Linux programming
lab-6

k.Bhavesh | 17MIS1079

System Calls For File Operations:

1.System calls for file oprn,read,write

coad:

```
#include<stdio.h>
#include<fcntl.h>
#include<errno.h>
extern int errno;
int main()
{
   int fd = open("tt.txt", O_RDONLY);
   printf("fd = %d\n", fd);
   if (fd ==-1)
   {
      printf("Error Number % d\n", errno);
      perror("Program");
   }
   return 0;
}
```

output:

```
root@Bhavesh-17mis1079:~# gcc l6.c
root@Bhavesh-17mis1079:~# ./a.out
fd = 3
```

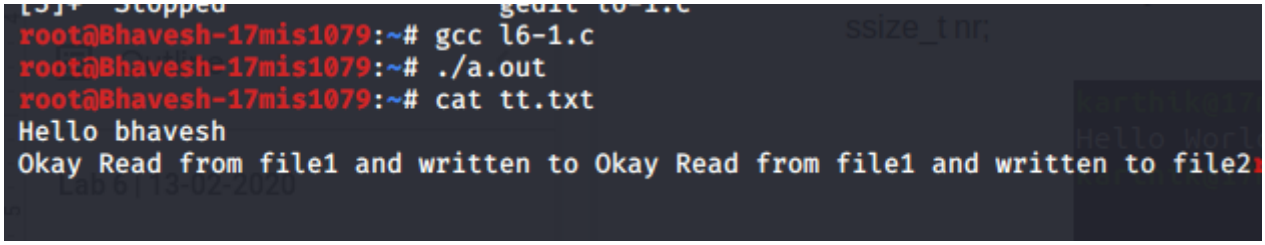
2)
Read & Write:

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("/root/tt.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:

```
root@Bhavesh-17mis1079:~# gcc l6-1.c
l6-1.c: In function 'main':
l6-1.c:10:10: warning: implicit declaration of function 'write'; did you mean 'fwrite'? [-Wimplicit-function-declaration]
   10 |     sz = write(fd, "I am Inevitable\n", strlen("I am Inevitable"));
      |          ^~~~~
      |          fwrite
root@Bhavesh-17mis1079:~# ./a.out
Read: Bad file descriptor
```

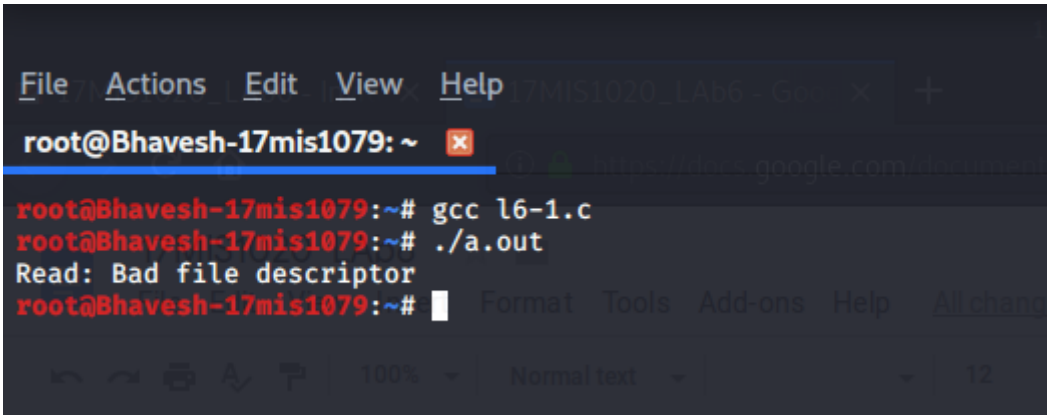
3. Do Non-Block read and write using system calls.

