The A/UX Web Server Deployment Guide

A SnakeOil Labs White Paper

By steve at snakeoillabs dot com

Table of Contents

Figures

Tables

“For six long years I've been in trouble
No pleasures here on earth I found
For in this world I'm bound to ramble
I have no friends to help me now.”

–The Soggy Bottom Boys, singing about the release of A/UX 3.0.1.

Licensing and Disclaimer

WARNING!: This guide describes a method for building a functional A/UX 3.0.1 Web Server running Apache 1.3.33. The guide focuses on functionality, not security. It is the author's opinion that the installation will need further fine tuning before a host built following this guide would be sufficiently hardened to place on a live network, let alone the Internet, and accepts no responsibility for any damage caused by doing so.

How to use this guide

Thanks for downloading this guide. If you're one of the four, five, maybe even six people out there on the Internet who want to set up an A/UX web server, then this guide is for you. To make things simpler, this document follows a few standard conventions.

Text in Courier is reserved for terminal sessions. This provides a sample walkthrough of commands to type and their usual responses. For example:

A larger courier font is used to denote relevant commands mid-sentence, such as `newconfig`, in order to separate the command from the rest of the text.

File and path names, such as **/etc/inittab**, are in bold. Individual references to filenames without paths like inittab aren't.

 **Note:**	The famous Mac Bomb symbol is used to denote a tip or note. Small notes like these provide hints and tips as well as insight drawn from bitter experience. Make sure you read them!

One small point; there are many instances where command lines are preceded by either a # or $ symbol. The # refers to the root user and the $ symbol refers to a regular user. In either case, don't type the symbol. To make matters even more confusing, most of the configuration files used in this guide use leading # symbols to define comments. It's confusing but it's the UNIX convention, not ours. Don't worry, you'll get used to it.

Credit Where Credit's Due

This guide simply wouldn't exist without the prior work of many others. In particular, I'd like to thank Scott from aux-penelope for letting me use his screenshots of the installation process and Nic Leymann for his work in porting the GNU tools used in this guide across to A/UX. Finally, I'd like to thank Jim Jagielski, who's FAQ, patches and all round coolness has made this possible. Without the work of these guys, my Quadra would probably be running NetBSD.

Introduction

For many people, Mac OSX is the Operating System of choice; the fusion of traditional Mac OS friendliness with the stability of BSD has been deemed by many commentators and pundits to be revolutionary. This 'revolution' is not unique, nor is it Apple's first attempt. The first version of A/UX was released in 1988, and was available on 55 floppies, a 150Mb tape, or pre-installed on an 80Mb hard drive. The last version of A/UX, 3.1.1, was released in 1995. My first experience with A/UX was in 2004; 9 years later. Needless to say, documentation and help was thin on the ground, so this guide was born.

Before You Start

A/UX is not a toy OS. It requires a lot of care and attention that by modern standards would be seen as insane. The OS itself is fairly stable once patched up and left alone, but was written by an infinite amount of Apple-branded monkeys chained to an infinite amount of Mac Classics. To make matters worse, they probably had to use PageMaker for a large portion of it. Bearing this in mind, there are some things you should be made aware of before you start:

A/UX only runs on certain hardware. A good place to check if yours is supported is http://www.aux-penelope.com/hardware.htm. If your hardware isn't on the list it doesn't mean that A/UX won't run, but don't be surprised if it doesn't. A/UX definitely shouldn't work on LCs, Quadra AVs or anything without an MMU or FPU. In terms of memory and space requirements, an 80Mb Hard Disk should be enough for a minimal install, although you'll need at least 120Mb if you want to do anything with the system. 4Mb of RAM is the official minimum to run A/UX. I'd be uncomfortable with less than 16Mb simply because of the amount of space MacOS 7 and the Finder take up. To put things into perspective, this guide was tested on a Quadra 650 with a 250Mb Hard Disk and 128Mb of RAM – Quite a powerhouse by comparison.

