Nachos

Over the years, numerous projects have been developed for teaching OSs. Many of these projects were motivated by the development of UNIX [Ritchie and Thompson 1974]. At Berkley, they implemented a new teaching OS and simulation environment, called

Nachos

(available from:
`http://www.cs.washington.edu/homes/tom/nachos`)

Nachos makes it possible to write portions of each of the major pieces of a modern OS:

–thread management
–files systems
–multiprogramming
–virtual memory
–networking

Any one can download and install Nachos from the above address. They have ports to:

–DOS/Borland
–Linux
–Solaris (Sun's UNIX based environment)
–RS/6000 (IBM large scale, high performance computer server)
–HP-UX (The Hewlett-Packard UNIX)

Nachos is implemented in a subset of C++. Object-oriented programming is becoming more popular, and C++ is a natural idiom for stressing the importance of modularity and clean interfaces in building OSs.

As a results of the designers emphasis on simplicity, the Nachos OS is about 2500 lines of code, about half of which are devoted to interface description and comments. (The hardware simulator takes up another 2500 lines, but you do not need to understand the workings of the simulator, to design parts of the OS.)

Conclusion:

If you download Nachos, you can learn about the principal OS ideas by adding functionality to the bare-bones Nachos system and improving the performance on benchmarks that are provided. This provides you with a working example of a UNIX-like OS, albeit overly simplistic.

Linux Kernel

Why not take advantage of a great OS whose code is available? Linux is a member of the large family of UNIX-like OSs. A relative newcomer experienced sudden spectacular popularity starting in the late 1990s, Linux joins such well-known commercial Unix OSs as:

–System V Release 4 (SVR4) developed at AT&T (now owned by Novell)
–4.4 BSD release from the University of California at Berkeley
–Digital Unix form Digital Equipment Corporation (now Compaq)
–AIX from IBM
–HP-UX from Hewlett-Packard
–Solaris from Sun Microsystems

Linux was initially developed by Linus Torvalds in 1991 as an operating system for IBM-compatible personal computers based on the Intel 80386 microprocessors. Linus remains deeply involved with improving Linux, keeping it up-to-date with various hardware developments and coordinating the activity of hundreds of Linux developers around the world. Over the years, developers have worked to make Linux available on other architectures, including:

–Alpha
–SPARC
–Motorola MC680x0
–Power PC
–IBM System/390

One of the more appealing benefits to Linux is that it isn't a commercial OS: its source code, under the **GNU Public License**, is open and available to anyone to study. The official site of the Linux kernel is:

`http://www.kernel.org/`

If you download it, you will be able to explore from top to bottom one of the most successful, modern OSs.

There are many **distributions** of Linux. Perhaps the most popular one is RedHat (current version 7.2). If you have a RedHat distribution, you can download the latest version of the kernel from:

`www.redhat.com`

as RPM files:

`kernel-2.4.7-10.src.rpm`
`kernel-source-2.4.7-10.i386.rpm`

Linux source code for all supported architectures is contained in about 4500 C and Assembly files stored in about 270 subdirectories; it consists of about 2 million lines of code, which occupy more than 58Mb of disk space.

What do the kernel numbers mean?

$$\underbrace{2}_{\text{version nr}} . \underbrace{4}_{\text{stable kernel}} . \underbrace{7}_{\text{patchlevel}} - \underbrace{10}_{\text{release}}$$

Thus, the naming convention is:

major.minor.patchlevel

Linus Torvalds maintains the “official” kernel release. By convention, the even-numbered kernel versions (2.0, 2.2, and so on) are “stable” releases, patches that contain only bug fixes and no new features. Odd-numbered kernel versions (2.1, 2.3, and so on) are “development” releases.

New releases are made very rapidly, sometimes daily!

The main reason is that nearly all device drivers are contained within the kernel code, and every time someone updates a driver, a new release is necessary. As the Linux community moves toward loadable device drivers, the maintainers of those drivers can release them independently of the main kernel alleviating the necessity of such rapid updates.

If you want to see the history of kernel releases, go to:

`http://www.memalpha.cx/Linux/Kernel/`

Linux Distributions

Some of the most popular distributions of Linux are:

–Caldera OpenLinux
–Debian GNU/Linux
–Mandrake Linux
–Slackware Linux
–S.u.S.E. Linux

These are only the best known distributions; there is about 160 distributions at the last count!

Read the Distributions-HOWTO, for all the information about the different Linux Distributions.

The different HOWTOs are available from the **Linux Documentation Project** web site:

`http://www.linuxdoc.org/`

We mentioned GNU on the last slide. The GNU project is coordinated by the Free Software Foundation, Inc.:

`http://www.gnu.org/`

its aim is to implement a whole OS freely usable by everyone. The availability of a GNU C compiler has been essential for the success of the Linux project.

Technically speaking, Linux is a true UNIX kernel, although it is not a full UNIX OS, because it does not include all the applications such as filesystem utilities, windowing systems and graphical desktops, system administrator commands, text editors, compilers, and so on. However, since most of these programs are freely available under the GNU General Public License, they can be installed into one of the filesystems supported by Linux.

In order to define a common user interface, UNIX-like kernels often share fundamental design ideas and features. Thus, understanding the Linux Kernel will help to understand other UNIX variants.

All commercial variants of UNIX were derived from either SVR4 or 4.4BSD; all of them tend to agree on some common standards like:

–IEEE’s POSIX (Portable Operating Systems based on UNIX)
–X/Open’s CAE (Common Application Environment)

The 2.2 and up versions of the Linux kernel aim to be compliant with the IEEE POSIX standard. This means that most existing UNIX programs can be compiled and executed on a Linux system with very little effort or even without the need of patches to the source code. Moreover, Linux includes all the features of a moder UNIX OS, like:

–virtual memory
–virtual filesystem
–lightweight processes
–reliable signals
–SVR4 interprocess communication
–support for Symmetric Multiprocessor (SMP)
–etc.

By itself, the Linux Kernel is not very innovative. When Linus Torvalds wrote the first kernel, he referred to some classical books on UNIX internals, like Maurice Bach’s:

The Design of the Unix Operating System

(Prentice Hall, 1986).