void interpreter(void) >**

**void** inperpreter**(void)** **{**

**FILE** **\***fp**;**

**char** fname**[**BUF\_SIZE**];**

**char** buf**[**BUF\_SIZE**];**

**int** ck**;**

printf**(**"실행 파일 명을 입력하세요 >> "**);**

fgets**(**fname**,** BUF\_SIZE**,** stdin**);**

fname**[**strlen**(**fname**)** **-** **1]** **=** '\0'**;**

**if** **((**fp **=** fopen**(**fname**,** "r"**))** **==** **0)** **{**

fprintf**(**stderr**,** "%s file open error!\n"**,** fname**);**

**return;**

**}**

**while** **(**fgets**(**buf**,** BUF\_SIZE**,** fp**)** **!=** NULL**)** **{**

printf**(**"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"**);**

buf**[**strlen**(**buf**)** **-** **1]** **=** '\0'**;**

fputs**(**buf**,** stdout**);**

ck **=** check**(**buf**);**

**if** **(**ck **<** **0)** **{**

printf**(**"\n문장 오류\n"**);**

**continue;**

**}**

printf**(**" -> "**);**

*//DEFUN에 있는 것을 변환하는 함수*

*//make\_defun();*

printf**(**"\nPrefix to Postfix : "**);**

*//Postfix 출력*

*//make\_postfix();*

*//postfix로 만드는 함수*

printf**(**"\nResult : %d\n"**,** calcul**(**buf**));**

**}**

**}**

- 사용자로부터 인터프리팅 할 내용이 저장된 txt 파일을 입력받는다.

- 파일에 쓰여진 내용에 DEFUN에 저장된 명령어가 포함되어있는 경우 환산한다.

- 이를 후위표기식으로 출력한 후, 인터프리팅하여 계산 결과를 출력한다.

**<void make\_defun(char buf[]) >**

- 프로그램 파일에서 읽어온 문장에 defun에 정의된 명령어가 포함되어있는 경우, 이를 환산하는 함수이다.

- defun에 정의된 명령어를 모두 MINUS, IF로만 구성되게 변환하는 작업을 하는 함수이다.

**<void make\_postfix(char buf[]) >**

- make\_defun() 함수로 변환함 프로그램 문장을 후위표기식으로 변환하는 함수이다.

- 함수를 통해 문장을 후위표기식으로 변환하여 사용자에게 출력하고, 계산을 할 수 있게 한다.