The Unix side of A/UX is based on SVR2.2, with elements of BSD and later SVR versions thrown in for good measure. It implements *some* POSIX extensions. Consequently you may find that your favourite piece of software may not compile cleanly, or at all. This gets worse with A/UX 3.1 or above. On the subject of compiling, A/UX came with a lot of things other UNIX providers charged extra for. Sockets, Streams, A TCP/IP stack, a C compiler and even an X11R5 implementation all come as standard with A/UX 3.0.1. It's not as great as you think. The C compiler is cursed for eternity never to compile any reasonably sized piece of code, the TCP/IP stack falls over and the X11 implementation source can be found in the dead-sea scrolls. Thankfully some nice people managed to port GCC across along with some libraries that make compiling a little easier.

Finally, make sure you've backed everything up beforehand as it's all going to get wiped.

Initial Startup

Before you start, make sure you've backed up your hardware and have the A/UX 3.0.1 Startup Disk and A/UX 3.0.1 Installation CD to hand. Insert the floppy and turn the system on. Insert the Installation CD. Pretty soon you'll see the familiar "Welcome to Macintosh" screen. After a couple of flickers the Apple Workgroup Server 95 splash screen will appear and A/UX will be running on your Mac. If you can't see the screen, then its possible you have hardware incompatibility issues that need to be resolved before you can continue. Assuming it all went well you should see the following screen shown in Figure 1.

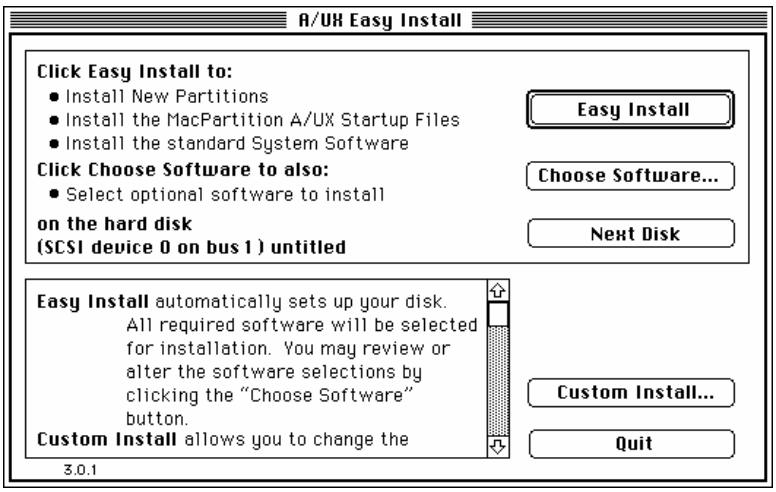

Figure 1: A/UX Easy Install Screen

Resist the urge to take the Easy Install option. It is the quickest way to get up and running but defines a tiny swap partition that is woefully insufficient for anything more than the most basic of tasks. Choose the Custom Install option and you should see something like Figure 2 shown below:

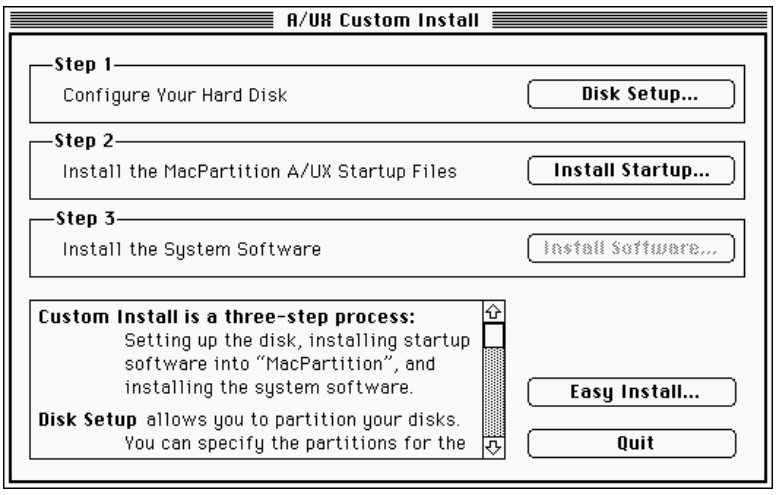

Figure 2: A/UX Custom Install Screen

Partitioning the Hard Drive

The first step in the A/UX 3-step install process is to prepare the hard disk. Click on "Disk Setup…" to launch the Apple HD SC Setup tool. It is advised that new disks are initialised prior to partitioning. Click on Initialise and follow the on-screen dialogs.

