OPERATING SYSTEMS

Storage Management - 7

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

OPERATING SYSTEMS

Storage Management - 7:
File System - Mounting, Sharing, Protection

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

OPERATING SYSTEMS

Course Syllabus - Unit 3

Unit 4: Storage Management

Mass-Storage Structur - Mass-Storage overview, Disk Scheduling, Swap-Space Management, RAID structure. File System Interface - file organization/structure and access methods, directories, sharing File System Implementation/Internals: File control Block (inode), partitions & mounting, Allocation methods.

Case Study: Linux/Windows File Systems

OPERATING SYSTEMS

Course Outline

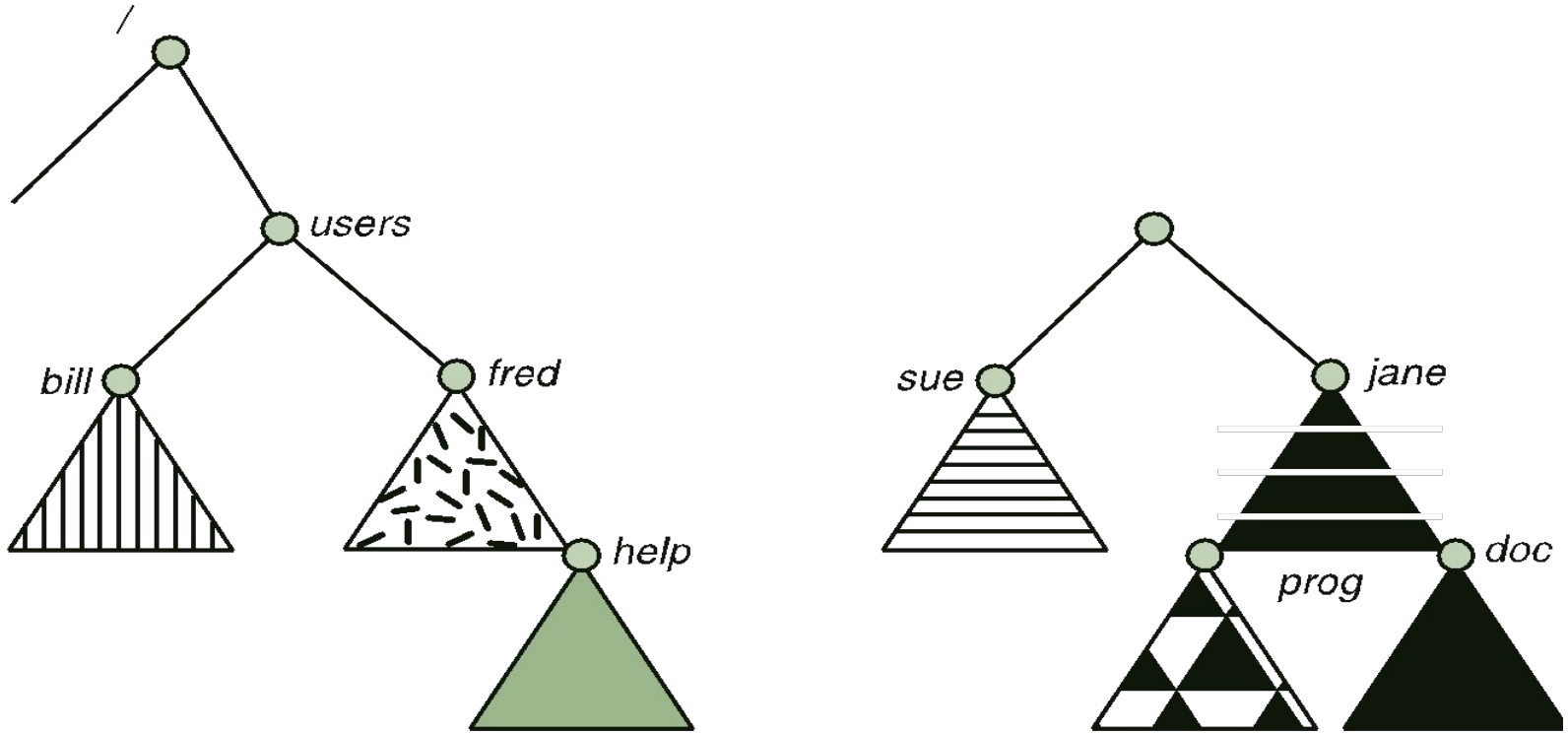
37	Mass-Storage Structure: Mass-Storage overview	12.1	82.1
38	Disk Scheduling – FCFS, SSTF, SCAN, C-SCAN, LOOK	12.4	
39	Swap-Space Management, RAID Structure	12.6,12.7	
40	File Concept, File Structure, Access Methods	10.1-10.2	
41	Directory and Disk Structure	10.3	
42	File-System Mounting, File Sharing, Protecting	10.4-10.6	
43	Implementing File-Systems: File control Block (inode), partitions & mounting	11.1,11.2	
44	Disk Space Allocation methods: Contiguous, Linked, Indexed	11.4	
45	Case Study: Unix/Linux File systems	11.8	
46	NFS	16.7	

