

1 File Operations

1.1 Open a File

To obtain the following output, **file created: inode_number**, whenever a new file is created the following changes were made in **minix/servers/vfs/open.c** inside the **common_open** function.

```
1  if (r == OK)
2  {
3      exist = FALSE; /* We just created the file */
4
5      /*-----*/
6      /* Lab 9 */
7      struct vmnt *virtual_mount;
8      virtual_mount = find_vmnt(vp->v_fs_e);
9      if (strcmp(virtual_mount->m_mount_path, "/home") == 0)
10     {
11         printf("file created: %llu\n", vp->v_inode_nr);
12     }
13
14     /*-----*/
15 }
```

1.2 Read and Write

To obtain the following outputs, **file read/write: inode_number; nbytes = nbytes; offset = offset**, whenever a file is read from or written to, the following changes were made in **minix/servers/vfs/read.c** inside the **read_write** function.

```
1  if (r >= 0)
2  {
3      /*-----*/
4      /* Lab 9 */
5
6      struct vmnt *virtual_mount;
7      virtual_mount = find_vmnt(vp->v_fs_e);
8      if (rw_flag == WRITING && strcmp(virtual_mount->m_mount_path, "/home") == 0)
9      {
10         printf("file write: %llu; nbytes = %d; offset = %llu\n", vp->v_inode_nr, size,
11                position);
12     }
13     else if (rw_flag == READING && strcmp(virtual_mount->m_mount_path, "/home") == 0)
14     {
15         printf("file read: %llu; nbytes = %d; offset = %llu\n", vp->v_inode_nr, size,
16                position);
17     }
18 }
```

```

15     }
16
17     /*-----*/
18
19     position = new_pos;
20     cum_io += cum_io_incr;
21 }

```

1.3 Delete File

To obtain the following outputs, **file deleted: inode_number**, whenever a file is deleted, the following changes were made in **minix/servers/vfs/link.c** inside the **do_unlink** function.

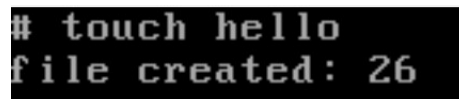
```

1  /*-----*/
2  /* Lab 9 */
3  // Directory lookup (vpm2 <-> mount node, vmp<->directory node)
4  lookup_init(&stickycheck, resolve.l_path, PATH_RET_SYMLINK, &vmp2, &vp);
5  // Set directory operation to read only
6  stickycheck.l_vmnt_lock = VMNT_READ;
7  stickycheck.l_vnode_lock = VNODE_READ;
8  // Look up inode of file to unlink to retrieve owner
9  vp = advance(dirp, &stickycheck, fp);
10
11  if (strcmp(vmp->m_mount_path, "/home") == 0)
12  {
13      printf("file deleted: %llu\n", vp->v_inode_nr);
14  }
15
16  if (vp != NULL)
17  {
18      unlock_vnode(vp);
19      put_vnode(vp);
20  }
21
22  /*-----*/

```

1.4 Outputs

The outputs of commands after building the minix kernel are shown below.



```

# touch hello
file created: 26

```

Figure 1: Output on running touch command

```
# echo helloworld >> hello
file write: 26; nbytes = 11; offset = 0
# echo helloworld >> hello
file write: 26; nbytes = 11; offset = 11
# echo helloworld >> hello
file write: 26; nbytes = 11; offset = 22
# echo helloworld >> hello
file write: 26; nbytes = 11; offset = 33
# echo helloworld >> hello
file write: 26; nbytes = 11; offset = 44
```

Figure 2: Output on appending text to file

```
# cat hello
file read: 26; nbytes = 4096; offset = 0
helloworld
helloworld
helloworld
helloworld
helloworld
file read: 26; nbytes = 4096; offset = 55
```

Figure 3: Output on reading file contents

```
# rm hello
file deleted: 26
```

Figure 4: Output on deleting the file