

Assembling the ASM file:

The assembly language code written by me is assembled to create an object file using the **compileAssembly** target, it uses the command:

**nasm -felf64 add.asm**

It creates an object file (add.o) and uses -felf64 flag as the OS I am using is ubuntu20.4 on a 64 bit machine.

In the code the label add is used to defined the add function being used in add.c and rdi and rsi registers contain variable1 and variable2 respectively and the rax register is used to store the final outcome of the addition. I have used rax,rdi,rsi as we are dealing with 64 bit long integers and using the complete registers

```
saatvik@saatvik-Lenovo-Legion-Y540-15IRH-PG0: ~/OS/Assignment0/Assignment0.2
saatvik@saatvik-Lenovo-Legion-Y540-15IRH-PG0:~/OS/Assignment0/Assignment0.2$ make compileAssembly
nasm -felf64 add.asm
saatvik@saatvik-Lenovo-Legion-Y540-15IRH-PG0:~/OS/Assignment0/Assignment0.2$
```

Assembling the C file:

I have used the same targets as Assignment 0.1 to get the object file of the prog-add.c file (prog-add.o)

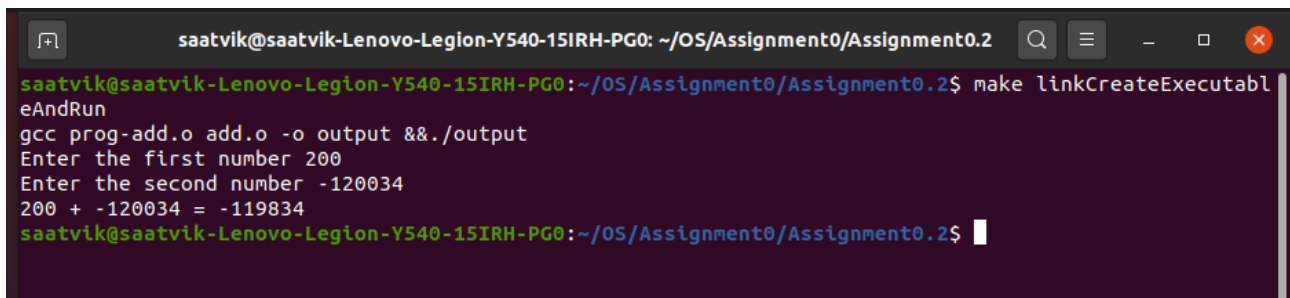
```
saatvik@saatvik-Lenovo-Legion-Y540-15IRH-PG0: ~/OS/Assignment0/Assignment0.2
saatvik@saatvik-Lenovo-Legion-Y540-15IRH-PG0:~/OS/Assignment0/Assignment0.2$ make createPreprocessed
gcc -E prog-add.c -o prog-add.i
saatvik@saatvik-Lenovo-Legion-Y540-15IRH-PG0:~/OS/Assignment0/Assignment0.2$ make createAssemblyProgram
gcc -S prog-add.i -o prog-add.s
saatvik@saatvik-Lenovo-Legion-Y540-15IRH-PG0:~/OS/Assignment0/Assignment0.2$ make createByteCode
gcc -c prog-add.s -o prog-add.o
saatvik@saatvik-Lenovo-Legion-Y540-15IRH-PG0:~/OS/Assignment0/Assignment0.2$
```

Linking C and ASM :

Linking is important to actually get the reference of add function from the add.o file. Till the time we don't linking the object file of prog-add.c and object file of add.asm using the linker the function definition of add is unresolved as we only have only added the function prototype in the prog-add.c file,it can be clearly seen here that add function is not converted into assembly as there is no function definition:

```
Assignment0.2 > prog-add.s
43      call    __isoc99_scanf@PLT
44      movq    -24(%rbp), %rdx
45      movq    -32(%rbp), %rax
46      movq    %rdx, %rsi
47      movq    %rax, %rdi
48      call    add@PLT
49      movq    %rax, -16(%rbp)
50      movq    -24(%rbp), %rdx
51      movq    -32(%rbp), %rax
52      movq    -16(%rbp), %rcx
53      movq    %rax, %rsi
54      leaq    .LC3(%rip), %rdi
```

