UNIX Command Cheat Sheets

Command	Description (short)	Example	Explanation
date	Writes the current date to the screen	date	Mon Nov 20 18:25:37 EST 2000
sort **infile**	Sorts the contents of the input file in alphabetical order	sort **names**	Sorts the contents of **names** in alphabetical order
who	Tells you who is logged onto your server	who	None
who am I	Tells you your user information	who am i whoami	None
clear	Clears the window and the line buffer	clear	None
echo **whatever I type**	Writes **whatever I type** to the screen.	echo **hey you!**	Writes **hey you!** to the screen
banner **big words**	Does the same thing as echo only in BIG words	banner **hey!**	Writes **hey!** in large letters on the screen
cat **file1 file2 file3**	Shows the three files in consecutive order as one document (can be used to combine files)	cat **cheese milk**	This prints the **cheese** file to the screen first and immediately follows it with the **milk** file.
df **system**	Reports the number of free disk blocks	df ~ df **$HOME**	Both commands will print the total kb space, kb used, kb available, and %used on the home system (your system).
head **file**	Prints the first 10 lines of the file to the screen	head **addresses**	Prints the first 10 lines of **addresses** to the screen
	Number of lines can be modified	head -25 **addresses**	Prints the first 25 lines of **addresses** to the screen
tail **file**	Prints the last 10 lines of the file to the screen	tail **test.txt**	Prints the last 10 lines of **test.txt** to the screen
	Number of lines can be modified here, too	tail -32 **test.txt**	Prints the last 32 lines of **test.txt** to the screen
more **input**	This prints to screen whatever is input—useful because it only shows one screen at a time. *scroll bar* continues to the next screen *return* moves one line forward *Q* quits G goes to the end 1G goes to the beginning Ctrl u moves up ½ screen Ctrl d moves down ½ screen	more **groceries**	This will list the **groceries** file to the screen.

Command	Description (short)	Example	Explanation
ls (-*option*-optional)	Lists all the nonhidden files and directories	ls	Lists all nonhidden files and directories in the current directory
		ls **bin**	Lists all nonhidden files and directories in the **bin** directory
ls -l or ll	Lists all nonhidden files and directories in long format	ls -l ll	Lists all nonhidden files and directories in the current directory in long format
		ls -l **work** ll **work**	Lists all nonhidden files and directories in the **work** directory in long format
ls -a	Lists all files and directories including hidden ones	ls -a	Lists all files and directories, including hidden, in the current directory
		ls -a **temp**	Lists all files and directories in the **temp** directory.
ls -r	Lists all files and directories in reverse alphabetical order	ls -r	Lists all nonhidden files and directories in the current directory in reverse alphabetical order
		ls -r **abc**	Lists all nonhidden files and directories in the **abc** directory in reverse alphabetical order
ls -t	Lists all nonhidden files in the order they were last modified	ls -t	Lists all the nonhidden files in the current directory in the order they were last modified from most recent to last
		ls -t **work**	Lists all the nonhidden files in the **work** directory in the order they were last modified from most recent to last
NOTE: Options can be combined using ls		**ls -al**	**Lists all files (including hidden (-a)) in long format (-l)**

Important Characters			
|	"pipe" directs the output of the first command to the input of another.	ls -l | more	Lists your files in long format one screen at a time
>	Sends the output of a command to a designated file	ls -l > **myfiles**	Prints your listing to a file named **myfiles**
>>	Appends the output of a command to a designated file	ls -l >> **allfiles**	Appends your filenames to the end of the **allfiles** file
&	Runs command in the background; you can still work in the window	xclock &	Runs xclock (a clock) allowing you to keep working
~	Designates the home directory ($HOME)	echo ~	Writes your home directory to the screen
<	Designates input from somewhere other than terminal	progA < **input1**	progA program gets its input from a file named **input1**
Wildcards	UNIX has a set of wildcards that it accepts.		
*	Any string of characters	ls ***.c**	Lists any file or directory (nonhidden) ending with **c**
?	Any one character	ls **file**?	Lists any file/directory with **file** and 1 character at the end
[ ]	Match any character in the brackets (a hyphen is used for ranges of characters)	ls **v**[6-9]**file**	Lists **v6file, v7file, v8file,** and **v9file**

