In Sem-I   OS (Advanced)   190031187

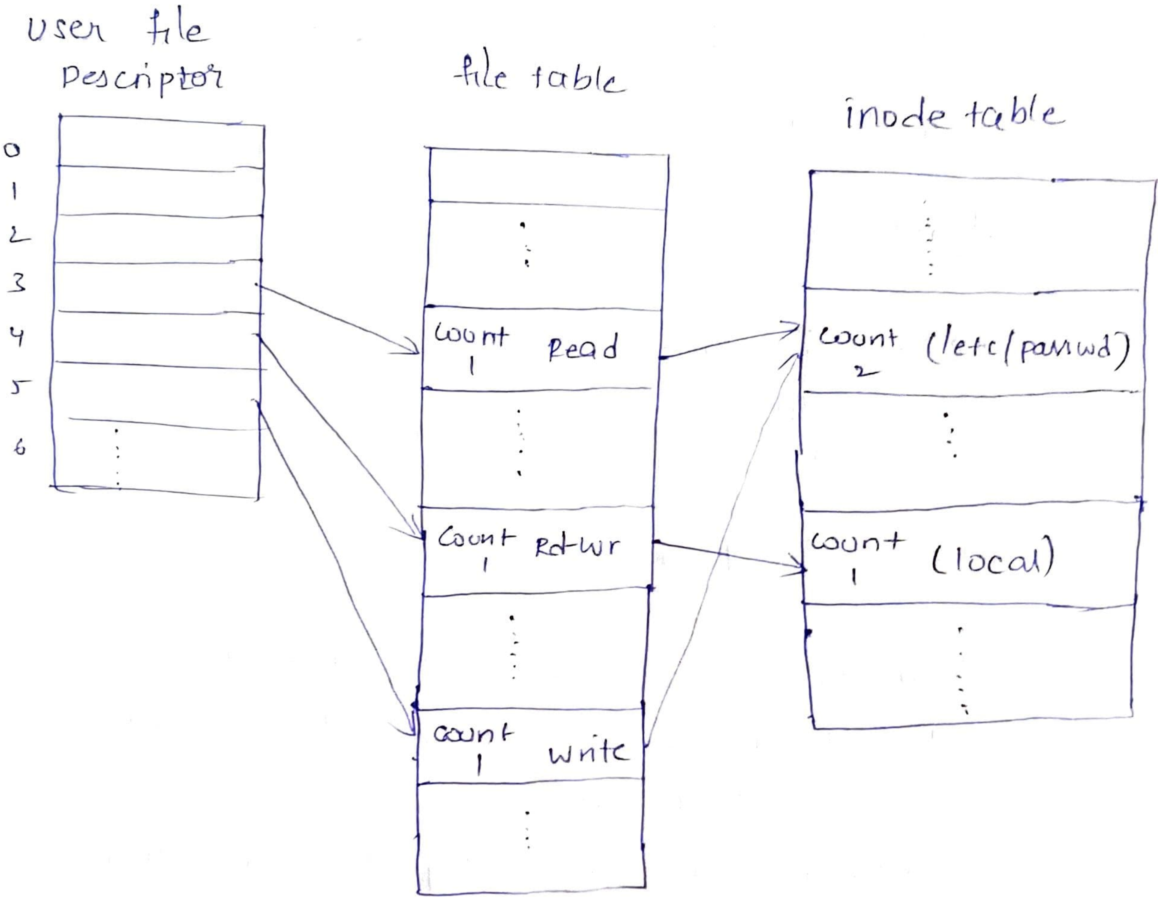
3. (i) unix uses three tables to hold data about open files they are user file descriptor table, inode table, file table

```
algorithm open

inputs    file name
          type of open
          file permissions

Output: file descriptor

    convert file name to inode (namei);
    if ( file does not exist or not permitted access )
        return error;
    allocate file table entry for inode, intialize
    Count, offset ;
    allocate user file descriptor entry, set pointer to
    file table entry ;
    if ( type of open specifies truncate file )
        free all file blocks ;
    unlock ( inode);
    return (file descriptor);
```

→ The kernel searches the filesystem for the file name parameter using namei.

