Open Computing ‘‘Hands-On’’ ‘‘PC-Unix Connection’’ Column: October 94

Making the Leap to DOS

Moving from Unix to DOS/Windows? Don’t panic--here’s a three-part survival guide

By Tom Yager

You’re hurt. You’re bitter. You just found out you have to switch from using a Unix system, your first love, to a (gasp!) PC. Maybe you’re a corporate user and your workstation application has migrated to Windows. Or perhaps you’re a developer, and your new project has you treading on this unfamiliar ground.

Don’t sweat it; the transition isn’t that hard. I’ve made it, and in fact I now jump effortlessly between Unix and DOS-based systems, both as a user and a developer. If you’re not looking forward to the switch, or you’ve made it, but you’re still a bit befuddled, consider this article your too-brief Unix user’s guide to DOS. My next column will discuss using Windows, and the one after that, the shift from a developer’s perspective.

Jurassic DOS

If you spend much time in Unix gossip circles, particularly on the Usenet, your opinion of PCs and of Microsoft’s operating-system hegemony is your biggest obstacle. It’s true, DOS doesn’t have Unix’s command-line power. But it’s not nearly as useless as popular Unix rhetoric would have you believe. Still, it does have its snake pits. Once you know where they are, it’s easier to get around.

The first thing that’s liable to drive you nuts is the way DOS deals with files. Names are too short, limited to eight characters, a period, and a three-character ‘‘extension.’’ You can create file names without extensions, but you’ll have to stick to eight characters. A period is mandatory between the eighth and ninth characters.

Only a few file extensions are meaningful under DOS. The .exe and .com extensions signify binary executable files. A .sys file holds a device driver or, in the case of config.sys, the system-configuration script. The .bat extension signifies a batch file, the inadequate DOS answer to the shell script.

Most DOS commands will not accept multiple file name arguments on the command line. You are, however, permitted to use wildcards. These follow a subset of the wildcarding rules of Unix: only ‘‘*’’ and ‘‘?’’ are used.

DOS has no security. This absence has several ramifications, but on the subject of files, it means there is no protection. Anyone with access to your machine can get at every file. The only file attributes are ‘‘read-only,’’ ‘‘archive,’’ and ‘‘hidden.’’ The archive bit is used by some backup utilities to determine which files have changed since the last backup. The hidden attribute hides a file from a normal directory listing. A simple switch added to the dir command (see the table) lists hidden files.

The DOS file system structure does not share Unix's single point of origin. There are multiple root directories--signified by backslash (backslash, just to annoy you)--attached to lettered drive specifiers. The letters are A to Z, with some conventions governing their use. A and B (or, in the more common DOS notation, A: and B:) point at your system's floppy drives. If you have one floppy, it will be A:, and B: will be an unmapped drive letter. Your system's primary hard drive is C:. Your boot sector, command shell, and boot-time scripts must be located here; the system won't go hunting for them.

DOS has the ability to split a drive into multiple partitions. This ability is partly a throwback to the days when you had to split up large DOS drives to get at all their space. Current DOS versions will let you cover a one-gigabyte drive with a single partition, if it suits you. Using a partition-per-drive scheme simplifies storage management, giving each physical drive its own letter.

Not every drive letter points to a physical drive. Your PC may load, at boot time, device drivers that masquerade as DOS drives. That's how DOS accommodates CD-ROM and most types of networks. The device driver generally takes the next available drive letter, but you can often specify which letter you want to use. Fixing all CD-ROM drives in the systems you use to F:, for example, makes desk-hopping easier.

Memory Muddle

The most vile DOS snake pit of all is memory management. DOS runs in the 80x86 processor's real mode, with access to a maximum of one megabyte of memory. Newer DOS programs run in protected mode, able to get at all of your PC's memory, but they often require a fairly large real-mode memory space for their primary code. What does that mean? You might gleefully load up your system with network drivers, a CD-ROM drive, and a little program that pops up to remind you of appointments. Each of these takes away from available program memory, and DOS has no virtual memory. When you try to launch that beastly DOS application, you might get a ''not enough memory'' message.

