Unix **COMMANDS** in a nutshell (useful commands for the Gian course)

Contents

Introduction....2

Short tutorial on Unix input/output (and pipe) 2

man: online manual....4

ls: list shows the files....4

cd: change directory....4

pwd: print working directory. 4

mkdir: Make directory....4

cat: display the content of a file (or concatenate more than one file) 4

head: shows the head of a text file 4

tail: shows the tail of a text file 5

more Displays the content of a text file page by page....5

wc: displays counts of a text file (chars / words / lines) 5

cp: copy a file to another....5

mv (move): used to rename a file or to move it across directories....5

rm: deletes file(s) 5

rmdir: deletes a directory, must be empty 6

diff : shows the difference between two files....6

grep: look for a string (or a pattern) in a text file 6

cut : takes a given column out of a text file 6

paste: “paste” two text files....6

sort: sort a text file according to alphabetical order....7

uniq: delete adjacent duplicates....7

echo: displays something on the screen 7

tar: manage archive (extract or create) 8

chmod: change permissions....8

Introduction

Each Unix command can use parameters and options

Usage: command [-o] [parameters]

You can add several options separating each of these by a space (e.g. wc -w –l) you can have more than one parameter separating each of these by a space (e.g: wc -l myText.en myText.hi)

e.g. command “cat” to display the content of a text file

```
cat textFile
```

e.g.

```
cat Icon2016/labs/WelcomePleaseRead.txt
```

```
This virtual machine contains open source software installed by

…

tercom: http://www.cs.umd.edu/~snover/tercom/
```

You can use one of its options:

e.g. cat **-n** Icon2016/labs/WelcomePleaseRead.txt

cat **-n** Icon2016/labs/WelcomePleaseRead.txt

```
     1      This virtual machine contains open source software ins
     2      If you intend to use it elsewhere please install the
…
```

Short tutorial on Unix input/output (and pipe)

Any command followed by:

> *filename* will redirect its output to a file (instead of displaying on screen)

>> *filename* will append its output at the end of the file

< *filename* will get its input from the file (instead of keyboard).

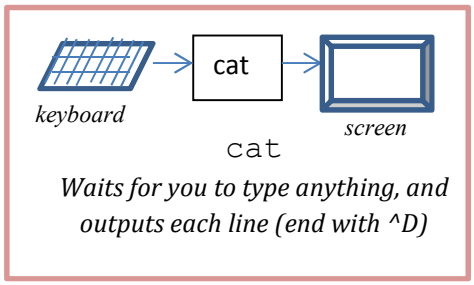
| pipe its output. The output or a command is fed as input to the other

& at the end of command will run command in background.

; will separate commands on same line.

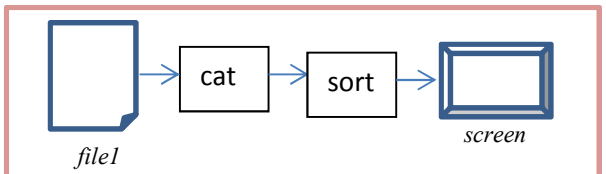
Usually a Unix command gets its input from keyboard and displays its output on screen:

But you can redirect the input from a file or redirect the output to a file, or redirect the output of a command to the input of another (pipe)

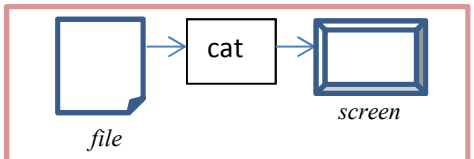


`cat`

Waits for you to type anything, and outputs each line (end with ^D)