Topic Outline

- File System Mounting
- File Sharing
- Protection

File System Mounting

- A file system must be **mounted** before it can be accessed
- A unmounted file system Figure b is mounted at a **mount point**



(a) (b)

File System Mounting

- Before you can access the files on a file system, you need to mount the file system. Mounting a file system attaches that file system to a directory (mount point) and makes it available to the system. The root (/) file system is always mounted. Any other file system can be connected or disconnected from the root (/) file system.

- The mount point is the directory (usually an empty one) in the currently accessible filesystem to which a additional filesystem is mounted.

- It becomes the root directory of the added directory tree, and that tree becomes accessible from the directory to which it is mounted (i.e., its mount point).

- Any original contents of a directory that is used as a mount point become invisible and inaccessible while the filesystem is still mounted.

File System Mounting

- The /mnt directory exists by default on all Unix-like systems. It, or usually its subdirectories (such as /mnt/floppy and /mnt/usb), are intended specifically for use as mount points for removable media such as CDROMs, USB key drives and floppy disks.

- On some operating systems, everything is mounted automatically by default so that users are never even aware that there is any such thing as mounting. Linux and other Unix-like systems can likewise be configured so that everything is mounted by default, as a major feature of such systems is that they are highly configurable.

File System Mounting

- However, they are not usually set up this way, for both safety and security reasons. Moreover, only the root user (i.e., administrative user) is generally permitted by default to mount devices and filesystems on such systems, likewise as safety and security measures.

- In the simplest case, such as on some personal computers, the entire filesystem on a computer running a Unix-like operating system resides on just a single partition, as is typical for Microsoft Windows systems. More commonly, it is spread across several partitions, possibly on different physical disks or even across a network. Thus, for example, the system may have one partition for the root directory, a second for the /usr directory, a third for the /home directory and a fourth for use as swap space. (Swap space is a part of HDD that is used for virtual memory, which is the simulation of additional main memory).

File System Mounting

- The only partition that can be accessed immediately after a computer boots (i.e., starts up) is the root partition, which contains the root directory, and usually at least a few other directories as well. The other partitions must be attached to this root filesystem in order for an entire, multiple-partition filesystem to be accessible. Thus, about midway through the boot process, the operating system makes these non-root partitions accessible by mounting them on to specified directories in the root partition.

