Assignment 8

Operating Systems Lab

Mohammad Sameer | 190010024 | 190010024@iitdh.ac.in

Abstract

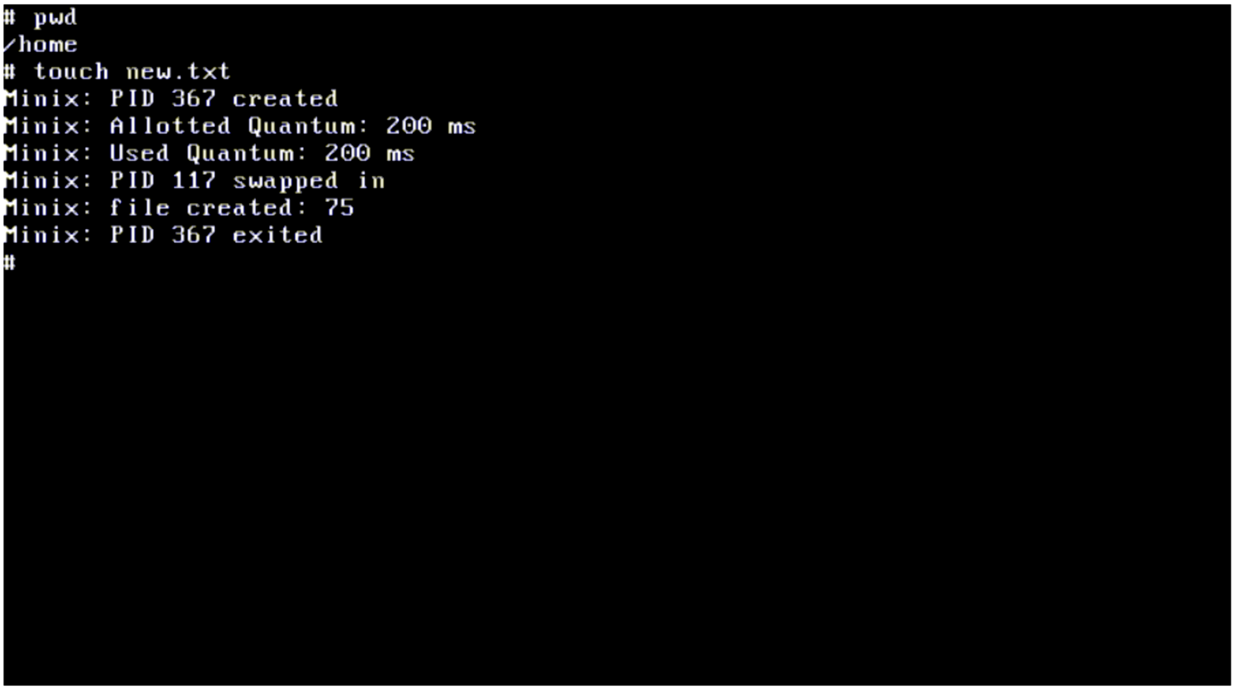
In this assignment we have to trace the working on the mfs file system, only for the file system mounted at /home.

File Creation

When a new file is created, we had to print the event along with its inode number. To do this the common_open function of file open.c located at minix/servers/vfs/open.c was modified to include the following lines of code:

```
struct vmnt *vmpPath;
vmpPath = find_vmnt(vp->v_fs_e);
if (strcmp(vmpPath->m_mount_path, "/home") == 0) {
    printf("Minix: file created: %llu\n", vp->v_inode_nr);
}
```

Also, since we use string comparison function, the string library was included whenever needed. The below figure shows the print statement in minix when a new file new.txt is created.



```
# pwd
/home
# touch new.txt
Minix: PID 367 created
Minix: Allotted Quantum: 200 ms
Minix: Used Quantum: 200 ms
Minix: PID 117 swapped in
Minix: file created: 75
Minix: PID 367 exited
#
```