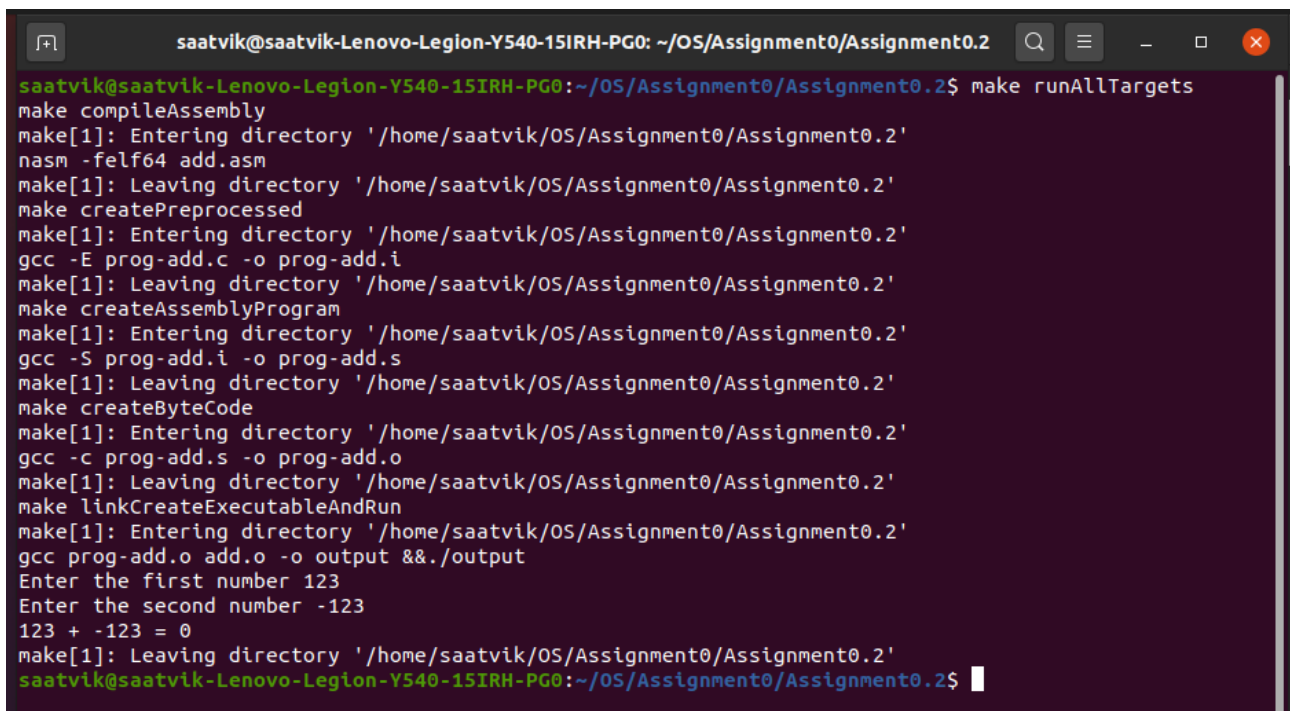
The **make linkCreateExecutableAndRun** command links both the files and resolves the add function reference and creates an executable (output) and then runs it. It uses the command:  
**gcc prog-add.o add.o -o output && ./output**



```
saatvik@saatvik-Lenovo-Legion-Y540-15IRH-PG0: ~/OS/Assignment0/Assignment0.2
saatvik@saatvik-Lenovo-Legion-Y540-15IRH-PG0:~/OS/Assignment0/Assignment0.2$ make linkCreateExecutableAndRun
gcc prog-add.o add.o -o output && ./output
Enter the first number 200
Enter the second number -120034
200 + -120034 = -119834
saatvik@saatvik-Lenovo-Legion-Y540-15IRH-PG0:~/OS/Assignment0/Assignment0.2$
```

In the end i have used the target **runAllTargets** which calls the other targets to first create an object file from add.asm and then create object file for prog-add.c and finally link both the object files to create an executable and run it. It uses the commands:

**make compileAssembly**  
**make createPreprocessed**  
**make createAssemblyProgram**  
**make createByteCode**  
**make linkCreateExectuableAndRun**



```
saatvik@saatvik-Lenovo-Legion-Y540-15IRH-PG0: ~/OS/Assignment0/Assignment0.2
saatvik@saatvik-Lenovo-Legion-Y540-15IRH-PG0:~/OS/Assignment0/Assignment0.2$ make runAllTargets
make compileAssembly
make[1]: Entering directory '/home/saatvik/OS/Assignment0/Assignment0.2'
nasm -felf64 add.asm
make[1]: Leaving directory '/home/saatvik/OS/Assignment0/Assignment0.2'
make createPreprocessed
make[1]: Entering directory '/home/saatvik/OS/Assignment0/Assignment0.2'
gcc -E prog-add.c -o prog-add.i
make[1]: Leaving directory '/home/saatvik/OS/Assignment0/Assignment0.2'
make createAssemblyProgram
make[1]: Entering directory '/home/saatvik/OS/Assignment0/Assignment0.2'
gcc -S prog-add.i -o prog-add.s
make[1]: Leaving directory '/home/saatvik/OS/Assignment0/Assignment0.2'
make createByteCode
make[1]: Entering directory '/home/saatvik/OS/Assignment0/Assignment0.2'
gcc -c prog-add.s -o prog-add.o
make[1]: Leaving directory '/home/saatvik/OS/Assignment0/Assignment0.2'
make linkCreateExecutableAndRun
make[1]: Entering directory '/home/saatvik/OS/Assignment0/Assignment0.2'
gcc prog-add.o add.o -o output && ./output
Enter the first number 123
Enter the second number -123
123 + -123 = 0
make[1]: Leaving directory '/home/saatvik/OS/Assignment0/Assignment0.2'
saatvik@saatvik-Lenovo-Legion-Y540-15IRH-PG0:~/OS/Assignment0/Assignment0.2$
```