→ It checks permission for opening the file after it finds the in-core inode and allocates an entry in the file table for the open file

Example



fd1 = open ("/etc / passwd", O_RDONLY);

fd2 = open ("local", O_RDWR);

fd3 = open ("/etc/passwd", O_WRONLY);

Now read () system call is uled to read data and copy to a buffer

read (fd, buffer, count)

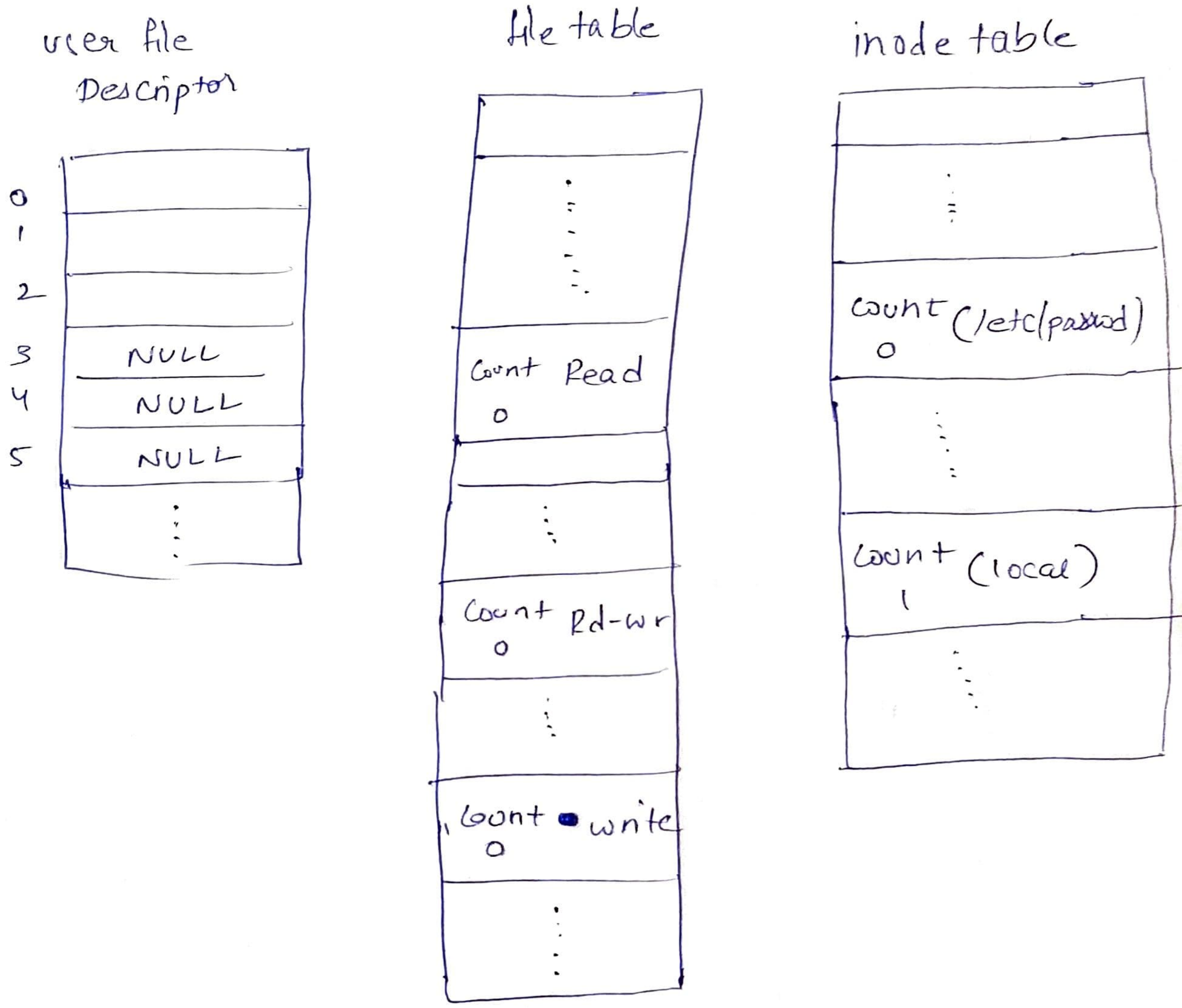
write () system call is used to write the data into the file

write (fd, buffer, count) ;

190031187

close () System call closes the file Descriptor

close (fd);



In sem-1 OSD (advanced) 190031187

2. There are multiple system calls to create a new file or directory:

mkdir creates a new directory. open with the O_CREATE flag create a new data file and

mknod creates new device file

```
mkdir("/dir");
fd= open("/dir/file", O-CREATE|O_WRONLY);
close(fd);
mknod("/console", 1, 1);
```

Here, mknod creates a file in the file system, but the file has no contents

Instead the file's metadata marks it as a records the major and minor device numbers. which uniquely identify a kernel device.