

UNIX for Mac Users

Guide for K12 Schools

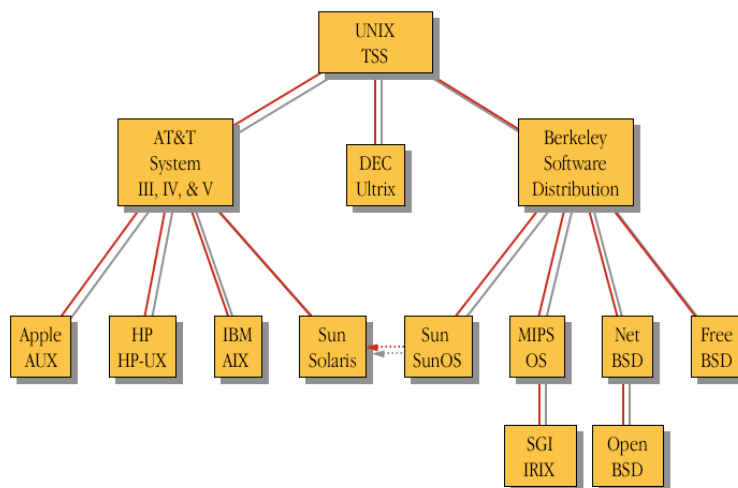
Introduction

As you learn Mac OS X, especially Mac OS X Server, you might realize that knowing a little bit of UNIX might be helpful when doing 'advanced' stuff. This document is aimed at traditional Mac users, who like the author, have little or no background in UNIX. It hopes to give you an orientation that will get you started.

History of UNIX

It is good to understand the roots of UNIX, to see how and why things were done. For example, the processing, RAM, etc. . . was so small, that abbreviated commands made a BIG difference in execution speed. Hence, we have short commands like `mkdir` which creates a folder. Early operating systems used punched cards to enter data, so when UNIX came along with it's command line interface (CLI) which gave real time interaction, it was BIG.

The diagram below shows some of the fragmented history of UNIX. Mac OS X is built on the Free BSD foundation, while our old Apple Network Servers ran IBM AIX, based on AT&T System V.



Further history about UNIX can be found at:

- <http://www.bell-labs.com/history/unix/> or
- http://vertigo.hsrl.rutgers.edu/ug/unix_history.html

Where Do I Start ?

As a traditional Macintosh User, you might be thinking, “where do I start?”

Mac OS X has a UNIX foundation, but for most users they enjoy the benefits of stability, performance and security without needing to know anything about UNIX, or indeed without having to go anywhere near it. For those of us with an interest, or the need to manage a Mac OS X Server, understanding a little bit is very handy indeed.

For example, a little understanding makes it easier to follow the Apple Knowledge Base articles, which sometimes contain a simple command, but could be daunting for a traditional Mac user. This document hopes to make this level of UNIX usage a little less daunting.

Using Terminal

The way you ‘interact’ with UNIX is via the command line interface (CLI), which is accessible by using the utility called ‘Terminal’ which you find in the Utilities folder on every Mac OS X computer. It looks like this:



Terminal has a ‘prompt’ where you can type text commands, and the results (if any) are displayed below. The type of ‘environment’ you are working in, known as the ‘shell’, controls the way the command line behaves. There are many shells available, but the default in Mac OS X is tcsh (sometimes called t-shell or tc-shell). Often UNIX experts have their favorite, but you can simply stick with the default.

It is good to remember the following things when you type commands:

Syntax: is very important, note especially these things:

- commands are case sensitive
- spelling errors can run the wrong command, so check what you type
- spaces can change the meaning on the command or how it executes
- * is a wildcard, replacing any filename. Good for recursive changes.

Spaces: if a file or folder name has a space, you need to add a \ before the space.

eg. to make a new folder called ‘trash me’, use the command `mkdir trash\ me`

Alternately, you can use “quotes”, eg. `mkdir "trash me"`

Man Pages: are the ‘help’ files for commands. `man` is short for manual, and can be typed before any command to show the manual for that command. For example, typing `man ls` displays the `man` page for the `ls` command. Use the return key and space bar to navigate.