Command	Description (short)	Example	Explanation
cd **directory**	Changes your current directory to the directory specified	cd **bin**	Changes directory to the **bin** directory
		cd .. cd ../..	Moves you to the directory that contains the directory you are currently in Ex. Current directory=/home/users/bob/bin execute cd .. New directory= /home/users/bob **or** executing cd ../.. New directory= /home/users.
		cd -	Moves you to the directory you just came from
		cd ~ cd	Both move you to your home directory (the directory you start from initially)
mkdir **dirname**	Creates a directory You can also designate where the directory is to reside.	mkdir **junk**	Makes a directory named **junk** in your current directory
		mkdir ~/**left**	Makes a directory in your home directory named **left**
rm **file1 file2 file3**	Removes (deletes) file(s)	rm **xyz**	Deletes a file named **xyz**
		rm **xyz abc**	Deletes the files named **xyz** and **abc**
		rm *	Deletes everything nonhidden
rm -i **file1 file2**	Prompts before deletion of files *******USE -i AT FIRST*******	rm -i *	Prompts at each nonhidden file and lets you decide whether or not to delete it
rm -f **file1 file2**	Forces deletion without prompt regardless of permissions	rm -f **program**	Removes the file **program** without regard to permissions, status, etc.
rm -r **directory** rm -R **directory**	Remove a directory along with anything inside of it	rm -r **bin** rm -R **bin**	Each of these will remove the **bin** directory and everything inside of it.
rmdir **directory**	Removes a directory like rm -r does if the directory is empty	rmdir **bin**	Removes the **bin** directory if it is empty
****dangerous**** rm -fR **name** rm -Rf **name**	This combination will force the removal of any file and any directory including anything inside of it	rm -Rf **c_ya**	Forces removal without prompts of the **c_ya** directory and anything inside of it
rm -Ri **directory**	Deletes the contents of a directory and the directory if it is empty by prompting the user before each deletion	rm -Ri **rusure**	Deletes anything in the directory called **rusure** that you verify at the prompt, and if you remove everything in the directory, you will be prompted whether you want to remove the directory itself or not
NOTE: Options can be combined using rm			
rmdir -p **directory**	Removes a directory and any empty parent directories above it (-pi does the same thing but it prompts before each removal)	rmdir -p **/home/bin/dir1**	Deletes the **dir1** directory; if **bin** directory is empty, it is deleted, and if **home** directory is empty it is also deleted

Command	Description (short)	Example	Explanation
cp **file1 newname**	Copies a file (file1) and names the copy the new name (newname)	cp **old new**	Makes a copy of the file/directory named **old** and names the copy **new**, all within the current directory **NOTE**: If you copy a file to a *newfile* name and *newfile* already exists, the *newfile* contents will be overwritten.
		cp **file dir2/**	Places a copy of **file** in **dir2/** and it retains its original name
		cp **../dir1/* .**	Copies everything from the **dir1** directory located just below where you currently are and places the copy "here" (**.**) in your current directory
cp -p **name target**	Preserves all permissions in the original to the target	cp -p **execut1 execut2**	Copies **execut1** executable file and calls the copy **execut2**, which also has executable permissions
cp -R **directory target**	Copies a directory and names the copy the new name (target)	cp -R **old/ junk/**	Makes a copy of the directory named **old** and names the directory copy **junk**
cp -f **name target**	Forces existing pathnames to be destroyed before copying the file	none	No example or description needed
mv **initial final**	Renames files and directories	mv **temp script_1**	Renames the file (or directory) **temp** to the name **script_1** in the current directory
	Also moves files to other directories	mv **script.exe ~/bin**	Moves the **script.exe** file to the **bin** directory that is in the home (**~**) parent directory *and* it keeps its initial name
	You can do multiple moves.	mv **script_1 script.exe ~/bin**	Moves both **script_1** and **script.exe** to the **bin** directory
pwd	Prints the current directory to the screen	pwd	May print something like "/home/bob"
pr (*option*) **filename**	Prints the specified file to the default printer (*options are not required but can be combined in any order*)	pr **userlist**	Prints the contents of **userlist** to the default printer
pr +k **filename**	Starts printing with page k	pr +5 **userlist**	Prints the contents of **userlist** starting with page 5
pr -k **filename**	Prints in k columns	pr -2 **userlist**	Prints the contents of **userlist** in 2 columns
pr -a **filename**	Prints in multicolumns across the page (use with -k)	pr -3a **userlist**1	Prints **userlist** in three columns across the page
pr -d **filename**	Prints in double space format	pr -d **userlist**	Prints **userlist** with double space format
pr -h "header" **filename**	Prints the file with a specified header rather than the filename	pr -h "users" **userlist**	Prints **userlist** with *users* as the header
NOTE: Options can be combined using pr			