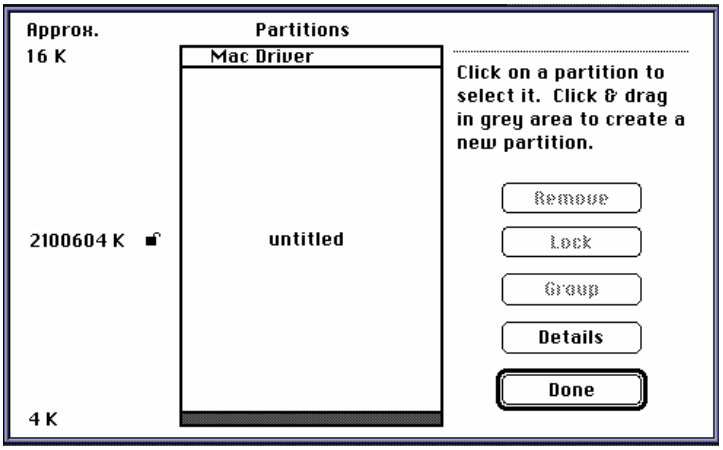

Figure 3: Partitioning a drive

Now it's time to partition the disk. When you click on "Partition…" you'll be taken to a screen with lots of options. Click on "Custom" to be taken to the partitioning screen shown in Figure 3. If you didn't initialise the disk you'll need to erase the existing partitions first. Leave the Mac Driver partition alone as it's used internally for MacOS.

Note:	You don't actually need a MacOS partition to run A/UX and it is possible to boot directly into A/UX, although doing this gets messy quickly. This guide only covers the default booting mechanism requiring a System 7 install.

The minimum reasonable drive size for a modern A/UX web server should be 250Mb. This is mainly because of the amount of swap space required in addition to the regular OS requirements. Table 1 provides a breakdown of the Partitions required.

Mount Point	Type	Description
Mac Driver	Macintosh Volume	Internal 16k Partition required by MacOS.
MacOS Boot Partition	Macintosh Volume	MacOS Boot Partition used to boot A/UX from MacOS. This needs to be at least 5Mb in size.
Swap	UNIX Swap slice 1	Swap partition. Should be set as high as possible. 131070K is preferable, but a minimum of 32768K should be used to avoid performance issues.
/	Unix Root & Usr slice 0	Where A/UX lives. This takes up all remaining available space. Separate partitions can be used for different areas (/usr, /opt etc.) if drive space permits.

Table 1: A/UX Partitioning Guide

To add a new partition, click at the top end of the grey space in the partition table. Click on Macintosh Volume and set the size to 5192. Click on OK to return to the partition management screen. Do the same for other partitions following the guidelines in Table 1. Figure 4 provides an example layout for a 2Gb drive. Once you're happy with the partition layout, click "Done" to return to the HD SC Setup tool. Click "Done" again to return to the main menu.

