**Linux Programming**

**Lab 6**

**17MIS1066**

**G Nanda Kishore Reddy**

System Calls For File Operations:

1.System calls for file oprn,read,write

Code:

open():

#include<stdio.h>

#include<fcntl.h>

#include<errno.h>

extern int errno;

int main()

{

    int fd = open("q1.txt", O\_RDONLY);

    printf("fd = %d\n", fd);

    if (fd ==-1)

    {

        printf("Error Number % d\n", errno);

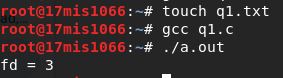
        perror("Program");

    }

    return 0;

}

Output:



Read & Write:

Code:

#include <stdio.h>

#include <sys/types.h>

#include <fcntl.h>

#include <string.h>

#include <errno.h>

#include <unistd.h>

int main()

{

    int fd;

    fd=open("/home/karthik/lab6/file1.txt",O\_RDWR);

    char word[50];

    read(fd,word,sizeof(word));

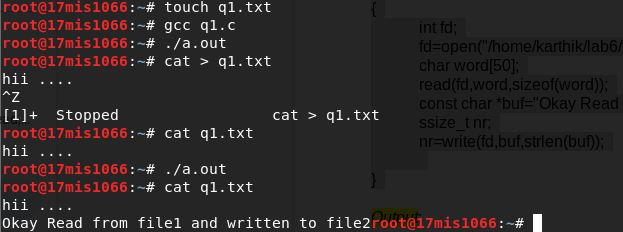
    const char \*buf="Okay Read from file1 and written to file2";

    ssize\_t nr;

    nr=write(fd,buf,strlen(buf));

}

Output:



2. Manage EINTR while accessing file using system calls.

Code:

#include<stdio.h>

#include<fcntl.h>

#include<errno.h>

#include<stdlib.h>

#include<string.h>

int main()

{

    int fd = open("file1.txt",  O\_RDONLY );

    int sz;

    sz = write(fd, "I am Inevitable\n", strlen("I am Inevitable"));

    if (sz == -1 && errno != EINTR)

    {

           perror("Read");

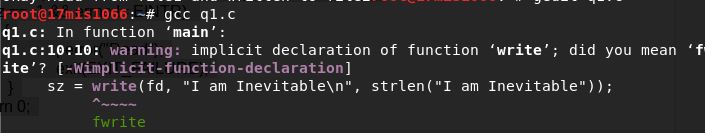
           exit(EXIT\_FAILURE);

       }

    return 0;

}

Output:



3. Do Non-Block read and write using system calls.

Code:

#include <stdio.h>

#include <sys/types.h>

#include <fcntl.h>

#include <string.h>

#include <errno.h>

#include <unistd.h>

int main()

{

    int fd,ret;

    fd=open("/home/karthik17mis1020/file1.txt",O\_RDWR);

    ssize\_t nr;

    char buf[BUFSIZ];

    start:

    nr=read(fd,buf,BUFSIZ);

    while(BUFSIZ!=0 && (ret = read(fd,buf,BUFSIZ))!=0)

    {

        if(nr==-1)

        {

            if(errno == EINTR)

            {

                goto start;

            }

            if(errno == EAGAIN)

            {

                continue;

            }

            else

            {

                perror("Read");

                break;

            }

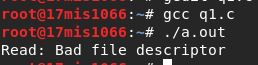
        }

    }

}

Output:

File

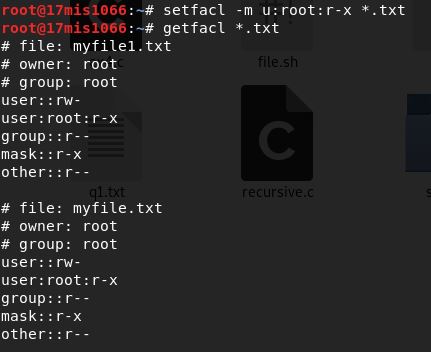
  
  
  
Permissions:

4. Disable Write permissions to user for all the files in specific folder.

Code:

So let’s create a directory called “acldemo” and create two files namely file1.txt,file2.txt in it.

Changing permissions to r-x i.e, disabling write permissions.



5.Set write permission for only one user on a file

