

UNIT 2.

LINUX Comparisons And The Vi editor



Structure

- 2.0 Introduction
- 2.1 Objectives
- 2.2 DOS- to-LINUX Comparisons
 - 2.2.1 significant differences
 - 2.2.2 DOS commands
 - Attrib, Backup, Msbackup, Restore, Chdir, Cd, Chkdsk, Copy, Xcopy, Date, Time, Del, Erase, Dir, Doskey, Edit, Edlin, Format, Help, Mem, Mkdir, Md, More, Path, Prompt, Ren, Rename
- 2.3 VI Editor quick references
 - 2.3.1 vi editor usage
 - 2.3.2 Simple Character Motion in vi
 - 2.3.3 Inserting Text into the File with i, a, o, and O
 - 2.3.4 Quick Reference for vi editor commands
 - 2.3.5 Quick Reference for ex mode commands
 - 2.3.6 The most common erase characters
- 2.4 Summary
- 2.5 Check Your Progress

2.0 Introduction

This chapter describes the basic differences and similarities between the Linux and the well-known Dos operating system. This is because the students are very much familiar with the previous Dos commands. This would help them to know easily about the Big boss Linux operating system.

For the benefit of the user the king of editors, visual editor vi is illustrated along with the other editors.

2.1 Objectives

At the end of this unit, you would be able to

- Understand the differences and similarities between DOS and LINUX Comparisons
- Know about visual editor vi
- Distinguish among different editors
- Describe the usage of Linux commands

2.2 DOS-to-LINUX Comparisons

This section will provide a translation of DOS commands into LINUX (where LINUX has *much* more to offer than DOS!) Much of DOS is based on LINUX, so some commands act the same in both systems, like *mkdir* to create a directory. Other commands are basically the same but have subtle differences. *cd* (or CD), for

example, changes directories in both DOS and LINUX, but, if you don't specify the directory to change to, DOS and LINUX take you to two different places. Other commands are similar but have different names, like LINUX's *ls* and DOS's *DIR*. Some DOS commands have absolutely no counterparts in LINUX (thank heavens), like the notorious *RECOVER* command, found in early versions of DOS.

Each commonly used DOS command will be compared against a suitable LINUX command with explanations on each.

2.2.1 Significant Differences

The most important difference between LINUX and DOS is-- the *slash!* As opposed to DOS, LINUX uses the *normal* slash instead of the backslash (**), with the normal slash being the forward slash (*/*). LINUX uses the forward slash for its absolute pathnames.

Another significant difference is LINUX's ability to utilize long filenames. One isn't only limited to eight character names with three letter extensions (AUTOEXEC.BAT). A feasible filename in LINUX may look like *a.terrific.filename.possible.only.in.LINUX*. Some of the older UNICES limit filenames to 14 characters, but any of the recent versions allow up to an incredible amount.

One last point while the topic is on filenames, remember that though DOS doesn't care about capitalization, LINUX does. LINUX is known as being *case sensitive*. That is to say, the file *News*, *news*, *NEWS*, and *NeWs* are all different and unique names according to LINUX.

2.2.2 DOS Commands

ATTRIB

The DOS *ATTRIB* command allows you to change the attributes of the files, such as whether they are hidden or read-only.

- *ATTRIB +H*
 - The DOS *ATTRIB +H* command makes a file hidden, so that it doesn't show up on normal directory listings. To do this in LINUX, rename the file with a filename that begins with a period. A file named *.secret*, for example, does not appear on *ls* listings. To ask for a listing of all files, including hidden, you need you give it a special switch with a command line like *ls -a*.
- *ATTRIB -H*
 - The DOS *ATTRIB -H* command, "unhide" files; the thing you do in LINUX is to rename the file so that they don't begin with a period. In LINUX, however, some files require this attribute to function correctly, namely *.login* and *.profile* (your shell executes these every time you login). They won't work otherwise.
- *ATTRIB +R* and *ATTRIB -R*
 - These commands control whether a file is read-only. In LINUX, every file is owned by someone; the owner can control who can read it (*r*), alter or change it (*w*), and, for a program, run it (*x*).

To set permissions on a file that you own so that no one can change them (not even you), type this line:

```
chmod ugo-w filename
```

To set permissions so that only you can change the file, type the following:

```
chmod go-w,u+w filename
```

BACKUP, MSBACKUP, RESTORE

In DOS, the BACKUP and MSBACKUP commands store files on floppies in a backup format. LINUX, however, system administrators take care of these tasks. If you are indeed the one who administers the machine, there are programs such as *tar* (for *tape archive*), *cpio* (for *c*opy *i*n or *o*ut), save and restore LINUX files to and from tapes and floppies.

CHDIR, CD

This command changes directories, or makes another directory the current working directory. DOS's CD command may take an argument of a directory name and move there; if this argument isn't given, then CD won't take you anywhere, but will tell you the current working directory. you the current working directory, or the same directory. In LINUX, however, if an argument of a directory isn't given, *cd* will take one to their home directory.

