

Unix Sampler

PEOPLE	
<code>whoami</code> <code>id</code>	Learn about yourself.
<code>who</code>	See who is logged on
<code>finger username</code>	Find out about the person who has an account called <code>username</code> on this host things like whether they are currently logged on to this host or not. Try substituting <code>username</code> with your account name, or the user name of one of your classmates.
<code>hostname</code>	Learn out about the host computer you're logged on to.
<code>grep pattern /etc/passwd</code>	<p>Search for <code>pattern</code> in the password file.</p> <p>Everyone with a user account on this host has a line in the <code>/etc/passwd</code> file. This command is useful if you know part of someone's real or account name and want to know more. Try substituting <code>pattern</code> with your account name.</p> <p>The colon-separated line includes the user name, placeholder for an encrypted password, unique user id, default group id, name, full pathname home directory, default shell.</p>
<code>su - username</code>	<p><i>Don't bother to type this in, but we include it FYI.</i></p> <p>If you know the password for another user account, you can change identities to be that person by using the superuser command. Warning: don't tell anyone your password. This command is typically used by people that have a shared project account or by a system administrator when changing identities to become the superuser known as "root" (<code>su - root</code>), who has permission to do anything and everything.</p>
THE UNIX SHELL	
<i>What is a "shell"?</i>	<p>A shell is a unix command line interpreter. It reads commands that a user types in at a terminal window and attempts to execute them.</p> <p>There are many different shell programs. You are given a default one with your user account, but you can change it. One of the first if not the first shell was called "sh". Others include csh (the C shell), ksh (the Korn shell), tcsh, zsh, and bash (the "born again" shell which is based on sh).</p>
<code>ps</code>	<code>ps</code> means "process status". This command reports what programs you are currently running. What shell are you running?
THE UNIX MANUAL	
<code>man</code>	<p>Use the <code>man</code> (manual) command to look up a command's syntax and usage. Understanding <code>man</code> pages takes practice but it's an invaluable skill if you work on a unix platform. Examples:</p> <ul style="list-style-type: none">• <code>man ps</code>• <code>man bash</code>• <code>man cp</code>

MOVING AROUND A UNIX FILE SYSTEM

<code>What is the "unix file system"?</code>	<p>It's an n-ary tree where all the files (including special files such as directories (folders), programs, and device drivers are kept).</p> <p>Here is a picture of a typical unix file system</p>
<code>pwd</code>	<p>pwd means "print working directory". It prints the <i>full path name</i> (starting the root) of your current directory.</p>
<code>cd</code> <code>full_or_relative_pathname</code>	<p>cd means "change directory"; it's used to change your location in the file system.</p> <p>A full path name is the complete path to a file, starting at the root directory with forward slashes as separators. Sample full pathnames:</p> <ul style="list-style-type: none">• /• /etc/passwd• /home4/b/bob• /usr/bin/dos2unix <p>A relative pathname is a path to a file starting at the current directory, with forward slashes as separators. A relative pathname never begins with a forward slash "/". Shell metacharacters "." and ".." mean current directory and parent directory, respectively. Sample relative pathnames:</p> <ul style="list-style-type: none">• ./hws/hw01• hws/hw01 (<i>same as pathname above</i>)• ../userB/pub• ../../projectX/docs
<code>cd</code> <code>cd ~</code>	<p>With no argument, the cd command will "bring you home" - back to your login directory, no matter where you are currently located in the file system.</p> <p>Because the tilde "~" is a shell metacharacter that means "home directory" "cd ~" will also bring you home.</p>

VIEWING, COPYING, MOVING, RENAMING, AND REMOVING FILES

<code>cat file</code>	Print contents of file to screen
<code>more file</code>	Print contents of file to screen one page at a time. Similar command: less
<code>ls directory/file</code>	List contents of a directory (current directory if no argument).: <code>ls</code> <code>ls -l</code> <code>ls -la</code>
<code>cp file(s) path</code>	cp means copy. The file(s) are copied to path.
<code>mv file(s) path</code>	mv means move. The file(s) are moved to path. Also used to rename a file
<code>rm file(s)</code>	rm means remove. It removes file(s). To remove directories, use the -r option (for a recursive remove).
<code>rmdir file</code>	rmdir means "remove directory". It will remove the directory <i>file</i> only if it is empty.

SHELL METACHARACTERS FOR MATCHING FILE NAMES

*	Matches any number of characters. Examples: <ul style="list-style-type: none"> • cp *.java Backup • rm *.class • rm -i *~
?	Matches any one character. Examples: <ul style="list-style-type: none"> • ls resume?.pdf
[characters]	Matches one character listed in square brackets. Examples: <ul style="list-style-type: none"> • ls resume[1-9].pdf • rm resume[ABC].pdf • ls resume[1-9A-Z].pdf

PIPES

<code>program1 program2</code> <code>program1 program2 program3</code>	The vertical bar " " is called the pipe symbol. Here the output from program1 becomes the input to program2 and so on.
<code>ls -la more</code>	See the full contents of a large directory one screen full at a time.
<code>cat file.txt java MyClass</code>	Execute the main method of the java class MyClass, feeding it file.txt as standard input.
<code>cat file.txt java MyClass sort more</code>	Execute the main method of the java class MyClass, feeding it file.txt as standard input, sorting the output, and viewing the sorted output one screen full at a time.

