**Assignment 4 (Bonus)**

* Device encryption file and code scripts are all in the same directory
* First created encdev.c file, created a file\_operation Structure & overridden/changed the reference of the following function pointers read, write, open, release. read function write function writes the data to writer global buffer generates a random bytes key of 16 bits and uses xor encryption to encrypt the data provided by the user, & store it to global variable, when user try to read the data give him that data along with the first 16bits as key
* Decdev.c file, created a file\_operation Structure same & overridden/changed the reference of the following function pointers read, write, open, release. read function the write function takes the encrypted message and uses first 16bits as the key and xor it again such that we again get what the original message was.
* For Description on how to compile open Commands Order.txt
* enter the text which you wanted to be encrypted in textToBeEncrypted.txt & follow Commands Order.txt
* the program will print encypted data in encrypted file .txt
* the program will print decrypted data in decryptedfile.txt
* the output will also be shown on console
* Please make sure to be root every time other wise you will not be able to load kernel module
* Two Errors are handled

1. If textToBeEncypted is missing
2. if any of the two encryption or decryption device is missing

///Make sure doing it everything as root

//For Creation of file & removal also

make

cd Encrypter

make

insmod Encrypter.ko

make clean

cd ../Decrypter

make

insmod Decrypter.ko

make clean

cd ..

//Steps to test

dmesg

mknod /dev/encdev c #majorNumber 1

mknod /dev/decdev c #majorNumber 1

./testFile

//Removing kernel module

rm /dev/encdev

rm /dev/decdev

rmmod Encrypter

rmmod Decrypter