CHKDSK

This DOS program checks a specified hard disk for logical errors then outputs how much disk space is left (as well as how much is used). LINUX has an equivalence in *df*, or *d*isk *f*ree. There is also a *du*, which stands for disk usage. *du* reports the amount of space used by the files in each directory and subdirectory.

COPY, XCOPY

The LINUX *cp* command works in almost the same way as DOS's COPY. In DOS, however, you don't have to give it an argument of a destination directory-- it will assume the current directory. LINUX requires a destination directory where you may give it a (.) for the current working directory. In LINUX, you may type in: *cp /home/sbl/school/os/linux/sgml/linux.sgml .*

XCOPY

copies directories recursively. LINUX uses a -r switch on it's *cp* command to copy recursively.

DATE, TIME

DOS's DATE and TIME display the date and time, respectively. LINUX's *date* outputs both the date and time.

DEL, ERASE

These DOS commands delete files. The LINUX *rm* command works the same way. Both delete files indefinitely.

DIR

This command lists files in a given directory. It is similar to LINUX's *ls* command, but with different options. To get a listing such as DOS's DIR, issue *ls -l*. This shows the size, date, and owner of every file in the directory.

DOSKEY

DOSKEY is lets you give the same command without retyping it. In LINUX, this is handled by shell, whichever you use. In *BASH*, press the up-arrow key or Ctrl-P and then press Enter.

- If you are using the C shell (csh), type *!!*.
- If you use the Korn shell (ksh), type *r*.
- If you use the Bourne shell (sh), sorry!

EDIT, EDLIN

EDLIN is a line editor for DOS, which came on earlier versions. It was eventually replaced by EDIT. To edit files in LINUX, you can use an editor such as *ed*, *vi*, *emacs*, or *pico*.

FIND DOS's FIND command doesn't compare to LINUX's *find*. Have a nice day.

FORMAT

This DOS command prepares a floppy disk for use. Many LINUX systems don't even use floppy disks, but there are programs available. Linux has a script called *fdformat*.

HELP

DOS's HELP command gives help on a specific command. LINUX utilizes "man" pages. To get help on a specified command, you may issue the command:

man some_program Ex: \$ **man man**

LINUX shells normally have a *help* command built in. This gives information on shell-specifics.

MEM

MEM tells you how much of the machine's memory resources are used. LINUX has a command called *free*.

MKDIR, MD

This command creates a new directory, just like LINUX's command.

MORE

This filter is just like LINUX's filter.

PATH

This tells DOS what directories to look in to for command names and programs. In LINUX, you use a shell variable to set this.

PROMPT

This changes your normal DOS prompt. LINUX, again, uses a shell environment variable to set this. The environment variable varies depending on the shell being used by the user.

REN, RENAME

This renames DOS filenames. To rename files in LINUX, you can use the *mv* command.

2.3 VI Editor quick reference

There are a number of editors that may be included with your LINUX system, including *ed*, *ex*, *vi*, and *EMACS*. The latter two use the entire screen, a big advantage, and both are powerful editors.

The *vi* editor is often looked at as archaic and difficult to learn, but it is in fact, rather simple if you know the main commands to use. This section will teach you nearly everything you will ever need to know about the *vi* editor.

2.3.1 vi editor usage

```
\home\user$ vi filename
```

If the file does not exist, it will be created for you.

There are three modes of *vi* that you need to understand.

- command mode - used to give commands to the editor.

- input mode - used to input text.

- ex mode - denoted by typing a *:* from command mode.

When you execute *vi*, you start in command mode and must tell *vi* what you want to do before inputting text.

2.3.2 Simple Character Motion in vi

Getting to a file isn't much good if you can't actually move around in it. Now you will learn how to use the cursor control keys in *vi*. To move left one character, press the *h* key. To move up, press the *k* key. To move down, press the *j* key. To move right a single character, use the *l* key. You can move left one character by pressing the Backspace key, and you can move to the beginning of the next line with the Enter key.

2.3.3 Inserting Text into the File with i, a, o, and O

Being able to move around in a file is useful. The real function of an editor, however, is to enable you to easily add and remove—in editor parlance, insert and delete—information. The *vi* editor has a special insert mode, which you must use in order to add to the contents of the file. There are four different ways to shift into insert mode, and you will learn about all of them in this section.

The first way to switch to insert mode is to enter the letter `i`, which, mnemonically enough, inserts text into the file. The other commands that accomplish more or less the same thing are `a`, which appends text to the file; `o`, which opens up a line below the current line; and `O`, which opens up a line above the current line.

2.3.4 Quick Reference for vi editor commands

Any command that ends in `<esc>` is a command that will put you into a specific mode until `<esc>` is pressed to back you out to command mode. If at any time you forget what mode you are in, or you get lost by typing the wrong character, you can press `<esc>` twice to ensure that you are in command mode.

