

# CS314: Lab Report

## Assignment 9

Devdatt N

200010012@iitdh.ac.in

26 March 2023

### 1 Introduction

We work on the mfs file system, only for the file system mounted at `/home` . We have to print statements when a file is created, read, written or deleted.

### 2 File Creation

For this, we have changed the file `open.c` at location `/usr/src/minix/servers/vfs` . We have included this particular piece of code to execute if file creation is successful:

```
    r = err_code;
//start
    if (r == OK){
        exist = FALSE;
        struct vmnt *vmpPath;
        vmpPath = find_vmnt(vp->v_fs_e);
        if (strcmp(vmpPath->m_mount_path, "/home") == 0)
            printf("File created: %llu\n200010012\n", vp->v_inode_nr);
    }
//end
```

Figure 1: Code Added

We can see that this has been done successfully:

```
#
#
# Roll Numbers: 2000100- 03 & 12
Time Quantum: 200, Time Quantum Executed: 6 Endpoint: 65563
touch hello
Minix: PID 212 created
Roll Numbers: 2000100- 03 & 12
Time Quantum: 200, Time Quantum Executed: 200 Endpoint: 32956
File created: 98
200010012
Minix: PID 212 exited
```

Figure 2: Results

We can see that a file with inode 98 has been created.

### 3 File Reading

For printing statements, while a file is being read, we have included the following code in the function `read_write()` :

```
/* filp_pos = position;
//start
struct vmnt *vmp;
vmp = find_vmnt(vp->v_fs_e);
if (rw_flag == WRITING && strcmp(vmp->m_mount_path, "/home") == 0)
    printf("File written: %llu; nbytes = %zu; offset = %llu\n200010012\n", vp->v_inode_nr, size, f->filp_pos);
if (rw_flag == READING && strcmp(vmp->m_mount_path, "/home") == 0)
    printf("File read: %llu; nbytes = %zu; offset = %llu\n200010012\n", vp->v_inode_nr, size, f->filp_pos);
//end
```

Figure 3: Code Added

We can see that this works fine:

```
# cat hello.txt
Minix: PID 248 created
Roll Numbers: 2000100- 03 & 12
Time Quantum: 200, Time Quantum Executed: 200 Endpoint: 32992
File read: 98; nbytes = 4096; offset = 7
200010012

hello
File read: 98; nbytes = 4096; offset = 7
200010012
Minix: PID 248 exited
#
```

Figure 4: Result

We are reading from a file `hello.txt` and the inode number is the same as displayed when we created the file as well, which is 98.

### 4 File Writing

We have to make changes in the same file `read.c` for this, as shown in the last screenshot for File Reading. We get these results after writing with the `vi` editor:

```
~
~
~
File written: 98; nbytes = 15; offset = 15
200010012
hello.txt: 3 lines, 15 charac
ters
Minix: PID 249 exited
#
```

Figure 5: Result

We can see that it works fine.

## 5 File Deletion

We will use `rm hello.txt` to remove the file. However, we have to first make some changes in the `do_unlink()` function in the `link.c` file which is located among other files in the same directory. These are the changes:

```
// start

lookup_init(&stickycheck, resolve.l_path, PATH_RET_SYMLINK, &vmp2, &vp);
stickycheck.l_vmnt_lock = VMNT_READ;
stickycheck.l_vnode_lock = VNODE_READ;

vp = advance(dirp, &stickycheck, fp);

if (strcmp(vmp->m_mount_path, "/home") == 0)
{
    printf("File deleted: %llu\n200010012\n", vp->v_inode_nr);
}
if (vp != NULL)
{
    unlock_vnode(vp);
    put_vnode(vp);
}

// end
```

Figure 6: Code Added

We can see that this works perfectly:

```
#
#
# rm hello.txt
Minix: PID 250 created
Roll Numbers: 2000100- 03 & 12
Time Quantum: 200, Time Quantum Executed: 200 Endpoint: 32994
File deleted: 98
200010012
Minix: PID 250 exited
# _
```

Figure 7: Result

## 6 Note

Something interesting that I noted that was when I use the `vim` editor and edit a file, after saving it the system just crashes. I'm looking into reasons why this would happen. However, it works with the `vi` editor. Perhaps this has something to do with how vim is made.