If you already have enough total memory to run the program, adding more won't help you. What you lack is sufficient memory below one megabyte. There are two ways to get it: you can start removing drivers and loadable programs (like your pop-up calendar) until your application runs again. Or you can ask newer versions of DOS (like the current MS-DOS 6.2) to fiddle with your CPU's memory-management hardware to move memory blocks around. It can sometimes load drivers into memory above one megabyte but fool DOS into thinking the memory is really below that magical mark.

Tuning memory yourself is a black art as dark as hacking sendmail rules. Try doing it automatically--DOS 6.x, for example, has a memmaker utility that sniffs around and attempts to reconfigure your system for maximum DOS program memory.

The last-resort approach, one that only recently became available, is the DOS 6.x notion of menu-selectable boot sequences. You can create one boot sequence that loads all your favorite drivers but doesn't leave enough room for that big DOS program to run. Then create another configuration that leaves some of the nonessential drivers out and optimizes your system for running that special program. DOS will put up a menu at boot time and let you choose from among two or more such configurations. The menu times out to a default in case you're not there to make a selection.

Gotta Have It

DOS gives you a pretty sparse command environment. Most savvy DOS users don't rely on the standard commands alone; there are a few essentials that you shouldn't let yourself be without. The first is DOS's most popular compression utility, pkzip, a shareware utility that compresses the daylights out of files or even whole directory trees. Its options will make Unix users smile with recognition--if anything, pkzip is overloaded with features, including password protection. Everybody who uses DOS has pkzip, and you can easily find the latest version (2.04g) on bulletin board systems, pay services, and on Internet archives.

You might find the lack of a good shell language too limiting. DOS's batch facility has made advancements, taking on conditional branching and some limited user-input support, but it's still no match for the Korn shell. Don't fret--ksh and all your other favorite Unix commands have been ported, and very well, to DOS. Grab yourself a copy of the MKS Toolkit for DOS. Fully configured, your machine will boot straight into a log-in prompt. You can even send and receive e-mail and news through UUCP. It's not Unix, but MKS did such a masterful job of masking the differences between Unix and their DOS look-alike command set that you'll be quite comfortable.

The thousands of freeware and shareware programs available for DOS seem irresistible, but they may carry viruses with them. Unlike the complex and imaginative programming required to infect a well-administered Unix system, a DOS machine is extremely vulnerable. To be absolutely safe, you shouldn't load any noncommercial program onto your system until you've completely backed up your hard drive. The DOS 6.x virus-scanning utility will scan freshly loaded programs for hundreds of known viruses. If it finds one, it can usually disarm it and repair any damage. If you don't discover the damage until later, often the only safe option is to restore your system from the most recent uninfected backup.

DOS is hobbled by a file system even more fragile than the old System V layout. Lose power in the middle of a file write operation, and all hell could break loose. This problem is yet another reason for frequent backups. DOS doesn't usually know when its file system is busted--you have to ask. The chkdsk utility sifts through the file system looking for various forms of floating debris, and it's a good idea to have this program run automatically at boot time. Otherwise, a simple file system error can compound, extending damage to several files.

It's likely that you'll feel the pinch for a while, but after you adjust, you should find DOS a reasonably affable working environment. If you get lonely for Unix, there's always the MKS Toolkit, and failing that, maybe the coworker who inherited the Unix machine you used to have will let you telnet into it for old times' sake.

Tom Yager is an Open Computing contributing editor. He can be reached through the Internet at tyager@maxx.net.

Table: DOS equivalents of Unix commands

Unix	*DOS*
ls -l	dir
ls -x	dir /w
more filename	more < filename
fsck	chkdsk or scandisk

cp	copy or xcopy
rn	rename
rm	delete
mv	copy then delete
tar vf dev file...	pkzip -a archive file...

Edited by Becca Thomas / Online Editor / UnixWorld Online / beccat@wcmh.com

Go To Content   Search

HTML markup by Tim Cooley
Last Modified: Wednesday, 23-Aug-95 17:26:17 PDT