

Basic Shell Commands

Jeremy Sanders

October 2011

1. **acroread** - Read or print a PDF file.
2. **cat** - Send a file to the screen in one go. Useful for piping to other programs
3. `cat file1` # list file1 to screen
4. `cat file1 file2 file3 > outfile` # add files together into outfile
5. `cat *.txt > outfile` # add all .txt files together
6. `cat file1 file2 | grep fred` # pipe files
7. **cc** - Compile a C program
8. `cc test1.c` # compile test1.c to a.out
9. `cc -O2 -o test2.prog test2.c` # compile test2.c to test2.prog
10. **cd** - Change current directory
11. `cd` # go to home directory
12. `cd ~/papers` # go to /home/user/papers
13. `cd ~fred` # go to /home/fred
14. `cd dir` # go to directory (relative)
15. `cd /dir1/dir2/dir3...` # go to directory (absolute)
16. `cd -` # go to last directory you were in
17. **cp** - Copy file(s)
18. `cp file1 file2` # copy file1 to file2
19. `cp file1 directory` # copy file1 into directory
20. `cp file1 file2 file3 ... directory` # copy files into directory
21. `cp -R dir1 dir2/` # copy dir1 into dir2 including subdirectories
22. `cp -pR dir1 dir2/` # copy directory, preserving permissions
23. **date** - Shows current date
24. `> date`
25. `Sat Aug 31 17:18:53 BST 2002`
26. **dvips** - Convert a dvi file to PostScript
27. `dvips document.dvi` # convert document.dvi to document.ps
28. `dvips -Ppdf document.dvi` # convert to ps, for conversion to pdf
29. **emacs** - The ubiquitous text editor
30. `emacs foo.txt` # open file in emacs
31. `emacsclient foo.txt` # open file in existing emacs (need to use
32. # M-x start server first)
33. **file** - Tells you what sort of file it is
34. `> file temp_70.jpg`
35. `temp_70.jpg: JPEG image data, JFIF standard 1.01,`
36. `resolution (DPI), 72 x 72`
37. **firefox** - Start Mozilla Firefox
38. **f77/f90** - Compile a Fortran 77/99 program
39. `f77 -O2 -o testprog testprog.f`
40. **gedit** - Gnome text editor
41. **gnuplot** - A plotting package.

42.grep - Look for text in files. List out lines containing text (with filename if more than one file examined).

```
43. grep "hi there" file1 file2 ... # look for 'hi there' in files
44. grep -i "hi there" filename     # ignore capitals in search
45. cat filename | grep "hi there"  # use pipe
46. grep -v "foo" filename          # list lines that do not include foo
```

47.gtar - GNU version of the tar utility (also called tar on Linux). Store directories and files together into a single archive file. Use the normal tar program to backup files to a tape. See `info tar` for documentation.

```
48. gtar cf out.tar dir1           # put contents of directory into out.tar
49. gtar czf out.tar.gz dir1       # write compressed tar, out.tar.gz
50. gtar tf in.tar                 # list contents of in.tar
51. gtar tzf in.tar.gz             # list contents of compressed in.tar.gz
52. gtar xf in.tar                 # extract contents of in.tar here
53. gtar xzf in.tar.gz             # extract compressed in.tar.gz
54. gtar xf in.tar file.txt ...    # extract file.txt from in.tar
```

55.gv - View a Postscript document with Ghostscript.

56.gzip / gunzip - GNU Compress files into a smaller space, or decompress `.Z` or `.gz` files.

```
57. gzip file.fits                # compresses file.fits into file.fits.gz
58. gunzip file.fits.gz           # recovers original file.fits
59. gzip *.dat                    # compresses all .dat files into .dat.gz
60. gunzip *.dat.gz               # decompresses all .dat.gz files into .dat
61. program | gzip > out.gz       # compresses program output into out.gz
62. program | gunzip > out        # decompresses compressed program output
```

63.info - A documentation system designed to replace `man` for GNU programs (e.g. `gtar`, `gcc`). Use cursor keys and return to go to sections. Press `b` to go back to previous section. A little hard to use.

```
64. info gtar                    # documentation for gtar
```

65.kill - Kill, pause or continue a process. Can also be used for killing daemons.

```
66. > ps -u jss
67. ...
68. 666 pts/1          06:06:06 badprocess
69. > kill 666          # this sends a ``nice'' kill to the
70.                    # process. If that doesn't work do
71. > kill -KILL 666    # (or equivalently)
72. > kill -9 666      # which should really kill it!
73.
74. > kill -STOP 667   # pause (stop) process
75. > kill -CONT 667   # unpause process
```

76.latex - Convert a tex file to dvi

77.logout - Closes the current shell. Also try ```exit"`.

