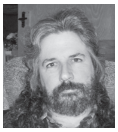
/dev/random
What is UNIX?

ROBERT G. FERRELL



Robert G. Ferrell is a fourth-generation Texan, literary techno-geek, and finalist for the 2011 Robert Benchley Society Humor Writing Award. rgferrell@gmail.com

I stumbled across this piece I wrote as a writing sample in 1997 in a forgotten directory on a seldom-accessed backup drive. I can't remember what I was auditioning for; whatever it was, I probably triggered some sort of mental aberration in the editor and she had to go to a sanitarium or monster truck rally. That happens with my writing on a fairly regular basis. Whatever the case, now that this vintage nonsense has had 16 years to ferment, I thought I'd pop it open and give y'all a hearty slurp. Remember: 1997.

* * * * *

I encounter an uncomfortably large number of people who, for one reason or another, want desperately to convince some unsuspecting slice of the population that they are technically knowledgeable, if not downright expert, at All Things UNIX. Most of them, however, have neither the aptitude nor the patience to endure the years of dedication it takes to get this sort of experience on their own. It is a bewildering fact that virtually all of these hapless souls somehow end up in my office seeking advice, training, or consolation (usually in that order). So that I might save my vocal cords untold wear and get in at least a minimal amount of work for the agency that pays my salary, I have come up with a short guide for all those who want to speak UNIX without actually knowing anything about it. A little time invested memorizing this and you can add at least $25 an hour onto your consulting fee. That will help to offset the psychiatric bills.

Q: What, exactly, is UNIX?

A: UNIX is an operating system that has been around since 1970, almost as long as that shrimp cocktail in the back of your fridge. When you push down those little buttons on the keyboard, magic pixies carry the scan code for each button to the keyboard controller fairy, who puts them all in an envelope, licks it, and shoots it up to the processor in one of those vacuum-powered hamster tube things, only digital. The processor steams open the envelope and arranges the scan codes like building blocks, creating a surprisingly realistic model of the Taj Mahal before reluctantly knocking it over and getting on with the business at hand. The scan codes are instructions to the program currently running in the foreground, which may be GUI or just slightly sticky.

Q: How does UNIX work?

A: UNIX runs as a series of processes. These processes can be started at boot time or later on by a user or another process. When one process starts a second one it is said to **fork**. The result of forking is, not surprisingly, a **child**, and the original is then a **parent**, which should be apparent. If the parent process dies, the child is orphaned and Social Services has to be called in. Processes that are supposed to be controlled by other processes but aren't are defunct and become eligible for a government bailout. Each process has an identifying number, the PID. The parent of a child process has another number, called a PPID. To understand why this is, you need a PhD. Users have a number, too, known as a UID (which should never under any circumstances be confused with an IUD). The most *useful* number for you to know is probably the one for the help desk.

Q: How about UNIX and the Internet?

A: UNIX and the Internet are like two socks in a shoe. While all other operating systems have to be clever and sweet talk their way onto the Internet, UNIX just strolls right in without even breaking a sweat. When UNIX systems communicate with one another over the Internet, they talk TCP/IP or UDP, through little revolving doors called ports. There are thousands of ports available, but a lot of them have been reserved for visiting diplomats or taken over by applications no one except other software developers and family members will ever use because the hinges squeak. This is called progress, and it gets people into trouble. Ports are identified by simple decimal numbers, such as 25. This is surprising and a little misleading, since virtually everything else about UNIX is in hex, octal, binary, or worse.

Q: What the heck are daemons, anyway?

A: There are special processes in UNIX that slink around in the background listening for requests for service. These are called daemons, because they can turn on you when you least expect it. Some of them peek out through those ports we just talked about, waiting for the odd packet to stray too close and slurp it up like a frog snatching a dragonfly. These processes are controlled by the all-powerful, all-knowing INETD, without which ARPANET would have been just (tremendously expensive and highly classified) cans with strings running between them. INETD is really a whole suite of listening programs started at the same time, including TCP/IP, FTP, UUCP, Telnet, RCP, and more. As a result, if your INETD is DOA, you can't even send out an SOS. LOL.

