Project Report On

Mini C COMPILER

PROJECT MEMBERS :

Nithin Kumar B	**4NM09CS066**
Dattatraya B Mukri	**4NM10CS103**
Raghavendra Salotagi	**4NM09CS078**
Mihir Nath Pandya	**4NM09CS052**
Nayan Kumar	**4NM09CS061**

PROJECT GUIDES

Mr. Sunil Kumar B L

Senior Lecturer, Dept. of CSE

ABSTRACT

This project aims to provide a new compiler for Mini C language that is a subset of C language. A compiler is a set code that convert high level program (here C program) to assembly level program or machine level program (binary code). In this project, we try to build a Mini C Compiler For Testing and implementing simple arithmetic instructions involving integers as operands. The code converts a subset of 'C' language to assembly language. It also spots syntactic errors. We intend to build a compiler which has efficient memory utilization and more compile time efficiency.

CONTENTS

INTRODUCTION

A compiler is essentially a translator that reads in some code in an input language and output the result in another language. Our project aims on developing a mini C compiler that follows ANSI C standards. The compiler takes in a C program as its input and produces the corresponding machine code as its output. The C compiler is developed using C++ language itself. There are mainly 6 phases for a compiler: the lexical analyzer, the syntax analyzer.

PROBLEM OVERVIEW

Our compiler targets the Intel-8086 architecture. The input to the compiler is a C programming language. The output is an assembly.
The main modules in our project are:

Module-1: This is a main module which will later call other modules appropriately.EXE version of this module can be used along with command line arguments to run the complete Mini-C compiler.
Module-2: This creates the lexical analyzer for tokenizing the input code. Also contains functions for detecting syntactic errors.

PROBLEM DEFINITION

Our project requires the following:

- An input file, which is a program in C language.
- Gcc compiler is used to develop the code of our project

REQUIREMENT ANALYSIS

This phase deals with understanding the problem, the goals and constraint etc. requirement analysis starts with some general "statement of need" or high level "problem statement ". During analysis the problem domain and the environment are modeled in an effort to understand the system behavior, constraints if the system, its inputs and its outputs etc. The basic purpose of the activity is to obtain any thorough understanding of what the software needs to provide. The understanding of requirements leads to requirement specification. The analysis produce large amount of information and knowledge with possible redundancies properly organizing and describing the requirements is an important goal of this activity.

HARDWARE REQUIREMENT

Processor	:	Pentium III 800MHz or higher.
RAM	:	128 MB and above
Hard Disk Drive	:	20 GB or higher
Keyboard	:	101/102 Natural Keyboard
Monitor	:	Resolution of 800 X 600

SOFTWARE SPECIFICATIONS

Operating System	:	Linux distribution
Language Used	:	C
Tools	:	GCC

IMPLEMENTATION

Module 1:

Test.mc is the input c-program. This module should create test.asm file which contains equivalent 8086 code for the input mini c-program only if the c-program does not contain any error.

Module 2:

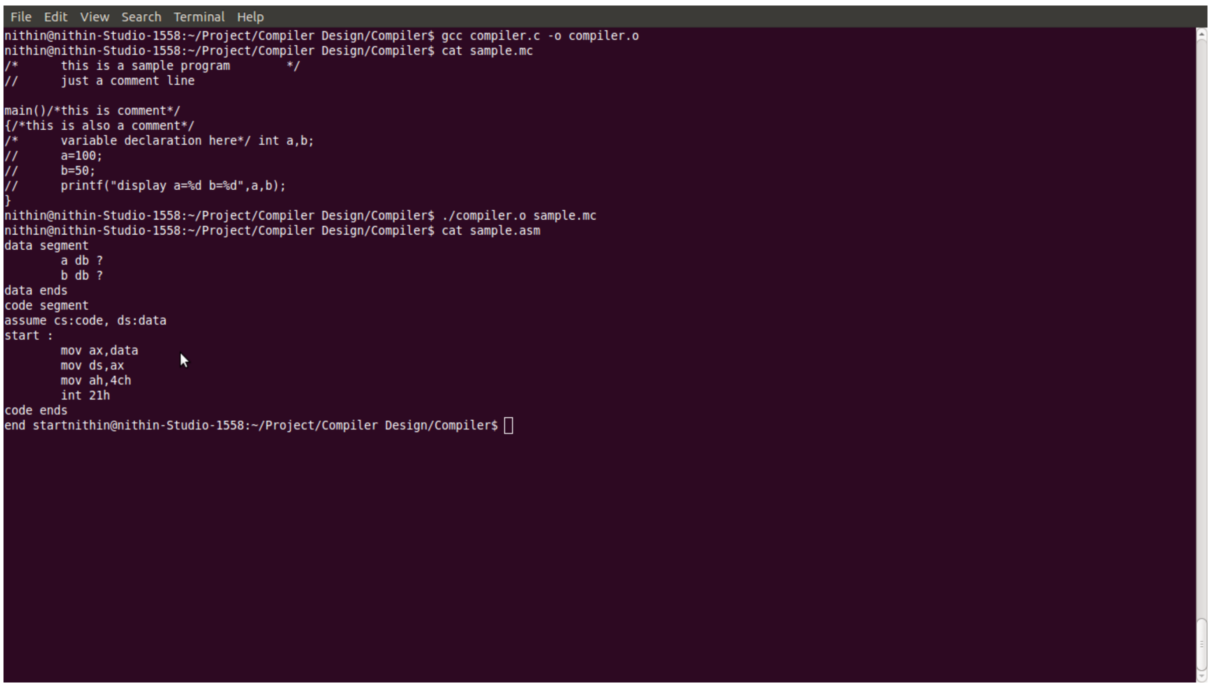
The main function is used to handle variable declaration, in our program the only allowed datatype is integer so the function mylex when encounters an 'int' in the line it recognizes it as the declaration line using the below given code.

Functions used:

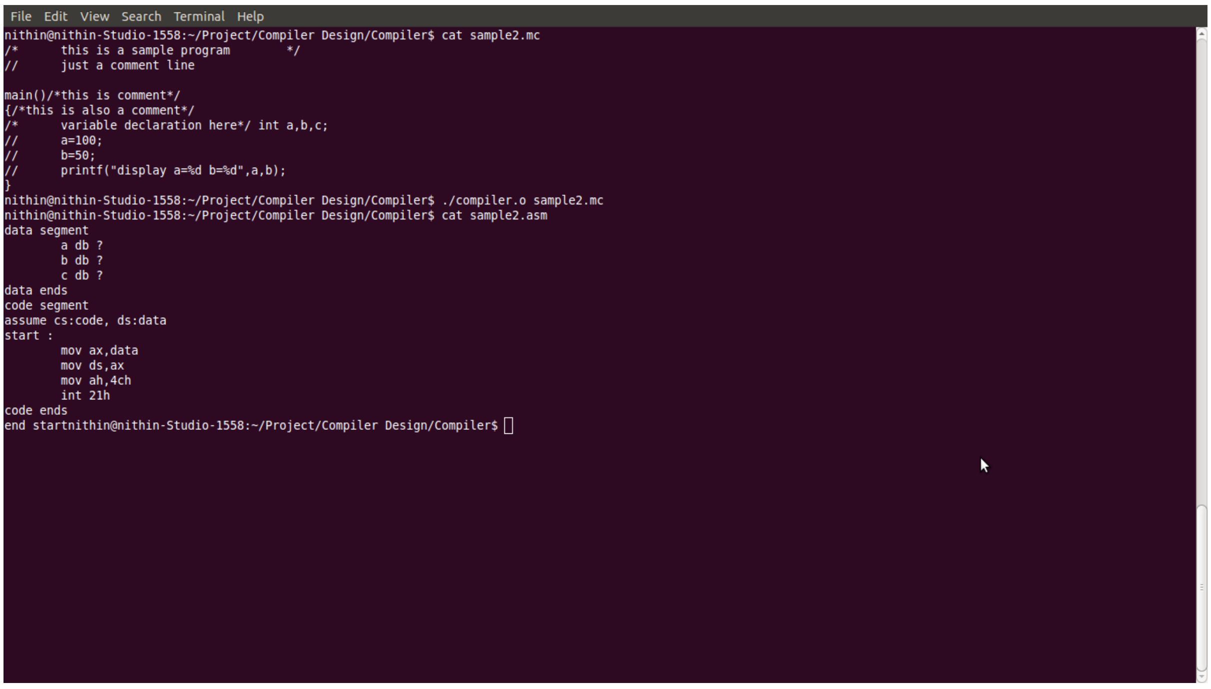
```
void error(int n,int num,char *str)
/*  for displaying errors
    n : error number
    num : line number
    str : string related to error       */

NODEPTR maketree(char *str,int x)
/*  creates a new node
    str : variable name
    x : variable value        */

void setleft(NODEPTR p,char *str,int x)
/*  insert the node onto the left of node pointed by "p"
    str : variable name
    x : variable value        */
```

```
void setright(NODEPTR p,char *str,int x)
/*   insert the node onto the right of node pointed by "p"
     str : variable name
     x : variable value        */

NODEPTR getnode()
//   create node by allocating dynamic memory

int check_comment(char *input_line)
//   removes comment from the input_line

void mylex(char *input_line,char tokens[100][200],int *num)
/*   for tokanizing the input_line
     input_line : string to be tokanized
     tokens : tokens stored in this array
     num : in the end will hold the number of tokens                    */

int identifier(char *str)
/*   check for ideintifier
     return 1 if str is identifier, else return 0          */

int match(char A,char B)
//   returns 1 if the open brackets in A match the closing brakets in B, else returns 0

int var_check(char *str)
//   return 1 if str contains valid data type, else return 0
```