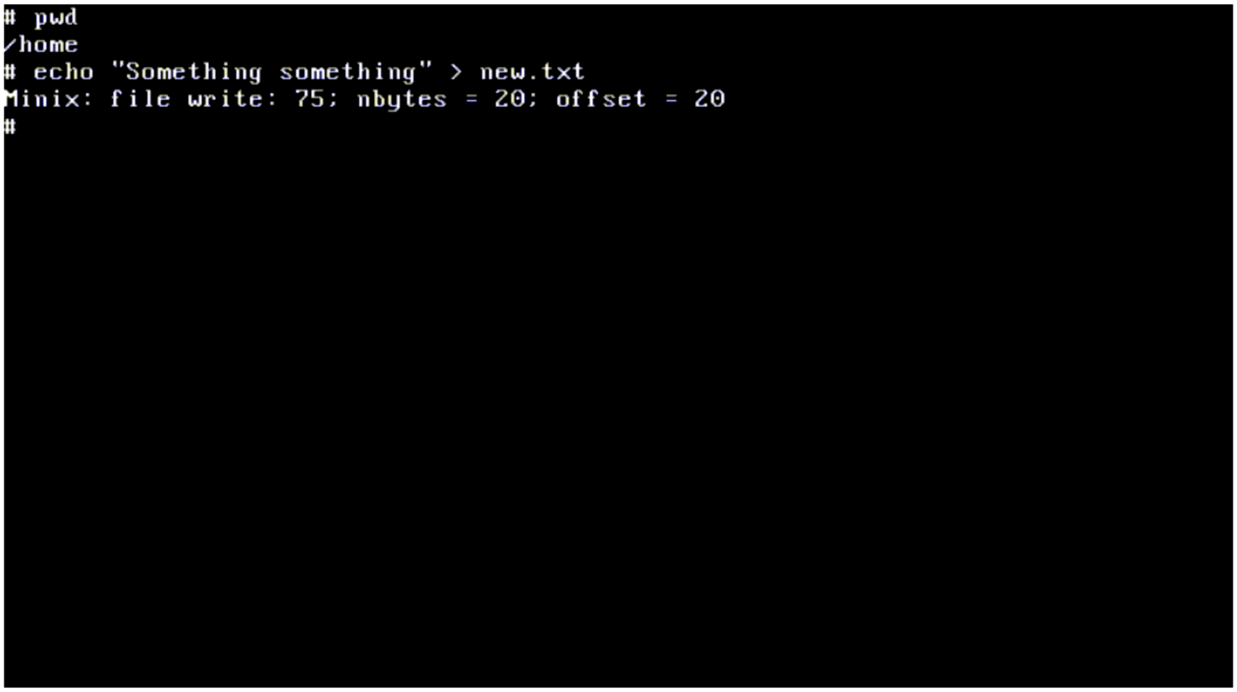
Figure 1: File Creation

File Writing and Reading

Whenever data is written/read in/from a file, we had to print the event along with its `inode` number, `bytes written` and `offset`. To do this the `read_write` function of file `read.c` located at `minix/servers/vfs/read.c` was modified to include the following lines of code:

```
struct vmnt *vmp;
vmp = find_vmnt(vp->v_fs_e);
if (rw_flag == WRITING && (strcmp(vmp->m_mount_path, "/home") == 0)) {
    printf("Minix: file write: %llu; nbytes = %zu; offset = %llu\n",
    vp->v_inode_nr, size, f->filp_pos);
}
if (rw_flag == READING && (strcmp(vmp->m_mount_path, "/home") == 0)) {
    printf("Minix: file read: %llu; nbytes = %zu; offset = %llu\n",
    vp->v_inode_nr, size, f->filp_pos);
}
```

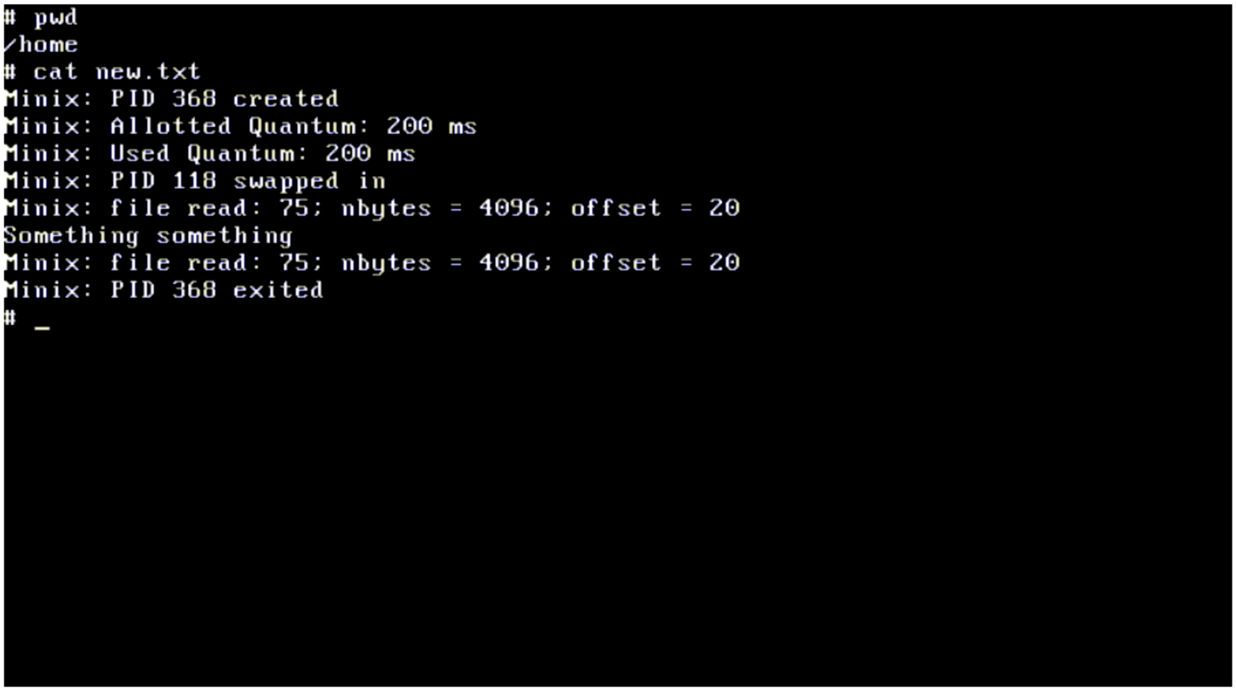
Also, since we use string comparison function, the string library was included whenever needed. The below figure shows the print statement in minix when a new line is written into `new.txt` file.



```
# pwd
/home
# echo "Something something" > new.txt
Minix: file write: 75; nbytes = 20; offset = 20
#
```

