**UNIX**

**A Brief History of UNIX**

by Sam Coniglio  
Contract Technical Writer  
Email: [spaceman@mindspring.com](mailto:spaceman@mindspring.com)   
September 07, 1999

<http://www.wordesign.com/unix/coniglio.htm>

In the beginning, there was AT&T.

Bell Labs’ Ken Thompson developed UNIX in 1969 so he could play games on a scavenged DEC PDP-7. With the help of Dennis Ritchie, the inventor of the “C” programing language, Ken rewrote UNIX entirely in “C” so that it could be used on different computers. In 1974, the OS was licensed to universities for educational purposes. Over the years, hundreds of people added and improved upon the system, and it spread into the commercial world. Dozens of different UNIX “flavors” appeared, each with unique qualities, yet still having enough similarities to the original AT&T version. All of the “flavors” were based on either AT&T’s System V or Berkeley System Distribution (BSD) UNIX, or a hybrid of both. During the late 1980’s there were several of commercial implementations of UNIX:

* Apple Computer’s A/UX
* AT&T’s System V Release 3
* Digital Equipment Corporation’s Ultrix and OSF/1 (renamed to DEC UNIX)
* Hewlett Packard’s HP-UX
* IBM’s AIX
* Lynx’s Real-Time UNIX
* NeXT’s NeXTStep
* Santa Cruz Operation’s SCO UNIX
* Silicon Graphics’ IRIX
* SUN Microsystems’ SUN OS and Solaris

... and dozens more.  
  
The Open Standards Foundation is a UNIX industry organization designed to keep the various UNIX flavors working together. They created operating systems guidelines called POSIX to encourage inter-operability of applications from one flavor of UNIX to another. Portability of applications to different gave UNIX a distinct advantage over its mainframe competition.

Then came the GUIs. Apple’s Macintosh operating system and Microsoft’s Windows operating environment simplified computing tasks, and made computers more appealing to a larger number of users. UNIX wizards enjoyed the power of the command line interface, but acknowledged the difficult learning curve for new users. The Athena Project at MIT developed the X Windows Graphical User Interface for UNIX computers. Also known as the X11 environment, corporations developed their own “flavors” of the UNIX GUIs based on X11. Eventually, a GUI standard called Motif was generally accepted by the corporations and [academia](http://www.wordesign.com/unix/coniglio.htm).

During the late 1990’s Microsoft’s Windows NT operating system started encroaching into traditional UNIX businesses such as banking and high-end graphics. Although not as reliable as UNIX, NT became popular because of the lower learning curve and its similarities to Windows 95 and 98. Many traditional UNIX companies, such as DEC and Silicon Graphics, abandoned their OS for NT. Others, such as SUN, focused their efforts on niche markets, such as the Internet.

Linus Torvalds had a dream. He wanted to create the coolest operating system in the world that was free for anyone to use and modify. Based on an obscure UNIX flavor called MINIX, Linus took the source code and created his own flavor, called Linux. Using the power of the Internet, he distributed copies of his OS all over the world, and fellow programmers improved upon his work. In 1999, with a dozen versions of the OS and many GUIs to choose from, Linux is causing a UNIX revival. Knowing that people are used to the Windows tools, Linux developers are making applications that combine the best of Windows with the best of UNIX.

**A Brief OS Comparison**

It would be impossible to create a [complete](http://www.wordesign.com/unix/coniglio.htm) comparison of commands without creating a book, so here are a few examples. I compare some common UNIX commands with MS-DOS, MS Windows and [Apple Macintosh](http://www.wordesign.com/unix/coniglio.htm). All of the OSs have similar functions, but implement them slightly differently.

**UNIX and MS-DOS**

At first glance, the UNIX command line looks a lot like MS-DOS command line. In fact, some of the commands are the same. Then subtle differences appear. The DOS prompt looks like C:\ while the UNIX prompt looks like “$” or “#” The slashes are different: DOS uses the backslash “\” while UNIX uses the forward slash “/ ”. Finally, for each UNIX command, there are a ton of switches to customize the output of the command.

|  |  |  |
| --- | --- | --- |
| **To ...** | **UNIX** | **MS-DOS** |
| display list of files | ls OR ls -l | dir/w dir |
| display contents of file | cat | type |
| display file with pauses | more | type <filename> | more |
| copy file | cp | copy |
| find string in file | grep OR fgrep | find |
| compare files | diff | comp |
| rename file | mv | rename OR ren |
| delete file | rm | erase OR del |
| delete directory | rmdir | rmdir OR rd |
| change file protection | chmod | attrib |
| create directory | mkdir | mkdir OR md |
| change working directory | cd | chdir OR cd |
| get help | man OR apropos | help |
| display date and time | date | date, time |
| display free disk space | df | chkdsk |
| print file | lpr | print |
| display print queue | lpq | print |
| kill a crashed program | kill |  |

**UNIX and Windows**

With the advent of Microsoft Windows, many of the DOS command line functions became abstract. Document icons represent files and folder icons represent directories. But Windows is still glued to its command line past. Just click on the “Start” button, select the “Run” command, type **cmd** in the text field, click the OK button, and poof!! Good old DOS command line appears in a new window.

