Doing More with SunOS: Quick Reference

This quick reference lists commands presented in this manual, including a syntax diagram and brief description.

1. Files

1.1. Filename Substitution

Wild Cards	? *
Character Class	[c...]
Range	[c–c]

c is any single character.

String Class	{str[, str]}

str is a combination of characters, wild cards, embedded character classes and embedded string classes.

Home Directory	~
Home Directory of Another User	~user
List Hidden Files	ls -[l]a

1.2. File Properties

Seeing Permissions	ls -l *filename*
Changing Permissions	chmod *nnn filename*
	chmod *c=p...[,c=p...] filename*

n, a digit from 0 to 7, sets the access level for the user (owner), group, and others (public), respectively. *c* is one of: u – user, g – group, o – others, or a – all. *p* is one of: r – read access, w – write access, or x – execute access.

Setting Default Permissions	umask *ugo*

ugo is a (3-digit) number. Each digit restricts the default permissions for the user, group and others, respectively.

Changing Modification Time	touch *filename*
Making Links	ln *oldname newname*
	ln -s *oldname newname*

Seeing File Types	ls -F

1.3. Encrypting Files

Source Files	crypt < *source* > *encrypted*
Editing	vi -x *encrypted*
Decrypting Files	crypt < *encrypted*	more
	crypt < *encrypted* > *text*

crypt asks for the encryption key.

1.4. Searching with more

Run more	more *filename*
Next Line	Return
Next 11 Lines	d
Next Page	SPACE
Search for Pattern	/*pattern*
Next Occurrence	n
Next File	:n

1.5. The Directory Stack

Change Directory, Push	pushd *directory*
Change to Top Directory, Pop	popd
Show Stack	dirs

2. Commands

2.1. Command-Line Special Characters

Quotes and Escape

Join Words	"..."
Suppress Filename, Variable Substitutions	'...'
Escape Character	\

Separation, Continuation

Command Separation	;
Command-Line Continuation	\Return

2.2. I/O Redirection and Pipes

Standard Output	>
	>!
Appending to Standard Output	>>
	>>!
Standard Input	<
Standard Error and Output	>&
Standard Error Separately	
	(*command* > *output*) >& *errorfile*
Pipes/Pipelines	*command*	*filter* [	*filter*] ...
Duplicating Displayed Output	
	command	tee *filename*

Filters

Word/Line Count	wc [-l]
First *n* Lines	head [-*n*]
Last *n* Lines	tail [-*n*]
Skip to Line *n*	tail [+*n*]
Show Nonprinting Characters	cat -v
Sort lines	sort [-*n*]
Format Paragraphs	fmt
Reverse Character Order	rev
Multicolumn Output	pr -t
List Spelling Errors	spell
Substitutions in Output Stream	
	sed -e "s/*pattern*/*string*/[g]"
Report-Generation	awk

2.3. Searching with grep

grep Command	grep "*pattern*" *filename*
	command	grep "*pattern*"

grep Search Patterns

beginning of line	^
end of line	$
any single character	.
single character in list or range	[...]

character not in list or range `[^...]`

zero or more of preceding character

    or pattern `*`

zero or more of any character `.*`

escapes special meaning `\`

3. C-Shell Features

3.1. History Substitution

The History List

Set Up History List `set history=n`

See History List `history [-h]`

Event Designators

Repeat Previous Command `!!`

Display Previous Command `!!:p`

Command Line *n* `!n`

n Commands Back `!-n`

Command Beginning with *str* `!str`

Command Containing *str* `!?str[?]`

All Arguments to Prev. Command `!*`

Last Argument to Prev. Command `!$`

First Argument to Prev. Command `!^`

n'th Argument `!:n`

Word Designators

All Arguments `:*`

Last Argument `:$`

First Argument `:^`

n'th Argument `:n`

Arguments *x* Through *y* `:x-y`

Modifiers

Print Command Line `:p`

Substitute Command Line `:[g]s/l/r/`

3.2. Aliases

alias Command `alias name 'definition'`

> *definition* can contain escaped history substitution event and word designators as placeholders for command-line arguments.

3.3. Variable Substitution

Creating a Variable `set var`

Assigning a Value `set var = value`

Expressing a Value `$var`

Displaying a Value `echo $var`

> *value* is a single word, an expression in quotes, or an expression that results in a single word after variable, filename and command substitution takes place.

Assigning a List `set var = (list)`

> *list* is a space-separated list of words, or an expression that results in a space-separated list.

Selecting the *n*'th Item `$var[n]`

Selecting all Items `$var`

Selecting a Range `$var[x-y]`

Item Count `$#var`

3.4. foreach Lists

Start foreach Loop `foreach var (list)`

> foreach prompts for commands to repeat for each item in *list* (with >), until you type `end`. Within the loop, `$var` stands for the current item in *list*.

3.5. Command Substitution

Replace Command with its Output on the

Command Line `` `...` ``

3.6. Job Control

Run Command in the Background `&`

Stop Foreground Job `Control-Z`

List of Background Jobs `jobs`

Bring Job Forward `%[n]`

Resume Job in Background `%[n]`

4. Processes

Listing `ps -[aux]`

Terminating `kill [-9] PID`

Timing `time command`

Scheduling `at time[a|p] script`

> *time* is a number up to 4 digits. *script* is the name of a file containing the command line(s) to perform.

5. Users

Seeing Who Is Logged In `who`

`w`

Changing Identities `su [username]`

Seeing Your User Name `whoami`
`who am i`
`who is this`

6. Managing Files

6.1. Looking Up Files

Standard Commands `whereis file`

Aliases and Commands `which command`

Describe Command `whatis filename`

Searching Out Files

`find dir -name name -print`

> *dir* is a directory name within which to search. *name* is a filename to search for.

6.2. Tracking Changes

Comparing Files `diff leftfile rightfile`

> diff prefixes a left angle-bracket (<) to selected lines from *leftfile* and a right angle bracket (>) to lines from *rightfile*.

Auditing Changes

Putting Files Under **sccs**	**mkdir SCCS**
	chmod 775 SCCS
	sccs create *filename*...
	rm , *
Checking Files Out	**sccs edit** *filename*...
Checking Files In	**sccs delget** *filename*...
Backing Files Out	**sccs unedit** *filename*...

Recovering Current Versions

sccs get SCCS

Reviewing Pending Changes

sccs diffs *filename*...

6.3. Automating Tasks

Create a Makefile	**vi Makefile**

A makefile consists of macro definitions and targets.

Test Makefile	**make -n** [*target*]
Run make	**make** [*target*]

6.4. Managing Disk Usage

Seeing Disk Usage	**df**
	du -s
	**du	sort -r -n**
	ls -l

Making A Tape Archive

tar -cv[f *drive*] *file*...

Extracting Archived Files

tar -xv[f *drive*] *file*...

7. Printing

7.1. The Printer Queue

List the Queue	**lpq**
Removing a Printer Job	**lprm** *job*
Removing Your Printer Jobs	**lprm -**
Selecting a Printer	**lpr -P***printer*
	l -P *rinter*

lprm -P*printer job*

7.2. Printing **troff** Output and Screen Dumps

troff Output	**lpr -t**
Screen Dumps	

screendump [| **rastrepl**] | **lpr -v**