SOFTENGER
Strive to Excel

Handling Ordinary Files

Unix Fundamentals

Handling Ordinary Files

wc : Counting lines words and characters

- wc infile
  - 3 20 103 infile
    - wc counts 3 lines , 20 words and 103 characters
      - A line is a group of characters not containing a newline
      - A word is a group of characters not containing a space , tab or newline
      - A character is the smallest unit of information , and includes a space , tab and newline.

Handling Ordinary Files

wc : Counting lines words and characters

- wc –l infile
  - 3 infile Number of lines
- wc –w infile
  - 20 infile Number of words
- wc –c infile
  - 103 infile Number of characters

Handling Ordinary Files

wc : Counting lines words and characters

- wc chap01 chap02 chap03
    - 305	4058	23179	chap01
    - 550	4732	28132	chap02
    - 377	4500	25221	chap03
    - 1232	13290	76532	total

- When used with multiple filenames , wc produces a line for each file as well as a total count .

Handling Ordinary Files

od : Displaying data in octal

- cat infile

```
I am pressing tab
ctrl-G ^G
ctrl-L ^L

bye
```

A tab has been pressed after the word tab

SOFTENGER
Strive to Excel

od : Displaying data in octal

- **od -b infile**

0	111	40	141	155	40	160	162	145	163	163	151	156	147	40	164	141
20	142	11	12	143	164	162	154	55	107	40	136	107	40	12	143	164
40	162	154	55	114	40	136	114	12	12	142	171	145	12			
55																

Handling Ordinary Files

od : Displaying data in octal

- **od –bc infile**

0	111	40	141	155	40	160	162	145	163	163	151	156	147	40	164	141
	I		a	m		p	r	e	s	s	i	n	g		t	a
20	142	11	12	143	164	162	154	55	107	40	136	107	40	12	143	164
	b	\t	\n	c	t	r	l	-	G		^	G		\n	c	t
40	162	154	55	114	40	136	114	12	12	142	171	145	12			
	r	l	-	L		^	L	\n	\n	b	y	e	\n			
55																

Handling Ordinary Files

- cmp : comparing two files

  cmp command needs a two filenames as an argument.

```
$ cmp chap01 chap02
  chap01 chap02 differ : char 9 , line1
```

- Two files are compared byte by byte and the first mismatch is echoed on the screen.
- By default it doesn't bother about the possible subsequent mismatches.

Handling Ordinary Files

- The –l option gives detail list of the byte number and the different bytes in octal for each character that differ in both files :

- cmp –l note1 note2

```
3      143      145         third character has ascii values 143 , 145
6      170      167
7      171      170
8      172      171
```

Handling Ordinary Files

- comm : What is common ?

- It need’s the two sorted files .
- Displays output in three columnar format :
  - ◈First column contains lines unique to first file
  - ◈Second column contains lines unique to second file
  - ◈Third column contains lines common to both

```
$ comm sort[12]
```

Handling Ordinary Files

- diff - differences in files

- The diff command compares two files, directories, etc, and reports all differences between the two. It deals only with ASCII files. It’s output format is designed to report the changes necessary to convert the first file into the second.

- Syntax

  diff [options] file1 file2

Handling Ordinary Files

- diff - differences in files

- For the mon.logins and tues.logins files above, the difference between them is given by:
  - % diff mon.logins tues.logins
  - 2d1
  - < bsmith
  - 4a4
  - > jdoe
  - 7c7
  - < mschmidt
  - ---
  - > proy

Note that the output lists the differences as well as in which file the difference exists. Lines in the first file are preceded by "< ", and those in the second file are preceded by "> ".

Handling Ordinary Files

- *date* command is used to display current date and time

  Formatting output :

  - $ date +%m
  - $ date +%h
  - $ date +"%h %m"

Handling Ordinary Files

- *touch* - change file timestamps
- Update the access and modification times of each FILE to the current time.
- Common options :

  -a change only the access time

  -m change only the modification time

Handling Ordinary Files

file : Knowing the File types

- file command is used to determine the type of file

  - file correctly identifies the basic file types (regular , directory or device)
  - file *

Cw.exe	dos executable (EXE)
archive.tar.gz	gzip compressed data
createuser.sh	commands text
fork.c	c program text
chaps.pdf	Adobe Portable Format

Handling Ordinary Files

find : Locating files

- **Syntax :**

  find *path_list* *selection_criteria* *action*

- How find operates ?
  - Recursively examines the pathe_list.
  - Matches for one or more files for selection criteria.
  - Takes action on selected files.
- Ex :
  - find /home -name a.out -print

    /home/sagar/scripts/a.out

    /home/sandeep/reports/a.out

Handling Ordinary Files

find : more examples

- find . –name "*.c" -print *all file with extension .c*
- find . -name '[A-Z]' -print*Single quotes also works*
- find . -inum 13975 -print *files with inode number 13975*
- find . -type d -print 2 > /dev/null *show all directories , redirect error to /dev/null*
- find $HOME -perm 777 -print *all file with permission 777*
- find . -mtime -2 -print *file that have modified in less than two days*
- find . -atime +365 -print *files not accessed for more than a year*

Handling Ordinary Files

find : operators (! , -o , -a)

- find . ! –name “*.c” -print *all file without extension .c*

- find /home \(-name “*.sh” -o -name “*.c” \) -print
  all file with extension .sh or .c

Handling Ordinary Files

find : options used by action components

- find . –name "*.c" -ls *show long listing (ls – lids) for all file with extension .c*

- find /home -type f -atime +365 -exec rm {} \;
  remove all ordinary file with that are not accessed for more than 365 days.

