

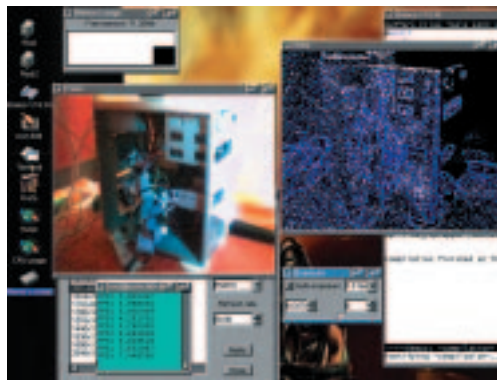
A new GPLed desktop OS

FAITH IN ATHEOS



RICHARD SMEDLEY

As mentioned in the first article in this series, there are many UNIX-like OSs out there, developed for many purposes. Those who wish to move UNIX to the desktop – and avoid the burden of the X Window System – will be curious to look at AtheOS, a GUI-based POSIX-leaning Operating system (OS), with modular kernel and a 64-bit journaled filesystem.



A self portrait of the AtheOS server



AtheOS Web browser in action



AtheOS version of XSpringies

It is often said that most Free Software projects are written to scratch a coder's particular itch. In the case of Atheos creator Kurt Skauen, this was to create a free *desktop* OS. Skauen felt that GNU/Linux and the BSDs, with their mishmash of toolkits (Gtk, Qt, Motif/Lesstif, Xt, fltk), failed on several levels to give the end user a consistent and easy to use environment.

The GUI has been an integral part of Atheos right from the beginning. Originally written, like the kernel, in C – the GUI was reworked in C++ as Skauen learned it, to meet the projects aims of an Object-Oriented API.

Kernel

Of course, the user interface (UI) is important for the end user but it is the kernel and the API that will interest the developer. The OS builds on firm foundations with a pre-emptable, multi-thread kernel coded with SMP in mind, for high-end workstations. Threads are scheduled independently of whether they are inside the kernel, as well as improving SMP performance, this reduces scheduling latency on all machines and results in the GUI feeling extremely responsive (dare one say *Amiga*-like?).

The kernel has been called microkernel-like. It is not a true microkernel, however construction is extremely modular – there are no device drivers built in. At boot time required device drivers are loaded by GRUB. Any driver can be loaded during runtime, including block devices or file systems. In keeping with the AtheOS philosophy drivers are written for a well-defined API, there is no need to learn everything about the kernel before you dive in and have a go.

File under buzzword

There has been coverage of journaled filesystems in this publication and elsewhere as the four

contenders to replace Linux ext2 (ReiserFS, ext3, JFS and XFS) shape up. AtheOS, like BeOS, was designed to do more than this from conception. The 64-bit journalled filesystem supports user attributes (MIME-type specification or a custom data stream – an icon, associated with the file for example). Indexing of file attributes allows extremely fast file look up. Applications can monitor files and directories for changes without any need for polling, thanks to node watching within the kernel.

History

The three BSDs, GNU/Hurd and Linux and other major OSS Operating Systems have developed over time with the input of a large number of people. AtheOS has been largely the spare-time work of one hacker. Kurt Skauen takes pride in preferring coding to talking about how such-and-such feature could be implemented. A clear single vision has produced fairly clean code and good APIs at both the device driver level and for application programming.

Since putting a TCP/IP stack into the kernel and running the *www.atheos.cx* website on AtheOS the project has been picked up by many of the geek sites on the Web. Interest is growing and the project may now be faced with how to deal with conflicting ideas about direction. So far new developers seem happy to follow Skauen's lead. TCP/IP stack is currently undergoing heavy improvements resulting in better connection for those with high latency.

All the code is licensed under the GNU GPL – indeed a complete set of GNU tools are every bit as useful for developing AtheOS as they were 10 years ago for Linux.

No browser conflicts

W3m was a fairly early port (most UNIX command line applications compile with little or no alteration) but as a graphical environment a modern graphical Web browser was needed. Impressed with the clean design of KHTML – the HTML renderer used in Konqueror – Skauen has ported it to AtheOS.

Calls to port KDE (and Qt) fall on deaf ears though. AtheOS with its high level API and OS-implemented GUI features demands the consistency of using its native toolkit. KHTML has been ported to use AtheOS native widgets: Each widget (checkbox, radio button *et al*) either uses a native AtheOS widget directly, or is presented with a Qt-like widget by a thin class written to wrap the AtheOS widget.

KHTML is not multithreaded and a total rewrite would have been difficult. Fortunately a workaround was available and *ABrowse*, the browser that uses KHTML, synchronises all browser windows with the same mutex, giving the impression of a single thread of execution, thus enabling most of the KHTML code to run unchanged.

Multithreading

The AtheOS project has found itself welcoming many former BeOS developers, which has led to some debate on the merits of pervasive multithreading. The problem under BeOS, also famous for its multithreading, was that developers were forced to use multithreading and the BeOS locking mechanism. In itself this wouldn't have been so bad but for various bugs and limitations – including the limited number of threads that an application may spawn. This limit greatly troubled the BeOS port of Mozilla. Another limitation was the messaging system which BeOS uses (*BLoopers*), in particular the limits on message queueing – which would also have been problematic even under a single-threaded OS. Under AtheOS threads can communicate through message ports (most common), shared memory, POSIX signals, semaphores, named and anonymous pipes, ptys and TCP/IP – and probably body language as well.

