LINUX VOICE
TUTORIAL

UNIX, LINUX AND HOW WE GOT WHERE WE ARE TODAY

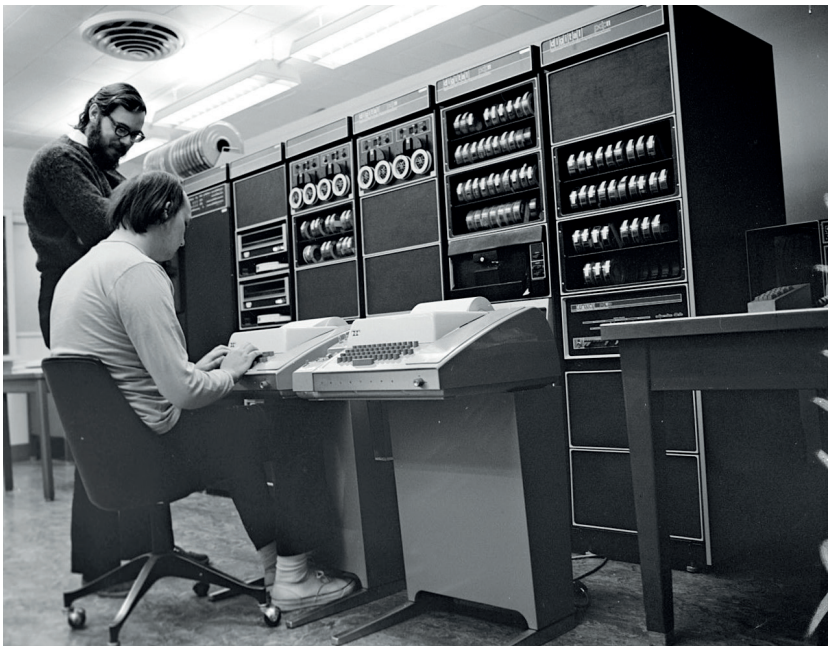
JULIET KEMP

Linux didn't just come from thin air you know. Take a look back at how we got from the 1970s to the OS we know and love today.

Unix is the oldest operating system that's still widely used today. Linux -- which, if you're reading this, you're presumably interested in -- is just one of the many Unix clones and descendants that are kicking around the computer world, alongside UNIX (the trademark is all-caps) itself. Unix and Unix-like systems have always been popular as servers, with a rather smaller population of 'users' alongside them; but with the rise of the smartphone, nearly a billion people worldwide now have a Unix-type box in their pocket. Pretty good for a 45-year-old Bell Labs side project!

Bell Labs: starting out

Back in 1968, Bell Labs was involved with a shared project called Multics, an early time-sharing operating system for the GE-645 mainframe. However, the project, while functional for those working on it, wasn't producing the widely-available OS that Bell was after, and they pulled out. The last of the Bell people working on it (Ken Thompson, Dennis Ritchie, Doug McIlroy and Joe Ossanna) were keen not to lose their own access to interactive computing, so spent 1969 partly trying to persuade management to buy them a computer, and partly developing what would become the Unix filesystem. In his spare time, Thompson rewrote a game called *Space Travel* (a sort of solar system simulator where the player also piloted a ship), together with a bunch of supporting packages, to run on a spare PDP-7 that was kicking around.



Ken Thompson (sitting) and Dennis Ritchie at a PDP-11. Possibly even working on C at the time!

Preparing programs for the PDP-7 was complicated, requiring them to be created on a GE 635 and the paper tapes carried by hand to the PDP-7. So Thompson began implementing a full operating system on the PDP-7: filesystem, processes, small utilities, and a simple shell. It was 1970 when Kernighan suggested calling the new system Unics or Unix – a play on Multics.

The filesystem developed on that first machine had i-nodes, directories, and device files, just like modern Unixes, but there were no path names (they were substituted by a complicated linking system). The system had processes, too, but they were very limited, with no forking or waiting. A fascinating 1979 paper by Dennis Ritchie (available online from Bell Labs – **http://cm.bell-labs.com/who/dmr/hist.html**) goes into detail about the various calls and processes.

In 1970–1 the system was rewritten for a new PDP-11, together with a text editor and formatter (*roff*). The machine began offering a text-processing service to the Patent department, and three typists from that department came to use it. This was an important part of demonstrating that Unix was genuinely useful, even if it got a little in the way of the programmers!