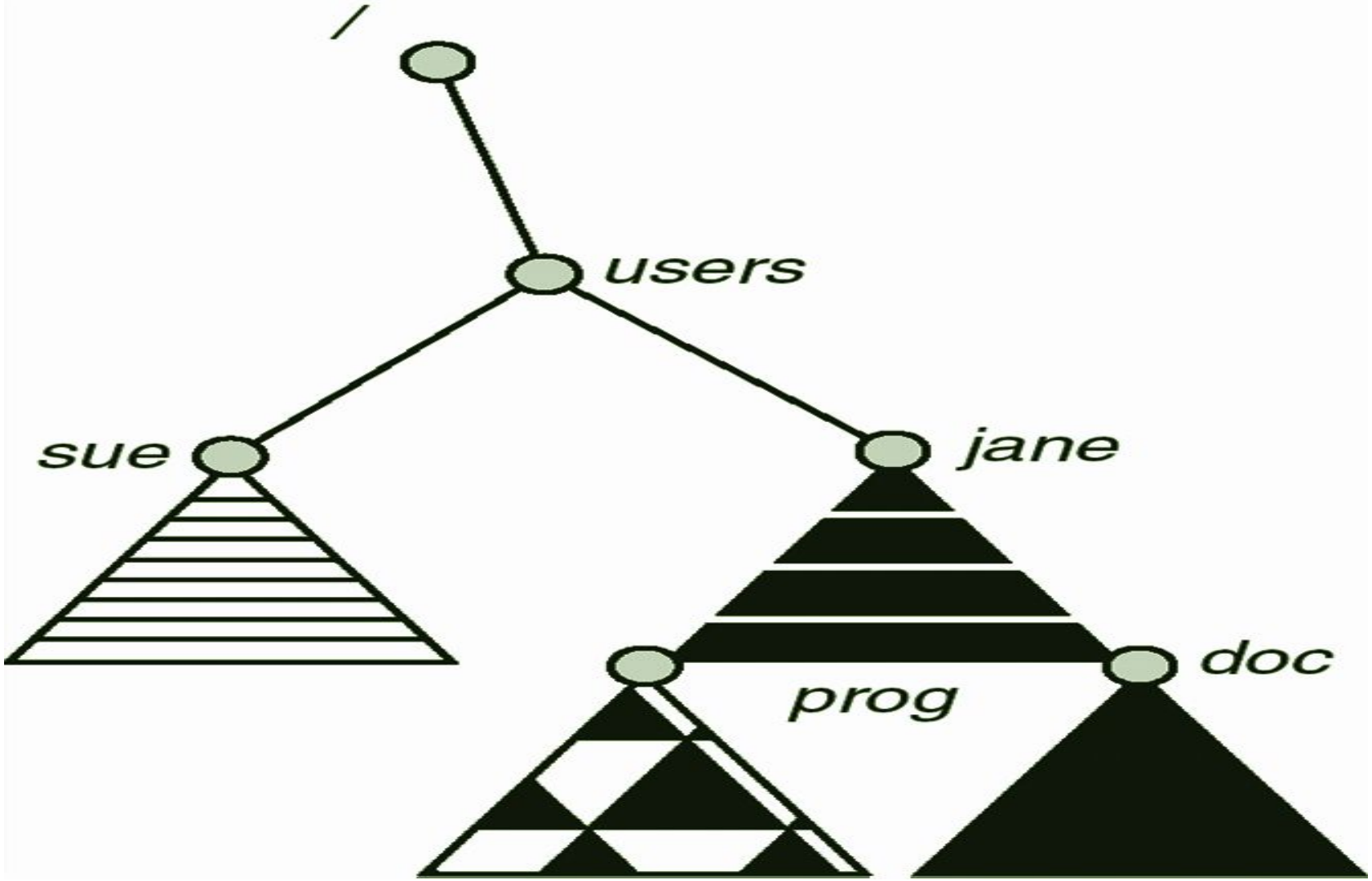
- Systems can be set up so that external storage devices can be mounted automatically upon insertion. This is convenient and is usually satisfactory for home computers. However, it can cause security problems, and thus it is usually not (or, at least, should not be) permitted for networked computers in businesses and other organizations. Rather, such devices must be mounted manually after insertion, and such manual mounting can only be performed by the root account.

File System Mounting

- Mounting can often be performed manually by the root user by merely using the mount command followed by the name of the device to be mounted and its mounting destination (but in some cases it is also necessary to specify the type of filesystem). For example, to mount the eighth partition on the first HDD, which is designated by /dev/hda8, using a directory named /dir8 as the mount point, the following could be used: mount /dev/hda8 /dir8

- Removing the connection between the mounted device and the rest of the filesystem is referred to as unmounting. It is performed by running the umount (with no letter n after the first u) command, likewise followed by the name of the device to be unmounted and its mount point. For example, to unmount the eighth partition from the root filesystem, the following would be used:
  umount /dev/hda8 /dir8

Mount Point



File Sharing

- Sharing of files on multi-user systems is desirable
- Sharing may be done through a **protection** scheme
- On distributed systems, files may be shared across a network
- Network File System (NFS) is a common distributed file-sharing method
- If multi-user system
  - **User IDs** identify users, allowing permissions and protections to be per-user
    Group IDs allow users to be in groups, permitting group access rights
  - Owner of a file / directory
  - Group of a file / directory

