# Lab 2:

## System Calls

### 1. syscall,getpid()

```
GNU nano 8.0
#include <syscall.h>
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
int main()
{
        long id1,id2;
        id1=syscall(SYS_getpid);
        printf("syscall(SYS_getpid)=%ld\n",id1);
        id2=getpid();
        printf("getpid() is %d\n",id2);
        return (0);
}
```

```
┌──(kali㉿kali)-[~]
└─$ gcc test.c

┌──(kali㉿kali)-[~]
└─$ ./a.out
syscall(SYS_getpid)=9443
getpid() is 9443
```

### 2.fork()

```
GNU nano 8.0
#include <stdio.h>
#include <syscall.h>
int main()
{
        pid_t id=fork();
        if(id==-1)
        {
                printf("Error!!");
        }
        else if(id==0)
        {
                printf("It is a child process\n");
                printf("Child process id: %d\n",getpid());
                printf("Parent process id: %d\n",getppid());
        }
        else
        {
                printf("It is a parent process\n");
                printf("Parent process id: %d\n"getpid());
                printf("Child process id: %d\n",id);
        }
}
```

```
┌──(kali㉿kali)-[~]
└─$ nano fork.c

┌──(kali㉿kali)-[~]
└─$ gcc fork.c

┌──(kali㉿kali)-[~]
└─$ ./a.out
It is a parent process
Parent process id: 10893
Child process id: 10894
It is a child process
Child process id: 10894
Parent process id: 10893
```

### 3.exec()

```
GNU nano 8.0
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
int main(int argc,char *argv[])
{
        printf("We are in hello.c\n");
        printf("process id of hello.c=%d\n",getpid());
        return 0;
}
```

```
 GNU nano 8.0                                                    h
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
int main(int argc,char *argv[])
{
        printf("process id=%d\n",getpid());
        char *args[]={"Hello","C","Programming",NULL};
        execv("./hello",args);
        printf("Back to execfile.c - This line will not be executed");
        return 0;
}
```

```
┌──(kali㉿kali)-[~]
└─$ nano execfile.c

┌──(kali㉿kali)-[~]
└─$ nano hello.c

┌──(kali㉿kali)-[~]
└─$ gcc hello.c -o hello

┌──(kali㉿kali)-[~]
└─$ gcc execfile.c -o execfile

┌──(kali㉿kali)-[~]
└─$ ./execfile
We are in hello.c
process id of hello.c=16279
```

# 4. Strace

```
┌──(kali㉿kali)-[~]
└─$ strace ls -l
execve("/usr/bin/ls", ["ls", "-l"], 0x7fffbe8fd248 /* 54 vars */) = 0
brk(NULL)                               = 0x55d18eb93000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f48da9da000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=89695, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 89695, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f48da9c4000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libselinux.so.1", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=182504, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 190160, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f48da995000
mmap(0x7f48da99c000, 114688, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7000) = 0x7f48da99c000
mmap(0x7f48da9b8000, 32768, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x23000) = 0x7f48da9b8000
mmap(0x7f48da9c0000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2b000) = 0x7f48da9c0000
mmap(0x7f48da9c2000, 5840, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f48da9c2000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P~\2\0\0\0\0\0"..., 832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0"..., 784, 64) = 784
newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=1933688, ...}, AT_EMPTY_PATH) = 0
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0"..., 784, 64) = 784
mmap(NULL, 1985936, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f48da7b0000
mmap(0x7f48da7d6000, 1404928, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x26000) = 0x7f48da7d6000
mmap(0x7f48da92d000, 348160, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x17d000) = 0x7f48da92d000
mmap(0x7f48da982000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1d1000) = 0x7f48da982000
mmap(0x7f48da988000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f48da988000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpcre2-8.so.0", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=633480, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 631688, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f48da715000
```