It was also in 1971 that Ken Thompson and Dennis Ritchie began working on the C programming language. Thompson had already developed a language called B, but although some general systems programs were written in it, the operating system basics were still in assembler (see page 106 for more on this minimalist way of coding). Once C was developed, it was used to rewrite the kernel into C in 1973 – the first time that an operating system had been written in anything other than assembler. This also let them demonstrate just how genuinely useful C was (and continues to be: it's still under the hood of plenty of programs).

In the mid-70s, UNIX began to be shipped out under licence, with its full source code included. It was versioned according to editions of the user manual, so Fifth Edition UNIX and UNIX Version 5 are the same. By 1978, when UNIX/32V was released for the new VAX system, over 600 machines were running some variety of UNIX, and various people (such as the BSD folk) were adapting it. At this point, an ongoing antitrust case still prevented AT&T from releasing a commercial product. When this was finally resolved in

1983, they released a commercial licence version of UNIX System V. Since the licence conditions weren't great for university use, BSD became more popular. (And, indeed, the BSD networking code made it back into the main Unix kernel.) Various other companies also developed their own versions of UNIX under licence; which turned in due course into the "Unix wars" where different companies tried to promote rival standards. POSIX was eventually the most successful, designed to be easy to implement on both BSD and System V.

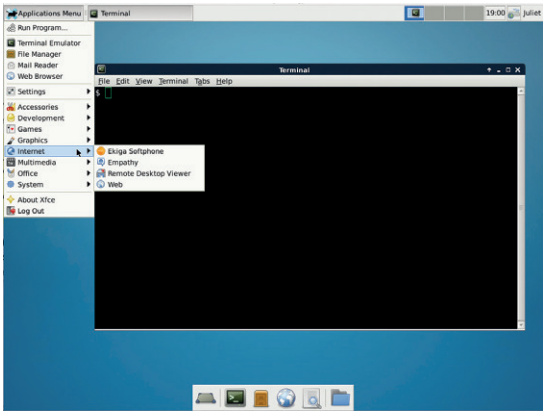
AT&T sold UNIX to Novell in the late 1980s, then (after UnixWare did badly) Novell transferred it to the X/Open Consortion, which now sets UNIX specification standards. Some parts of the licensing business were also sold to SCO, which in due course led to the SCO/Linux legal action (see boxout).

There are five commercial UNIX-certified OSes still available: OS X, HP-UX, Solaris, Inspur K-UX (used on mainframes), and AIX. Unfortunately you can't try out HP-UX or IBM's AIX without going through HP/IBM (or a partner) and spending a large sum of money; and for Inspur K-UX you need a mainframe. But there's more below on OS X and Solaris.

BSD Unix

The University of California, Berkeley, had a Unix Version 4 system running in 1974, and when Ken Thompson was there in 1975 he helped install Version 6. As more people, and other universities, became interested in the system, Bill Joy, a Berkeley grad student, started creating the Berkeley Software Distribution. This was an add-on to Version 6 Unix which included a Pascal compiler and the *ex* line editor (written by Joy). This was possible because Unix was still being released with full source code at the time. The second release, 2BSD, in 1979, included the text editor *Vi* and the C shell *csh*, both still available on Unix systems. (I wrote this article in *Vim*, an extended version of *Vi*, which dates back to 1991.)

BSD became increasingly popular as it improved – it was the OS of choice for VAX minicomputers (used for timesharing) at the start of the 80s. It included



I couldn't get Gnome running on FreeBSD but Xfce worked fine – the apps in the menu are probably from the abortive Gnome install though.

GNU tools

Here's just a few of the GNU tools you may be using:

- The Bash shell.
- The coreutils package, which provides ls, mv, rm, cat, and so on.
- The boot loader Grub.
- The sysutils package, which provides utilities to manage users and groups.
- tar and gzip.
- grep for searching through text files.
- make and autotools for building software.
- glibc, the C library, underlies a huge range of user software. You may never use it directly, but it's essential to your system.
- The GNU Compiler Collection compiles languages including C, C++, and Java.
- The graphics program Gimp.
- The Gnome desktop project (although this is effectively a separate entity now, it is officially a GNU project).
- The venerable text editor GNU Emacs… as well as scientific software, desktop software, internet software, and a whole plethora of development tools.

delivermail (a *sendmail* precursor) and the *curses* library, among other useful bits of software.