I could go on ad infinitum, as many of my friends will readily attest, but in the interests of brevity I will now turn to a list of the essential terminology you absolutely must be able to bandy about to impress and, if necessary, confuse your clients.

cat: Reads a file and prints it to the screen, or combines files, or appends to a file. Likes milk. Opposite of dog().

cc: Compile a C program. Or copy a message to someone. CC is a cool command to toss casually into conversation because it has so darn many options, like -dalign, -fnonstd, -qp, -W[p02abl], -xsbfast, and -xstrconst.

chmod: Make files inaccessible or render them non-executable. Then magically fix them for your awestruck and deeply grateful clients. Example: *chmod 000 | find / * -name -print*

cmp: This one is good just to impress people with its voluminous and highly cryptic output. Example: *cmp -l /etc/disktab /usr/adm/messages*.

cof2elf: I don't want to explain what this one does. I just like the way it sounds, like something you might hear in Elrond's infirmary.

cpio: Archives files onto or off of a disk. Use in conjunction with r2d2. Also has a plethora of options. Example: *cpio -i bBcCdEfHIkmMrRsStuvV*. Gesundheit.

crontab: Run a command or shell script at a set time. Not to be taken internally.

crypt: Bury a file so deep in gobbledygook that its meaning can't ever get out. A favorite of legislators and instruction book authors worldwide.

df: See how many devices are attached to the system, and how much can be deleted from them.

du: Report how much disk space is being wasted by useless fluff such as /vmUNIX.

egrep: An old version of *grep*. Or a long-legged water bird if your mouf is foo ob peanub bubba.

find: A career-enhancing command with more options than Georgia has peaches. A well-written find command can approach a Perl script for unreadability:

```
find . \? -mtime -4 | xargs grep [0o]bfuscate -o -prune -perm 444
-exec lp {} \;
```

ftp: Share those unwanted files and programs. The Internet equivalent of dumping stuff out on the curb.

grep: Feel around in a file for something. Not legal in all states. Be careful you don't prick your finger and get AIDS.

head: Print the first ten lines or so. A useful command, mostly for its puerile suggestive value.

hostid: Spits back a mysterious-looking hexadecimal number for no apparent reason.

kill: The ultimate aggressive sysadmin tool. Looks bad, is worse.

ln: A great way to confuse any file system hopelessly. Example: *ln -f pwd kill -9 1*.

nohup: Disallow current or former drill instructors from logging onto the system.

od: A command that is as descriptive as it is functional. Example: *od -bv kernel.o. > /etc/inittab*.

pack: A good, well-rounded command. Example: *pack -f -* *.

pg: A great way to ensure that important material is read. Find a nice, long text file and then try *pg -r -1* file.

red: Print files with all characters far to the left of center (warning: process obsoletes itself after a while).

rksh: The most versatile and useful shell. Add the line SHELL=/bin/rksh to /.profile right now.

stty: One of the all-time great UNIX commands. Virtually any combination of options and modes is a veritable work of poetic `art`. Example: *stty -a cstopb parodd -ixoff -olcuc flusho stappl ctrl-char dsusp cooked -dtrxoff x cibrg r setctbrg*.

talk: A clever little utility that might get you committed. Example: *talk | whoami*.

tar: Rolls up your files into a small, viscous black ball for storage. Surprisingly, it is not GUI.

timex: A version of **rollx** that uses a plainer font and a lot fewer system resources.

touch: Update and otherwise control files. Leave yourself open to various lawsuits and possible criminal indictment in the process. See **grep**.

tee: Balance the ball on a little wooden pegleg. Then hit that sucker hard. Yelling "fore!" optional.

truss: Trace system parameters during program execution. Primarily for customer support, especially of the lower abdomen and pelvic region. Example: _/usr/bin/hernia > truss_.

uuglist: Stops the system clock abruptly.

vc: Destroys the target file, then disappears into the operating system before you can kill it.

wall: A collection of bricks. Or Just Another Perl Hacker.

wc: A little chickadee that really doesn't do much of anything. Example: _wc | find / * -print_.

who: Checks your system for owls.

yacc: Checks your system for yaks.

xargs: Checks your system for, um, zargs.