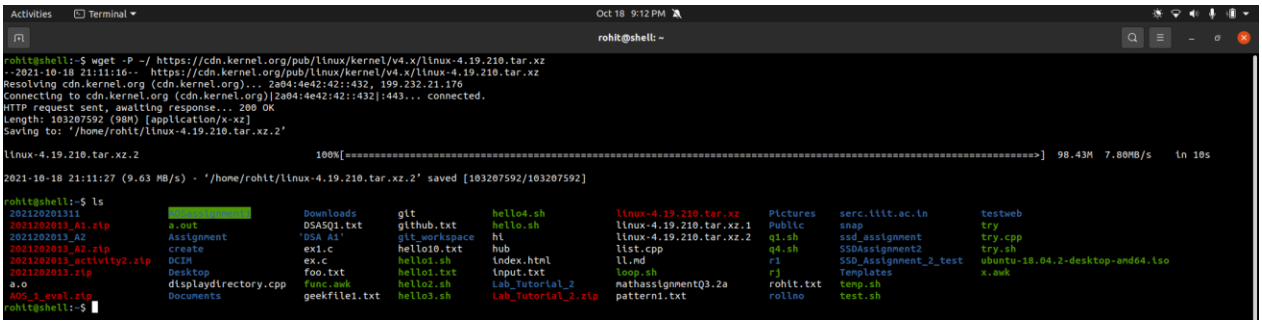
Assignment 2

Rohit Jangir(2021202013)

Step1:- I downloaded linux kernel 4.19.210 using “wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.19.210.tar.xz”



Step2:- Extracted kernel using this

```
rohit@shell:~$ tar -xvf linux-4.19.210.tar.xz
```

Now, we implement 4 system calls for different purposes and below are listed one by one with explanation.

Q1:- Created a system call “**rohithello()**” which when called print “hello world” on kernel log(dmesg).

-> I made directory name rohithello containing rohithello.c which is containing code to print hello world and Makefile to get rohithello.c compiled and included in kernel source code.

```
rohit@shell:~$ cd linux-4.19.210
rohit@shell:~/linux-4.19.210$ mkdir rohithello
rohit@shell:~/linux-4.19.210$ cd rohithello
rohit@shell:~/linux-4.19.210/rohithello$ touch rohithello.c
rohit@shell:~/linux-4.19.210/rohithello$ touch Makefile
rohit@shell:~/linux-4.19.210/rohithello$
```

-> Here I edited system Makefile because we need to tell the compiler that we have new system call which is present in rohithello directory.

```
PHONY += prepare0

ifeq ($(KBUILD_EXTMOD),)
core-y          += kernel/ certs/ mm/ fs/ ipc/ security/ crypto/ block/ rohithello/

vmlinux-dirs    := $(patsubst %/,%,$(filter %/, $(init-y) $(init-m) \
                     $(core-y) $(core-m) $(drivers-y) $(drivers-m) \
```

-> Here I have added system call to system call header file.

```
        return old;
}
asmlinkage long rohithello(void);
```

-> Added new “rohithello” system call entry to system call table syscall_64.tbl , here 64 refers to 64bit system.

```
546     x32     preadv2                         __x32_compat_sys_preadv64v2
547     x32     pwritev2                    __x32_compat_sys_pwritev64v2
548     64      rohit1                      rohithello
```

->Code of rohithello.c

```
1 #include <linux/kernel.h>
2
3 SYSCALL_DEFINE0(rohithello)
4 {
5         printk("Hello world\n");
6         return 0;
7 }
```

->code of Makefile

```
obj-y := rohithello.o
```

Q2:- System Call “rohitprint(string)” will take string as parameter and print it along message to linux log

-> Made file “rohitprint.c” in directory rohithello

```
rohit@shell:~/linux-4.19.210/rohithello$ touch rohitprint.c
rohit@shell:~/linux-4.19.210/rohithello$
```

-> Made changes in Makefile in rohithello directory

```
1 obj-y := rohithello.o rohitprint.c
```

-> Here I have added system call to system call header file.

```
asmlinkage long rohithello(void);
asmlinkage long rohitprint(char *);

#endif
```

-> Added new “rohitprint” system call entry to system call table syscall_64.tbl

```
547     x32     pwritev2                __x32_compat_sys_pwritev64v2
548     64      rohit1                  rohithello
549     64      rohit2                  rohitprint
```

->Code of rohitprint.c

```c
#include <linux/syscalls.h>
#include <linux/kernel.h>

SYSCALL_DEFINE1(rohitprint, char *,buf)
{
        char buffer[256];
        long copied = strncpy_from_user(buffer,buf, sizeof(buffer));
        printk("Rohit's system call returned this\n");
        return 0;
}
```

Q3:- This “rohitprocess()” system when called print parent and current process id. The file containing code is in rohithello named rohitprocess.c

```
rohit@shell:~/linux-4.19.210/rohithello$ touch rohitprocess.c
rohit@shell:~/linux-4.19.210/rohithello$
```

-> Accordingly edited Makefile in rohithello to get it compiled and included in source code of kernel

```
obj-y := rohithello.o rohitprint.c rohitprocess.o
```

-> Here I have added system call to system call header file.

```
asmlinkage long rohithello(void);
asmlinkage long rohitprint(char *);
asmlinkage long rohitprocess(void);
```

> Added new “rohitprocess” system call entry to system call table syscall_64.tbl

```
548     64      rohit1                  rohithello
549     64      rohit2                  rohitprint
550     64      rohit3                  rohitprocess
```

-> Code for “rohitprocess”

```c
#include <linux/syscalls.h>
#include <linux/kernel.h>
#include <linux/cred.h>
#include <linux/sched.h>

SYSCALL_DEFINE0(rohitprocess)
{
        printk("Parent Process Id: ",current->parent->pid);
        printk("Child Process Id: ",current->pid);
        return 0;
}
```

Q4:- This system call “rohitgetpid” calls another predefined system call “task_tgid_vnr(current)” which will return current process id.

-> Making rohitgetpid

```
rohit@shell:~/linux-4.19.210/rohithello$ touch rohitgetpid.c
rohit@shell:~/linux-4.19.210/rohithello$
```

-> Updating Makefile

```
obj-y := rohithello.o rohitprint.c rohitprocess.o rohitgetpid.o
```

-> Adding system call entry to system table

```
390 549     64      rohit2                  rohitprint
391 550     64      rohit3                  rohitprocess
392 551     64      rohit4                  rohitgetpid
```

-> Adding system call to system call header file

```
asmlinkage long rohithello(void);
asmlinkage long rohitprint(char *);
asmlinkage long rohitprocess(void);
asmlinkage long rohitgetpid(void);
```

-> The following system call have following code

```c
#include <linux/syscalls.h>
#include <linux/kernel.h>
#include <linux/cred.h>
#include <linux/sched.h>

SYSCALL_DEFINE0(rohitgetpid)
{
        return task_tgid_vnr(current);
}
```