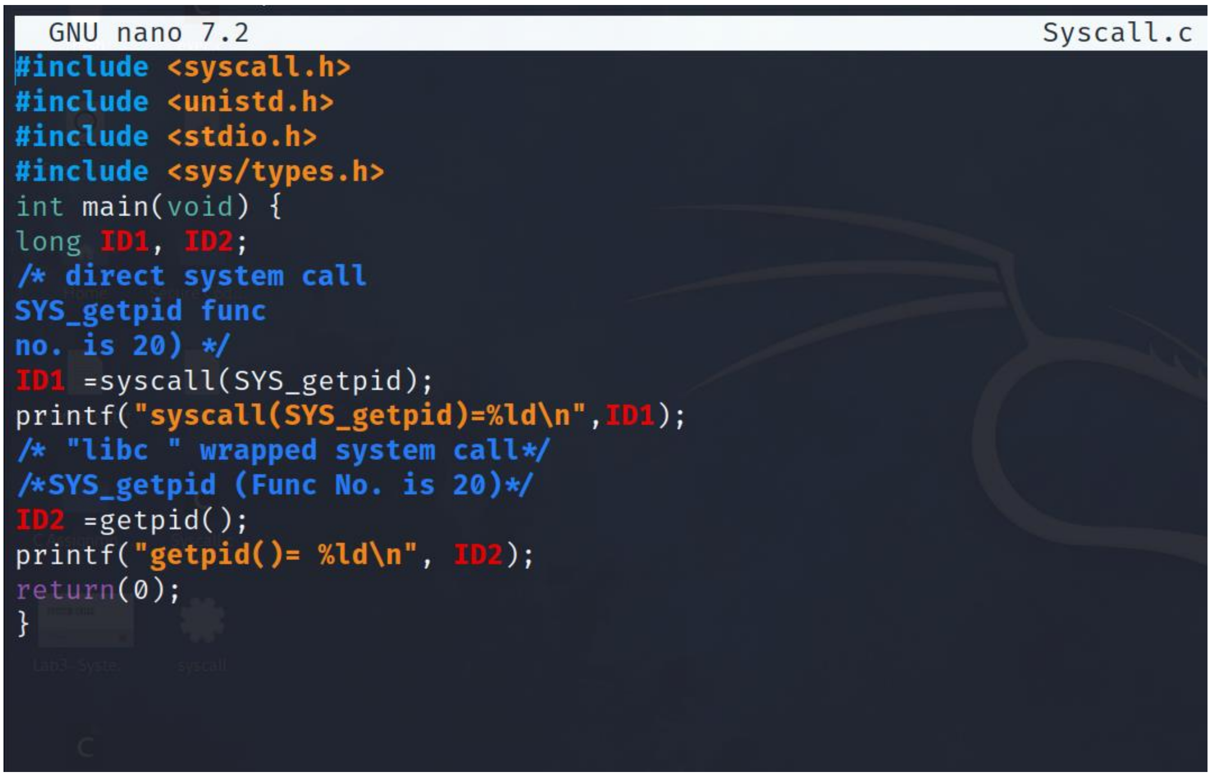
SYSTEM CALLS

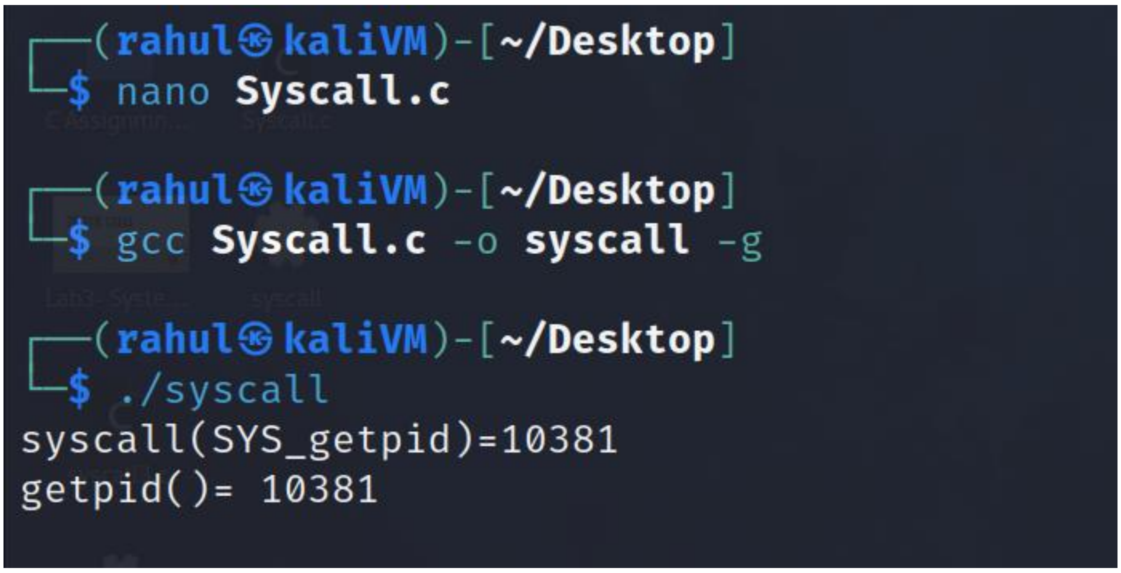
Rahul M Menon
CB.EN.P2CYS23015

GETPID



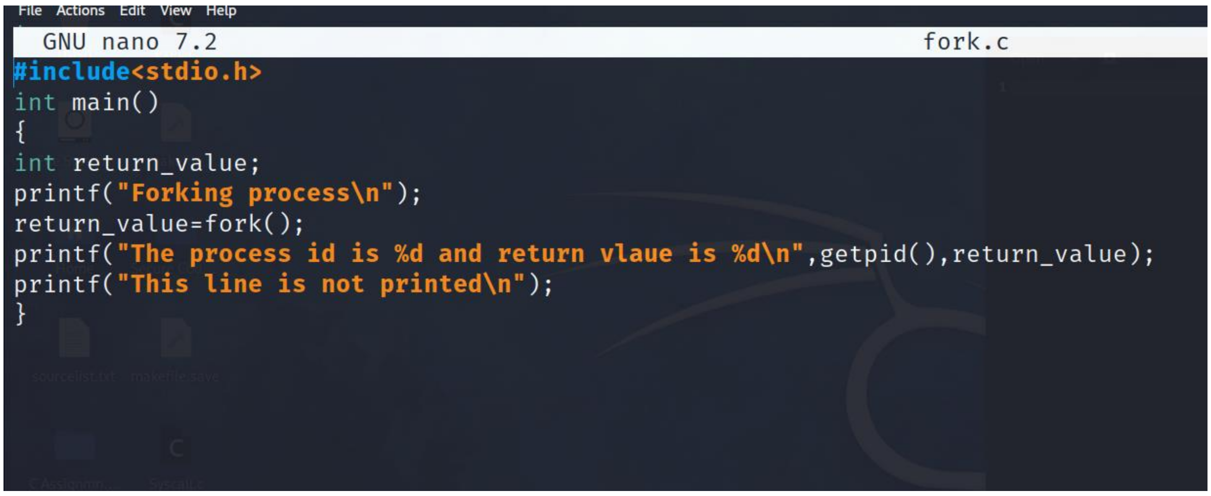
```
GNU nano 7.2                                                    Syscall.c
#include <syscall.h>
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
int main(void) {
long ID1, ID2;
/* direct system call
SYS_getpid func
no. is 20) */
ID1 =syscall(SYS_getpid);
