

Unix Tools for Climate Dynamics (and everything else!)

Jean-Yves Peterschmitt / LMCE / 12-97
peterschmitt@cea.fr

or "How to deal with ASCII data files without writing a program"

The Unix system provides several standard tools that can be used to manipulate text files

more	diff	diff
sed	sort	...

You can save a lot of time if you know that those programs exist, and where to find a user's manual

```
man command_name  
"UNIX in a Nutshell", O'Reilly & Associates, Inc
```

A few things you should know!

stdin/stdout? What is that?

- "stdout" is the standard output. This is where a program will write its results. The screen, by default
- "stdin" is the standard input. This is where a program will read its data. A file name, usually

The output of a program can be

- sent to a file by redirecting it with ">"
ls > my_file
- sent to the standard input of another program through a "pipe", "|"
ls | lpr

Note : some programs can't recognize automatically that they are getting their data through a pipe. You have to specify "-" instead of the regular file name

```
zcat mytarfile.tar.Z | tar tvf -
```

What are "regular expressions"?

A regular expression is a sequence of normal and special characters used to match text patterns. It allows the user to look for certain text patterns in files, replace them, ...

Some metacharacters :

- . Any single character (use \. if you want to match a ".")
- * Any number of the single preceding character (0* -> any number of 0)
- ^ or \$ Beginning (^) or end (\$) of the line
- [] Any one of the enclosed characters (use a hyphen for a range of chars)
- ^[0-9.]* will match any positive real number at the beginning of a line

Useful commands...

"Viewing" a file

cat	send the content of a file to stdout	grep	look for a string (regular expression) in a file
head	display the beginning of a file	options : -i ignore uppercase/lowercase distinction, -v get lines that don't match regexp	
head -n -15 my_file	displays the first 15 lines	grep -i "Ac" my_prog.f	will display the comment lines (lines starting with C or c)
tail	display the end of a file	wc	count the number of lines/words/characters in a file
tail -n -15 my_file	displays the last 15 lines	wc -l my_file	will display the number of lines of my_file
tail -n +15 my_file	displays the file, starting at line 15	sdiff	display a side by side comparison of 2 files
diff	compare the content of 2 files	sdiff -w80 file1 file2	will display the differences in a 80-column window
comm	display the lines common and unique to 2 sorted files	uniq	remove duplicate adjacent lines from a file

Changing a file

sort	sort the content of a file in alphabetical order	sort -n -r my_file > newfile	will sort the file in reversed numerical order
options : -n	sort in numerical order, -r	reverse the order	
sed	perform changes on a file (and much much more)	sed 's/./g' my_file > newfile	will change all "." in "."
awk	perform (very) complex operations on a file	awk '{print \$1, \$3}' my_file	will extract the 1st and the 3rd column of a file
awk 'BEGIN {tot = 0 ; nb = 0} NF == 1 {tot += \$1 ; nb++} END {print "mean = " tot/nb}'	my_file		will compute the mean of a column of data stored in my_file, skipping empty lines
paste	merge corresponding lines of 2 or more files into vertical columns	paste file1 file2 > my_file	will paste file1 one and file2 (if file1 and file2 have one column of data, my_file will have 3 columns)

Combining commands

The commands shown above can be combined by using pipes. Several pipes can be used on the same line!

```
tail -n +5 my_file | head -n -3 | wc -l | wc -l  
will show the number of empty lines of my_file
```