REDIRECTING INPUT AND OUTPUT WITH SHELL METACHARACTER >, >>, and <

<code>program > file</code>	The ">" redirects the command output to a file. It will create a new file or overwrite a file having the same name (if permissions/quota system permits).
<code>program >> file</code>	The ">>" appends/adds the command's output to a file. It will create a new file or add to an existing file with the same name as <i>file</i> (if permissions/quota system permits).
<code>program < file</code>	The "<" redirects input. Here the command is run and its input is taken from <i>file</i> .
<code>redirecting "standard error"</code>	It's also possible to redirect error messages from a command or program to file. The mechanism varies from shell to shell.

CREATING FILES AND DIRECTORIES

<code>mkdir directory</code>	mkdir means "make directory". It creates a new directory (if permissions and the host's quota system allows it). <i>directory</i> may be a full or relative pathname.
<code>emacs and vi</code>	Text editors worshipped by rival cults.
<code>touch file</code>	Updates the specified file's last modification time, or creates an empty file if the specified file doesn't already exist. It comes in handy sometimes, e.g. to quickly create some files to experiment with, or to ensure that a source file gets rebuilt when using make.

FINDING FILES

<code>find . -print</code>	Starting with the current directory, recursively print the names of all files beneath it.
----------------------------	---

<code>find ../src - print</code>	Starting with the sibling directory src, recursively print the names of all file beneath it.
<code>find . - print grep pattern</code>	List all files below whose pathname contains pattern.
<code>find . -type d -exec chmod 775 {} \;</code>	Change the mode/permissions of all directories under the current one to rwxrwxr-x.
<code>find . -type f -exec chmod 660 {} \;</code>	Change the mode/permissions of all directories under the current one to rw-rw----.

ARCHIVING COMPRESSING FILES

Create and compress and archive:

<code>tar cvf somename.tar dir1</code>	tar means "tape archive". It was originally used to back up files/directories to a magnetic tape. Although tapes aren't used much anymore, tar is still widely used both for creating an archive (which is a file with potentially lots of files contained inside it that keeps track of file hierarchies) and for simply compressing files to make the process of transporting/downloading them quicker. The options cvf mean create verbosely (print file names as they are added file . Here we're creating a new file called "somename.tar" which is our new archive. A copy of all the files in the directory called dir will be added to the archive. tar recursively visits every file under dir to do this.
<code>tar cvf somename.tar file1 file2 dir2</code>	Here's another example showing that you can give a list of files/directories put in the archive. In the case file1, file2, and all the files in dir2 are added
<code>gzip somename.tar</code>	Usually people compress the archive further by using gzip. This makes the archive smaller and easier to transport. This will rename the file to somename.tar.gz. If gzip doesn't work, try zip.

Decompress and unbundle and archive:

<code>gunzip somename.tar.gz</code>	Unzips the file. Renames the file to somename.tar. If gunzip doesn't work, try unzip.
<code>tar xvf somename.tar</code>	The x option means extract. This command copies all the files in the archive in to the current directory, recreating the exact file hierarchy of the files that were archived (e.g. file1, file2, dir2).
<code>ls -l</code>	Check to see that the files were extracted.
<code>rm somename.tar</code>	Usually you want to remove the archive to free up the space it's taking.

NETWORK CONNECTIVITY

<code>ping eniac.seas.upenn.edu</code> <code>ping www.upenn.edu</code>	Test network connectivity with another host.
---	--

COPYING FILES FROM ONE UNIX MACHINE TO ANOTHER

<code>scp file login@remotehostname:dest</code>	scp means "secure copy". Copies file on the current host to the destination path on remotehost. The example below copies ClassX.java (in the current directory on the unix machine I'm logged on to) in to my eniac account's cse100/hw8 directory.
---	--

		<pre>scp ClassX.java mylogin@eniac.seas.upenn.edu:cse100/hw08</pre>
	<pre>scp login@remotename:file dest</pre>	<p>Copies file on remotehost to current host.</p> <pre>scp mylogin@eniac.seas.upenn.edu:cse100/hw08/ClassX hw08</pre>
REMOTE LOGIN		
	<pre>ssh remotehost</pre>	<p>Use "secure shell" to log on to a remote host. You'll be prompted for a password; the same user name is assumed. Example:</p> <pre>ssh blue.seas.upenn.edu</pre>
	<pre>ssh login@remotehost</pre>	<p>Use "secure shell" to log on to a remote host. You'll be prompted to enter the password for the "otherlogin" account. Example:</p> <pre>ssh otherlogin@blue.seas.upenn.edu</pre>