

## Questions and Answers on Unix File Management

### 1. How do we change File Access Permissions?

Answer: Every file has following attributes in it:

- owner's user ID
- owner's group ID

File access mode word

(r w x) - (r w x) - (r w x)

(user permission) - (group permission) - (others permission)

For changing the access mode, we always use `chmod(filename,mode)`.

### 2. What are links and symbolic links in UNIX file system?

Answer: A link is the second name for a file. Links are used to assign more than one name to a file, but it cannot be used to assign a directory more than one name or to link file-names on different computers. Symbolic link 'is' a file that only includes the name of another file in it. Operation on the symbolic link is may directed to the file pointed by the it. Both the limitations of links are eliminated in symbolic links.

**Commands use for linking files are:**

- Link "ln filename1 filename2"
- Symbolic link "ln -s filename1 filename2"

### 3. What is a FIFO?

Answer: These are other wisely called as 'named pipes'. FIFO (first-in-first-out) is a special file can be said to be data transient. Once data is read from named pipe, it cannot be read again. Also, data is read only in the order written. It is used in inter-process communication where a process writes to one end of the pipe and the other reads from the other end.

### 4. How do you create special files like named pipes and device files?

Answer: The system called `mknod` creates special files according to the following sequence.

- Kernel assigns new inode,
- It sets the file type for indicating that the file is a pipe, directory or special file,
- If it is a device file, it makes the other entries like major, minor device numbers.

**For example:**

If the device is a disk, major device number always refers to the disk controller and minor device number is the disk.

#### **5. Give a brief idea about representation of devices in UNIX.**

Answer: All devices are represented by files which are called special files that are located in /dev directory. That's why, device files and other files are named and accessed in the same way. A 'regular file' is only an ordinary data file in the disk. A 'block special file' represents a device with characteristics similar to a disk. A 'character special file' represents a device with characteristics similar to a keyboard.

#### **6. Discuss briefly about the directory representation in UNIX.**

Answer: A Unix directory is a file containing and also a correspondence between file names and inodes. A directory is a special file which are maintains by the kernel. Only kernel can modifies directories, but processes can read directories. The contents of a directory are a list of file name and inode number pairs. When new directories are created, kernel makes two entries named '.' and '..' System call for creating directory is mkdir

#### **7. What are the Unix system calls for I/O?**

Answer: The Unix system calls for I/O are as follows:

1. open (pathname,flag,mode) – open a file
2. creat (pathname,mode) – create a file
3. close(filedes) -to close an open file
4. read(filedes,buffer,bytes) -to read data from an open file
5. write(filedes,buffer,bytes) -to write data to an open file
6. lseek(filedes,offset,from) - position an open file
7. dup(filedes) -to duplicate an existing file descriptor
8. dup2(oldfd,newfd) –to duplicate to a desired file descriptor
9. fcntl(filedes,cmd,arg) -to change properties of an open file
10. ioctl(filedes,request,arg) -to change the behaviour of an open file
11. The difference between fcntl anf ioctl is that the former is intended for any open file, while the latter is for device-specific operations.

#### **8. Discuss the mount and unmount system calls.**

Answer: The privileged mount system call is usually used to attach a file system to a directory of another file system; the unmount system call detaches a file system. When we mount another file system on to our directory, we are essentially splicing one directory tree onto a branch in another directory tree.

The first argument to mount call is the mount point, i.e, a directory in the current file naming system. The second argument is the file system to mount to that point. When you insert a cdrom to your unix system's drive, the file system in the cdrom automatically mounts to `"/dev/cdrom"` in your system.

### **9. How does the inode map to data block of a file?**

Answer: Inode consists of 13 block addresses. The first 10 which are direct block addresses of the first 10 data blocks in the file. The 11th address points to a 1-level index block. The 12th address points to a 2-level index block. The 13th address points to a 3-level index block. This provides a really large maximum file size with efficient access to large files, but also small files are accessed directly in 1 disk read.

### **10. Define 'inode'?**

Answer: All UNIX files have its description which are stored in a structure called 'inode'. The inode includes info about the size of file, location of file, time of last access, time of last modification of file, permission to access and so on. Directories are also defined as files and have an associated inode. In addition for describing about the file, the inode includes pointers to the data blocks of the file. If the file is large, inode includes indirect pointer to a block of pointers to additional data blocks. A block is typically 8k.

**EDUCATION**  
Success Comes in Way...