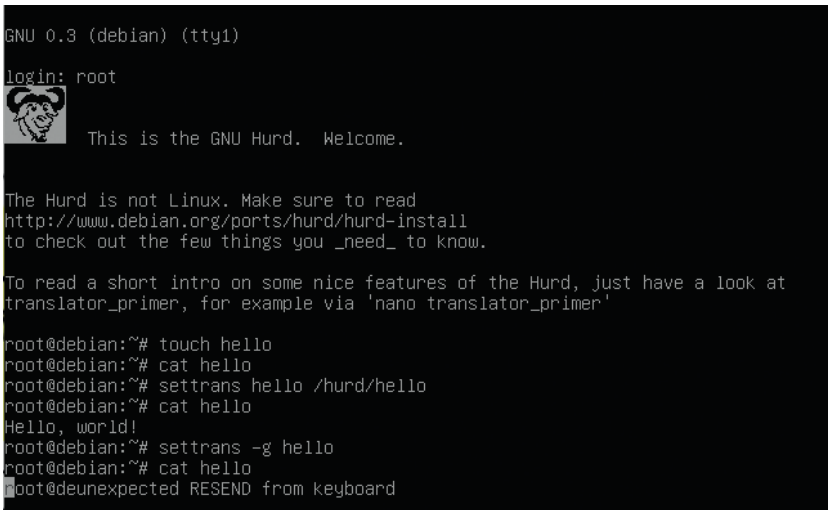
In 1989, BSD released its networking code separately, under the BSD licence. Prior to this, all BSD releases included AT&T Unix code, so post-1983 had required an (expensive) AT&T licence. The networking code had been developed entirely outside this, and various people were interested in acquiring it separately. The general BSD distribution was continuing to improve, and in 1990 the BSD team decided to rewrite all the AT&T-dependent code, resulting in Networking Release 2 in 1991, a freely available OS that was the basis for ports to Intel 80386 architecture and which would later become NetBSD and FreeBSD.

Networking is probably the BSD team's most important contribution to computing. Berkeley sockets, the first Internet Protocol libraries available for Unix, became the standard internet interface. The POSIX API is basically Berkeley with a few changes, so all modern OSes have some implementation of the Berkeley interface.

Unfortunately, the then-owners of the Unix copyright sued in 1992, and while the lawsuit was settled in 1994 largely in the favour of BSD (only three of the 18,000 files had to be removed and a handful more modified), development slowed massively during those two years. As it happens, this was while Linux was being developed. The slow release of 386BSD was part of what prompted Torvalds to create the Linux kernel.

The modern operating systems of FreeBSD, OpenBSD, and NetBSD are all descendants of the 386 port and of 4.4BSD-Lite. They in their turn have various descendants, including SunOS and Mac OS X. Most of these are open source and available under the BSD Licence. *Sendmail*, *Vi*, *curses*, and *csh* are a few of the BSD programs and utilities still in use today.

Of the currently available BSDs, FreeBSD is probably the most friendly to the average non-developer user (though they all have good points, and NetBSD has the distinction that you can install it on a toaster). You can download FreeBSD from **https://www.freebsd.org/where.html**, which also has good user documentation. FreeBSD's install is text-based and will be familiar if you've installed Debian in text mode.



The Hurd running on *VirtualBox*. No graphical desktop! (Though X is supported.) The 'translator' trial from the README is shown.

LV PRO TIP

The Free Software Foundation argue that as GNU software makes up a significant part of a 'Linux' system (more than the kernel does in many systems), it should be referred to as GNU/Linux. This isn't reflected in mainstream usage.

The basic install is deliberately very sparse; afterwards you'll need to install any packages you want. The binary package management system is *pkg*, so to get the Gnome desktop I logged into the new system as root, then typed:

```
$ /usr/sbin/pkg    # this bootstraps pkg itself
$ pkg install xorg
$ pkg install xfce
$ echo "exec /usr/local/bin/startxfce4" > /home/juliet/.xinitrc
```

pkg search name is a useful command to find other available packages, and the documentation and other support for FreeBSD seems good. However, don't expect to log on immediately into a fully-featured desktop system; it expects you to decide the details of what you install for yourself.

BSD to SunOS to Solaris

In 1982, Bill Joy, one of the main BSD developers, joined three Stanford graduates to found Sun Microsystems. Their first generation of workstations and servers were based around a design created by Andy Bechtolsheim (co-founder of Sun Microsystems) while still studying at Stanford. The very first software was Sun UNIX 0.7, based on UniSoft Unix v7, but a year later, SunOS 1.0, based on 4.1BSD, was released.

SunOS continued to be based on BSD until the final update on SunOS4 in 1994. One of their major developments was the creation, in 1984, of the NFS (Network File System) protocol, allowing client computers to access files (largely) transparently over a network. NFS is an open standard so can be implemented by anyone. It's still used in modern networks, especially Unix and Unix-like ones.