printf("syscall(SYS_getpid)=%ld\n",ID1);
/* "libc " wrapped system call*/
/*SYS_getpid (Func No. is 20)*/
ID2 =getpid();
printf("getpid()= %ld\n", ID2);
return(0);
}
```

OUTPUT



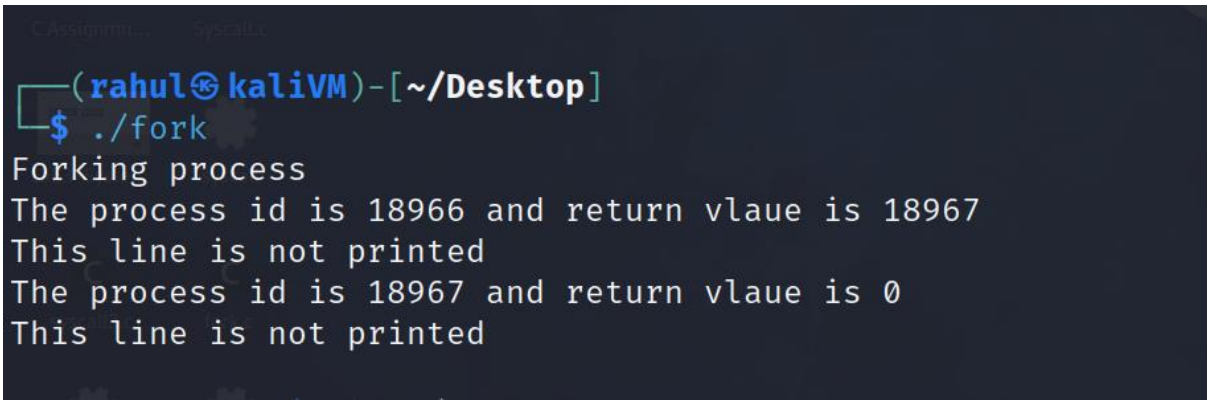
```
┌──(rahul㉿kaliVM)-[~/Desktop]
└─$ nano Syscall.c

┌──(rahul㉿kaliVM)-[~/Desktop]
└─$ gcc Syscall.c -o syscall -g

┌──(rahul㉿kaliVM)-[~/Desktop]
└─$ ./syscall
syscall(SYS_getpid)=10381
getpid()= 10381
```