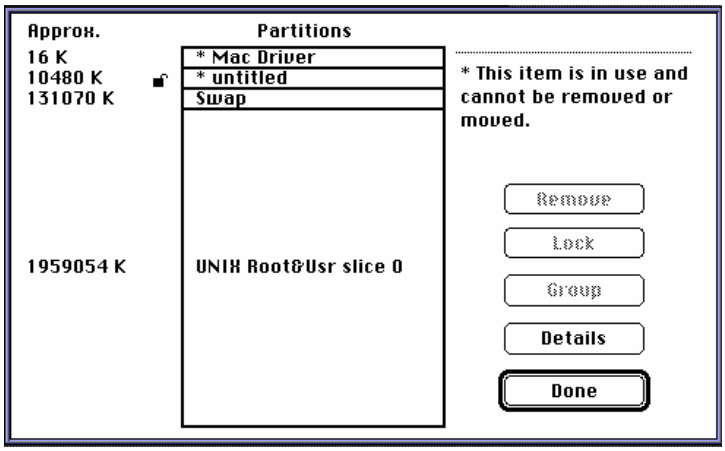

Figure 4: Example Partitioning for 2Gb Drive

Installing MacOS 7

The next stage is to install the minimal MacOS 7 boot system that will launch A/UX. From the main installer menu, click “Install Startup…” to reach the screen shown in Figure 5.

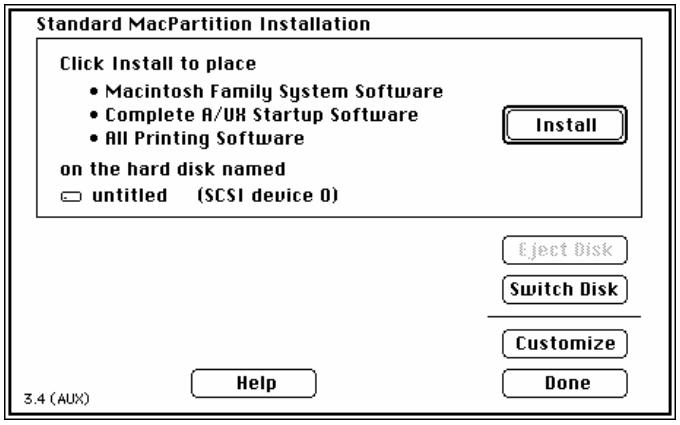

Figure 5: Startup Installation Menu

Because the MacOS partition is so small, it’s preferable to choose a minimal MacOS installation to reflect this. Click “Customize” to reach the Startup software selection screen. The safest option to select is “Min System for any Macintosh” and “Minimal A/UX Startup Software” using Shift-Click, followed by Install.

 **Note:**	Whatever System software you have will have been wiped during the partitioning phase. If you know your hardware won’t work with A/UX’s version of System 7, you can skip this stage. See the note on Installing A/UX packages later.

Installing A/UX Packages

With the startup partition ready it's time to install the individual packages. A/UX has a large amount of packages available, almost all of which is completely out of date. Click on "Install Software…" from the main menu. The "Choose Software" menu contains a large list of software available for installation. Most of this software is outdated, unnecessary or just plain buggy. Table 2 provides details on the packages available.

Package	Space Required	Essential?	Comments
Core A/UX System	39117K	Yes	Core Kernel and Utilities
More UNIX Utilities	1298K	No	Whereis and other useful but non-essential tools.
UNIX Printing Utilities	1419K	No	Outdated Unix LPR Services
Accounting & Admin	597K	No	System accounting and Tape Administration tools
UNIX Text Processing	2449K	No	Text processing tools (nroff, troff etc.) and Unix fonts. Requires printing to install.
Networking Capability	3442K	Yes	Basic network client utilities such as an FTP, NFS and rlogin.
UUCP	1258K	No	Unnecessary
Network Server Capability	1305K	Yes	Additional network server config files, including extra Sendmail config information.
Commando Dialogs	1298K	No	Graphical command-line builder
Manual Pages	2878K	Yes	Required reading
Basic C Programming	3330K	Yes	Contains header files and common libraries as well as the awful included cc
Debugging and Version Control	1381K	Yes	Essential debugging tools like sdb and dbx
Extended C Programming	2200K	Yes	Lex, yacc, termcap and curses (unless you want to roll your own)
Basic Macintosh Programming	987K	No	Tools and headers for developing Mac OS programs
Misc. Programming Languages	527K	No	Fortran 77, Basic, SNOBOL and Extended Fortran.
MacX	3710K	No	X-Windowing System for A/UX
X Client Sampler	1251K	No	Sample X-Windows Clients
X11 Server & More	12080K	No	X11R4 Server, manual pages, widgets etc.
X Programming	2683K	No	Tools and libraries for X11 Programming. Needs X Client Sampler
Quicktime	721K	No	Early Quicktime support

Package	Space Required	Essential?	Comments
Games	770K	No	BSD Games, including hunt the wumpus
Terminal Information	430K	Yes	Termcap info for most terminals including ansi and vt terminal types
Shell Layering	24K	No	SysV Shell Layering protocol

Table 2: A/UX Software Packages

When you've made your selection, click Install. A minimal install takes about 15-20 Minutes on a Quadra 650. Once this is complete click restart to reboot the system.

