



LINUX PROGRAMMING

Prepared and Presented by
Mr. T.P.SHEKAR
Assoc. Professor
Computer Science and Engineering

Presentation overview

- Text processing utilities-cat ,tail ,head
- Sort , nl, grep ,egrep ,fgrep
- cut, paste, join, tee
- Comm, cmp ,diff
- Backup utilities-tar, cpio
- Awk, tr, pg

Text processing utilities

- **cat:** cat is used to create the files.
- `$cat> filename`

Type some text here

Press ctrl+d

\$

Cat can also be used to display the contents
Of a file.

`$cat filename`

- Cat can also concatenate the contents of 2 files and store them in third file.
- `Cat>file1 file2>new file`
- To append the contents of two files into another file use
- `Cat>file1 file2>>new file`

- **tail:**

tail command displays the end of the file. It displays the last ten lines by default.

```
$tail file
```

To display last 3 lines use

```
$tail -n 3 file    or
```

```
$tail -3 file
```

We can also address the lines from the beginning of the file instead of the end.

The + count allows to do that.

- **head:**
- head command as the name implies, displays the top of the file. When used without an option, it displays the first 10 lines of the file.
- `$head file`
- We can use `-n` option to specify a line count and display, say first 3 lines of the file.
- `$head -n 3 file` or
- `$head -3 file`

- **Sort:**
- Sort can be used for sorting the contents of a file.
\$sort shortlist
- Sorting starts with the first character of each line and proceeds to the next character only when the characters in two lines are identical.
- **Sort options:**
- With -t option sorts a file based on the fields.


```
$sort -t "|" +2 shortlist
```

The sort order can be reversed with `-r` option.

Sorting on secondary key:

U can sort on more than one field i.e. u can provide a secondary key to sort.

If the primary key is the third field and the secondary key the second field, we can use

```
$sort -t \ | +2 -3 +1 shortlist
```


- Numeric sort (-n):
- To sort on number field use sort with -n option.
- `$sort -t: +2 -3 -n group1`
- Removing duplicate lines (-u):
- The -u option u purge duplicate lines from a file.

- **nl:**
- nl is used for numbering lines of a file.
- Nl numbers only logical lines –those containing something other apart from the new line character.
- `$nl file`
- nl uses a tab as a default delimiter, but we can change it with `-s` option.
- `$nl -s: file`
- nl won't number a line if it contains nothing.

- **Grep:** globally search for a regular expression and print.
- Grep scans a file for the occurrence of a pattern and depending on the options used, displays
- Lines containing the selected pattern.
- Lines not containing the selected pattern (-v).
- Line numbers where pattern occurs (-n)
- No. of lines containing the pattern (-c)

- File names where pattern occurs (-l)

Syntax:

grep option pattern filename(s)

Egrep: extended grep

Egrep extended set includes 2 special characters + and ?.

+ --matches one or more occurrences of the pervious character.

?-- matches zero or more occurrences of the pervious character.

- **fgrep**: fast grep
- If search criteria requires only sequence expressions, fgrep is the best utility.
- Fgrep supports only string patterns, no regular expressions.
- To extract all the lines that contain an apostrophe use fgrep as follows:
- `$fgrep "" file`

- **Cut:** slitting the file vertically

U can slice a file vertically with cut command.

- Cutting columns(-c):

Cut with -c option cuts the columns.

To extract first 4 columns of the group file :

```
$cut -c 1-4 group1
```

The specification -c 1-4 cuts columns 1 to 4.

Cutting fields:

To cut 1st and 3rd fields use

```
$cut -d: -f1,3 group1
```


- **Paste:** pasting files
- What u cut with the cut can be pasted back with paste command-but vertically rather than horizontally. u can view two files side by side by pasting them.
- To join two files calc.lst and result.lst use
- `$paste -d= calc.lst result.lst`

- **Join:**
- is a command in Unix-like operating systems that merges the lines of two sorted text files based on the presence of a common field.
- The join command takes as input two text files and a number of options. If no command-line argument is given, this command looks for a pair of lines from the two files having the same first field (a sequence of characters that are different from space), and outputs a line composed of the first field followed by the rest of the two lines.
- `$join file1 file2`

tee:

Unix tee command breaks up its input into two components; one component is saved in a file, and other is connected to the standard output.

tee doesn't perform any filtering action on its input; it gives exactly what it takes.

tee can be placed any where in a pipeline.

u can use tee to save the output of the who command in a file and display it as well:

\$who |tee user.lst

- The tee command reads standard input, writes its content to standard output and simultaneously copies it into the specified file or files.
- **Flags**
- **-a** Adds the output to the end of File instead of writing over it.
- **-i** Ignores interrupts.

- **Comm:**

- Suppose if u have 2 list of people, u are asked to find out the names available in one and not the other or even those common to both. Comm is the command that u need to for this work.
- It requires two sorted file and lists the differing entries in different columns.
`$comm file1 file2`
- Comm display a three-column output.

- **Cmp:** comparing two files
- The two files are compared byte by byte and the location of the first mismatch is echoed to the screen using cmp.
- Cmp when invoked without options it does not bother about possible subsequent mismatches.

`$cmp group1 group2`

If two files are identical cmp displays no message, but simply returns the \$ prompt.

- **diff:** converting one file to another
- diff takes different approach to displaying the differences.
- It tells u which lines in one file have to be changed to make two files identical.
- When used with the same files it produces a detailed output.
- `$diff group[12]`
- diff uses certain special symbols with its instructions to indicate the changes needed to make two files identical.

Backup utilities

- **tar**: the tape archive program
- Tar doesn't normally write to the standard output but creates an archive in the media.
- Tar accepts file and directory names as arguments.
- It operates recursively.
- It can create several versions of same file in a single archive.
- It can append to an archive without overwriting the entire archive.

- -c option is used to copy files to backup device.
- `$tar -cvf /dev/rdisk/foq18dt /home/sales/sql/*.sql`
- The verbose option (-v) shows the no. of blocks used by each file.
- Files are restored with the -x (extract) key option.
when no file or directory name is specified it restores all files from the backup device.

- **Cpio:** copy input-output
- Cpio command copies files to and from a backup device. It uses standard input to take the list of filenames.
- It then copies them with their contents and headers into stream which can be redirected to a file or a device.
- Cpio can be used with redirection and piping.
- Cpio uses two options -o (output) and -i (input) either of which must be there in the command line.

- **Awk: Aho, Weinberger and Kernighan**
- Awk is not just a command, but a programming language too.
- Syntax:
awk options 'selection criteria {action}' file(s)
- Simple filtering
awk '/Simpsons/ { print }' homer |Simpsons
- Splitting a line into fields
awk -F "|" '/Simpsons/ {print \$1}' homer

tr: translating characters

- **tr** command manipulates individual characters in a character stream.

`tr options expr1 expr2 < standard input`

It takes input only from the standard input, it does not take input a file name as its argument.

- When executed, the program reads from the standard input and writes to the standard output. It takes as parameters two sets of characters, and replaces occurrences of the characters in the first set with the corresponding elements from the other set.

- **Examples:**

- `$ tr "[a-z]" "z[a-y]" < computer.txt`
- `$tr -d '/' <shortlist | head -3`
- `$tr '|' '\012' <shortlist | head -6`
- `$tr '/' '~-' < shortlist |head -3`
- **pg:**
- pg is used to display the output of a file in page by page.
- `$pg file1`