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("/root/tt.txts",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:

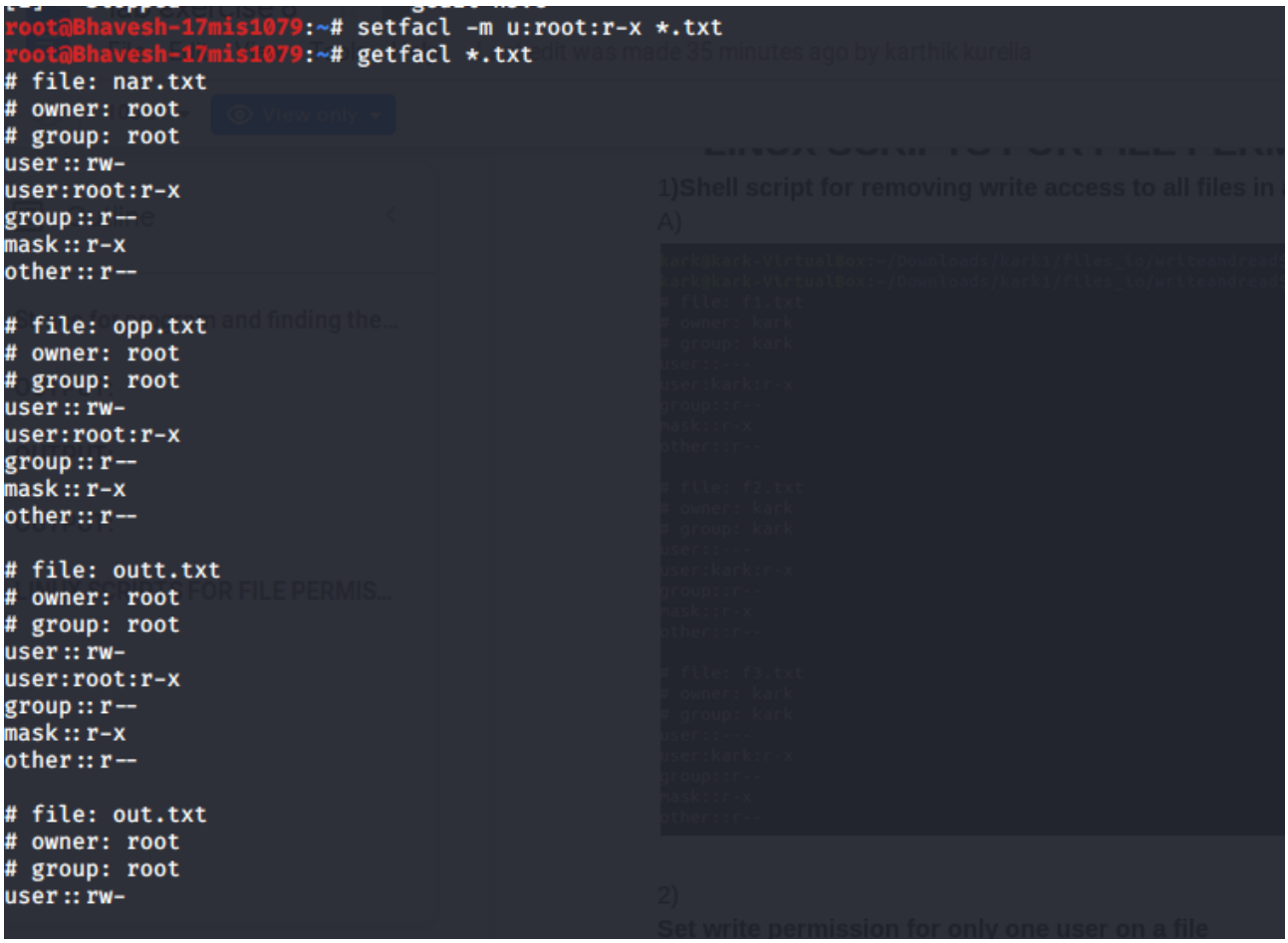


File Permissions:

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

linux scripts fro file permission using ACL:

1)shell script to remove write access to all the files



2)set write premisssion to only one user on th file