

Bash Basics

Bash is a command shell that gives a lot more flexibility than the standard Windows command shell. This document gives just enough information to get started using it, plus a few common commands not actually part of bash which you are likely to want to use. A comprehensive beginners manual for bash can be found at <http://www.tldp.org/guides.html>.

Useful Tips

- File paths use "/" character, NOT "\" character as in Windows.
- You can access windows drives as /cygdrive/drive (see below for examples).
- Scroll up and down through past commands using the up or down arrow.
- Edit a line using left or right arrows, delete and/or enter text, then hit return to execute the line.
- If your command needs a file name you can type just the beginning of it (enough to be unique) and press tab. Bash will complete the rest of the file name for you.
- To abandon a line, or to stop execution of a program, type control c.
- To direct standard output into a file, use the ">" symbol followed by the file name. This will overwrite the contents of the file, if it already exists.
- To append standard output to a file (after any data that's already in it) use ">>" followed by the file name.
- The * character is a file name wild-card - matches any string of characters (even a dot) in a file name.
- The "|" character is a pipe - pipe the standard output of one command into the standard input of another.
- To use a special character (such as > or * for example) in a word of a command enclose the whole word in quotes.
- The current directory can be accessed as "." (dot).
- The parent directory of the current directory can be accessed as ".." (two dots).
- When you type in a command, bash looks in the search path (environment variable PATH) for that command and prints an error message ("command not found") if it can't find it.
- The current directory is often not included in the search path. If you want to execute something from the current directory, prefix it with its path (eg ./myfile).

Some Useful Commands (not part of bash)

ls - list files in current directory (use "**ls -l**" to get more details on each file)

cd - change directory. Without any argument goes to your home directory, otherwise goes to the directory you name.

rm - remove (delete) a file (cannot then be undeleted). To recursively delete a directory and all files below it use "**rm -r**"

cp - copy a file

mv - move a file - use to move a file to a different directory or just to rename a file.

mkdir - make a directory in the current directory

rmdir - remove a directory (must be empty - other wise use rm -r)

less - display the contents of a file on the screen, one screenfull at a time. Press space bar to get next screen. Use up/down arrows to move up/down a line. Use "page up", "page down" keys to scroll a page up or down. Use G (or "End" key) to go to the end of the file, 1G (or "Home" key) to go to the start. Use "/string" to find string in the file. Press q to quit.

grep - look for a string in a file or files

Examples

\$ ls *.txt	lists all files whose name ends in .txt
\$ ls -l results*	lists details of all files whose name begins with results
\$ cd newdir	changes to directory (folder) newdir
\$ cd ..	changes to next directory up
\$../doit	execute the file called doit which can be found in the next directory up
\$../doit > res.txt	execute doit and put its output into file res.txt in the current directory
\$../doit less	execute doit and pipe its output through less (display one screen at a time)
\$ mv ../doit .	move doit from the next directory up to the current directory
\$ mv doit fred	rename file doit to the name fred
\$ cp fred doit	copy file fred to file doit (would overwrite doit if it already existed)

\$ `rm do*` delete all files whose name starts with do in current directory (eg doit)

\$ `mkdir files` make a directory called files in the current directory

\$ `grep "fred*" *` look in all files in current directory for lines containing the string fred* and print them out

\$ `grep -R "fred*" *` as above, but recursively look through all files in all directories

\$ `grep -R "fred*" * | less` as above, but pipe the results into less (display a screen at a time)

\$ `ls /cygdrive/c/user` list files in C:\user folder

\$ `cp -r foo /cygdrive/g` Recursively copy the "foo" folder and all files below it to drive g: (which might be, for example, your usb memory stick).

PRB 25/10/05