Options: often a command will have a option, or switch, to change how the command works. The switch is often a dash, followed by a letter. For example, the `ls` command has a switch `-al`, which changes the way the directory listing is presented.

Running Commands: When you type a command in the shell, it needs to know where to look for the real executable command. The shell will look up in a list of directories called the PATH. If it doesn’t find a match, you will need to tell it EXACTLY where to look.

Security: What commands you are allowed to type, depends who you have logged in as. Some commands are only available to the systems administrator, also known as ‘root’. By typing `sudo` at the start of the command, and entering the root password, you can run a command as root without being logged in as root.

Editing Commands: can be achieved by using the left arrow key to get to the place you wish to delete characters, or where you wish to type new ones. Unfortunately, you cannot use the mouse to help editing. But cut and paste does work !

Beware: some commands can make a horrible mess, even delete your hard disk. Terminal does not have the usual ‘do you want to do that’ warnings, or undo, you find in the GUI interface. If unsure, check man page.

Navigation: finally, to get around your file system, the **cd** command (change directory is used). This command is explained later in the document, but remember:

- **cd *directoryname*** takes you into that directory
- **cd ~** takes you to your home directory
- **cd /** takes you to the root of your hard disk
- **cd ..** takes you back one level

Cool Features of Terminal

Some nice features that you should be aware of are:

Drag & Drop: makes entering long pathnames easy. In the finder, drag a folder or file into the Terminal window and the path to that item will be entered at the prompt.

Multiple Windows: you are not limited to one window. Having more than one window allows you to do things like have a man page open while typing a command in another window, or show a directory listing in one while typing a command in another.

Command Completion: if you hit the tab key mid-command, it will try to complete the command. If there is more than one command with the same beginning, it will give you options.

Automatic History: allows you to use up and down arrow keys to get back to previous commands you have typed, edit them, and run them again.

Transparent Dock: you can change colours, transparency, via ‘Window Settings...’ in the Terminal menu.

A few Commands to Try

Practice is the best way to learn UNIX, so the following commands are a good place to start. These commands give you a chance to explore Terminal and will be useful in the future:

Type **ls** and press return. To see how switches work, type **ls -al**. See how the information is different, in both content and layout? Now type **man ls** to see what the **-a** and **-l** switches do.

Type **cd /** to get you to the root (or top level) of your hard disk. Do an **ls -al** to see what is there. Type **cd ~** to get you to your home directory. Now, do an **ls** to check what directories you have, and now type **cd Documents** to get into your Documents folder (remember the capitals). Try **cd ..** to go back to your home directory. Now, try **cd Doc** and hit tab.

Type **mkdir Trash\ Me** to create a folder in Documents titled “Trash Me”. Now, to enter that new directory, type **cd Trash\ Me**. Ok, type **cd ..** to go back to Documents.

Now, type **MKDIR Trash\ Me** and see what happens. See, case is important. Type **cd /**

Open a Finder window and navigate to your home directory. In Terminal, type **cd** (then before you press return, drag the Documents folder onto the Terminal window to enter its path) and now press return. Press the UP arrow key on the keyboard – cool eh!

Type **cd ~/Documents** to get back there quickly. Do an **ls** to check “Trash Me” is still there. Type **rm Trash\ Me** and press return. Do an **ls** and it should be gone.

