Common File System Commands

`ls`	List names of all files in current directory
`ls` *filenames*	List only the named files
`ls -t`	List in time order, most recent first
`ls -l`	Long listing, more information. Also `ls -lt`
`ls -t`	List by time last used. Also `ls -lu`, `ls -lut`
`ls -r`	List in reverse order. Also `ls -rt`, `ls -rlt`, etc
`ed` *filename*	Edit named file
`cp` *file1 file2*	Copy *file1 file2*. Overwrite old *file2* if it exists
`mv` *file1 file2*	Move *file1 file2*. Overwrite old *file2* if it exists
`rm` *filenames*	Remove named files, irrevocably
`cat` *filenames*	Print contents of named files
`pr` *filenames*	Print content with header, 66 lines per pages (default)
`pr -`*n* *filenames*	Print in *n* columns
`pr -m` *filenames*	Print named files side by side in multiple columns
`wc` *filenames*	Count lines, words, and characters for each file
`wc -l` *filenames*	Count lines for each file
`grep` *pattern filenames*	Print lines matching *pattern*
`grep -v` *pattern filenames*	Print lines not matching *pattern*
`sort` *filenames*	Sort files alphabetically by line
`tail` *filename*	Print last 10 lines of file
`tail -`*n filename*	Print last *n* lines of file
`tail +`*n filename*	Start printing file at line *n*
`cmp` *file1 file2*	Print location of first difference
`diff` *file1 file2*	Print all differences between files

Shell Metacharacters

`>`	*prog* `>` *file* direct standard output to *file*
`>>`	*prog* `>>` *file* append standard output to *file*
`<`	*prog* `<` *file* take standard input from *file*
`|`	p_1 `|` p_2 connect standard output of p_1 to standard input of p_2
`<<here`	*here document*: standard input follows, up to next `here` on a line by itself
`*`	Match any string of zero or more characters in filenames
`?`	Match any single character in filenames
`[`*ccc*`]`	Match any single character from `[`*ccc*`]` in filenames. Ranges like `0-9` or `a-z` are legal
`;`	Command terminator: p_1 `;` p_2 does p_1, then p_2
`&`	Like `;` but does not wait for p_1 to finish
`` `…` ``	Run command(s) in … ; output replaces `` `…` ``
`(…)`	Run command(s) in … in a sub-shell
`{…}`	Run command(s) in … in current shell (rarely used)
`$1`, `$2`, *etc*	`$0` … `$9` replaced by arguments to shell file
`$`*var*	Value of shell variable *var*
`${`*var*`}`	Value of *var*; avoids confusion when concatenated with text
`\`	`\`*c* take character *c* literally, `\`*newline* discarded
‘…’	Take … literally
”…”	Take … literally after `$`, `` `…` `` and `\` interpreted
#	Text after # is a comment
var=*value*	Assign *value* to variable *var*
p_1 `&&` p_2	Run p_1; if successful, run p_2
p_1 `||` p_2	Run p_1; if unsuccessful, run p_2

Shell I/O Redirections

> *file*	direct standard output to *file*
>> *file*	append standard output to *file*
< *file*	take standard input from *file*
p_1 | p_2	connect standard output of program p_1 to input of p_2
^	obsolete synonym for |
n>*file*	direct output from file descriptor *n* to *file*
n>>*file*	append output from file descriptor *n* to *file*
n>&*m*	merge output from file descriptor *n* with file descriptor *m*
n<&*m*	merge input from file descriptor *n* with file descriptor *m*
<<*s*	here document: take standard input until next *s* at beginning of a line; substitute for $, `…`, and \
<<*s*	here document with no substitution
<< '*s*'	here document with no substitution

`grep` and `egrep` Regular Expressions
(decreasing order of precedence)

c	any non-special character *c* matches itself
c	turn off any special meaning of character *c*
^	beginning of line
$	end of line
.	any single character
[…]	any one of characters in …; ranges like a-z are legal
[^…]	any single character not in …; ranges are legal
n	what the *n*'th \(…\) matched (grep only)
*r**	zero or more occurrences of *r*
r+	one or more occurrences of *r* (egrep only)
r?	zero or more occurrences of *r* (egrep only)
$r_1 r_2$	r_1 followed by r_2
$r_1 \mid r_2$	r_1 or r_2 (egrep only)
\(*r*\)	tagged regular expression *r* (grep only); can be nested
(*r*)	regular expression *r* (egrep only); can be nested
	No regular expression matches a newline.

Shell Built-in Variables

`$#`	the number of arguments
`$*`	all arguments to shell. "`$*`" is a single word
`$@`	similar to `$*`. "`$@`" is identical to the list of the arguments to shell
`$-`	options supplied to the shell
`$?`	return value of the last command executed
`$$`	process-id of the shell
`$!`	process-id of the last command started with `&`
`$HOME`	default argument for `cd` command
`$IFS`	list of characters that separate words in arguments
`$MAIL`	file that, when changed, triggers "you have mail" message
`$PATH`	list of directories to search for commands
`$PS1`	prompt string, default '$ '
`$PS2`	prompt string for continued command line, default '> '

Shell Pattern Matching Rules

`*`	match any string, including the null string
`?`	match any single character
`[`*ccc*`]`	match any of the characters in *ccc* `[a-d0-3]` is equivalent to `[abcd0123]`
”…”	match … exactly; quotes protect special characters. Also ’…’
`\`*c*	match *c* literally
`a|b`	in `case` espressions only, matches either `a` or `b`
`/`	in filenames, matched only by an explicit `/` in the expression; in `case`, matched like any other character
`.`	as the first character of a filename, is matched only by an explicit `.` in the expression