FORK



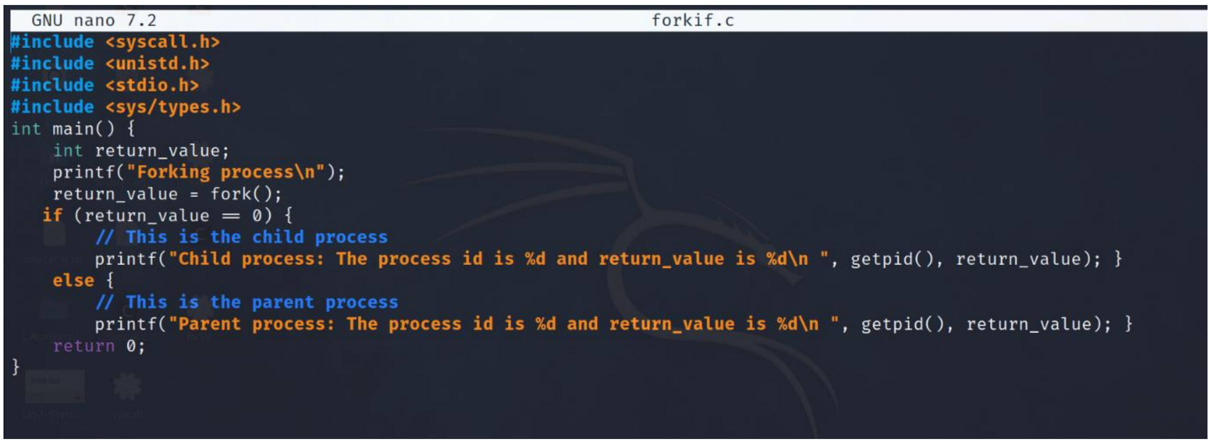
```c
#include<stdio.h>
int main()
{
int return_value;
printf("Forking process\n");
return_value=fork();
printf("The process id is %d and return vlaue is %d\n",getpid(),return_value);
printf("This line is not printed\n");
}
```

OUTPUT



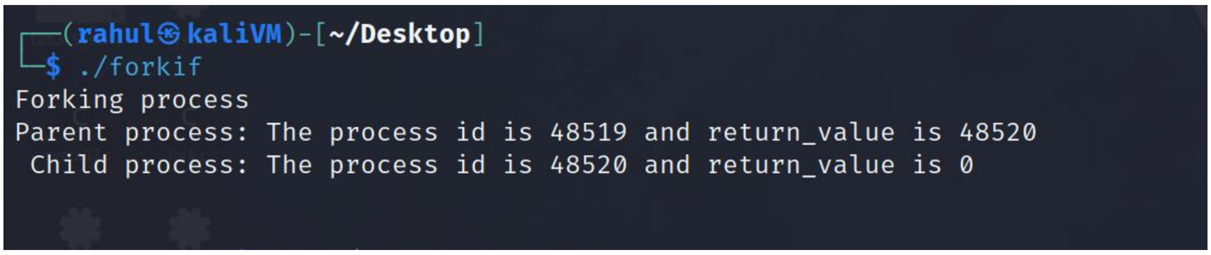
```
┌──(rahul㉿kaliVM)-[~/Desktop]
└─$ ./fork
Forking process
The process id is 18966 and return vlaue is 18967
This line is not printed
The process id is 18967 and return vlaue is 0
This line is not printed
```