Useful Commands

Command	Common Options	Examples	What does it do ?
pwd		pwd	Prints absolute path of current directory
ls	-a, -l, -F	ls -al <i>directoryname</i> ls -alF	Lists files in nominated directory, default is current directory
cd		cd cd ..	Changes to home directory Changes up one directory
chmod	u/g/o , +/- / =, r/w/x, -R	chmod g+rw <i>filename</i>	Changes permissions for owner/user, group, and everyone/other
chown	-R	chown <i>username filename</i>	Changes ownership of the file or directory
chgrp	-R	chgrp <i>groupname filename</i>	Changes the group associated with the file or directory
mv		mv <i>filename1 filename2</i>	Moves or renames files
rm	-R, -f , -i	rm -iRf <i>directoryname</i> rm -f <i>filename</i>	Removes files, or entire directory trees (with -R)
cp	-R, -p	cp <i>filename1 filename2</i> cp -Rp <i>directory1 directory2</i>	Copy files or directories (with -R). -p preserves ownership and permissions.
mkdir		mkdir <i>directoryname</i>	Creates a new directory
who		who	Shows which users are logged in
w		w	Shows users logged on with additional info
ps	-ax	ps ps -ax	Lists current processes running on system
top		top	Info on processes, memory, system load
kill	-HUP (=1) , through to -KILL (=9)	kill -9 <i>processID</i> kill -HUP <i>processID</i>	Sends a kill signal to a process. HUP will restart daemons, and let normal programs shut down. -9 is enforced immediately.
ifconfig	-a , up , down, inet, netmask	ifconfig -a ifconfig <i>interfacename</i> down	Allows display and control of Ethernet interface settings
df	-k	df -k	Shows info on mounted disks
su	-	su su - su <i>username</i>	Changes which user you are at the command line. Defaults to root. "-" means load their customised command line environment
sudo		sudo rm -f <i>filename</i>	Runs the command that follows with root powers. You need to authenticate with your password before it is run.
grep		grep "string" <i>filename</i>	Searches for strings in files
find		find . -name <i>filename</i>	Searches directories for filenames
more		more <i>filename</i>	Displays a file to the screen, one page at a time. Scroll down with space, q to quit.
less		less <i>filename</i>	Displays a file to the screen, one page at a time, arrow keys are used to scroll up/down.
script		script <i>filename</i>	Saves everything you type up until "exit" or Control-D in a file . Default filename is "typescript".
>		ls -al > <i>outputfile</i>	Saves output of a command to a file
>>		pwd >> <i>outputfile</i>	Appends output of command to a file
		ls -al more ps -ax grep <i>processname</i>	Pipes (or plugs) commands together, so the output of one is used as the input of the next one.
printenv		printenv	Displays details of your current environment
man		man <i>command</i>	Get manual page information on a command

Why is Terminal so Good ?

There are a number of reasons why people think the Terminal is so powerful. Firstly, sometimes there is no GUI way of doing something, so it is good having a command line to use. Secondly, and more importantly, it is efficient for certain kinds of tasks.

For example: Imagine using Workgroup Manager to make a particular folder 'Read Only' in 500 home directories? How long would this take? Hours? With the Terminal, it can be done with one command in less than a minute.

A command like `chmod o-w /yourpathname/*/Documents/Trash\ Me/` is very powerful. Look at `man chmod` to understand this command.

Getting Help

There are a number of people who can help you with Mac OS X Server:

Your Apple Education Reseller
Apple Professional Services
Apple Certified Training: Mac OS X & Mac OS X Server
<http://www.apple.com.au/training/>

Resources

A variety of websites, white papers and tools that will help:

Learning UNIX for Mac OS X – a fantastic book designed to teach UNIX basics to traditional Macintosh users:
<http://www.oreilly.com/catalog/linxmacosx2/>
Learning the Terminal Tutorial
<http://www.oreillynet.com/pub/ct/51>
Terminal Basics – a guide, which can be downloaded at:
<http://homepage.mac.com/rgriff/>
Mac OS X CLI Tutorial
<http://www.macosx.org/clitutorial.html>
Mac OS X UNIX Webpage
<http://www.apple.com/macosx/jaguar/unix.html>
UNIX Development on Mac OS X
<http://developer.apple.com/unix/index.html>
Mac OS X Admin Guide – version 10.2.3 can be downloaded at:
http://www.apple.com/server/pdfs/MacOSXServer_AdminGuide_121902.pdf
Mac OS X Support Webpage
<http://www.info.apple.com/usen/macosx/>
Mac OS X Server List
<http://lists.apple.com/mailman/listinfo/macos-x-server>