Figure 2: File Writing

The below figure shows the print statement in minix when `new.txt` file is read.



```
# pwd
/home
# cat new.txt
Minix: PID 368 created
Minix: Allotted Quantum: 200 ms
Minix: Used Quantum: 200 ms
Minix: PID 118 swapped in
Minix: file read: 75; nbytes = 4096; offset = 20
Something something
Minix: file read: 75; nbytes = 4096; offset = 20
Minix: PID 368 exited
# _
```

Figure 3: File Reading

File Deletion

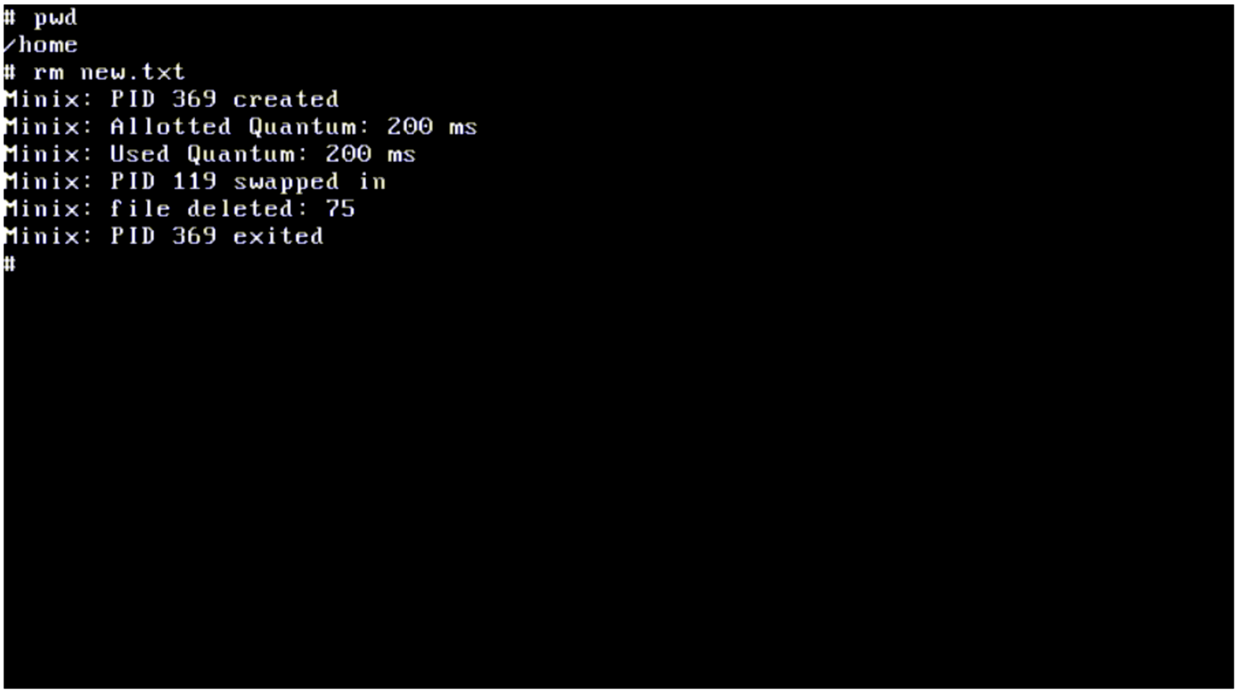
When an existing file is deleted, we had to print the event along with its `inode` number. To do this the `do_unlink` function of file `link.c` located at `minix/servers/vfs/link.c` was modified to include the following lines of code:

```
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("Minix: file deleted: %llu\n", vp->v_inode_nr);
}

if (vp != NULL){
    unlock_vnode(vp);
    put_vnode(vp);
}
```

Also, since we use string comparison function, the string library was included whenever needed. The below figure shows the print statement in minix when `new.txt` file is deleted.



```
# pwd
/home
# rm new.txt
Minix: PID 369 created
Minix: Allotted Quantum: 200 ms
Minix: Used Quantum: 200 ms
Minix: PID 119 swapped in
Minix: file deleted: 75
Minix: PID 369 exited
#
```

Figure 4: File Deletion