Note: The version of MacOS 7 installed by A/UX might not be compatible with your Mac. If this is the case, re-install a minimal MacOS 7 system from your original MacOS 7 disks, then install the "Minimal A/UX Startup" software from the A/UX "Install Startup…" Dialog.

First Boot and Network Configuration

When you reboot, MacOS 7 should start, followed by the Apple Workgroup Server 95 (A/UX). The first time it boots it will re-build the kernel according to your hardware. When it is finished, click Reboot on the dialog to reboot the server. Once the system reboots, the next step is to configure networking and re-building the kernel again. Open a CommandShell from the chooser and type `newconfig` to configure a new kernel. The newconfig tool enables you to configure all detectable network interfaces through a series of prompts. You may have noticed that default routes and dns servers weren't asked for. It's time to edit some files before rebooting. Append the following line to /etc/rc:

```
/usr/etc/route add default default.route.goes.here 1
```

Replace default.route.goes.here with the IP address of your default router. Secondly, open up **/etc/resolv.conf** and add the following:

```
domain my.domain
nameserver primary.dns.server.ip
```

Of course, replace my.domain with the domain name you use and primary.dns.server.ip with your DNS Server's IP address. The final piece of work to do is to go through **/etc/inittab**. Unlike modern unix systems which use /etc/init.d or /etc/rc.conf to manage service startup, A/UX uses inittab for almost everything in startup. In particular, the following changes need to be made:

```
net9:2:respawn:/etc/inetd #Set to "respawn" for networking
net6:2:wait:/etc/syslogd #Set to "wait" to run a syslog daemon
```

Inetd is largely controlled through **/etc/servers** on A/UX, not inetd.conf. It enables a lot of services that really don't need to be there. Comment them all out except for the telnetd line.

After rebooting you should be able to ping other hosts from the A/UX server. More importantly, you should be able to telnet *to* the A/UX server.

 **Note:**	If you're not happy with the drab black and white desktop, select the control panel from the Finder and open the Monitors Control Panel. In the monitor characteristics section select Colors instead of Grays.

Post-Install Tidy Up

So far, so good. The system is up and running but there's still some way to go before the system is ready for use as a web server. The first time you telnet to the A/UX server, you should notice two things: Firstly that you can login as root. This is bad. The other issue is that you can login as root with no password. This is even worse. To change the root user's password, use the passwd command with no arguments.

A quick cat of **/etc/passwd** reveals the following unneeded accounts:

```
Guest::90:90:A/UX Guest account:/users/Guest:/bin/csh
start:PG/qLJaYo/6mo:100:100:Initial login:/users/start:/bin/csh
```

Disable these accounts by replacing the lines in **/etc/passwd** with the following:

```
Guest:*:90:90:A/UX Guest account:/users/Guest:/bin/noshell
start:*:100:100:Initial login:/users/start:/bin/noshell
```

This isn't the reason that the desktop is automatically logged into upon boot. That's due to the Mac Autologin feature. Disable it by executing the following:

```
rm "/mac/sys/Login System Folder/Preferences/Autologin"
```

To create an account for day-to-day tasks use the friendly and interactive `adduser`. I prefer to set my user account to the staff group, with friends in perm. Create the personal System folder if you plan to log on locally and say no when asked if the user needs to set a password on logon, which will then prompt you for a password.

There are a whole host of issues surrounding relaxed passwords, the lack of a native shadow suite and unstable daemons (not to mention the OS). The best source of information in this area is the A/UX faq (http://www.faqs.org/faqs/aux-faq/) by Jim Jagielski.