- find /home -type f -atime +365 -ok rm {} \;
  remove all ordinary file with that are not accessed for more than 365 days. Confirm before removing

Filters

pr :

Converts text file for printing.

- Ex : pr dept.lst
- pr –k prints output in k columns.
- Other options :

-d	Doublespaces input
-n	Number lines
-o	Offsets lines by n spaces. (increase left margin)

- Ex : pr -t –n –d –o 10 dept.lst

Filters

head :

Displaying the beginning of a file.

- Ex : head dept.lst *shows first ten lines of the file*
- Ex : head -3 dept.lst *shows first three lines of the file*

Filters

tail :

Displaying the end of a file.

- Ex : tail -3 dept.lst *shows last three lines of the file*
- Ex : tail +11 dept.lst *11th line onwards till the end of file.*
- Ex : tail -f dept.lst *monitor file for growth.*

Filters

cut :

Slitting a file vertically.

- Ex : cut -c 6-22,24-32,55- dept.lst *cut columns from 6 to 22 , 24 to 32 and 55 to end of file .*

- Ex : cut -d “|” -f 2,3 dept.lst *cut fields 2 and 3 separated by the delimiter “| ”.*

Filters

paste :

Merge lines of file.

- Ex: cut -d "|" -f 2,3 dept.lst>cutlist1 *cut fields 2 and 3 separated by the delimiter "|".*

  cut -d "|" -f 7,8 dept.lst>cutlist2 *cut fields 7 and 7 separated by the delimiter "|".*

  paste cutlist1 cutlist2

whatever was cut in cutlist1 & cutlist2 is viewed by pasting vertically rather than horizontally.

Handling Ordinary Files

- uniq – Locate repeated and non repeated lines

- Options :

-u	Selects Non-Repeated lines.
-d	Selects Duplicate lines.
-c	Counts Frequency of occurrence.

- Note :

1. uniq requires a sorted file as input.

2. If uniq is provided with two filenames as arguments , it reads the first file and write its output in second file.

Handling Ordinary Files

SOFTENGER
Strive to Excel

- Examples for uniq:

```
$ cat dept.lst
01|accounts|6213
01|accounts|6213
02|admin|5423
03|marketing|6521
03|marketing|6521
04|personnel|2365
05|production|9876
06|sales|1006
```

```
$ uniq dept.lst
01|accounts|6213
02|admin|5423
03|marketing|6521
04|personnel|2365
05|production|9876
06|sales|1006
```

Handling Ordinary Files

- Examples for uniq:

```
$ uniq -u dept.lst
02|admin|5423
04|personnel|2365
05|production|9876
06|sales|1006
```

```
$  uniq -d dept.lst
01|accounts|6213
03|marketing|6521
```

```
$ uniq -c dept.1st
2 01|accounts|6213
1 02|admin|5423
2 03|marketing|6521
1 04|personnel|2365
1 05|production|9876
1 06|sales|1006
```

Handling Ordinary Files

- uniq – Locate repeated and non repeated lines

- Options :

-u	Selects Non-Repeated lines.
-d	Selects Duplicate lines.
-c	Counts Frequency of occurrence.

- Note :

1. uniq requires a sorted file as input.

2. If uniq is provided with two filenames as arguments , it reads the first file and write its output in second file.

Filters

sort :

Sort lines of text files.

- Ex: sort dept.lst *sort list in ascending order.*
- Ex: sort -t "|" –k 2 dept.lst *sort list in ascending order on second filed.*
- Ex: sort -t "|" -r –k 2 dept.lst *sort list in descending order on second filed.*
- Other Options :

-c	:	to check if file is sorted.
-u	:	Removes repeated lines.

Filters

tr :

Translating Characters.

- Ex: tr ‘|/’ ‘~-’ < emp.lst *replace | with ~ and / with -*
- Ex: tr -d ‘|/’ < emp.lst *delete all occurrences of | and /*
- Ex: tr -s ‘ ’ < emp.lst *squeeze all space characters*
- Ex: tr -cd ‘|/’ < emp.lst *delete all characters except | and /*
- Ex: tr ‘|’ ‘\012’ < emp.lst *replace | with \n character*
  \n can be used instead of \012
  (\012 is an octal value for \n)

Filters

SOFTENGER
Strive to Excel

Quiz ?

Display the frequency of each word in the document emp.lst in descending order of frequency ?

Filters

SOFTENGER
Strive to Excel

Answer :

1. Convert all space and tabs to new line

- tr “\040\011” “\012\012” < emp.lst

Filters

Answer : *(.. continued)*

2. Delete all non-alphabetic characters except new line character ‘\n’.

```
tr   “\040\011”        “\012\012”   <  emp.lst    |

tr   -cd        “[a-zA-Z\012]”
```

Filters

Answer : *(.. continued)*

3. Sort the output and pipe it to uniq –c

```
tr     "\040\011"          "\012\012"    <  emp.lst     |

tr     -cd        "[a-zA-Z\012]"   |

sort              |          uniq      -c
```

Filters

Answer : *(.. continued)*

4. Sort the output in reverse numeric sequence

```
tr    “\040\011”        “\012\012”   <  emp.lst     |

tr    -cd        “[a-zA-Z\012]”                     |

sort             |          uniq      -c            |

sort             -nr
```