In the late 1980s, AT&T and Sun began a joint project to merge BSD, System V, and Xenix, resulting in Unix System V Release 4 (SVR4). In 1991, Sun replaced SunOS4 with Solaris, based on SVR4 instead of on BSD. (So, still a Unix derivative, just with a different parent.) Solaris included OpenWindows (a GUI) and Open Network Computing. SunOS (current release SunOS5.11) still exists as the core of Solaris (current release Solaris 11.2), but the Solaris brand is used externally.

Solaris has always been heavily associated with Sun's own SPARC hardware, but it's also used on i86pc machines worldwide, and is supported by several of the major server manufacturers including Dell, IBM and Intel. Linux distros are also available for SPARC and i86pc hardware. Since 2007, Sun has also supported the open source OpenSolaris project, although it is now known as Solaris 11 Express.

Solaris 11 is free (but not freely licenced) to download for personal use. I tried out the live CD, which was very slow to download, but once there, booted fine on a 64-bit virtual machine.

The basic terminal commands of Solaris 11 are the same as in Linux, and you can find further documentation on the Oracle website. The differences between Solaris and Linux become (in my experience) more noticeable as you delve further into the guts of the system; the average desktop user may not notice anything beyond the difference in package availability.

NeXTSTEP/Mac OS/Darwin

NeXT was founded by Steve Jobs after he was pushed out of Apple in 1985. They developed NeXTSTEP, an object-oriented, multitasking OS to run on their workstations, based on Unix (including some BSD code) and various other bits and pieces. Tim Berners-Lee developed the first browser, *WorldWideWeb*, on a NeXT cube running NeXTSTEP; *Doom* and *Quake* were also developed on NeXT machines. In 1993, OPENSTEP was created by Sun and NeXT; before Apple decided to use it as the basis of what would become Mac OS X, and bought out NeXT. (Jobs, of course, returned to Apple along with his company.)

Mac OS X is based on the XNU (X is Not Unix) kernel, developed for NeXTSTEP, with all the usual Unix commands and utilities available on the command line. The kernel has code from FreeBSD along with other improvements and changes. It's POSIX compliant, which means that many BSD/Linux/Unix packages can be recompiled for OS X with a bit of work (as with HomeBrew, Fink, and other similar projects). The core of OS X is released as the open-source Darwin. iOS is also based on OS X, and Android (Linux-based) and iOS between them have a 90% share of the smartphone market. So it's very likely that your smartphone runs a Unix-like OS, giving Unix today an unprecedented userbase.

The GNU Project

Richard Stallman started the GNU Project in 1983, with the aim of creating "a sufficient body of free software [...] to get along without any software that is not free". GNU stands for 'GNU's Not Unix': the proposed GNU operating system was Unix-like, but Unix was proprietary and GNU was to be free.

The first piece of software released by the GNU project was *GNU Emacs* (an implementation of the existing *Emacs* text editor). They had a debugger, parser, and linker; they also needed a free C compiler

SCO/Linux lawsuit

Various bits of Unix were sold on to Novell in 1993, which then sold parts of it again to what became SCO. In 2003, SCO filed a lawsuit against IBM for $1 billion (later $5 billion), claiming that IBM had transferred SCO property into Linux. Another four major lawsuits followed.

SCO's right to be identified as the 'owner' of UNIX was challenged by Novell, so SCO sued them too. Assorted legal wranglings followed. SCO also claimed that some UNIX code had been transferred line-for-line into Linux, but seemed reluctant to specify the details.

In 2010, after several court rulings and a jury trial, Novell was found to be the owner of the UNIX copyright, and has announced that "We don't believe there is Unix in Linux". As of December 2014, SCO's case against IBM for 'devaluing' its version of UNIX remains open, though now with a reduced scope.

and tools. By 1987 they had an assembler, nearly the *GCC* C compiler, *GNU Emacs*, and a bunch of utilities, together with an initial kernel. Although the rest of the software development carried on at a decent pace, by 1992 they had all the major components except the kernel. The GNU Hurd kernel started development in 1990, based on the Mach microkernel, but for various reasons moved very slowly (it is still not ready for production environments, although the existence of Linux has doubtless slowed development).

