

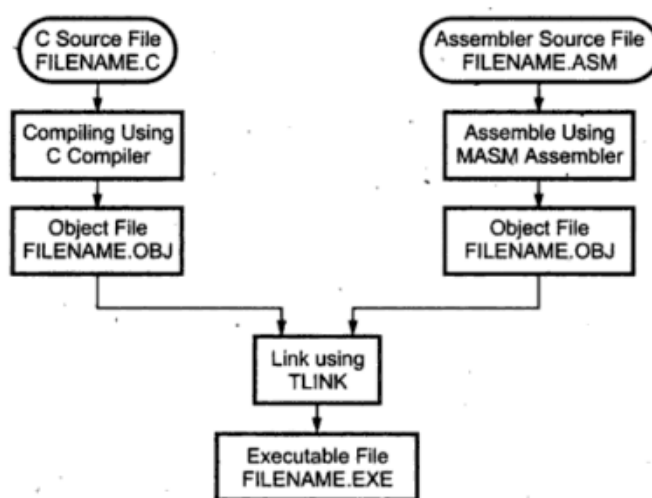
Experiment 07 : Mixed Mode Programming

Learning Objective: Student should be able to Write a mixed language code for designing calculator.

Tools: TASM/MASM/ Turbo C/C++

Theory:

1. There are times when programs need to call programs written in other languages referred as mixed language programming. For example, when a particular subprogram is available in a language other than language you are using, or when algorithms are described more naturally in a different language, you need to use more than one language.
2. Mixed-language programming always involves a call to a function, procedure, or subroutine. Mixed-language calls involve calling functions in separate modules. Instead of compiling all source programs with same compiler, different compilers or assemblers are used as per the language used in the programs.
3. Microsoft C supports this mixed language programming. So it can combine assembly code routines in C as a separate language.
4. C program calls assembly language routines that are separately assembled by-MASM (MASM Assembler). These assembled modules are linked with the compiled C modules to get executable file. Fig shows the compile, assemble and link processes using C compiler, MASM assembler, and TUNIC.



Compiler, assembly and link processes

```
#include<stdio.h>
```

```
void main() {
```

```
    int a = 3, b = 3, c;
```

```
    asm {
```

```
        mov ax,a
```

```
        mov bx,a
```

```
        add ax,bx
```

```
        mov c,ax
```

```
    }
```

```
    printf("%d",c);
```

```
}
```

1. Assembly Language can be Written in C .
2. C Supports Assembly as well as Higher Language Features so called “Middle Level Language”.
3. As shown in above Program , “asm” Keyword is written to indicate that “next followed instruction is from Assembly Language”.
4. Opening Curly brace after “asm” keyword tells that it is the “Start of Multiple Line Assembly Statements” i.e “We want to Write Multiple Instructions”
5. Above Program Without “Opening and Closing Brace” can be written as – [“asm” keyword before every Instruction]

Application: Use of mixed mode programming to write modular program.

Design:

CODE:

```

#include <iostream.h>
#include <stdio.h>
#include <conio.h>
#include <process.h>

void main()
{

int op1,op2,ans,choice;
clrscr();

cout<<"EXP 7 - by COMP A, Roll No 68"<<endl;
cout<<"Calcutor Program"<<endl;
while(1)
{
cout<<endl<<endl<<"Calcutor Menu:"<<endl;
cout<<endl;
cout<<"1. Addition"<<endl;
cout<<"2. Subtraction"<<endl;
cout<<"3. Exit"<<endl;
cout<<"Your Choice: ";
cin>>choice;

if (choice==3)
{
exit(0);
}

cout<<endl<<"Enter Operand one: ";
cin>>op1;
cout<<endl<<"Enter Operand second: ";
cin>>op2;

switch(choice)
{
    case 1: asm mov ax,op1;
            asm add ax,op2;
            asm mov ans,ax;
            cout<<endl<<"Answer: "<<ans;
            break;

    case 2: asm mov ax,op1;
            asm sub ax,op2;
            asm mov ans,ax;

```

```
        cout<<endl<<"Answer: "<<ans;
        break;

    case 3: exit(0);
        break;

    default: cout<<endl<<"Invalid choice";
        break;
}
}
getch();
}
```

Result and Discussion:

Menu :

1. Addition
2. Subtraction
3. Exit

Choice: 1

Enter the two numbers: 3124 1134

4258

Menu :

1. Addition
2. Subtraction
3. Exit

Choice: 2

```
1. Addition
2. Subtraction
3. Exit
Choice: 1

Enter the two numbers: 3124 1134

4258
Menu:

1. Addition
2. Subtraction
3. Exit
Choice: 2

Enter the two numbers: 2134
1234

900
Menu:

1. Addition
2. Subtraction
3. Exit
Choice: 3
```

Learning Outcomes: The student should have the ability to

- LO1: To develop the understanding of Mixed mode programming.
- LO2: Develop the program in mixed language for Intel 8086 processor.
- LO3: Demonstrate the execution and debugging of mixed language program.

Course Outcomes: Upon completion of the course students will be able to make use of instructions of 8086 to build assembly and Mixed language programs.

Conclusion:

The objective of implementation of a calculator using mixed programming was successfully attained.

Viva Questions:

1. Write short note on mixed mode programming.

For Faculty Use

Correction Parameters	Formative Assessment [40%]	Timely completion of Practical [40%]	Attendance / Learning Attitude [20%]	
Marks Obtained				