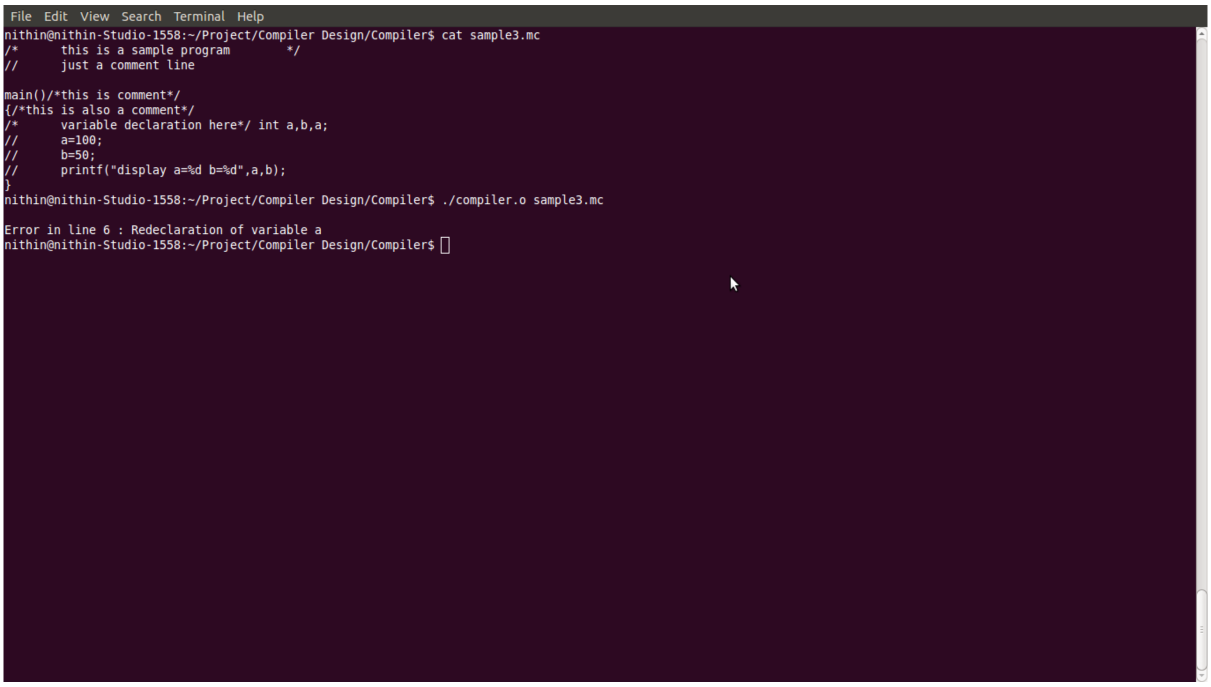
SNAPSHOTS



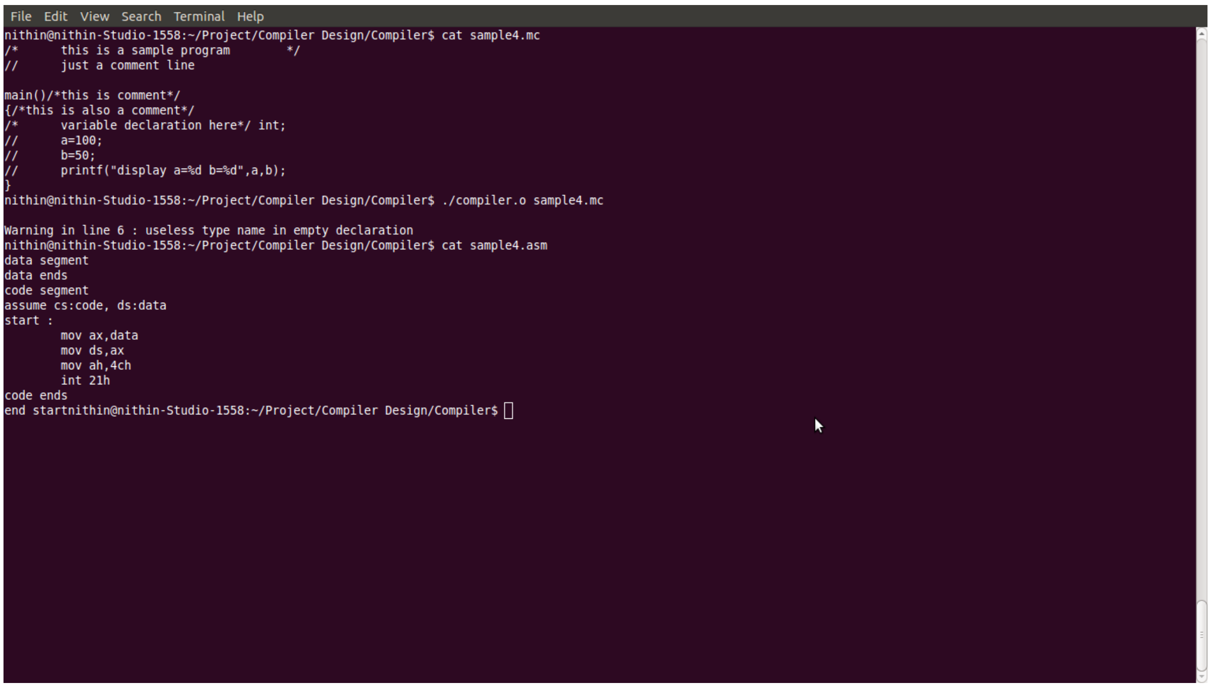
```
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ gcc compiler.c -o compiler.o
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ cat sample.mc
/*      this is a sample program        */
//      just a comment line

main()/*this is comment*/
{/*this is also a comment*/
/*      variable declaration here*/ int a,b;
//      a=100;
//      b=50;
//      printf("display a=%d b=%d",a,b);
}
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ ./compiler.o sample.mc
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ cat sample.asm
data segment
        a db ?
        b db ?
data ends
code segment
assume cs:code, ds:data
start :
        mov ax,data
        mov ds,ax
        mov ah,4ch
        int 21h
code ends
end startnithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ 
```