Installing Updates and Replacements

With the system up and running its time to replace some of the more outdated daemons and services. Depending upon your requirements you may need to replace a variety of programs. In any case, you'll need to replace the ageing syslogd, which is known to 'miss' certain syslog events. The easiest replacement is the one to be found on Jagubox which, whilst old, will do the job for now. The bad news is that Jagubox went down some time ago. The good news is that (at the time of writing) the relevant tools and patches for this guide are contained within a tarball available from www.snakeoillabs.com.

Using ftp, connect to your Jagubox mirror and download syslogd-1.2.tar.gz from jagubox/Daemons. There's no gzip on A/UX so you'll need to download it. Nic Leymann has an unarchived gzip that you can download via wget then ftp onto the A/UX system. For the sake of cleanliness and continuity, any gnu tools should be placed in **/usr/local/gnu/bin.** So, as root, do the following:

```
# mkdir /usr/local/gnu /usr/local/gnu/bin /usr/local/gnu/sbin
# cp gzip /usr/local/gnu/bin
# PATH=$PATH:/usr/local/gnu/bin
# export PATH
# chmod +x /usr/local/gnu/bin/gzip
# gzip -d syslogd-1.2.tar.gz
# tar xvf syslogd-1.2.tar
x syslogd-1.2/syslogd.c, 31203 bytes, 61 tape blocks
x syslogd-1.2/syslog.h, 4705 bytes, 10 tape blocks
x syslogd-1.2/syslogd, 52932 bytes, 104 tape blocks
x syslogd-1.2/Makefile, 95 bytes, 1 tape blocks
x syslogd-1.2/README, 578 bytes, 2 tape blocks
# cd syslogd-1.2
# cp syslogd /usr/local/gnu/sbin
# mv /usr/include/syslog.h /usr/include/syslog.h.bak
# cp syslog.h /usr/include/syslog.h
```

Open up **/etc/inittab**. Append a copy of the existing syslogd line, but refer to **/usr/local/gnu/sbin/syslogd** instead of **/etc/syslogd**. Comment out the old syslogd entry and save then quit. Now reboot the system and when it comes back up, you should be running the new syslogd. When you check **/usr/adm/messages**, you may notice that no entry was created when you logged in over telnet. Thankfully, **/usr/adm/sulog** at least records attempts to su to root.

 **Note:**	Global PATH settings are kept in **/etc/sysinitrc**. Add **/usr/local/gnu/bin** and **/usr/local/gnu/sbin** to the PATH declaration to avoid having to re-export the PATH on each login.

Installing Basic GNU Tools

By now you should've realised that this isn't quite as easy as setting up Fedora Core. Even an OpenBSD system is generally friendlier (and certainly less scary from a security perspective). You're going to be spared the torture associated with the use of the archaic and arcane C compiler suite, as its time to bring some GNU tools (courtesy of Nic Leymann) onto the system.

To make things easier, we'll start with GNU Wget. Wget can be downloaded from Nic's web site, although the easiest way to get hold of it is in the A/UX Guide support package at www.snakeoillabs.com. Nic includes a readme file with most of his packages, you should take a look with more before installing any of them. Thankfully in most cases you just need to copy the directories contained within the tarball to **/usr/local/gnu**, which is what we're going to do here, although it just wouldn't be A/UX if you didn't have to be careful how you copy files and directories across.

```
# cp -r bin etc info /usr/local/gnu/
# wget
wget: missing URL
Usage: wget [OPTION]... [URL]...

Try `wget --help' for more options.
```

Note:

The above command correctly copies the **bin**, **etc** and **info** directories (and their contents) into **/usr/local/gnu**. If you use **bin/ etc/** or **info/** then the contents of those directories are copied into **/usr/local/gnu/**.