<code>i</code>	-	insert before cursor <code><esc></code>
<code>I</code>	-	insert at beginning of line <code><esc></code>
<code>a</code>	-	append (add) text after cursor <code><esc></code>
<code>A</code>	-	append text at end of line <code><esc></code>
<code>x</code>	-	delete one character at the current cursor position
<code>r</code>	-	replace a single character at the cursor
<code>R</code>	-	replace characters until <code><esc></code> is pressed
<code>#d</code>	-	delete # of lines
<code>dd</code>	-	delete current line
<code>D</code>	-	delete remainder of line after cursor

2.3.5 Quick Reference for ex mode commands

<code>:w</code>	-	write the current buffer to the file
<code>:w filename</code>	-	write the buffer to filename
<code>:wq</code>	-	write and quit the editor
<code>:q</code>	-	quit the editor
<code>:q!</code>	-	force quit if <code>:q</code> doesn't work

Table 2.1: Useful more Commands

Command	Description
---------	-------------

<code>[SPACE]</code>	Display next page.
<code>[RETURN]</code>	Display next line.
<code>Nf</code>	Move forward <i>n</i> pages.
<code>B</code>	Move backward one page.
<code>Nb</code>	Move backward <i>n</i> pages.
<code>/word</code>	Search forward for <i>word</i> .
<code>?word</code>	Search backward for <i>word</i> .
<code>V</code>	Start the vi editor at this point.
<code>[CTRL-L]</code>	Redisplay current page.
<code>H</code>	Help.
<code>:n</code>	Go to next file on command line.
<code>:p</code>	Go back to previous file on command line.
<code>Q</code>	Quit more (before end of file).

2.3.6 The most common erase characters

The most common erase characters are

- [BACKSPACE]
- [DELETE], [DEL], or [RUBOUT] key
- [CTRL-H]

[CTRL-H] is called a control character. To type a control character (for example, [CTRL-H], hold down the [CTRL] key while pressing the letter "h". (This is like the way you make an uppercase letter: hold the [SHIFT] key while pressing a letter key.) In the text, we will write control characters as [CTRL-H], but in the examples, we will use the standard notation: ^H. This is not the same as pressing the ^ (caret) key, letting go, and then typing an H!

The key labeled [DEL] may be used as the interrupt character instead of the erase character. (It's labeled [DELETE] or [RUBOUT] on some terminals.) This key is used to interrupt or cancel a command, and can be used in many (but not all) cases when you want to quit what you're doing. Another character often programmed to do the same thing is [CTRL-C].

Some other common control characters are: [CTRL-U]

Erases the whole input line; you can start over.

[CTRL-S]

Pauses output from a program that is writing to the screen.

[CTRL-Q]

Restarts output after a pause by [CTRL-S].

[CTRL-D]

2.4 Summary

In LINUX, the screen editor of choice is called vi.

In some ways, an editor is like another operating system living within LINUX. If you're used to Windows or Macintosh editors, you'll be unhappy to find that vi doesn't know anything about your mouse. Once you spend some time working with vi, however, it will grow on you.

The first is that vi is a modal editor. A mode is like an environment. Different modes in vi interpret the same key differently. For example, if you're in insert mode, pressing the A key adds an a to the text, whereas in command mode, pressing the A key enters a, a single key abbreviation for the append command. If you ever get confused about what mode you're in, press the Esc key on your keyboard. Pressing Esc always returns you to the command mode (and if you're already in command mode, it beeps to remind you of that fact).

- EMACS is a modeless editor. In EMACS, the A key always adds the letter a to the file. Commands in EMACS are all indicated by holding down the Ctrl key while pressing the command key; for example, Ctrl+C deletes a character.

The vi command by itself starts the editor, ready for you to create a new file. The vi command with a filename starts vi with the specified file, so you can modify that file immediately.

2.5 Check Your Progress

I. Choose the correct answer

1. Linux uses _____ slash.
 - a. Normal
 - b. Back
 - c. Double
 - d. None of the above
2. The following are the Linux commands _____
 - a. ATTRIB
 - b. BACKUP
 - c. RESTORE
 - d. None of the above
3. The equivalent for COPY in Linux is _____
 - a. mv & rm
 - b. cp
 - c. both a & b
 - d. none of the above.

II. Say True or False

1. In Linux operating system filenames are limited to 8 characters. True/False
2. Linux operating system is case sensitive. True/False

III. Essay Type Questions:

1. list out the comparisons between DOS-to-LINUX.
2. Explain different modes of vi-editor.

IV. Further readings and other activities

1. Get more information using help commands about Vi, ed, red, gEdit, vim, Emacs, ex, sed, awk, mtools
2. Install from Linux source cd, The dos emulation package by name DOSEMU and work on all dos emulation commands.
3. For further readings you can refer the following books
 1. Title: *Power Tools*, second edition,
Author: Jerry Peek, Tim O'Reilly, Mike Loukides, and others; O'Reilly & Associates; ISBN 1-56592-260-3; 1997. A huge collection of tips, techniques, and concepts for making intermediate users into advanced users.
 2. Title: *Learning the vi Editor*
Author: Linda Lamb; O'Reilly & Associates; ISBN 0-937175-67-6; 1992. A complete introduction to **vi**, structured like *LINUX in a Nutshell*.

Reference e-mails: raomvp@yahoo.com roopasindhe@lycos.com
URL / Web Site: <http://www.raomvp.bravepages.com>

Good-Luck