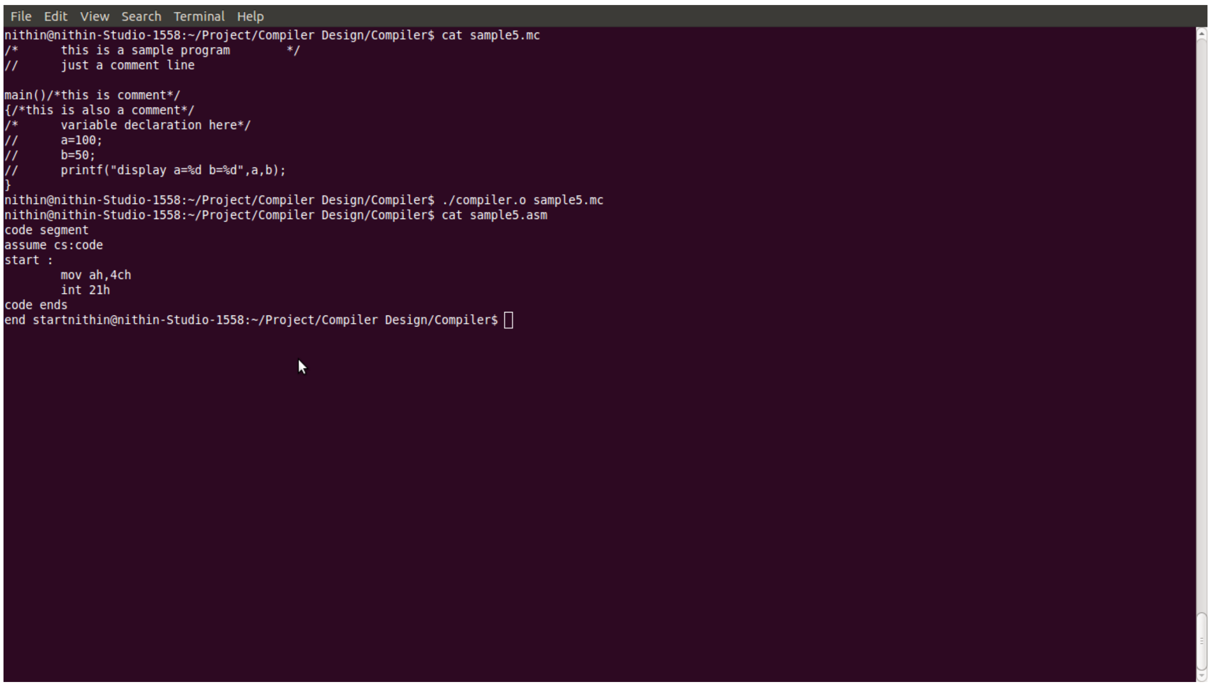
```
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ cat sample2.mc
/*      this is a sample program        */
//      just a comment line

main()/*this is comment*/
{/*this is also a comment*/
/*      variable declaration here*/ int a,b,c;
//      a=100;
//      b=50;
//      printf("display a=%d b=%d",a,b);
}
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ ./compiler.o sample2.mc
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ cat sample2.asm
data segment
        a db ?
        b db ?
        c db ?
data ends
code segment
assume cs:code, ds:data
start :
        mov ax,data
        mov ds,ax
        mov ah,4ch
        int 21h
code ends
end startnithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ 
```



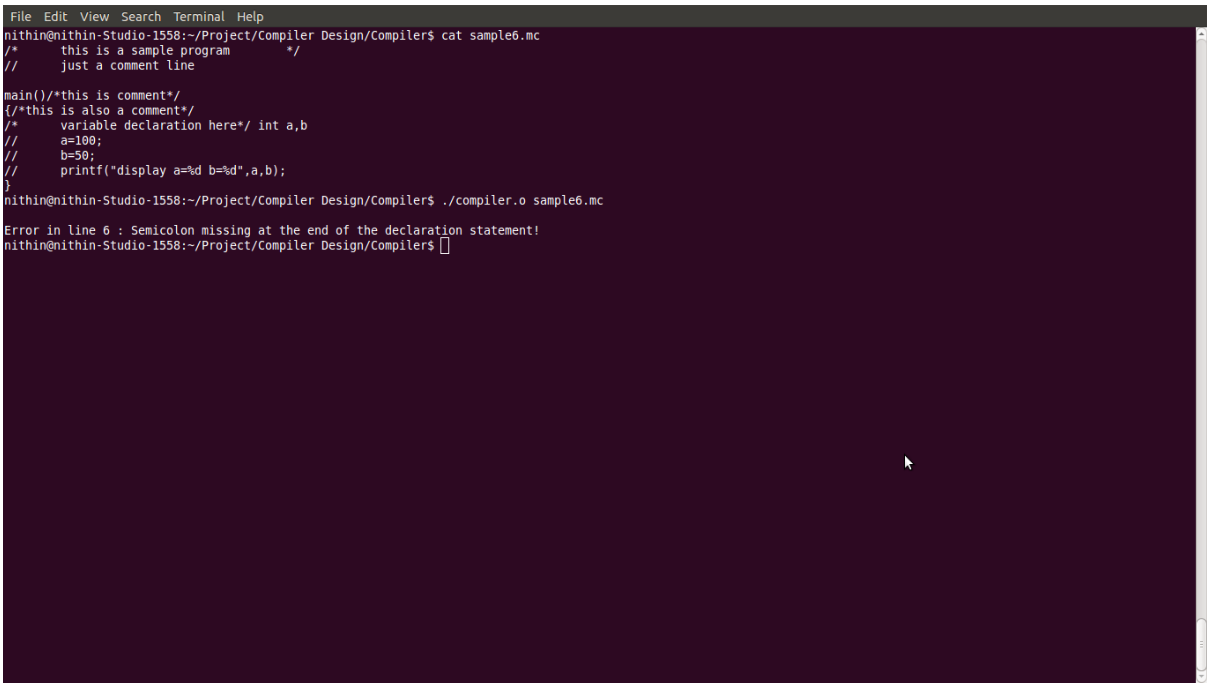
```
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ cat sample3.mc
/*      this is a sample program        */
//      just a comment line

main()/*this is comment*/
{/*this is also a comment*/
/*      variable declaration here*/ int a,b,a;
//      a=100;
//      b=50;
//      printf("display a=%d b=%d",a,b);
}
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ ./compiler.o sample3.mc

Error in line 6 : Redeclaration of variable a
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$
```