**<int calcul(char buf[]) >**

**int** calcul**(char** buf**[])** **{**

**int** cur**,** x **=** **0;**

**int** len **=** strlen**(**buf**);**

**char** tmp**[10];**

memset**(**tmp**,** **0,** **10);**

**for** **(**cur **=** **0;** cur **<** len**;** **)** **{**

**if** **(**buf**[**cur**]** **==** '('**)** **{**

cur**++;**

**continue;**

**}**

**else** **if** **(**buf**[**cur**]** **==** 'M'**)** **{**

cur **+=** **5;**

opst**.**push**(**'M'**);**

**}**

**else** **if** **(**buf**[**cur**]** **==** 'I'**)** **{**

cur **+=** **2;**

opst**.**push**(**'I'**);**

**}**

**else** **if** **(**buf**[**cur**]** **==** '-'**)** **{**

x **=** **0;**

memset**(**tmp**,** **0,** **10);**

tmp**[**x**++]** **=** buf**[**cur**++];**

**while** **(**'0' **<=** buf**[**cur**]** **&&** buf**[**cur**]** **<=** '9'**)**

tmp**[**x**++]** **=** buf**[**cur**++];**

inst**.**push**(**atoi**(**tmp**));**

**}**

**else** **if** **(**'0' **<=** buf**[**cur**]** **&&** buf**[**cur**]** **<=** '9'**)** **{**

x **=** **0;**

memset**(**tmp**,** **0,** **10);**

**while** **(**'0' **<=** buf**[**cur**]** **&&** buf**[**cur**]** **<=** '9'**)**

tmp**[**x**++]** **=** buf**[**cur**++];**

inst**.**push**(**atoi**(**tmp**));**

**}**

**else** **if** **(**buf**[**cur**]** **==** ' ' **||** buf**[**cur**]** **==** '\n'**)** **{**

cur**++;**

**continue;**

**}**

**else** **if** **(**buf**[**cur**]** **==** ')'**)** **{**

**char** op **=** opst**.**top**();**

opst**.**pop**();**

**int** b **=** inst**.**top**();**

inst**.**pop**();**

**int** a **=** inst**.**top**();**

inst**.**pop**();**

**if** **(**op **==** 'M'**)**

inst**.**push**(**a **-** b**);**

**else** **if** **(**op **==** 'I'**)** **{**

**if** **(**a **<=** **0)**

inst**.**push**(0);**

**else**

inst**.**push**(**b**);**

**}**

cur**++;**

**}**

**}**

**int** res **=** inst**.**top**();**

inst**.**pop**();**

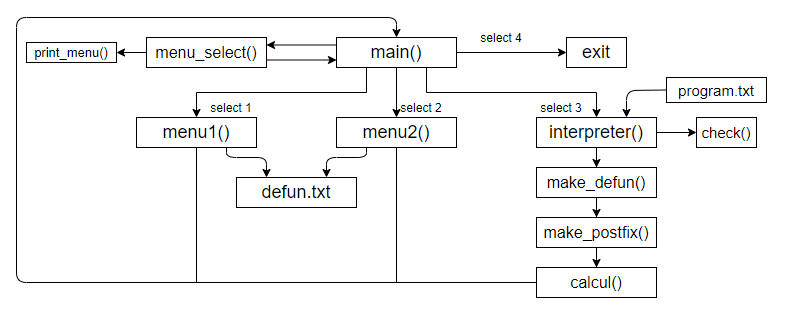
**return** res**;**

**}**

- make\_postfix() 함수를 통해 후위표기식으로 변환된 프로그램 문장을 이용하여 결과 값을 계산하는 함수이다.

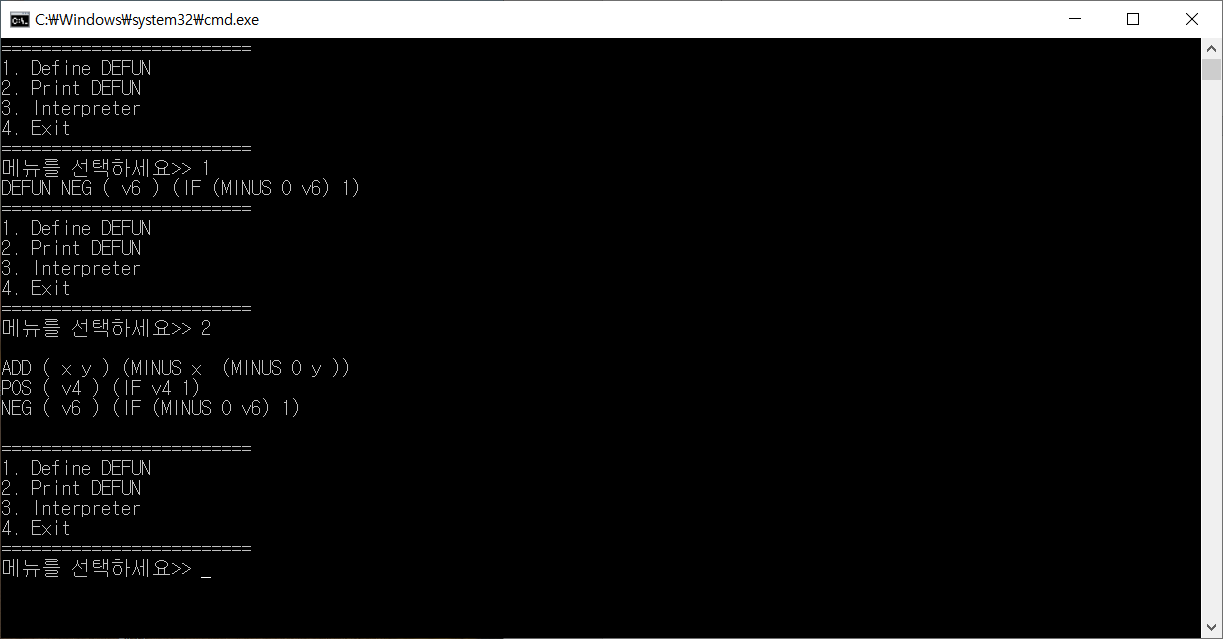
- 결과값을 계산하여 리턴한다.

4. 함수 흐름도



- 프로그램에서 사용한 함수들의 흐름도를 flow chart 형식으로 작성하였다.