78.lp - Sends files to a printer

```
79. lp file.ps # sends postscript file to the default printer
80. lp -dlp2 file.ps # sends file to the printer lp2
81. lp -c file.ps # copies file first, so you can delete it
82. lpstat -p lp2 # get status and list of jobs on lp2
83. cancel lp2-258 # cancel print job lp2-258
84.
85. lpr -Plp2 file.ps # send file.ps to lp2
86. lpq -Plp2         # get list of jobs on lp2
```

87. `lprm -Plp2 1234` # delete job 1234 on lp2

88. `ls` - Show lists of files or information on the files

89. `ls file` # does the file exist?
90. `ls -l file` # show information about the file
91. `ls *.txt` # show all files ending in .txt
92. `ls -lt` # show information about all files in date order
93. `ls -lrt` # above reversed in order
94. `ls -a` # show all files including hidden files
95. `ls dir` # show contents of directory
96. `ls -d dir` # does the directory exist?
97. `ls -p` # adds meaning characters to ends of filenames
98. `ls -R` # show files also in subdirectories of directory
99. `ls -l` # show one file per line

100. `man` - Get instructions for a particular Unix command or a bit of Unix.

Use space to get next page and q to exit.

101. `man man` # get help on man
102. `man grep` # get help on grep
103. `man -s1 sort` # show documentation on sort in section 1

104. `more` - Show a file one screen at a time

105. `more file` # show file one screen at a time
106. `grep 'frog' file | more` # Do it to output of other command

107. `mv` - Move file(s) or rename a file

108. `mv file1 file2` # rename file1 to file2
109. `mv dir1 dir2` # rename directory dir1 to dir2
110. `mv file1 file2 file3 ... directory` # move files into directory

111. `nano` - very simple text editor. Warning - this program can introduce extra line breaks in your file if the screen is too narrow!

112. `nice` - Start a process in a nice way. Nice levels run from -19 (high priority) to 19 (low priority). Jobs with a higher priority get more CPU time. See `renice` for more detail. You should probably be using the grid-engine to run long jobs.

113. `nice +19 myjob1` # run at lowest priority
114. `nice +8 myjob2` # run at lowish priority

115. `openoffice.org` - a free office suite available for Linux/Unix, Windows and Mac OS X.

116. `passwd` - change your password

117. `pine` - A commonly used text-based mail client. It is now called `alpine`. Allows you to send and receive emails. Configuration options allow it to become quite powerful. Other alternatives for mail are `mozilla mail` and `mutt`, however I suggest you stick to `alpine` or `thunderbird`.

118. `printenv` - Print an environment variable in `tcsh`

119. `setenv MYVARIABLE Fred`
120. `printenv MYVARIABLE`
121. `printenv` # print all variables

122. `ps` - List processes on system

123. `> ps -u jss` # list jss's processes
124. 934 pts/0 00:00:00 bash
125. ^^^^^ ^^^^^ ^^^^^ ^^^^^
126. PID output CPU time name

```

127. > ps -f          # list processes started here in full format
128. > ps -AF         # list all processes in extra full format
129. > ps -A -l        # list all processes in long format
130. > ps -A | grep tcsh # list all tcsh processes
131.      pwd - Show current working directory
132. > pwd
133. /home/jss/writing/lecture
134.      quota - Shows you how much disk space you have left
135. > quota -v
136. ...
137.      renice - Renice a running process. Make a process interact better with
other processes on the system (see top to see how it is doing). Nice levels run
from -19 (high priority) to 19 (low priority). Only your own processes can be
niced and they can only be niced in the positive direction (unless you are root).
Normal processes start at nice 0.
138. > ps -u jss | grep bigprocess      # look for bigprocess
139. 1234 pts/0      99:00:00 bigprocess
140. > renice 19 1234                    # renice PID 1234 to 19
141.      rm - Delete (remove) files
142. rm file1      # delete a file (use -i to ask whether sure)
143. rm -r dir1     # delete a directory and everything in it (CARE!)
144. rm -rf dir1    # like above, but don't ask if we have a -i alias
145.      rmdir - Delete a directory if it is empty (rm -r dirname is useful if it is
not empty)
146. rmdir dirname
147.      staroffice - An office suite providing word processor, spreadsheet,
drawing package. See Users' Guide on how to install this. This is a commercial
version of the openoffice office package - use openoffice.org on linux.
148.      setenv - Set an environment variable in tcsh.
149. setenv MYVARIABLE Fred
150. echo Hi there $MYVARIABLE
151.      tar - Combine files into one larger archive file, or extract files from that
archive (same as gtar on Linux).
152. tar cvf /dev/rmt/0 ./              # backup cwd into tape
153. tar tvf /dev/rmt/0                 # list contents of tape
154. tar xvf /dev/rmt/0                 # extract contents of tape
155.      thunderbird - Start mozilla thunderbird.
156.      top - Interactively show you the ``top'' processes on a system - the ones
consuming the most computing (CPU) time. Press the ``q'' key in top to exit.
Press the ``k'' key to kill a particular process. Press ``r'' to renice a process.

```

About this document ...

Basic Shell Commands

This document was generated using the [LaTeX2HTML](#) translator Version 2008 (1.71)

Copyright © 1993, 1994, 1995, 1996, [Nikos Drakos](#), Computer Based Learning Unit, University of Leeds.

Copyright © 1997, 1998, 1999, [Ross Moore](#), Mathematics Department, Macquarie University, Sydney.

The command line arguments were:

latex2html -split 0 -font_size 10pt -no_navigation commands_basic.tex

The translation was initiated by Jeremy Sanders on 2011-10-02

https://www-xray.ast.cam.ac.uk/~jss/lecture/computing/notes/out/commands_basic/