Mixed Language Programming using Assembly and C

- C generates an object code that is extremely fast and compact, but it is not as fast as the object code generated by a good programmer using assembly language.
- It is true that the time needed to write a program in assembly language is much more than the time taken in higher level languages like C.
- However, there are special cases where a function coded in assembly language reduce execution time.

Example: The floating point math package must be coded in assembly language as it is used frequently and its execution speed will have great effect on the overall speed of the program that uses it.

- There are also occasions where some hardware devices need exact timing and then it is necessary to write assembly level programs to meet such strict timing restrictions.
- In addition, certain instructions cannot be executed in higher level languages like C

Example: C does not have an instruction for performing bitwise rotation operation. Thus, in spite of C being very powerful, routines must be written in assembly language to:

- Increase the speed and efficiency of the routine.
- Perform machine specific functions not available in Microsoft C or in Turbo C.
- Use 3rd party routines.

Combining C and assembly:

There are 2 ways of combining C and assembly language.

Method 1:

- Here built-in inline assembler is used to include assembly language routines in the C program without any need for a specific assembly. Such assembly language routines are called in line.
- They are compiled right along with C routines, rather than being assembled separately and then linked together using linker modules provided by the C compiler
- Turbo C has inline assembler.

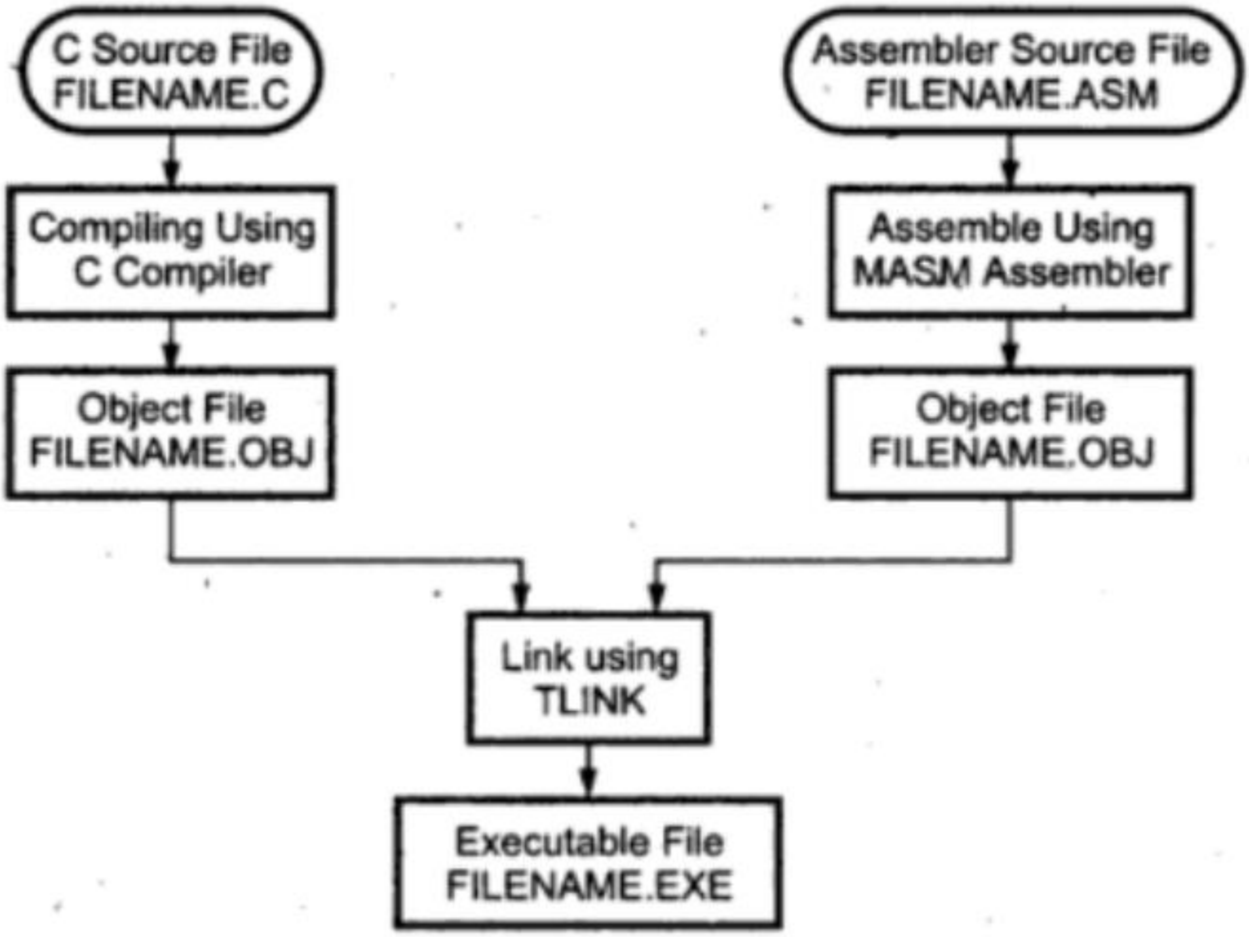
Method 2:

- There are times when programs written in one language have to call modules written in other languages
- This is called as mixed language programming.

Example: when a particular subroutine is available in a language, different from the language currently used in a program, or when algorithms are described in a different language, we need to use more than one languages

- Mixed language calls involve calling functions in separate modules.
- Instead of compiling all source programs using the same compiler, different compilers, assemblers are used as per that used in the program.

- Microsoft C supports mixed language programming, therefore it can combine assembly language routines in C as a separate language.
- C program calls assembly language routines that are separately assembled by MASM or TASM.
- These assembled modules are linked with the compiled C modules to get the combined executable file.



Compiler, assembly and link processes

PROGRAMMING EXAMPLE:

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

void main()
{
int n,i,c,p,j;
clrscr();
printf("ENTER THE NUMBER:");
scanf("%d",&n);
printf("ENTER THE POWER:");
scanf("%d",&p);

asm mov cx,n;
asm mov ax,n;

for(i=1;i<p;i++)
{
   for(j=1;j<n;j++)
   {
   asm add ax,cx;

   }
asm mov cx,ax;
}
asm mov c,cx;
printf("ANSWER IS:%d\n",c);
getch();
}
```