5. 실행 결과

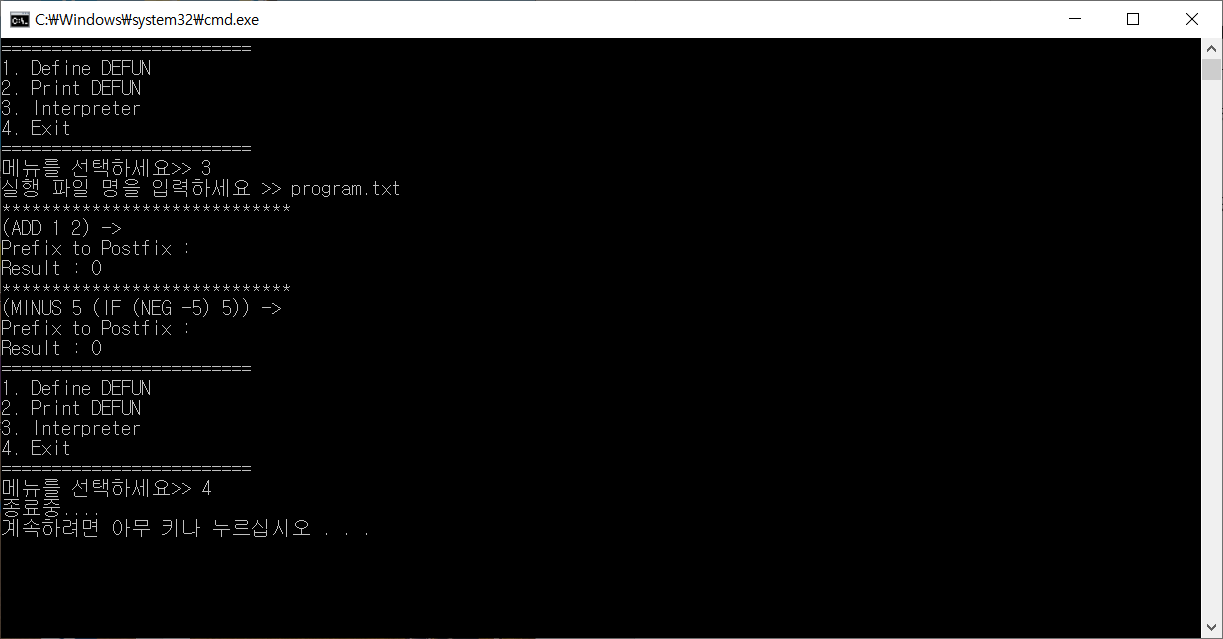


- 1번 메뉴를 선택하고, 새로운 DEFUN을 입력하는 과정이다.

- 사용자로부터 새로운 명령어를 입력받아서 defun.txt에 저장한다.

- 이어서 2번 메뉴를 선택하면, 1번 메뉴를 선택하고 입력한 새 명령어가 추가 되어있다.

- 사용자가 과거에 추가한 명령어도 defun.txt에 저장하고 있다가 2번 메뉴를 선택했을 때 함꼐 출력한다.



- 3번 메뉴를 선택한 실행 화면이다.

- program.txt 파일을 읽은 후에, 각 문장에 대해서 DEFUN을 적용하여 문장을 변환한다.

- 변환한 문장을 후위표기식으로 한번 더 변환한다.

- 후위표기식으로 표현된 식을 통해 계산을 진행한다.

- 계산한 결과를 사용자에게 출력하여 준다.

- 4번 메뉴를 선택하면 프로그램을 종료한다.

6. 전체 소스코드

#include <iostream>

#include <algorithm>

#include <vector>

#include <cstdio>

#include <cstdlib>

#include <cstring>

#include <stack>

#define BUF\_SIZE 1024

using namespace std;

void print\_menu(void);

int menu\_select(void);

void menu1(void);

void menu2(void);

void inperpreter(void);

int calcul(char buf[]);

int check(char buf[]);

stack <char> opst;

stack <int> inst;

int main(void)

{

int choose;

while (1) {

choose = menu\_select();

if (choose == 1) {

menu1();

}

else if (choose == 2) {

menu2();

}

else if (choose == 3) {

inperpreter();

}

else if (choose == 4) {

printf("종료중....\n");

break;

}

}

return 0;

}

void menu1(void) { //1번 메뉴용 함수

FILE \*fp;

char input[BUF\_SIZE];

memset(input, 0, sizeof(input));

if ((fp = fopen("defun.txt", "a+")) == 0) {

fprintf(stderr, "fopen error for defun.txt\n");

return;

}

fseek(fp, 0, SEEK\_END);

fgets(input, BUF\_SIZE, stdin);

fputs(input + 6, fp);

fclose(fp);

return;