With Wget on-board and some experience in installing packages you're now ready to try installing gcc. Nic Leymann ported gcc-2.8.1 to A/UX some time ago, but for our needs it'll do the job. Use wget to download it either from his site or extract it from the SnakeOil Labs support tarball. then `gzip -d` and untar the archive. Please note that gcc is large, so this may take a while. Once this has finished, look at the **README** and do the following:

```
# mv gcc281 /usr/local/gnu
# ln ../gcc281/bin/gcc gcc
# ln ../gcc281/bin/gcc cc
# ln ../gcc281/bin/as as
# ln -s ../gcc281/bin/c++ c++
# ln -s ../gcc281/bin/g++ g++
# ln -s ../gcc281/bin/gcov gcov
# ln -s ../gcc281/bin/ld ld
# ln -s ../gcc281/bin/protoize protoize
# ln -s ../gcc281/bin/unprotoize unprotoize
```

Note:

The local C compiler's cc command lives in **/bin**. Don't forget to `chmod -x /bin/cc` otherwise you'll have problems compiling most software with two different compilers at the same time!

Congratulations, you now have gcc installed! Don’t pop the champagne cork just yet though as there’s still more work to be done. A fair portion of library routines present in modern OSes don’t exist in A/UX. Thankfully, someone (namely Jim Jagielski) wrote a library called libUTIL that makes things a bit more bearable. Unfortunately, getting hold of libUTIL is difficult to say the least. Thankfully its contained within the guide support tarball. Download and unarchive the libUTIL tarball. Check both the **README** and **README.FIRST** before doing anything. First we need to copy some headers across.

```
# mkdir /usr/local/include
# mv regex.h sysexits.h strftime.h /usr/local/include/
# mv man /usr/local/gnu
# cp libUTIL.a /usr/local/gnu/gcc281/lib
```

Now open **/usr/local/gnu/gcc281/lib/gcc-lib/m68k-apple-aux3.0.1/2.8.1/specs** and look for the line that reads ‘*libgcc:’. On the line below that reads ‘–lgcc’ change that to ‘–lgcc –lUTIL’. You may now pop the champagne cork.

Preparing the System for use as a Web Server

Before you install your Web Server you need to determine your requirements. These are especially important on A/UX, mainly because of poor memory management, mostly because of resource limitations. My Quadra has roughly 96Mb of RAM available to A/UX (the MacOS side and Finder take up the rest). A basic Apache process takes about 500Kb of memory. Throw in mod_perl and you're looking at a megabyte. Add PHP and you're looking at between 4 and 64Mb per process, dependant upon configuration.

Before we install Apache we need to add a new user. Apache spawns as root, but drops privileges to a defined user and group later. Unfortunately this user is the 'nobody' user by default. Under A/UX, the account has a high user ID (UID) assigned. The UID is so high that A/UX won't allow Apache to change group via the setgid() call. This means that although Apache can change users, it can't change group and would have to run as root, something Apache won't do. One solution to this problem is shown below:

```
# adduser
Enter user's login name (e.g. fred) www

Suggested values are shown in brackets.
To accept, press RETURN; otherwise, type the new value.

User's full name (e.g. Fred Smith) [] Web User
Office address (e.g. mail stop) []
Office telephone (extension) []
Home telephone []

Initial group  (create a new group by entering a new name).
Currently available groups are:
Special (system) groups:
  root      daemon    bin       sys       adm       uucp      utmp
  lp        mail      nuucp

Regular (user) groups:
  staff     perm      temp      contract  guest     project

Initial group [gp1000] daemon
Shell [/bin/csh] /bin/noshell
Home directory [/users/www] /usr/local/apache
Create a personal System Folder? [yes] no

The account for www will be created with the following attributes:

Login name: www (uid 1001)      Real name: Web User
Office address:         ext.:   Home phone:
Home directory: /usr/local/apache       Shell: /bin/noshell
Group daemon (gid 1)
The common System Folder will be used.

OK to create account? [yes]
Require user www to set password on initial login? [yes] no
Specify password for www:
New password:
Re-enter new password:
Enter next user's login name (e.g. fred) [enter empty line to quit]
#
```

Now open up **/etc/passwd** and replace the DES string for the www user with an asterisk. Assuming all went well we're now ready for the final part of the equation, installing Apache!

