Chapter 2

File Management

School of Computer Science,
MIT-World Peace University,
Pune, Maharashtra, India.

Amol D. Vibhute (PhD)
Assistant Professor

Email ID:- amol.vibhute@mitwpu.edu.in

Roadmap of Chapter

- Introduction
- File Management,
  - Listing Files, Meta characters, Hidden Files, Creating Files, Editing Files,
- Display Content of a File,
- Counting Words in a File,
- Copying Files,
- Renaming Files,
- Deleting Files
- Standard Unix Streams.

Introduction:

- All **data** in Unix is organized into **files**.
- All **files** are organized into **directories**.
- These **directories** are organized into a tree-like structure called the **filesystem**.
- In Unix, there are three basic types of files –
  - **Ordinary Files** − An ordinary file is a file on the system that contains data, text, or program instructions. We will see ordinary files.
  - **Directories** − Directories store both special and ordinary files. For users familiar with Windows or Mac OS, Unix directories are equivalent to folders.
  - **Special Files** − Some special files provide access to hardware such as hard drives, CD-ROM drives, modems, and Ethernet adapters. Other special files are similar to aliases or shortcuts and enable you to access a single file using different names.

File Management:

- **Listing Files:**
  - To list the files and directories stored in the current directory, we can use following command.
  - $ls.
  - The command ls supports the -l option which would help you to get more information about the listed files –$ls –l.
    - amol.vibhute@Q06S0181 ~
    - $ ls -l
    - total 0
    - drwxr-xr-x+ 1 amol.vibhute Domain Users 0 Feb 24 08:58 amol.txt
    - drwxr-xr-x+ 1 amol.vibhute Domain Users 0 Feb 24 08:59 demo.txt
    - amol.vibhute@Q06S0181 ~
    - $

Cont.…

- Here is the information about all the listed columns –
  - **First Column**: Represents the file type and the permission given on the file.
  - **Second Column**: Represents the number of memory blocks taken by the file or directory.
  - **Third Column**: Represents the owner of the file. This is the Unix user who created this file.
  - **Fourth Column**: Represents the group of the owner. Every Unix user will have an associated group.
  - **Fifth Column**: Represents the file size in bytes.
  - **Sixth Column**: Represents the date and the time when this file was created or modified for the last time.
  - **Seventh Column**: Represents the file or the directory name.

Cont.…

- In the **ls -l** listing example, every file line begins with a **d**, **-,** or **l**. These characters indicate the type of the file that's listed.

Prefix	Description
-	Regular file, such as an ASCII text file, binary executable, or hard link
b	Block special file. Block input/output device file such as a physical hard drive
c	Character special file. Raw input/output device file such as a physical hard drive.
d	Directory file that contains a listing of other files and directories.
l	Symbolic link file. Links on any regular file.
p	Named pipe. A mechanism for inter process communications.
s	Socket used for inter process communication.

Metacharacters:

- Metacharacters have a special meaning in Unix.
  - ***** and **?** are metacharacters.
- We use ***** to match **0** or more characters, a question mark (**?**) matches with a single character.

  For Example −$ls ch*.doc

  Displays all the files, the names of which start with **ch** and end with **.doc** –
- Here, ***** works as meta character which matches with any character. If you want to display all the files ending with just **.doc**, then you can use the following command –$ls *.doc
  - Some other metacharacters are brackets and hypen may be used.
- **Note**: Do not create file names containing metacharacters.

Hidden Files:

- An invisible file is one, the first character of which is the dot or the period character (.). Unix programs (including the shell) use most of these files to store configuration information.
- Some common examples of the hidden files include the files −
  - **.profile** − The Bourne shell (sh) initialization script,
  - **.kshrc** − The Korn shell (ksh) initialization script,
  - **.cshrc** − The C shell (csh) initialization script,
  - **.rhosts** − The remote shell c.onfiguration file,
- To list the invisible files, specify the **-a** option to **ls** –
  - $ ls -a

    . .bash_history .bashrc .profile demo.txt

    .. .bash_profile .inputrc amol.txt
- **Single dot (.)** − This represents the current directory.
- **Double dot (..)** − This represents the parent directory.

