Lab Assignment 1

- **STEPS FOLLOWED :--**

1. Login into baadal vm ->
   - ssh <kerberos username>@baadal.iitd.ac.in
   - ssh baadalvm@<private ip of my vm on baadal>

2. Login into the proxy and export it
   - sudo lynx https://proxy62.iitd.ernet.in/cgi-bin/proxy.cgi
   - export http_proxy=<IP of my proxy after logging in>:3128
   - export https_proxy=<IP of my proxy after logging in>:3128

3. Initial test of the version of kernel

```
baadalvm@baadalvm:~$ uname -r
5.4.0-48-generic
```

4. Downloading the recent version of the kernel along with all the packages required to compile it and then extracting the package
   - sudo wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.8.1.tar.xz
   - sudo apt-get install gcc &&
     sudo apt-get install libncurses5-dev &&
     sudo apt-get install bison &&
     sudo apt-get install flex &&
     sudo apt-get install libssl-dev &&
     sudo apt-get install libelf-dev &&
     sudo apt-get install dwarves &&
     sudo apt-get update
   - sudo tar -xvf linux-5.8.1.tar.xz -C/usr/src
   - cd /usr/src/linux-5.8.1

5. Executing the Syscall part
   - sudo mkdir mySyscall
   - cd mySyscall
   - sudo vim mySyscall.c
     Then using the insert functionality of vim inserting the following text in the file and then saving it using :wq

```
#include <linux/kernel.h>
asmlinkage long sys_hello(void)
{
    printk("Hello world This is abide.YYYY\n");
    return 0;
}
```

   - sudo vim Makefile
     Then using the insert functionality of vim inserting the following text in the file and then saving it using :wq

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

- cd ../
- sudo vim Makefile
  Then using the insert functionality of vim appending mySyscall to the core-y line and then saving it using :wq
  core-y += kernel/ certs/ mm/ fs/ ipc/ security/ crypto/ block/ mySyscall/
- cd include/linux/
- sudo vim syscalls.h
  Then using the insert functionality of vim add the following prototype in it and then save it using :wq
  asmlinkage long sys_mySyscall()
- cd ../
- cd ../
- cd arch/x86/entry/syscalls/
- sudo vim syscall_64.tbl
  Then using the insert functionality of vim add the following syscall to the table at the end and then save it using :wq
  548 <tab> 64 <tab> mySyscall <tab><tab> sys_mySyscall
- cd /usr/src/linux-5.8.1

6. Compiling the new version of the kernel and rebooting the system
   - sudo make menuconfig
     General setup → Local version → Wrote the string "nikita.0413"
   - sudo make -j2
   - sudo make modules_install
   - sudo make install
   - sudo update grub
   - sudo reboot

- **TEST :--**

1. uname -r

```
baadalvm@baadalvm:/usr/src/linux-5.8.1$ uname -r
5.8.1nikita.0413
```

2. => cd /usr/src/linux-5.8.1/mySyscall
   => sudo vim mySyscallTest.c
   Wrote the following code in it and then saved it.

```
#include <stdio.h>
#include <linux/kernel.h>
#include <sys/syscall.h>
#include <unistd.h>
int main()
{
    long int mySyscallTest = syscall(548);
    printf("System call sys_mySyscall returned %ld\n", mySyscallTest); return 0;
```

```
}
```

=> sudo gcc mySyscallTest.c
=> sudo ./a.out

```
baadalvm@baadalvm:/usr/src/linux-5.8.1/mySyscall$ sudo gcc mySyscallTest.c
baadalvm@baadalvm:/usr/src/linux-5.8.1/mySyscall$ sudo ./a.out
System call sys_mySyscall returned 0
```

=> sudo dmesg | sudo tee -a dmesgoutput.txt

The dmesgoutput.txt file contains the effect in dmesg

=> cd ../
=> sudo make clean
=> sudo make mrproper
=> sudo mv linux-5.8.1 linux_updated
=> cd ~
=> sudo tar -xvf linux-5.8.1.tar.xz -C/usr/src
=> cd /usr/src/
=> sudo mv linux-5.8.1 linux_old
=> sudo diff -ur linux_old linux_updated | sudo tee -a diff_using_ur.patch
=> sudo diff -ruN linux_old linux_updated | sudo tee -a diff_using_ruN.patch

SUBMITTED BY :-
-> Nikita Bhamu
-> 2018CS50413