FORK If else



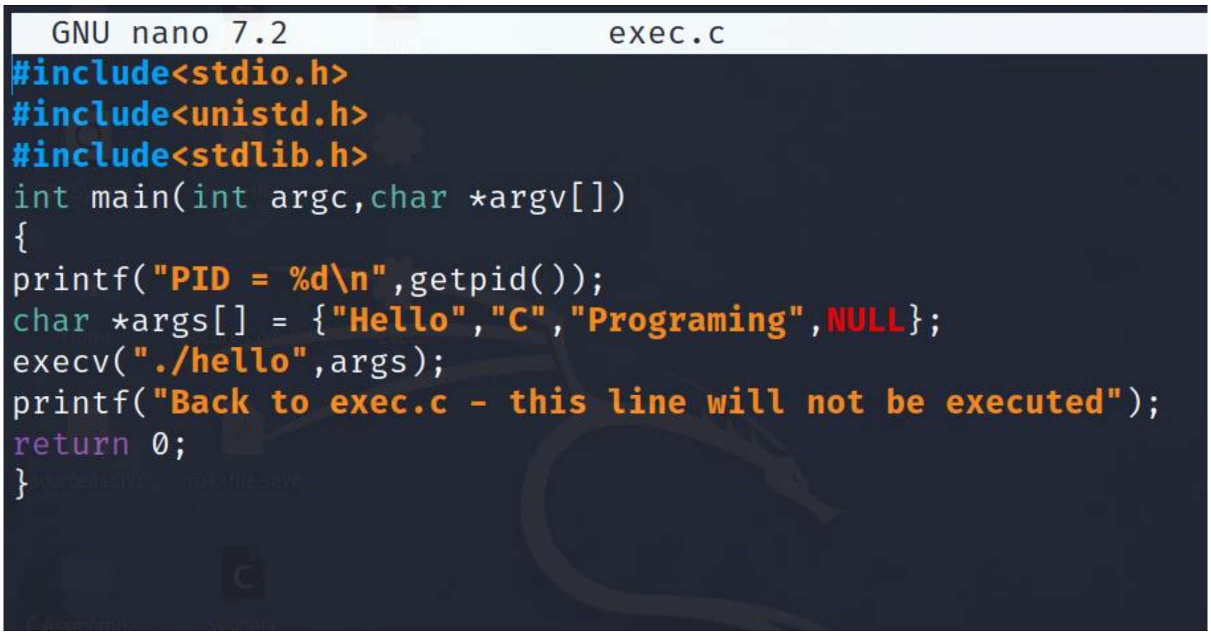
```
GNU nano 7.2                                                    forkif.c
#include <syscall.h>
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
int main() {
    int return_value;
    printf("Forking process\n");
    return_value = fork();
   if (return_value == 0) {
        // This is the child process
        printf("Child process: The process id is %d and return_value is %d\n ", getpid(), return_value); }
    else {
        // This is the parent process
        printf("Parent process: The process id is %d and return_value is %d\n ", getpid(), return_value); }
    return 0;
}
```

OUTPUT



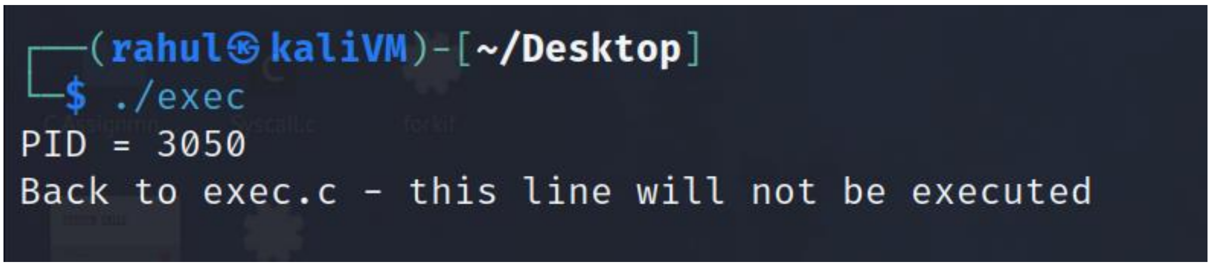
```
┌──(rahul㉿kaliVM)-[~/Desktop]
└─$ ./forkif
Forking process
Parent process: The process id is 48519 and return_value is 48520
 Child process: The process id is 48520 and return_value is 0
```