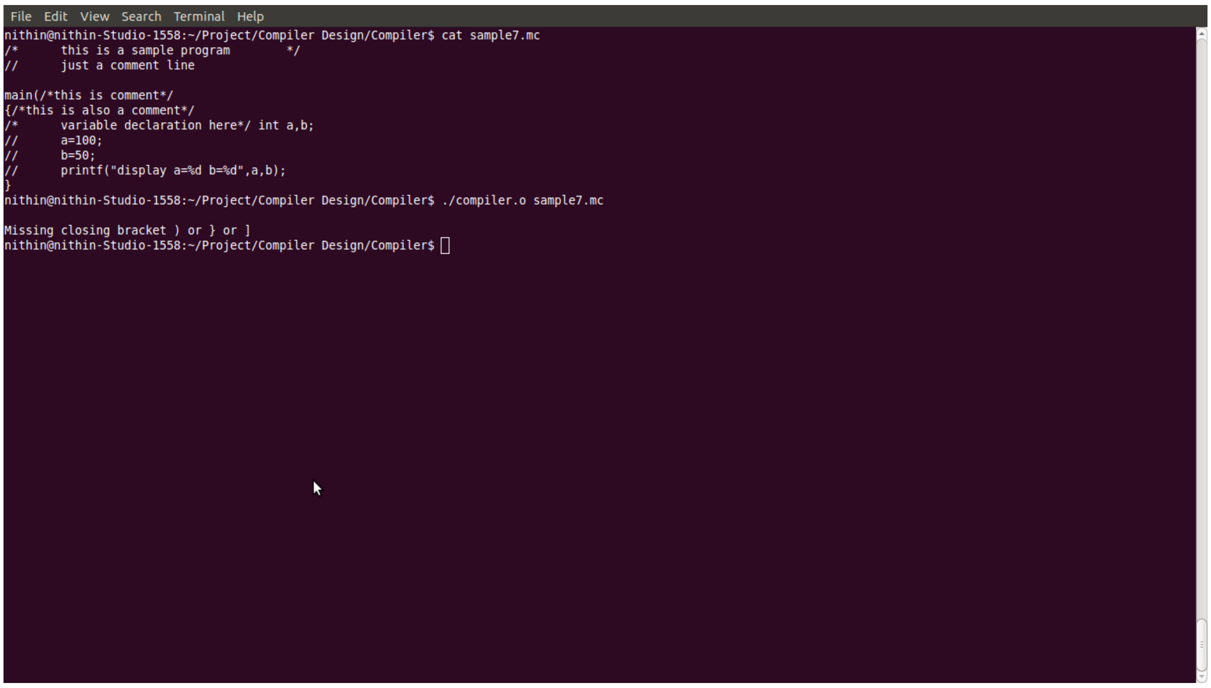
```
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ cat sample4.mc
/*      this is a sample program        */
//      just a comment line

main()/*this is comment*/
{/*this is also a comment*/
/*      variable declaration here*/ int;
//      a=100;
//      b=50;
//      printf("display a=%d b=%d",a,b);
}
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ ./compiler.o sample4.mc

Warning in line 6 : useless type name in empty declaration
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ cat sample4.asm
data segment
data ends
code segment
assume cs:code, ds:data
start :
        mov ax,data
        mov ds,ax
        mov ah,4ch
        int 21h
code ends
end startnithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$
```