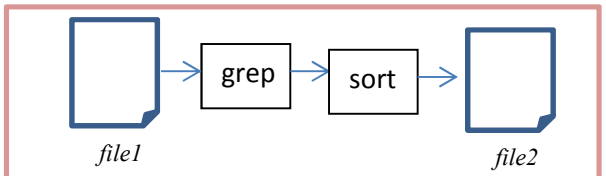


`cat < file1 | sort`



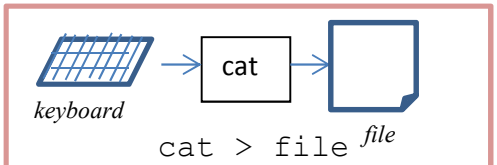
`cat < file`

or `cat file`



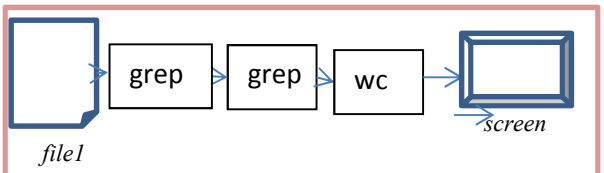
`grep string file1 | sort > file2`

Alphabetical list of lines containing ***string***



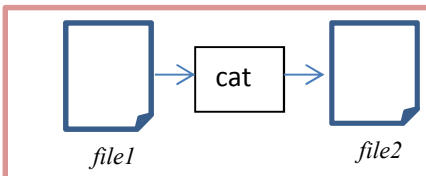
`cat > file`

Creates a new file, type anything, end with ^D and the file will be created



`grep string1 file1 | grep string2 | wc`

Count the number of lines in file1 that contain both string1 and string2



`cat < file1 > file2`

Copy file1 into file2 (see command cp)

List of useful Unix commands

man: online manual

Man is the "online" manual, when you want to see all the options of a given command.

- e.g. man sort: shows a manual of the sort command and explains the option

ls: list shows the files

"ls" alone: list the files that are in the current directory

"`ls directory`": to list the content of another directory

e.g.:

```
ls /home/Icon2016
```

```
labs
```

Options

- ls -l list all files, with permissions (rwx), size, date

cd: change directory

```
cd Icon2016/labs
```

will change directory from current to Icon2016/labs.

Remember to check your current directory with pwd

Cd without any parameter returns you to your home directory (/home/smt)

pwd: print working directory.

pwd will show the place where you are currently in the Unix directories.

```
pwd
```

```
/home/smt
```

```
cd Icon2016
pwd
```

```
/home/smt/Icon2016
```

mkdir: Make directory

```
mkdir tmp
```

creates a directory (in your current dir), you courld specify e.g.

```
mkdir /tmp/myNewDir
```

cat: display the content of a file (or concatenate more than one file)

```
cat /home/smt/Desktop/Welcom*
```

to display the file beginning by "Welcom" on the descktop.

head: shows the head of a text file

```
head -n textFile
```

Display the first N lines from a text file.

tail: shows the tail of a text file

```
tail -n textFile
```

displays the last n lines of a text file

Note that, using the unix “pipe”, you can combine the tail and head:

e.g. display only the 20th line:

```
head -20 textFile | tail -1
```

more Displays the content of a text file page by page

```
more textFile
```

wc: displays counts of a text file (chars / words / lines)

Options

- wc -l filename will print total number of lines in a file.
- wc -w filename will print total number of words in a file **(n. of words may not be accurate for indic languages! It is useful mainly for English)**
- wc -c filename will print total number of characters in a file **(due to utf8 encoding, n. of chars may not be accurate for other languages than English)**

You can combine options: wc -lw parallel-corpora/ur-en/training.en will display the number of English words in the training text, and the number of characters

cp: copy a file to another

e.g. save a file

```
cp file file.new
```

copy the file to another directory

```
cp file otherDirectory/
```

mv (move): used to rename a file or to move it across directories

- mv file1 file2 renames file1 to file2
- mv directory /tmp/ : moves an entire directory elsewhere
- mv /tmp/*txt . : more all the files ending by “txt” to current directory

rm: deletes file(s)

Prefer using “rm -i” at the beginning (will ask for confirmation)

- rm file delete the file (without anyway to get it back)
- rm -i *.txt deletes all files ending by “txt”

rmdir: deletes a directory, must be empty

diff : shows the difference between two files.