File Sharing - Remote File Systems

- Uses networking to allow file system access between systems
  - Manually via programs like FTP
  - Automatically, seamlessly using **distributed file systems**
  - Semi automatically via the **world wide web**
- **Client-server** model allows clients to mount remote file systems from servers
  - Server can serve multiple clients
  - Client and user-on-client identification is insecure or complicated
  - **NFS** is standard UNIX client-server file sharing protocol
  - **CIFS** is standard Windows protocol
  - Standard operating system file calls are translated into remote calls
- Distributed Information Systems **(distributed naming services)** such as LDAP, DNS, NIS, Active Directory implement unified access to information needed for remote computing

File Sharing - Failure Mode

- All file systems have failure modes
  - For example corruption of directory structures or other non-user data, called **metadata**
- Remote file systems add new failure modes, due to network failure, server failure
- Recovery from failure can involve **state information** about status of each remote request
- **Stateless** protocols such as NFS v3 include all information in each request, allowing easy recovery but less security

File Sharing - Consistency Semantics

- Specify how multiple users are to access a shared file simultaneously
  - Similar to process synchronization algorithms
    - Tend to be less complex due to disk I/O and network latency (for remote file systems
  - Andrew File System (AFS) implemented complex remote file sharing semantics
  - Unix file system (UFS) implements:
    - Writes to an open file visible immediately to other users of the same open file
    - Sharing file pointer to allow multiple users to read and write concurrently
  - AFS has session semantics
    - Writes only visible to sessions starting after the file is closed

Protection

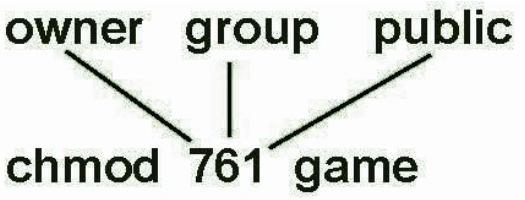
- File owner/creator should be able to control:
  - what can be done
  - by whom
- Types of access
  - **Read**
  - **Write**
  - **Execute**
  - **Append**
  - **Delete**
  - **List**

Access Lists and Groups

- Mode of access: read, write, execute
- Three classes of users on Unix / Linux

	RWX		
a) **owner access**	7	⇒	1 1 1
	RWX		
b) **group access**	6	⇒	1 1 0
	RWX		
c) **public access**	1	⇒	0 0 1

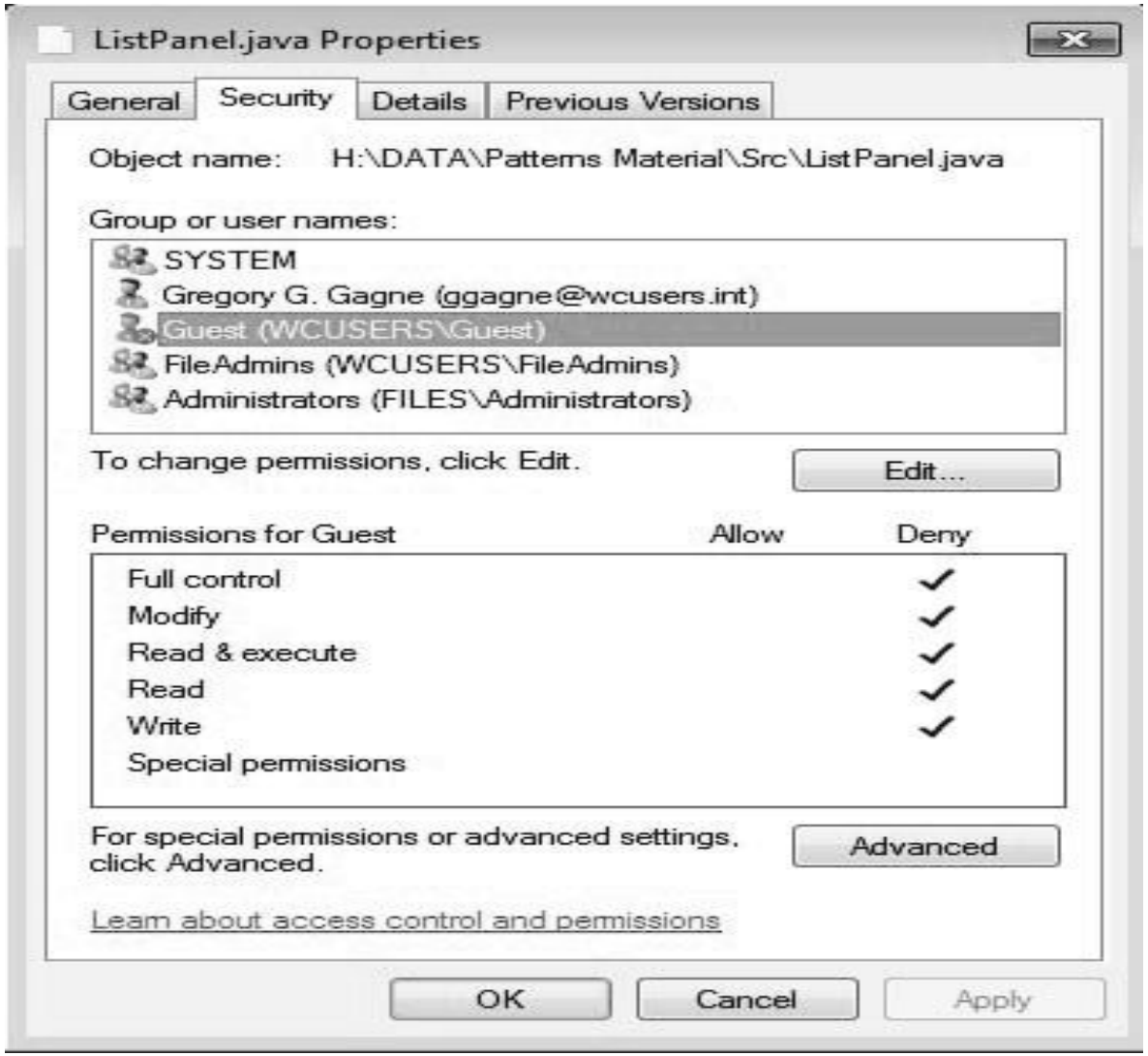
- Ask manager to create a group (unique name), say G, and add some users to the group.
- For a particular file (say *game*) or subdirectory, define an appropriate access.