Command	Description (short)	Example	Explanation
lpconfig **printer_id queue**	Configures remote printers to a local print queue	lpconfig **prntr1 bobprt**	Configures a printer named **prntr1** to accept print requests from a local queue named **bobprt**
lpconfig -r **queue**	Removes the said queue from the local system	lpconfig -r **bobprt**	Removes **bobprt** queue from the local system *if* the person removing the queue is the owner or "root"
lpconfig -d **queue**	Makes the said queue the default queue	lpconfig -d **vpprnt**	Makes **vpprnt** the default print queue
lpstat (-*options*)	Prints printer status information to screen (*options not required*)	lpstat	Prints status of all requests made to the default printer by the current server
lpstat -u"**user1, user2**"	Prints the status of requests made by the specified users	lpstat -u"**bob**"	Prints status of all requests made by the user with the id **bob**
lpstat s	Prints the queues and the printers they print to	none	None
lpstat -t	Shows all print status information	none	None
lpstat -d	Shows the default printer for the lp command	none	None
lpstat -r	Lets you know if the line printer scheduler is running	none	None
lp (-*option*) **file**(**s**)	Like pr, this prints designated files on the connected printer(s) (*options not required and options may be combined*).	lp **junkfile**	Prints the file **junkfile** to the default printer in default one-sided, single-sided, single-spaced format
lp -d*dest* **file**(**s**)	Prints the file(s) to a specific destination	lp -dbobsq **zoom**	Sends the file **zoom** to the *bobsq* print queue to print
lp -n*number* **file**(**s**)	Allows user to designate the number of copies to be printed	lp -n5 **crash**	Prints five copies of **crash** in default settings
lp -t*title* **file**(**s**)	Places *title* on the banner page	lp -t*Bobs* **cash**	Prints *Bobs* on the banner page of the file printout named **cash**
lp -o*option* **file**(**s**)	Allows printer-specific options to be used (i.e., double-sided or two pages per side, etc.)	lp -od **output**	Prints the **output** file double-sided on the printout
		lp -obold **output**	Prints **output** in bold print
		lp -ohalf **output**	Divides the paper into two halves for printing **output**
		lp -oquarter **output**	Prints four pages of **output** per side of paper
		lp -olandscape **output**	Prints **output** in landscape orientation
		lp -oportrait **output**	Prints **output** in portrait orientation
NOTE: Options can be combined using lp			
cancel **request_id**	Stops print jobs or removes them from the queue (**request_ids** are obtained using lpstat)	cancel **5438**	Stops the print job with the id **5438** whether it is printing or if it is sitting in the queue
cancel -a **printer**	Removes all print requests from the current user on the specified printer	cancel -a **bobsprt**	Removes all the requests from the current user to the printer named **bobsprt**
cancel -u **login_id**	Removes any print requests queued belonging to the user	cancel -u **bob**	Cancels all queued print requests for user **bob**

Command	Description (short)	Example	Explanation
ps	Shows certain information about active processes associated with the current terminal	ps	Shows a listing of process IDs, terminal identifier, cumulative execution time, and command name
ps -e	Shows information about *all* processes	ps -e	Shows a listing of process IDs, terminal identifiers, cumulative execution time, and command names for all processes
ps -f	Shows a *full* listing of information about the processes listed	ps -f	Shows UID (user or owner of the process), PID (process ID--use this number to kill it), PPID (process ID of the parent source), C (processor utilization for scheduling), STIME (start time of the process), TTY (controlling terminal for the process), TIME (cumulative time the process has run), and COMMAND (the command that started the process)
ps -u **user_id**	Shows all processes that are owned by the person with the pertinent user_id	ps -u **bob**	Shows all the processes that belong to the person with the userid **bob**
ps -ef	Shows all processes in a full listing	ps -ef	Shows all current processes in full listing
kill **process_id**	Stops the process with the said **id**	kill **6969**	Kills the process with PID **6969**
kill -9 **process_id**	Destroys the process with the said **id**	kill -9 **6969**	PID # **6969** doesn't have a chance here.
grep **string file**	Searches input file(s) for specified string and prints the line with matches	grep **mike letter**	Searches for the string **mike** in the file named **letter** and prints any line with **mike** in it to the screen
grep -c **string file**	Searches and prints only the number of matches to the screen	grep -c **hayes bankletter**	Searches the file **bankletter** for the string **hayes** and prints the number of matches to the screen
grep -i **string file**	Searches without regard to letter case	grep -i **hi file1**	Searches **file1** for **hi**, **Hi**, **hI**, and **HI** and prints all matches to the screen
grep -n **string file**	Prints to the screen preceded by the line number	grep -n **abc alpha**	Searches **alpha** for **abc** and prints the matches' lines and line numbers to the screen
grep -v **string file**	All lines that do not match are printed	grep -v **lead pencils**	Prints all lines in **pencils** that *do not* contain the string **lead**
grep -x **string file**	Only exact matches are printed	grep -x **time meetings**	Prints only lines in **meetings** that match **time** exactly
	grep is useful when you use it in a | "pipe"	ps -ef | grep **bob**	Finds all processes in full listing and then prints only the ones that match the string **bob** to the screen
	You can also redirect its output to a file.	grep -i **jan b_days**>**mymonth**	Searches the file **b_days** for case-insensitive matches to **jan** and places the matching lines into a file called **mymonth**

