

Linux programming

lab-6

k.Bhavesh | 17MIS1079

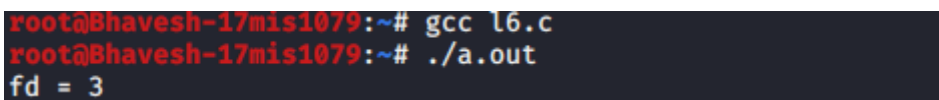
System Calls For File Operations:

1.System calls for file oprn,read,write

code:

```
#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";

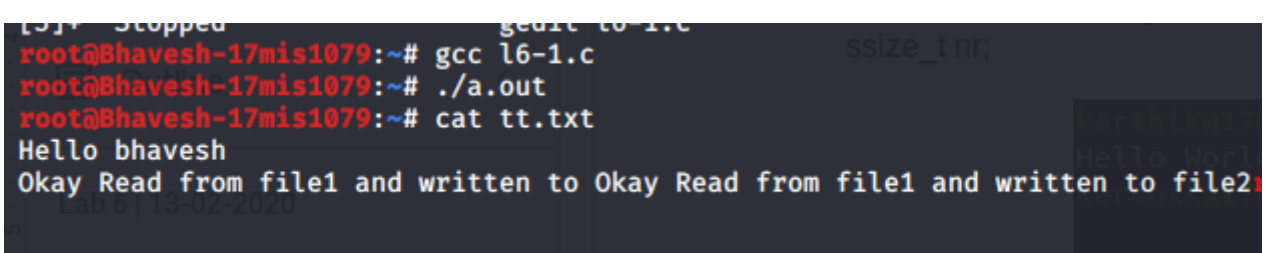
ssize_t nr;

nr=write(fd,buf,strlen(buf));

}

```

output:



```

[5] ~$ gcc l6-1.c
root@Bhavesh-17mis1079:~# gcc l6-1.c
root@Bhavesh-17mis1079:~# ./a.out
root@Bhavesh-17mis1079:~# cat tt.txt
Hello bhavesh
Okay Read from file1 and written to Okay Read from file1 and written to file2

```

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>
```

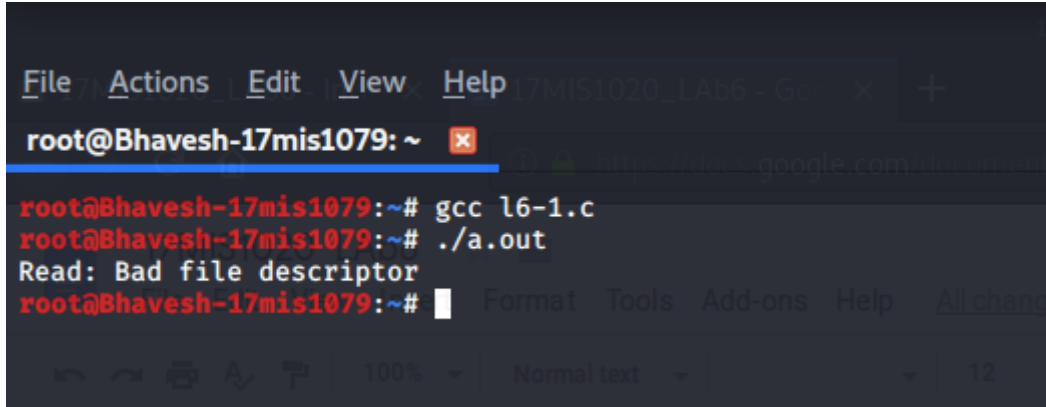
```
int main()
{
    int fd,ret;
    fd=open("/root/tt.txts",O_RDWR);
    ssize_t nr;
    char buf[BUFSIZ];
    start:
    nr=read(fd,buf,BUFSIZ);
    while(BUFSIZ!=0 && (ret = read(fd,buf,BUFSIZ))!=0)
    {
        if(nr==-1)
        {
            if(errno == EINTR)
            {
```

```

        goto start;
    }
    if(errno == EAGAIN)
    {
        continue;
    }
    else
    {
        perror("Read");
        break;
    }
}
}
}
}

```

output:



```

File Actions Edit View Help [7MIS1020_Lab6 - Google Chrome] x +
root@Bhavesh-17mis1079: ~ x
root@Bhavesh-17mis1079:~# gcc l6-1.c
root@Bhavesh-17mis1079:~# ./a.out
Read: Bad file descriptor
root@Bhavesh-17mis1079:~#
Format Tools Add-ons Help All changes
100% Normal text 12

```

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

```
root@Bhavesh-17mis1079:~# setfacl -m u:root:r-x *.txt
root@Bhavesh-17mis1079:~# getfacl *.txt
# file: nar.txt
# owner: root
# group: root
user::rw-
user:root:r-x
group::r--
mask::r-x
other::r--

# file: opp.txt
# owner: root
# group: root
user::rw-
user:root:r-x
group::r--
mask::r-x
other::r--

# file: outt.txt
# owner: root
# group: root
user::rw-
user:root:r-x
group::r--
mask::r-x
other::r--

# file: out.txt
# owner: root
# group: root
user::rw-
```

1)Shell script for removing write access to all files in A)

```
#!/bin/bash
# file: f1.txt
# owner: kark
# group: kark
user::r--
user:kark:r-x
group::r--
mask::r-x
other::r--

# file: f2.txt
# owner: kark
# group: kark
user::r--
user:kark:r-x
group::r--
mask::r-x
other::r--

# file: f3.txt
# owner: kark
# group: kark
user::r--
user:kark:r-x
group::r--
mask::r-x
other::r--
```

2)
Get write permission for only one user on a file

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

```
root@Bhavesh-17mis1079:~# setfacl -m u:root:-w- v3.txt
root@Bhavesh-17mis1079:~# getfacl v3.txt
# file: v3.txt
# owner: root
# group: root
user::rw-
user:root:-w-
group::r--
mask::rw-
other::r--
```