Usage: diff file1 file2

grep: look for a string (or a pattern) in a text file

In NLP, that is one of the most useful command ;-)

usage:

```
grep 'string'  textFile
```

```
grep 'indi'  parallel-corpora/hi-en/training.en
```

(usually you combine with a pipe:)

```
grep 'indi'  parallel-corpora/hi-en/training.en | more
```

cut : takes a given column out of a text file

Option -c is for columns and -f for fields. It is entered as

```
cut -f n textFile
```

=> will display only the column n. The textFile is supposed to be tab-separated

- -d’char’ use the “char” as a field separator (e.g. cut -d' ' -f 1 parallel-corpora/ur-en/training.en|more shows only the first word of each line).
- -f n,m will show the fields n and m (use 2,3 to show 2nd and 3rd, 2-4 to show from 2dn to 4th, 5- to show all fields from column 5)

paste: “paste” two text files

It concatenates each lines of two text files (and separates them by a tab).

e.g. show line-by-line the English and Hindi sentences in the trainset:

paste then running this command

```
paste parallel-corpora/hi-en/training.en parallel-corpora/hi-en/training.hi | more
```

```
initially, it ran on 6 routes which joined most of Delhi's parts.      प्रारंभिक  अवस्था में इसकी योजना छह मार्गों पर चलने की है जो दिल्ली के ज्यादातर हिस्से को जोड़ेगी।
At present situation it is programed to run in six lanes which joins maximun are as of delhi.       प्रारंभिक अवस्था में इसकी योजना छह मार्गों पर चलने की है जो दिल्ली के ज्यादा तर हिस्से को जोड़ेगी।
```

Note that the two files are supposed to have the same number of lines

sort: sort a text file according to alphabetical order

e.g.

Options:

- -r sort in reverse order
- -u do not display identical lines (useful in NLP for duplicate removal).

uniq: delete adjacent duplicates

Sometimes used in NLP to delete duplicate adjacent lines (sort -u is equivalent to sort | uniq)

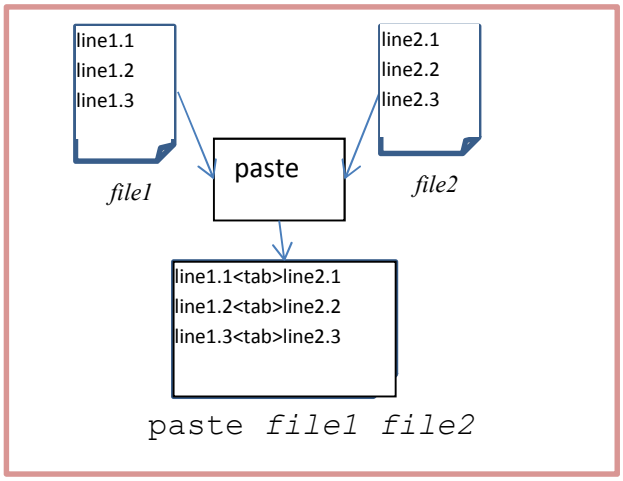
echo: displays something on the screen

```
echo "a text"
```

It displays it parameter (usually a text between quotes") on the screen (on its standard output)

can be used to "feed" a one-line text file. E.g. asking for the translation of a hindi sentence:

```
echo "मैं दुकान जाएगा" | mosesdecoder/bin/moses -f parallel-corpora/hi-en/trainMoses/models/moses.ini
```



tar: manage archive (extract or create)

tar manages archives (like the "zip" on windows), it combines creation and extraction

tar -c => will create an archives

tar -x => will extraction

tar -t => to look at the contents

add "z" to add (de)compression

e.g. List all files from the archive:

```
tar -tzvf indic-corpora.tar.gz
```

Extract all files from the archive:

```
tar -xzvf indic-corpora.tar.gz
```

To create a compressed mini archive containing only the "hindi" corpus:

```
tar -czvf myHindiArchive.tgz parallel-corpora/hi-en
```

chmod: change permissions

chmod command is used to change permissions on a file.

In practice you may not have to use it.

chmod +x File ==> give execution right on a file (e.g. a bash ".sh", python ".py", or perl script ".pl")

chmod -w File ==> Will forbid to write in the filename

synopsis: chmod [+-] [rwx] (r for read, x for execution, w fr write)