EXEC



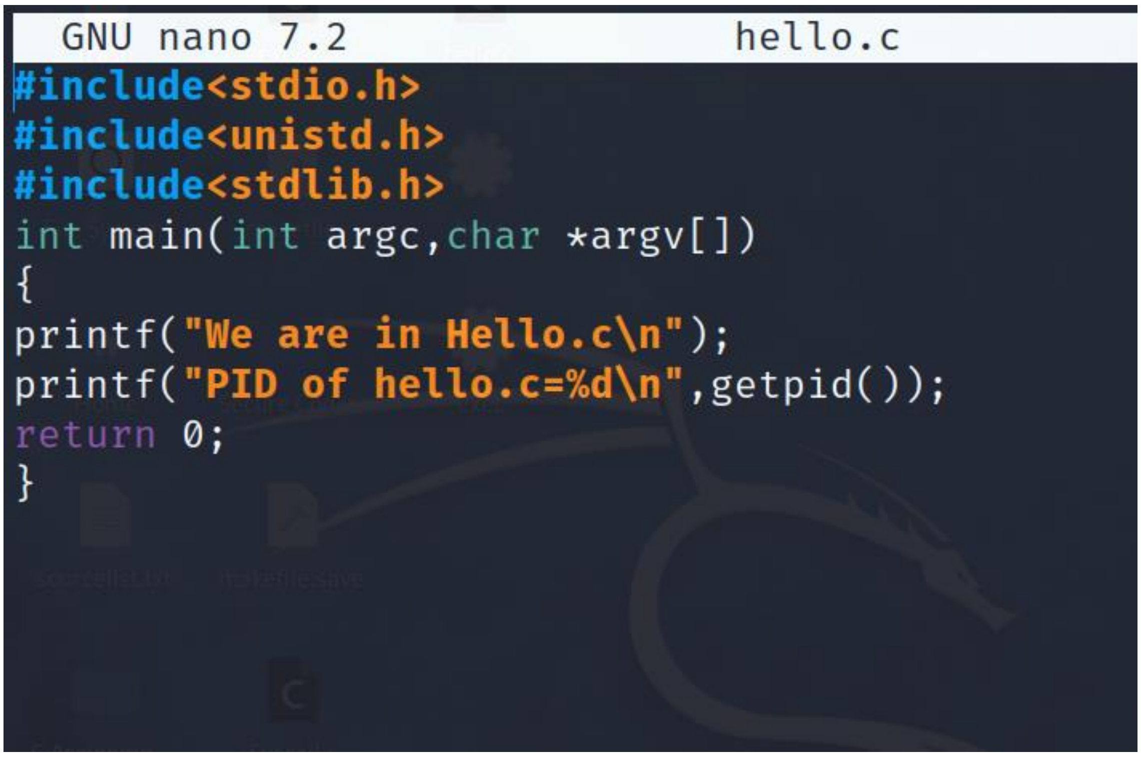
```
GNU nano 7.2                    exec.c
#include<stdio.h>
#include<unistd.h>
#include<stdlib.h>
int main(int argc,char *argv[])
{
printf("PID = %d\n",getpid());
char *args[] = {"Hello","C","Programing",NULL};
execv("./hello",args);
printf("Back to exec.c - this line will not be executed");
return 0;
}
```

OUTPUT



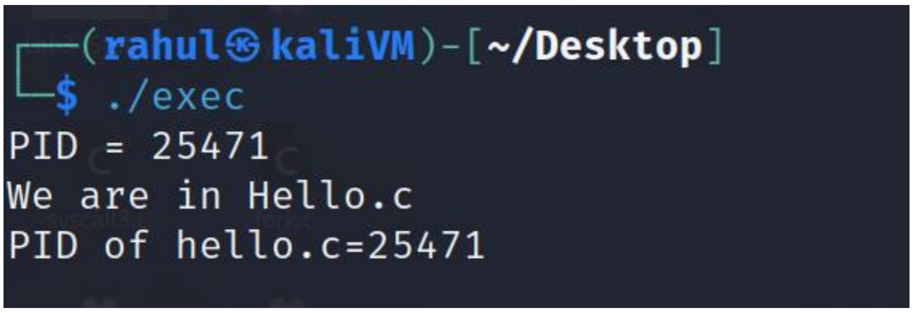
```
┌──(rahul㉿kaliVM)-[~/Desktop]
└─$ ./exec
PID = 3050
Back to exec.c - this line will not be executed
```

Hello.c



```
  GNU nano 7.2                    hello.c
#include<stdio.h>
#include<unistd.h>
#include<stdlib.h>
int main(int argc,char *argv[])
{
printf("We are in Hello.c\n");
printf("PID of hello.c=%d\n",getpid());
return 0;
}
```

OUTPUT



```
┌──(rahul㉿kaliVM)-[~/Desktop]
└─$ ./exec
PID = 25471
We are in Hello.c
PID of hello.c=25471
```