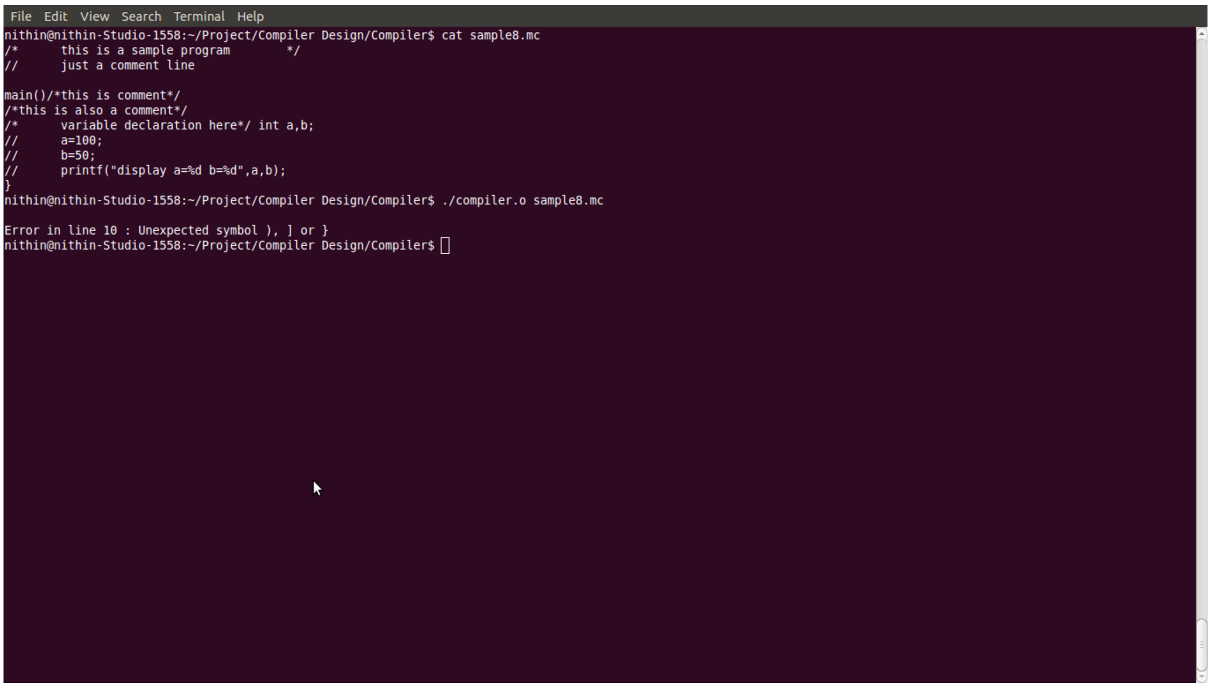
```
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ cat sample5.mc
/*      this is a sample program        */
//      just a comment line

main()/*this is comment*/
{/*this is also a comment*/
/*      variable declaration here*/
//      a=100;
//      b=50;
//      printf("display a=%d b=%d",a,b);
}
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ ./compiler.o sample5.mc
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ cat sample5.asm
code segment
assume cs:code
start :
        mov ah,4ch
        int 21h
code ends
end startnithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ 
```



