Survival guide to the Unix shell bash

Command syntax

Every Unix command has a name, a set of options, and a set of arguments:

name options arguments

Options are normally preceded by a minus sign `-` to distinguish them from arguments. A list of common Unix commands is given in the attached table.

Files and directories

`/`	Root of file system
`~`	Home directory
`.`	Current (or working) directory
`..`	Parent of current directory
name	File *name* in current directory
/f/g/h/name	Full path (file) name

The most important file processing commands are: `ls`, `more`, `mv`, `cp`, `rm`.

The most important directory processing commands are: `pwd`, `cd`, `mkdir`, `rmdir`

File permissions

Every file has a type, an owner, a group and a set of permissions.

The main file types are plain (text) files and directories.

Every file may have read/write/execute permission for user (owner), group and other users. These are displayed by the `ls -l` command and changed by the `chmod` command.

Directories can be stacked:

`dirs`	Display directory stack
`pushd` *dir*	Push and move to directory *dir*
`pushd +`*n*	Move to *n*th directory on stack
`pushd`	Exchange top and previous directory on stack
`popd`	Pop to previous directory on stack

`cd` changes top directory on stack.

Filename expansion

Sets of file names may be abbreviated:

`*`	Any string of characters
`?`	Any single character
`[`*string*`]`	Any character in *string*
`{`*string1*`,`*string2*`,...}`	The given set of (sub)strings

Autocompletion

Pressing *TAB* while entering a file name automatically completes the file name. If there are more than one possible completion, it beeps. To see the possible completions, type `^D`. Then type one or more characters, and press *TAB* again.

File redirection

command `<` *from*	*command* reads standard input from file *from*.
command `>` *to*	*command* writes standard output to file *to*, overwriting old file *to*
command `>>` *to*	*command* appends standard output to file *to*
command `>&` *to*	*command* writes stdout and stderr to file *to*

(This last construct also works for appending and piping (below); just add an "&" character after ">>" or "|".)

Command composition

command1 `;` *command2*	Execute *command1* then *command2*
command1 `|` *command2*	Pipe standard output from *command1* to standard input of *command2*

Job control

command `&`	Execute *command* in background
`^Z`	Suspend foreground process (job)
`stop %`*n*	Suspend background job *n*
`kill %`*n*	Terminate background job *n*
`fg %`*n*	Resume job *n* in foreground
`bg %`*n*	Resume job *n* in background
`jobs`	Display current jobs

History mechanism

`history`	Display last *n* commands
`!!`	Repeat previous command
`!`*m*	Repeat command *m*
`!`*cmd*	Repeat last command starting with string *cmd*
`!^`	First argument (or option) of previous command
`!$`	Last argument of previous command
`!*`	All arguments of previous command
`^`*old*`^`*new*	Repeat previous command with string *old* replaced by string *new*

Previous commands may also be found, edited, and executed using the up/down arrow keys.

.bashrc

Commands can be renamed and new commands defined:

```
alias name=command
```

The *command* may contain `;`, `|`, `!*` (as above). For example:

```
alias ll='ls -l \!* | more'
alias m=more
alias rm='rm -i'
```

Many variables can be defined, for example:

```
set ignoreeof notify noclobber
```

It is normal to put such definitions in the file `~/.bashrc`.

Regular expressions

These are used by *grep*, *sed*, *ed*, *vi*, and many other commands.

`.`	Any character
`^`	Start of line
`$`	End of line
c	Character *c*
`[`*string*`]`	Any character in *string*, *e.g.*, `[a-zA-Z0-9]`
`[^`*string*`]`	Any character not in *string*
e`*`	0 or more occurences of the regular expression *e*
e`+`	1 or more occurences of the regular expression *e*
`\(`*e*`\)`	Regular expression *e*
`\`*n*	Bracketed regular expression starting with *n*th \(
e1 e2	Regular expression *e1* followed by regular expression *e2*

Control characters

DELETE	Delete (or rubout) last character typed
`^W`	Delete last word typed
`^U`	Delete last line typed
`^S/^Q`	Suspend/resume output to screen
`^C`	Terminate current process
`^Z`	Suspend current process
`^D`	End of (standard input) file

Common Unix commands

`man` *command*	Display manual entry for *command*
`ls`	List (display) names of all files in current directory
`ls` *file* `...`	List named files or files in named directories. Options follow:
`-a`	List all, including files whose names start with .
`-d`	List (as) directory, not files in directory
`-l`	List long, more information
`-t`	List in time order, most recent first
`file` *file* `...`	Display (estimated) type of named files
`cp` *file1 file2*	Copy *file1* to *file2*, overwrite old *file2* if it exists
`ln` *file1 file2*	Link *file1* to *file2* (*file1* and *file2* are now aliases)
`ln -s` *file1 file2*	Symbolic link from *file1* (a directory) to *file2*
`mv` *file1 file2*	Move (rename) *file1* to *file2*, overwrite old *file2* if it exists
`rm` *file* `...`	Remove named files, irrevocably
`cp/mv/rm -i` *file* `...`	Overwrite/remove named files only after confirmation
`chmod` *int file* `...`	Change permission of named files
`pwd`	Print working (current) directory
`cd` *dir*	Change (move) to named directory
`mkdir` *dir*	Make a new named directory

`rmdir` *dir*	Remove named directory, irrevocably
`pushd` *dir*`/popd/dirs`	Push/pop/display directories on stack
`ed` *file*	Edit named file
`vi` *file*	Visually edit named file
`cat` *file* `...`	Concatenate (display) contents of named files
`more` *file* `...`	Display contents of named files, a screen at a time
`less` *file* `...`	Less is more, enables backward movement in files
`head` *file*	Display first 10 lines of named file
`head -`*n file*	Display first *n* lines of named file
`tail` *file*	Display last 10 lines of named file
`tail +`*n file*	Start displaying at line *n*
`pr` *file* `...`	Display contents of named files with headers
`pr -`*n file* `...`	Display contents of named files with headers in *n* columns
`lpr` *file* `...`	Print contents of named files (text files only!)
`lpq`	Display printer status
`lprm` *job*	Remove *job* from printer queue
`lpr/lpq/lprm -P`*printer*	Use named printer
`awk '`*commands*`'` *file* `...`	Apply *awk* commands to named files
`bc`	Arbitrary precision arithmetic calculator
`cal` *month year*	Display calendar for given month and year
`cmp` *file1 file2*	Compare named files
`date`	Display current date and time
`diff` *file1 file2*	Display all differences between named files
`fmt` *file* `...`	Format named files
`grep` *pattern file* `...`	Print lines of named files matching *pattern*
`logout`	Terminate session
`mail`	Read mail
`mail` *user*	Send mail to *user*
`passwd`	Change your password
`pine`	A user-friendly mail client
`sed '`*commands*`'` *file* `...`	Stream editor: apply *ed* commands to named files
`sed -f` *program file* `...`	Apply commands in file *program* to named files
`set` *variable* `[=` *value*`]`	Set or assign shell variable
`sort` *file* `...`	Sort named files alphabetically by line
`stty ...`	Set terminal characteristics
`tr` *string1 string2*	Copy standard input to standard output, translating all chars in *string1* into corresponding chars in *string2*
`wc` *file* `...`	Count lines, words and characters of named files
`who`	Display who is using the machine

More information

Read the (extensive) `bash` manual:

```
man bash
```