|  |  |  |
| --- | --- | --- |
| **To ...** | **UNIX** | **Microsoft Windows** |
| display list of files | ls OR ls -l | Double-click on My Computer, or activate [Windows Explorer](http://www.wordesign.com/unix/coniglio.htm). In View menu choose “Details” |
| display contents of file | cat | Double-click on an icon |
| display file with pauses | more | Click on the scroll arrows |
| copy file | cp | Select Icon, choose Copy from Edit menu, or from right mouse menu |
| find string in file | grep OR fgrep | Not in OS, but third party tools available |
| compare files | diff | Not in OS but third party products available |
| rename file | mv | Click on file’s name, or choose Rename from right mouse menu |
| delete file | rm | Selec t icon and press Delete key |
| delete directory | rmdir | Select icon and press Delete key |
| change file protection | chmod | Select icon, choose Properties from right mouse menu, click on Read-only checkbox |
| create directory | mkdir | Choose New->Folder from the File menu |
| change working directory | cd | Double-click on a folder icon |
| get help | man OR apropos | Select Help menu |
| display date and time | date | Date and time displayed on task bar |
| display free disk space | df | Click on disk icon, and capacity pie chart appears (Win98 Explorer mode) or space available displays in info bar (Windows Explorer) |
| print file | lpr | Ctrl-p (most programs) |
| display print queue | lpq | Double-click on printer icon on task bar |
| kill a crashed program | kill | Ctrl-Alt-Del, select [application](http://www.wordesign.com/unix/coniglio.htm) from list, click End Task button |

**UNIX and Macintosh**

Apple’s Macintosh operating system was designed from the ground up as graphical OS. There are no command line shells like MS Windows. This make life easy for people new to computers, but it can be frustrating for DOS and UNIX experts. With the upcoming MacOS X, a Macintosh/UNIX hybrid, Apple may satisfy both new [computer](http://www.wordesign.com/unix/coniglio.htm) users and command line wizards.

|  |  |  |
| --- | --- | --- |
| **To ...** | **UNIX** | **Macintosh** |
| display list of files | ls OR ls -l | Double-click on a folder or disk icon. In View menu choose “as List” |
| display contents of file | cat | Double-click on an icon |
| display file with pauses | more | Click on the scroll arrows |
| copy file | cp | Select Icon and press command-D |
| find string in file | grep OR fgrep | Command-F activates “Sherlock” tool, select “Find by Content” tab |
| compare files | diff | Not in OS but third party products available |
| rename file | mv | Click on file’s name |
| delete file | rm | Click and drag icon to trash, or command-delete |
| delete directory | rmdir | Click and drag icon to trash, or command-delete |
| change file protection | chmod | Get Info on a file, click on lock button |
| create directory | mkdir | Command-N |
| change working directory | cd | Double-click on a folder icon |
| get help | man OR apropos | Select Help menu, or command-? |
| display date and time | date | Date and time displayed on menu bar |
| display free disk space | df | Free space listed on top of each window |
| print file | lpr | Command-P |
| display print queue | lpq | Double-click on printer icon on desktop |
| kill a crashed program | kill | Command-Option-Esc |

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

Link: http://www.wilsonmar.com/1opsys.htm

Dec 17 at 9:22 AM

MS-DOS dominated the IBM PC compatible market between 1981 and 1995, or until about 2000 including the partially MS-DOS-based Microsoft Windows (95, 98, and Millennium Edition).

**MS-DOS UNIX\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

CD CD

> (Default Prompt) $(Default Prompt)

CLS (Clear the Screen) CLEAR (Clear the Screen)

VER (What Version of OS or DOS uname -r (What Version of OS or Kernel

am I using? )

am I using? )

echo $SHELL {What directory is the shell located? (e.g., /bin/bash)}

**Unix File Permissions**

If you create a UNIX shell script and run it, you will get a "Permission denied" message if you don't explicitly assign execute permissions with a command such as:

chmod +x myscript.sh

Code Permission Octal Decimal

r-- Read only 0100 4

-w- Write only 0010 2

--x eXecute only 0001 1

rwx Read, Write, eXecute 0111 7

In Unix systems, 3 permissions can be granted: read, write, and execute (abbreviated as r, w, and x).

Each permission is stored as one binary bit within a 4-bit number.

Unix commands specify all three permissions at once using the decimal equivalent of the 4 bit "octal" number, so called because one can count to 8 using 4 binary bits.

u

user g

group o

other

ls -l - rwx r-x r--

chmod binary 4+2+1 4+0+1 4+0+0

chmod octal 7 5 4

umask binary 0+0+0 0+2+0 0+2+1

umask octal 0 2 3

command ls -l (with the -long option) lists the 3 permissions for each category of ownership (user, group, other) along with the name of the owner and the KB of its password file.

An octal number is used to store the permission of each category of ownership.

Umask filename without argument echoes current permissions settings as an octal number.

chmod 754 filename sets permissions using "7" for user, "5" for group, "4" for other.

umask 023 filename (user mask) cause all new files to be created as if "chmod 754" were executed after their creation.

chmod ugo+rwx filename sets permissions for all category of ownership.

"-" instead of "+" removes the permissions.

chmod a+r filename sets permissions for all users.

In Solaris, to modify the default group membership of a file:

chgrp group file