Samiksha Modi

2019331

Dr. Sambuddho Chakravarty

Operating Systems

Combining C and Assembly

How the Program Works

We have a C program prog-add.c and an ASM routine add.asm. We create the object file for both using the following commands. `gcc -c prog-add.s` and `nasm -f elf64 add.asm`

Then we link prog-add.o and add.o with the command `gcc -o sum prog-add.o add.o` to create an executable called sum which can be executed from the command line using `./sum`

```
samiksha@samikshas-dell:~/Desktop/Assignment 2$ make
gcc -E prog-add.c -o prog-add.i
gcc -S prog-add.i -o prog-add.s
gcc -c prog-add.s -o prog-add.o
nasm -f elf64 add.asm
gcc -o sum prog-add.o add.o
./sum

Input a, b: 45 -43
Sum=2
samiksha@samikshas-dell:~/Desktop/Assignment 2$
```

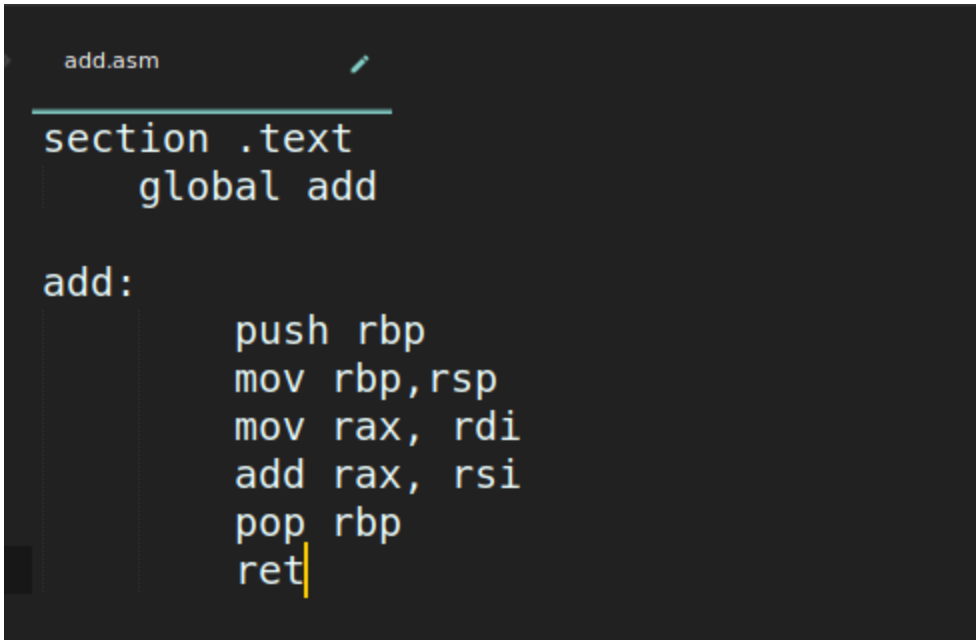
In order to pass the two arguments from prog-add.c we define the function prototype

```
long long int add (long long int, long long int);
// function in NASM assembly language
```

in the C program. After taking the input in the C program we call the add ASM routine.

The arguments passed through the main of the C program are now present in the registers. Since we have two integers the registers used for passing the arguments are rdi and rsi.

- We first push rbp into the stack to save the address of the previous stack frame.
- We then move rsp to rbp so that we have the address of the current stack frame.
- We then calculate our sum.
    - We move the contents of rdi to the rax register (mov rax,rdi)
    - Then add the contents of rsi register (add rax,rsi)
    - The sum is now stored in the rax register.
- We then pop rbp to restore the previously pushed stack base pointer address and then return.



After returning to our C program we just print the sum using printf.

prog-add.c

```c
#include <stdio.h>

long long int add (long long int, long long int);
// function in NASM assembly language

int main(int argc, char *argv[ ])
{
    long long int a, b,sum;

    printf("\nInput a, b: ");
    scanf("%lld %lld", &a, &b);
    sum=add(a, b);
    printf("Sum=%lld\n",sum);
    return 0;
}
```

Makefile

Makefile

```
all:
    gcc -E prog-add.c -o prog-add.i
    gcc -S prog-add.i -o prog-add.s
    gcc -c prog-add.s -o prog-add.o
    nasm -f elf64 add.asm
    gcc -o sum prog-add.o add.o
    ./sum


clean:
    rm *.o *.i *.s sum
```

References

1. https://man7.org/linux/man-pages/man1/gcc.1.html
2. https://linux.die.net/man/1/nasm
3. https://stackoverflow.com/questions/37581530/passing-argument-from-c-to-assembly