Command	Description (short)	Example	Explanation
vuepad **filename**	Opens **filename** for editing/viewing in the vuepad editor	none	None
vi **filename**	Text editor that exists on every UNIX system in the world	none	None
emacs **filename**	Another text editor	none	None
compress **filename** uncompress **filename**	Compresses the file to save disk space. Expands a compressed file	none none	None None
awk	UNIX programming language	none	None
eval `resize`	Tells the target computer that you've resized the window during telnet	none	None
chexp # **filename**	Keeps the file(s) from expiring (being erased) on the target computer for # days	chexp 365 **nr***	Keeps the target computer from deleting all files starting with **nr** for 1 year (365 days)
		chexp 4095 **nr***	Makes all files whose name starts with **nr** never expire or be deleted (infinite)
qstat	Displays the status of a process that has been submitted the Network Queuing System (basically a batch job)	qstat	Shows the status of the requests submitted by the invoker of the command—this will print request-name, request-id, the owner, relative request priority, and request state (is it running yet?)
		qstat -a	Shows all requests
		qstat -l	Shows requests in long format
		qstat -m	Shows requests in medium-length format
		qstat -u **bob**	Shows only requests belonging to the user **bob**
		qstat -x	Queue header is shown in an extended format
xterm xterm -option xterm +option	Opens a new window (x-terminal) for you to work -option sets the option +option resets the option to default	xterm	This opens another window like the one you are currently working in. **USING XTERM WILL ELIMINATE A LOT OF DESKTOP CLUTTER. I STRONGLY SUGGEST YOU LEARN TO USE IT IN YOUR SCRIPTS.**
xterm -help	Displays the xterm options	xterm -help	Shows the options available

Command	Description (short)	Example	(Explanation)
xterm -e **program**	Executes the listed program in the new xterm window—when the program is finished, the new xterm window goes away	xterm -e **myprog.exe**	This opens an xterm window and executes the program **myprog.exe** from that window so that you may still work in your present window.
xterm -sb	Opens an xterm that saves a set number of lines when they go off the top of the page and makes them accessible with a scroll bar	xterm -sb	Puts a scroll bar on the right side of the page for reviewing past lines in the window NOTE: When clicking in the scroll bar, the left button scrolls down, the right scrolls up, and the middle snaps the scroll bar to the mouse position for dragging up and down.
xterm -sl **number**	Specifies the **number** of lines to be saved once they go off the top of the screen (default is 64)	xterm -sl **1000**	The xterm will save **1,000** lines of work once it has moved off the immediate viewing area; it can be accessed using the scroll bar.
xterm -geom **x**x**y**+**px**+**py**	This option allows you to specify the size **x pixels** by **y pixels** and placement **position x** by **position y** of the new window when it opens. Position +0+0 is the top left-hand corner of the screen, and the bottom right is approx. +1200+1000 depending on your resolution. Note: The size of the window takes precedence over position, so if you position it too close to the side of the screen, it will position at the edge with the correct size.	xterm -geom **80x80+0+50**	The first command will open a window **80** pixels wide by **80** pixels tall and position its top left-hand corner at **0** pixels to the right of the left edge and **50** pixels down from the top of the screen.
		xterm -geom **10x35+300+500** xterm -geom **5x5+0+0**	The second command will open a window **10** pixs wide by **35** pixs tall and position its top left-hand corner **300** pixs from the left edge and **500** pixs down from the top. The third command will make a **5** by **5** window and position its top left-hand corner at the top left-hand corner of the screen. **xterm will not compromise size when positioning.**
xterm -title **label**	Allows you to label your window's top title bar	xterm -title **SCRIPTS**	Opens an xterm window with the title **SCRIPTS** (default is whatever follows the -e option)

xterm -(areas) **color**	Allows you to modify different colors in your xterm window	xterm -bg **white** xterm -bd **huntergreen** xterm -fg **red**	The first command sets the background color to **white.** The second command sets the window border color to **huntergreen**. The third command window sets the text color to **red**.
xterm -fn **font**	Sets the font in the new xterm window	xterm -fn **courr18**	Sets the font to **courr18** (default is *fixed*)
xterm -iconic	Starts the new xterm as an icon (double-click to maximize)	xterm -iconic -title **xyz**	Opens an xterm in iconic form with the title **xyz**
NOTE: Options can be combined using xterm			