You can try out the GNU/Hurd project courtesy of Debian. Note that it is not yet complete and it's not recommended for production use. If you just want to give it a quick go, you can get a virtual image thus, and run it on *KVM*:

```
$ wget http://ftp.debian-ports.org/debian-cd/hurd-i386/current/README.txt
$ tar zxf debian-hurd.img.tar.gz
$ kvm -no-kvm-irqchip -drive file=debian-hurd*.img,cache=writeback -m 1G
Or on VirtualBox if you convert it to the correct format:
$ VBoxManage convertfromraw debian-hurd*.img debian-hurd.vdi --format vdi
```

(This information from **http://ftp.debian-ports.org/debian-cd/hurd-i386/current/README.txt**; more detailed information available there.)

The Hurd's notion of 'translators' is new to me: a translator basically translates between one sort of data structure and another, for example from disk storage to the traditional filesystem. Check out the GNU Hurd website (**https://www.gnu.org/software/hurd/index.html**) for more information on this and other features of the Hurd. If you want to install the Hurd, the instructions suggest that this is a lot like installing Linux was about 15 years ago when I first tried it out, and requires a fair amount of messing around with text files and configuring by hand. (Ah, nostalgia…) Currently only about 50% of the Debian packages are available for the Hurd.

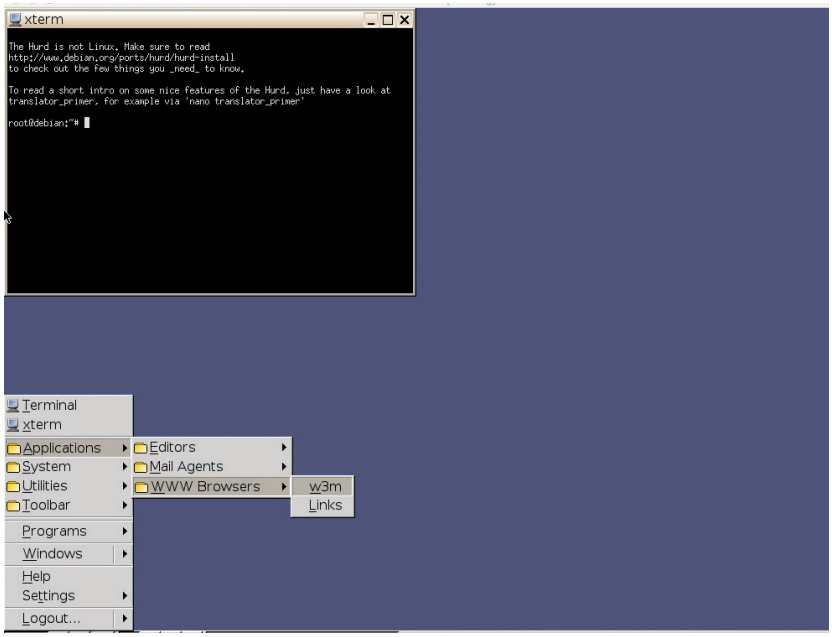
Linux

Finally, we come, of course, to Linux. Linux is not, in fact, an actual Unix variant. It's related to Minix, which was system-call compatible with Seventh Edition Unix but was created from scratch. In 1991, Linus Torvalds was irritated by the lack of a free kernel (GNU Hurd didn't exist and BSD were having problems), so started writing one. He developed it on a Minix system using the GNU C compiler, and was influenced by many Minix design decisions, but there was no actual code overlap (see the boxout for SCO's legal claims). The first release was on 25 August 1991. Unlike Minix's microkernel (a microkernel has as little software as possible in the kernel and moves functions like device drivers and filesystems into userspace), Linux has a monolithic kernel, where all the operating system is in kernel space.

Initially, it was just a kernel distribution, the idea being that you would also get hold of the GNU tools and that would give you a full system. GNU and Linux also had different licences. In Dec 1992 Linux was released under the GNU GPL, which in due course meant the whole thing could be distributed as an integrated system.

From there… well, there are a huge number of Linux distros, you can build your own, and you're reading a whole magazine dedicated to Linux. While it isn't Unix, it's largely Unix-compatible (it adheres to POSIX standards even if uncertified) and broadly speaking, if you know Linux you can find your way around Unix (though as any sysadmin will tell you there are a fair few gotchas in the details of utilities and syntax).

If you're interested in exploring the various Unixes further, try out some of the systems I tried, or one of the many others. For more on Unix, here's a cool (but huge) Unix family tree; there are also links at the bottom of this page – **www.levenez.com/unix**. And here's a Unix timeline (**www.unix.org/what_is_unix/history_timeline.html**).



Here's the Hurd after running **startx** from the console. All very basic by default (no graphical browser here…).

Juliet Kemp is a scary polymath, and is the author of Apress's *Linux System Administration Recipes*.