In AtheOS an application may spawn approximately 16.2 million threads, processes, message ports, global-semaphores, memory-areas or any other object which carries a global ID. For threads the task-switching attributes of intel-compatible CPUs which AtheOS uses limits this further to around 8000. However this could easily be removed.

In real world use, multiple threads may slow down processes but they will leave the User Interface *feeling* more responsive. For a desktop OS this benefits the end user.

AtheOS newbie install

AtheOS is still at an early development stage and lacks a friendly installer – or indeed the ability to read CD-ROM drives. Nevertheless what has been done is impressive enough. For the curious we have included it on the cover CD. Some PCs will not be compatible but if you have a video card from Matrox, nVidia or S3 you will be able to really appreciate the graphical environment. On all the machines I tried I had to use the vesa driver so video was *very* slow. Do I have to remind you to back up anything important before installing an experimental OS onto your hard drive?

Unless you already have an AtheOS File System (AFS) somewhere on your PC, you will need to create a FAT (DOS) partition and copy across the base installation *base-atheos-0.3.xx.tgz* to this partition. Take three blank floppy disks and copy the following files to them:

```
atheos-0.3.x.boot.01
atheos-0.3.x.boot.02
atheos-0.3.x.data.01
```

After booting from the floppies AtheOS drops you into a Bash shell inside a terminal emulator (*aterm*). Typing help will give you the commands available.

Buzzword-compliant

AtheOS boasts a number of desirable features for a geek OS: 64-bit journalled filesystem; kernel with built-in TCP/IP stack and support for SMP; fair degree of POSIX compliance – will run most UNIX CLI tools; client-server GUI protocol, using native messaging system; fine-grained multithreading. If that doesn't tempt you to have a look, nothing will.

If you have yet to prepare a partition to install AtheOS type

```
Diskmanager &
```

at the shell to invoke the AtheOS partitioning tool. If you have a partition set aside you can use *Diskmanager* in order to modify the partition size and change it to AFS the native AtheOS file system.

Now mount the FAT file system where you copied the base install package (in the following case hda0):

```
mkdir dos
bash-2.03$ mount /dev/disk/bios/hda/0 dos
```

note the heirarchical structure of /dev, with disk block drivers under /dev/disk. AtheOS reads disk data from BIOS as this is the only disk driver finished thus far. AtheOS 0.35 is some improvement over 0.34 in this driver and seems better with my (old and flaky) hardware.

Now we come to formatting our AtheOS partition:

```
bash-2.03$ format /dev/disk/bios/hda/1 afs
MyAtheOSPartition
bash-2.03$ mkdir afs
bash-2.03$ mount /dev/disk/bios/hda/1 afs
```

Please make sure that it is definitely hda1 (for example) you want to format before typing the above. Now unpack the base installation

```
bash-2.03$ cd /afs
bash-2.03$ tar -xvpzf /dos/atheos-base
-0.3.xx.tgz
```

You will now find yourself with an */atheos* directory containing the OS and a */boot* directory containing GRUB. We must now configure this bootloader to boot AtheOS. It would be a good idea to play around with GRUB under Linux first if LILO is your normal bootloader, and to make boot floppies for any other OS that you have installed.

Edit the */afs/boot/grub/menu.lst* file – you will find *jed*, an emacs-like editor, at the command line. For the above configuration you need

menu.lst as follows:

```
title AtheOS
root (hd0,1)
kernel /atheos/sys/kernel.so 2
root=/dev/disk/bios/hda/1
module /atheos/sys/drivers/fs/afs
module /atheos/sys/drivers/dev/disk/bios
```

This will tell GRUB to boot partition number 2 on the first drive through the “root (hd0,1)” command and where to find the kernel, the boot block-device driver and the boot-FS driver on that partition.

Save and exit and you are ready to reboot with the floppy disk to install the bootloader. Before GRUB starts loading up AtheOS hit <ESC> and then type “C” to enter the GRUB shell from which you can install GRUB. Type

```
root (hd0,1)
```

in order to tell GRUB where to find the config file and the second-stage loader. To install in the master boot record (mbr) type

```
setup (hd0)
```

to put GRUB on the AtheOS partition. You will need to point your current boot loader there:

```
setup (hd0,1)
```

Congratulations. Now reboot to be welcomed by the login prompt. Default root password is “root” and default user is *guest* (password=“guest”). Change it in *etc/passwd*.

Once up and running installing software is fairly straightforward, the binaries each unpack to their own folders. Emacs is the included editor and ports of Apache, php and Python mean that some will already find this a usable OS.

The idea of a GUI-based UNIX and the amount of work already implemented have generated considerable interest already and lead to a number of third-party applications appearing. Now check out the websites for more information and have fun.

The Future

Atheos is a work in progress. It will be sometime before it is ready for general use, though it might be hitting desktops before the Fresco-based Berlin display server reaches maturity for UNIX.

Nevertheless it is a fascinating project in itself and anyone interested in taking their first steps towards kernel hackerdom would be well advised to start with some much needed device drivers for this OS. At the moment most of the advanced features of Atheos are not fully implemented. If you have an app to write or port over now would be a good time to get involved to grow your project with the OS. One day you may be porting AtheOS apps to Gtk on Linux. ■

Info

AtheOS homepage
AtheOS Developer's mailing list

<http://www.atheos.cx>
<http://lists.sourceforge.net/lists/listinfo/atheos-developer>
<http://www.gnu.org/software/grub/grub.html>

GRUB
Brent Newhall's site hosts third-party applications
Other AtheOS sites

<http://www.kamidake.org>
<http://mnemo.nu/>
www.benews.com, www.osnews.com and
www.slashdot.org

Interviews and reviews of AtheOS