**LAB ASSIGNMENT NO. 1**

Name: Dhawal Sakharwade

Roll No. : - 56

PRN no. : 12111400

**Problem Statement : Create a symbol table for input file written in 'C" language.**

Code:

# include <stdio.h>

# include <stdlib.h>

# include <string.h>

# include <ctype.h>

# define MAX\_VARIABLES 100

# define MAX\_NAME\_LENGTH 50

struct Variable {

char name[MAX\_NAME\_LENGTH];

char type[MAX\_NAME\_LENGTH];

int length;

int address;

};

struct Variable symbolTable[MAX\_VARIABLES];

int symbolCount = 0;

void addToSymbolTable(const char\* name, const char \*type, size\_t length, int address, int index){

if(index < MAX\_VARIABLES){

strcpy(symbolTable[index].name, name);

strcpy(symbolTable[index].type, type);

symbolTable[index].length = length;

symbolTable[index].address = address;

index++;

}

}

void updateSymbolTable(const char \*type, size\_t length, int index){

if(index < MAX\_VARIABLES){

strcpy(symbolTable[index].type, type);

symbolTable[index].length = length;

index++;

}

}

int main(){

FILE\* file = fopen("example.c", "r");

char line[100];

int address = 1000;

int index = 0;

char\* type = NULL;

while(fgets(line, sizeof(line), file)){

// printf("1");

char \*token = strtok(line, " ();,\n\t");

// printf("%s \n", token);

while(token != NULL){

// printf("%s \n", token);

if(strcmp(token, "int") == 0 || strcmp(token, "char") == 0 || strcmp(token, "float") == 0 || strcmp(token, "double") == 0){

size\_t size = -1;

if(strcmp(token, "int") == 0){

size = sizeof(int);

type = "int";

} else if(strcmp(token, "char") == 0){

size = sizeof(char);

type = "char";

} else if(strcmp(token, "float") == 0){

size = sizeof(float);

type = "float";

} else if(strcmp(token, "double") == 0){

size = sizeof(double);

type = "double";

}

token = strtok(NULL, " ();,\n\t");

// printf("%s \n", token);

while(token != NULL && token[0] != ';'){

// printf("%s \n", token);

//! Handling the case if the syntax used for writing the array is array [100];

//? Updating the symbol table if the syntax for writing the array is array [100]

if(token[0] == '['){

int length = atoi(strtok(token, "["));

printf("%d ", length);

updateSymbolTable(strdup("array"), size \* length, index - 1);

address += size \* length;

}

//! Handling the case if the syntax used for writing the array is array[100];

if(strchr(token, '[') != NULL){

char \*name = strtok(token, "[");

int length = atoi(strtok(NULL, "]"));

printf("%d", length);

} else if(isalpha(token[0]) || token[0] == "\_"){

if(strcmp(token, "main") != 0){

addToSymbolTable(token, strdup(type), size, address, index);

address += size;

index ++;

}

}

token = strtok(NULL, " ();,\n\t");

}

} else {

char\* temp = strdup(token);

token = strtok(NULL, " ();,\n\t");

printf("%s \n", token);

}

token = strtok(NULL, " ();,\n\t");

}

}

fclose(file);

printf("Symbol Table:\n");

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

printf("| %-10s | %-10s | %-10s | %-10s |\n", "Symbol", "Type", "Length", "Address");

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

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

printf("| %-10s | %-10s | %-10d | %-10d |\n", symbolTable[i].name, symbolTable[i].type, symbolTable[i].length, symbolTable[i].address);

}

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

}

**Input File:**

**#include <stdio.h>**

**int globalVar = 42;**

**float DJJ;**

**void exampleFunction(int param1, char param2) {**

**int localVar = param1 + globalVar;**

**printf("Result: %d\n", localVar);**

**}**

**int main() {**

**int x = 10;**

**char y = 'A';**

**exampleFunction(x, y);**

**return 0;**

**}**

**Output:**

**Symbol Table:**

**-------------------------------------------------**

**| Symbol | Type | Length | Address |**

**-------------------------------------------------**

**| globalVar | int | 4 | 1000 |**

**| DJJ | float | 4 | 1004 |**

**| param1 | int | 4 | 1008 |**

**| char | int | 4 | 1012 |**

**| param2 | int | 4 | 1016 |**

**| localVar | int | 4 | 1020 |**

**| param1 | int | 4 | 1024 |**

**| globalVar | int | 4 | 1028 |**

**| x | int | 4 | 1032 |**

**| y | char | 1 | 1036 |**

**-------------------------------------------------**