Creating Files:

- Used to create ordinary files on any Unix system,
- We can use the vi editor using the **vi** command.
- Syntax: **$ vi filename**
- The above command will open a file with the given filename. Now, press the key **i** to come into the edit mode.
- Once you are in the edit mode, you can start writing your content in the file as in the following program –
  - **Ex**: Hello this is the Unix file. I have created it for showing the demo.
- Once you are done with the program, follow these steps −
  - Press the key **esc** to come out of the edit mode.
  - Press two keys **Shift + Z** together to come out of the file completely.
- You will now have a file created with filename in the current directory.
  - $ vi filename

- **Note**: We can create new file using **touch** command also.

Editing Files:

- We can edit an existing file using the **vi editor**.
  - For instance: $ vi filename
- Once the file is opened, you can come in the edit mode by pressing the key **i** and then you can proceed by editing the file. If you want to move here and there inside a file, then first you need to come out of the edit mode by pressing the key **Esc**. After this, you can use the following keys to move inside a file –
  - **l** key to move to the right side.
  - **h** key to move to the left side.
  - **k** key to move upside in the file.
  - **j** key to move downside in the file.
- So using the above keys, you can position your cursor wherever you want to edit. Once you are positioned, then you can use the **i** key to come in the edit mode. Once you are done with the editing in your file, press **Esc** and finally two keys **Shift + ZZ** together to come out of the file completely.

Display Content of a File:

- We can use the **cat** command to see the content of a file.
- For ex: $ cat filename
  - Hello this is the Unix file. I have created it for showing the demo.

- We can display the line numbers by using the **-b** option along with the **cat** command as follows –

- Ex: $ cat -b amol
  1 This is the Unix file. I created it for first time.
  2 I am going to save this content in it.

Counting Words in a File:

- We can use the **wc** command to get a count of the total number of lines, words, and characters contained in a file.
  - For ex: $ wc filename (amol)
  - 14 30 145 amol
- Here is the detail of all the four columns −
- **First Column**: Represents the total number of lines in the file.
- **Second Column**: Represents the total number of words in the file.
- **Third Column**: Represents the total number of bytes in the file. This is the actual size of the file.
- **Fourth Column**: Represents the file name.
- We can give multiple files and get information about those files at a time.
- Syntax: $ wc filename1 filename2 filename3

Copying Files:

- To make a **copy** of a file use the **cp** command.
  - Syntax: & cp source_file destination_file

- $ cp filename copyfile
- $
- You will now find one more file **copyfile** in your current directory. This file will exactly be the same as the original file **filename**.

Renaming Files:

- To change the name of a file, use the **mv** command.
  - Syntax: $ mv old_file new_file
- The following command will rename the existing file **filename** to **new file**.
- $ mv filename new file

- The **mv** command will move the existing file completely into the new file. In this case, you will find only **new file** in your current directory.

Deleting Files:

- To delete an existing file, use the **rm** command.
  - Syntax: $ rm filename
- This will completely remove or delete the existing file.
- **Caution**: A file may contain useful information. It is always recommended to be careful while using this **Delete** command.
- We can remove multiple files at a time with the command given below –
  - $ rm filename1 filename2 filename3

Standard Unix Streams:

- Under normal circumstances, every Unix program has three streams (files) opened for it when it starts up −
  - **stdin** − This is referred to as the *standard input* and the associated file descriptor is **0**.
    - This is also represented as **STDIN**. The Unix program will read the default input from **STDIN**.
  - **stdout** − This is referred to as the *standard output* and the associated file descriptor is **1**.
    - This is also represented as **STDOUT**. The Unix program will write the default output at **STDOUT.**
  - **stderr** − This is referred to as the *standard error* and the associated file descriptor is **2**.
    - This is also represented as **STDERR**. The Unix program will write all the error messages at **STDERR**.

References:

- Unix Shell Programming: Yashwant Kanitkar, BPB Publications, New Delhi.

Thank You !!!