}

void menu2(void) {

FILE \*fp;

char buf[BUF\_SIZE];

if ((fp = fopen("defun.txt", "r+")) == 0) {

fprintf(stderr, "fopen error for defun.txt\n");

return;

}

printf("\n");

while (fgets(buf, BUF\_SIZE, fp) != NULL) {

fputs(buf, stdout);

memset(buf, 0, sizeof(buf));

}

printf("\n");

return;

}

void inperpreter(void) {

FILE \*fp;

char fname[BUF\_SIZE];

char buf[BUF\_SIZE];

int ck, res = 0;

printf("실행 파일 명을 입력하세요 >> ");

fgets(fname, BUF\_SIZE, stdin);

if(fname[strlen(fname) - 1] == '\n')

fname[strlen(fname) - 1] = '\0';

if ((fp = fopen(fname, "r")) == 0) {

fprintf(stderr, "%s file open error!\n", fname);

return;

}

while (fgets(buf, BUF\_SIZE, fp) != NULL) {

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

if (fname[strlen(buf) - 1] == '\n')

buf[strlen(buf) - 1] = '\0';

fputs(buf, stdout);

ck = check(buf);

if (ck < 0) {

printf("\n문장 오류\n");

continue;

}

printf(" -> ");

//DEFUN에 있는것을 변환하는 함수

//make\_defun();

printf("\nPrefix to Postfix : ");

//Postfix 출력

//make\_postfix();

//postfix로 만드는 함수

//res = calcul(buf);

printf("\nResult : %d\n", res);

}

}

int calcul(char buf[]) {

int cur, x = 0;

int len = strlen(buf);

char tmp[10];

memset(tmp, 0, 10);

for (cur = 0; cur < len; ) {

if (buf[cur] == '(') {

cur++;

continue;

}

else if (buf[cur] == 'M') {

cur += 5;

opst.push('M');

}

else if (buf[cur] == 'I') {

cur += 2;

opst.push('I');

}

else if (buf[cur] == '-') {

x = 0;

memset(tmp, 0, 10);

tmp[x++] = buf[cur++];

while ('0' <= buf[cur] && buf[cur] <= '9')

tmp[x++] = buf[cur++];

inst.push(atoi(tmp));

}

else if ('0' <= buf[cur] && buf[cur] <= '9') {

x = 0;

memset(tmp, 0, 10);

while ('0' <= buf[cur] && buf[cur] <= '9')

tmp[x++] = buf[cur++];

inst.push(atoi(tmp));

}

else if (buf[cur] == ' ' || buf[cur] == '\n') {

cur++;

continue;

}

else if (buf[cur] == ')') {

char op = opst.top();

opst.pop();

int b = inst.top();

inst.pop();

int a = inst.top();

inst.pop();

if (op == 'M')

inst.push(a - b);

else if (op == 'I') {

if (a <= 0)

inst.push(0);

else

inst.push(b);

}

cur++;

}

}

int res = inst.top();

inst.pop();

return res;

}

int check(char buf[])

{

int lcnt = 0, rcnt = 0;

int len = strlen(buf);

//괄호 갯수 체크

for (int i = 0; i < len; i++) {

if (buf[i] == '(')

lcnt++;

else if (buf[i] == ')')

rcnt++;

}

if (lcnt > rcnt) {

printf("')' 갯수가 부족합니다.\n");

return -1;

}

else if (lcnt < rcnt) {

printf("'(' 갯수가 부족합니다.\n");

return -1;

}

// -가 여러개 쓰인 경우 체크

for (int i = 0; i < len; i++) {

if (buf[i] == '-') {

if (buf[i + 1] == '-') {

printf("'-'사용이 잘못되었습니다.\n");

return -1;

}

}

}

return 1;

}

int menu\_select(void) {

char input\_c = 0;

int input\_i = 0;

print\_menu();

while (1) {

scanf("%c", &input\_c);

getchar();

if (input\_c == '1' || input\_c == '2' || input\_c == '3' || input\_c == '4') {

input\_i = input\_c - '0';

return input\_i;

}

else {

printf("다시 입력하세요\n");

printf("메뉴를 선택하세요 >> ");

}

}

}

void print\_menu(void) {

printf("=========================\n");

printf("1. Define DEFUN\n");

printf("2. Print DEFUN\n");

printf("3. Interpreter\n");

printf("4. Exit\n");

printf("=========================\n");

printf("메뉴를 선택하세요>> ");

}