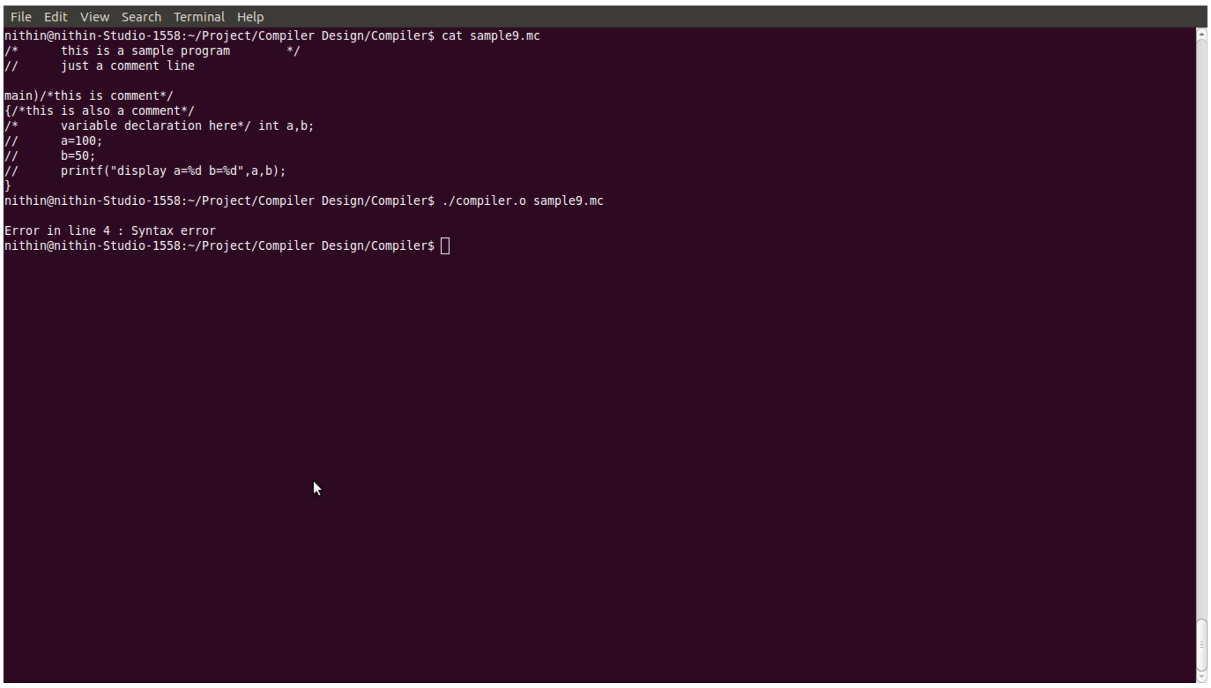
```
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ cat sample6.mc
/*      this is a sample program        */
//      just a comment line

main()/*this is comment*/
{/*this is also a comment*/
/*      variable declaration here*/ int a,b
//      a=100;
//      b=50;
//      printf("display a=%d b=%d",a,b);
}
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ ./compiler.o sample6.mc

Error in line 6 : Semicolon missing at the end of the declaration statement!
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ 
```



```
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ cat sample7.mc
/*      this is a sample program        */
//      just a comment line

main(/*this is comment*/
{/*this is also a comment*/
/*      variable declaration here*/ int a,b;
//      a=100;
//      b=50;
//      printf("display a=%d b=%d",a,b);
}
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ ./compiler.o sample7.mc

Missing closing bracket ) or } or ]
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ 
```



```
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ cat sample8.mc
/*      this is a sample program        */
//      just a comment line

main()/*this is comment*/
/*this is also a comment*/
/*      variable declaration here*/ int a,b;
//      a=100;
//      b=50;
//      printf("display a=%d b=%d",a,b);
}
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ ./compiler.o sample8.mc

Error in line 10 : Unexpected symbol ), ] or }
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ 
```



```
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ cat sample9.mc
/*      this is a sample program        */
//      just a comment line

main)/*this is comment*/
{/*this is also a comment*/
/*      variable declaration here*/ int a,b;
//      a=100;
//      b=50;
//      printf("display a=%d b=%d",a,b);
}
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ ./compiler.o sample9.mc

Error in line 4 : Syntax error
nithin@nithin-Studio-1558:~/Project/Compiler Design/Compiler$ 
```

CONCLUSION

Our project uses the concept of efficient memory utilization and time efficiency to build the compiler. It can and can support some of the compiler options provided by gcc. Our project is an attempt to make a smaller, faster c compiler using conventional programming techniques. Our project can use only simple C programs involving arithmetic operations and integers and generate an equivalent 8086 assembly code.