CSB 353: Compiler Design

LAB 1

Submitted By:

Name: PREM KUMAR

Roll No: 191210037

Branch: CSE

Semester: 6 th

Submitted To: Dr. Shelly Sachdeva

Department of Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY DELHI

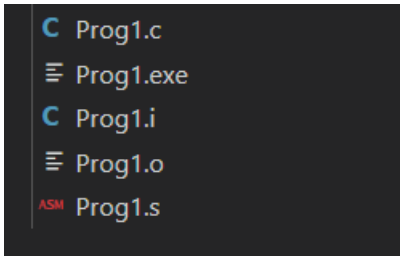
2019-2023

Ques 1. Identify the files created while compilation and execution of a C program.

Sol.

Compiling Prog1.c (C Program File)

```
PS C:\Users\Prem\Desktop\6thSem\CSB353> gcc .\Prog1.c -o Prog1 -save-temps
PS C:\Users\Prem\Desktop\6thSem\CSB353>
```



- When we compile a .c file, it is first passed to the preprocessor for code substitutions.
- The Preprocessor substitutes codes from header files and macros defined by #define. All of these header files and macros are substituted from preprocessor directives. After substituting the code, the preprocessor generates the .i file.
- The .i file contains substituted code which further passed to the compiler which converts the .i file into a .s file which contains assembly language code.
- The .s file is then passed to the assembler for further processing which converts the .s file into .o file which contains the object code.
- The .o file is an incomplete object file as it does not contain references to external subroutines and therefore cannot be executed directly by the operating system.
- When we execute a .c file, then the .o file is passed through a linker which performs linking and generates a .exe file.
- The .exe file is a complete object file with references to external subroutines and it can be executed directly by the operating system.

.Ques2. Create a C program for generating a Lexical Analyzer to identify the following tokens:

- Keywords; Examples-for, while, if etc.
- Identifier; Examples-Variable name, function name, etc.
- Operators; Examples '+', '++', '-' etc.
- Separators; Examples ',' ';' etc.

Program:

CSB353 > C lab1.c > main()

```c
#include <stdbool.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
bool isValidOperator(char ch)
{
    if (ch == '+' ||
        ch == '-' ||
        ch == '*' ||
        ch == '/' ||
        ch == '>' ||
        ch == '<' ||
        ch == '=')
        return true;
    return false;
}

bool isValidDelimiter(char ch)
{
    if (ch == ' ' ||
        ch == ',' ||
        ch == ';' ||
        ch == '(' ||
        ch == ')' ||
        ch == '[' ||
        ch == ']' ||
        ch == '{' ||
        ch == '}' || isValidOperator(ch))
        return true;
    return false;
}

```

```
bool isValidKeyword(char *str)
{
    if (!strcmp(str, "main") ||
        !strcmp(str, "int") ||
        !strcmp(str, "char") ||
        !strcmp(str, "if") ||
        !strcmp(str, "else") ||
        !strcmp(str, "for") ||
        !strcmp(str, "while") ||
        !strcmp(str, "break") ||
        !strcmp(str, "continue") ||
        !strcmp(str, "return"))
        return true;
    return false;
}

bool isValidDigit(char ch)
{
    int asciiValue = ch;
    if (asciiValue >= 48 && asciiValue <= 57)
        return true;
    return false;
}
bool isValidLetter(char ch)
{
    int asciiValue = ch;
    if ((asciiValue >= 65 && asciiValue <= 90) || (asciiValue >= 97 && asciiValue <= 122))
        return true;
    return false;
}
```

```c
bool isvalidIdentifier(char *str)
{
    int i, len = strlen(str);
    if (len == 0)
        return false;
    if (isValidLetter(str[0]))
    {
        for (i = 1; i < len; i++)
        {
            if (!isValidLetter(str[i]) && !isValidDigit(str[i]))
            {
                return false;
            }
        }
        return true;
    }
    else
    {
        return false;
    }
}

bool isValidInteger(char *str)
{
    int i, len = strlen(str);
    if (len == 0)
        return false;
    for (i = 0; i < len; i++)
    {
        if (!isValidDigit(str[i]))
            return false;
    }
    return true;
}
```

```c
bool isRealNumber(char *str)
{
    int i, len = strlen(str);
    bool hasDecimal = false;
    if (len == 0)
        return false;
    for (i = 0; i < len; i++)
    {

        if (str[i] == '.')
            hasDecimal = true;
        else if (!isValidDigit(str[i]))
            return false;
    }
    return hasDecimal;
}
char *generateSubString(char *str, int start, int end)
{
    int i;
    char *subStr = (char *)malloc(sizeof(char) * (end - start + 2));
    for (i = start; i <= end; i++)
        subStr[i - start] = str[i];
    subStr[end - start + 1] = '\0';
    return subStr;
}
```

```c
void detectAllTokens(char *str)
{
    int start = 0, end = 0;
    int length = strlen(str);
    while (start <= end && end <= length)
    {
        if (!isValidDelimiter(str[end]))
            end++;
        if (isValidDelimiter(str[end]) && start == end)
        {
            if (isValidOperator(str[end]))
                printf("Valid operator : '%c'\n", str[end]);
            end++;
            start++;
        }
        else if (isValidDelimiter(str[end]) && start != end || (end == length && start != end))
        {
            char *subStr = generateSubString(str, start, end - 1);
            if (isValidKeyword(subStr))
                printf("Valid keyword : '%s'\n", subStr);
            else if (isValidInteger(subStr))
                printf("Valid Integer : '%s'\n", subStr);
            else if (isRealNumber(subStr))
                printf("Real Number : '%s'\n", subStr);
            else if (isvalidIdentifier(subStr) && !isValidDelimiter(str[end - 1]))
                printf("Valid Identifier : '%s'\n", subStr);
            else if (!isvalidIdentifier(subStr) && !isValidDelimiter(str[end - 1]))
                printf("Invalid Identifier : '%s'\n", subStr);
            start = end;
        }
    }
    return;
}
```

```c
int main()
{
    char str[100] = "int a = b + 100; ";
    printf("All Tokens in '%s' are - \n", str);
    detectAllTokens(str);

    return 0;
}
```

Output:

```
PS C:\Users\Prem\Desktop\6thSem> cd "c:\Users\Prem\Desktop\6thSem\CSB353\" ; if ($?) { gcc lab1.c -o lab1 } ; if ($?) { .\lab1 }
All Tokens in 'int a = b + 100; ' are -
Valid keyword : 'int'
Valid Identifier : 'a'
Valid operator : '='
Valid Identifier : 'b'
Valid operator : '+'
Valid Integer : '100'
PS C:\Users\Prem\Desktop\6thSem\CSB353>
```