Attach a group to a file

chgrp G game

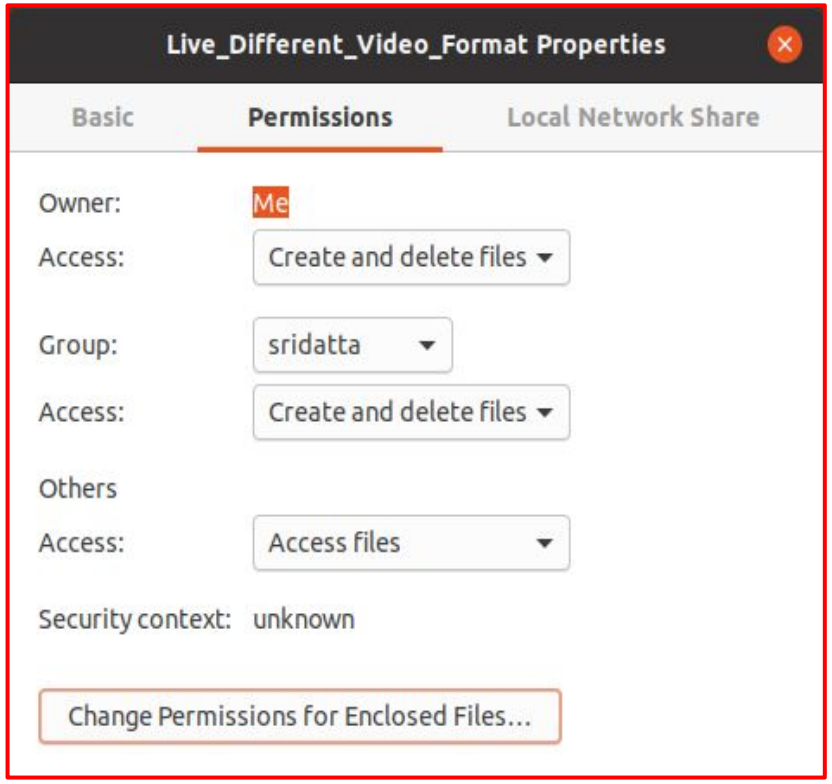
Access Lists and Groups - Windows Example

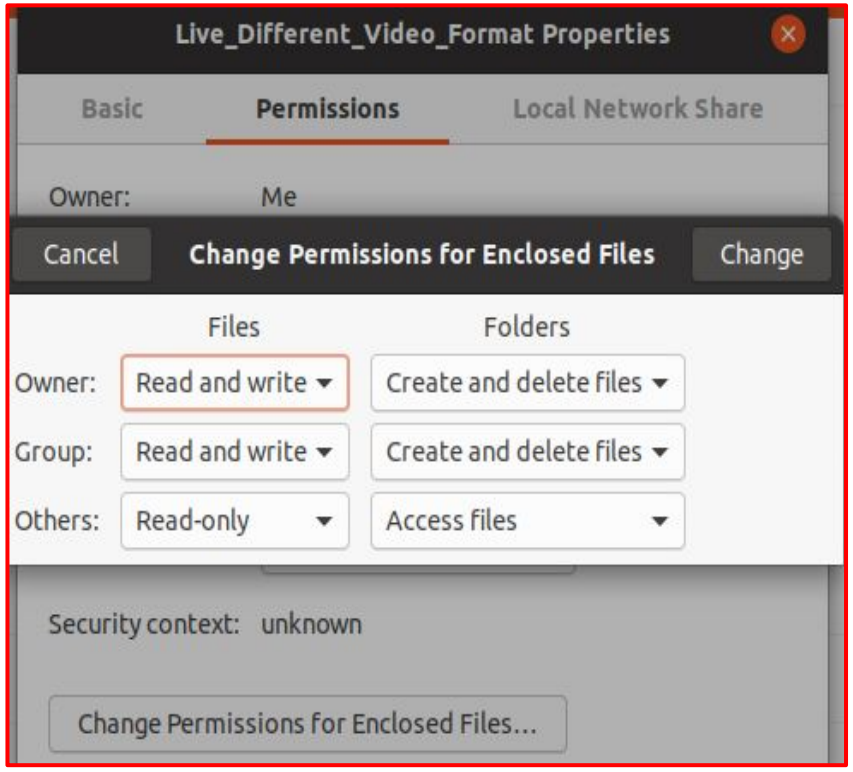


Access Lists and Groups - Linux Example

```
sridatta@sridatta:~$ ls -l
total 155248
drwxr-xr-x 26 sridatta sridatta      4096 Aug 23 19:18  anaconda3
drwxrwxr-x  6 sridatta sridatta      4096 Oct  9 05:25  AQRS_2020_Final_Sep6_2020
drwxrwxr-x  7 sridatta sridatta      4096 Sep 19 19:15  Aspirations_Python
drwxr-xr-x  6 sridatta sridatta      4096 Oct 24 05:35  Desktop
drwxr-xr-x  2 sridatta sridatta      4096 Sep 24 09:50  Documents
drwxr-xr-x  3 sridatta sridatta      4096 Oct 24 05:19  Downloads
drwxrwxr-x  2 sridatta sridatta      4096 Aug 11 08:05  DTP
-rw-rw-r--  1 sridatta sridatta  36920918 Oct 21 20:36 'From the COEs Desk.m4v'
drwxrwxr-x  2 sridatta sridatta      4096 Sep 25 06:28  Live_Different_Video_Format
drwxr-xr-x  2 sridatta sridatta      4096 Aug 11 04:51  Music
drwxrwxr-x  2 sridatta sridatta      4096 Aug 11 08:04  OBS_Output
drwxrwxr-x  4 sridatta sridatta      4096 Oct 18 19:34  PESU_DD_Status
drwxr-xr-x  3 sridatta sridatta      4096 Aug 20 11:08  Pictures
drwxr-xr-x  2 sridatta sridatta      4096 Aug 11 04:51  Public
drwxrwxr-x  8 sridatta sridatta      4096 Aug 11 08:37  rtl8188fu
drwxrwxr-x  9 sridatta sridatta      4096 Sep  6 05:30  rtl8821ce
drwxr-xr-x  9 sridatta sridatta      4096 Sep 24 06:24  snap
drwxrwxr-x  2 sridatta sridatta      4096 Sep 20 06:54  StaticLinkingDemo
drwxr-xr-x  2 sridatta sridatta      4096 Aug 11 04:51  Templates
drwxrwxr-x  2 sridatta sridatta      4096 Sep 19 22:01  thumbnail
-rw-rw-r--  1 sridatta sridatta 121964532 Oct  7 20:42  UE18CS302_071020_RC7_NVP_Compressed.mp4
drwxr-xr-x  3 sridatta sridatta      4096 Sep 23 21:28  Videos
sridatta@sridatta:~$
```

Access Lists and Groups - Linux Example





- File System Mounting
- File Sharing
- Protection

THANK YOU

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com