Compiling and Installing Apache

Finally, 15 pages after you've started you've got to compiling Apache. This is so much easier on RedHat yet considerably more rewarding on A/UX. The latest version of Apache can be downloaded from http://www.apache.org. In this tutorial we're going to install version 1.3.33. The 1.3.x branch is historically known to work well under A/UX.

Download and unarchive the tarball from your local mirror (this may take some time), then run `./configure –prefix /usr/local/apache`. Now run `make && make install` and expect to wait a while. Once this has completed, open up /usr/local/apache/conf/httpd.conf and make the following changes, outlined in the table below:

Directive	Old Setting	New Setting	Comments
Port	8080	80	This should be set to whatever port you want Apache to listen on.
User	nobody	www	This is the user Apache will change to
ServerName	(unset)	Your FQDN	Set this to the Fully Qualified Domain Name used by this system

Table 3: Configuration Changes Needed

Once you've done this, type `/usr/local/apache/bin/apachectl start` as root. Fire up a web browser and point it at your new Web Server. If all goes well, you should see something like Figure 6 below.

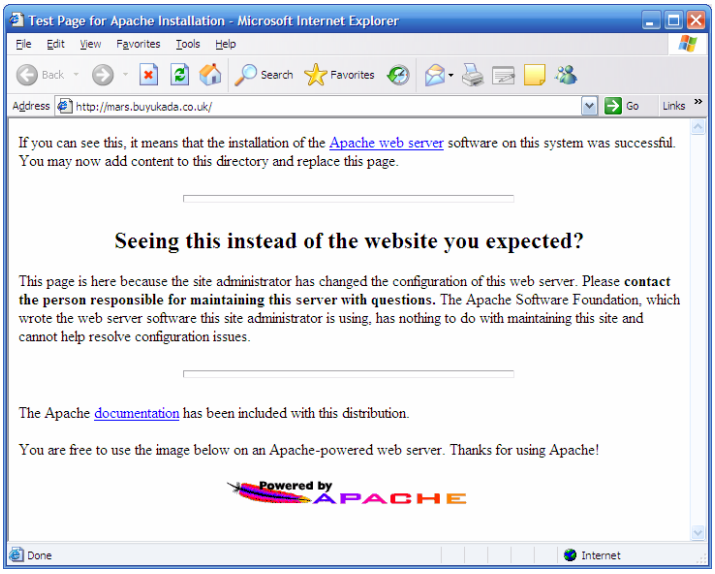

Figure 6: Default Apache Home Page

What now?

Congratulations, you've made it! You've now got a fully functional Web Server running on A/UX. As you bask in the glow of your own glory, you may want to consider adding the following:

Some form of interactive session to replace the ageing default telnetd. Jim Jagielski's replacement works but running telnet is hardly the best solution. Ports of older versions of SSH can be found on the Internet. Those with 030s need not apply – just generating the keys will be painfully slow.

Some means of uploading files to the server. There are plenty of options. Nic Leymann managed to get Samba to compile but it hasn't been tested. The built in FTP daemon is functional, but not exactly up to date. Wu-FTPd claims to support A/UX , although the current 2.6.2 build does not compile under an A/UX 3.0.1 built using this guide. NFS may also be an option, although A/UX's implementation may not be as stable as other OSes.

If your A/UX system is on for any real period of time you may notice the clock running out of sync. A popular NTP daemon called xntpd has historically supported A/UX. Whether or not it still works is unknown, but it may be worth a try.

Further Reading

The Usenet Newsgroup comp.unix.aux should be your first port of call, preferably through an archive such as google groups. It's not really active any more, although it can be a useful source of information.

www.aux-penelope.com is probably the best current resource for information on A/UX. In addition to the installation guide (which accounts for the majority of screenshots in this one, thanks Scott!) there's compatibility information and notes on other versions of A/UX.

http://home.earthlink.net/~gamba2/index.html is the place to go for older Mac system disk image downloads.

http://www.nleymann.de/appleAUX/index.htm is the home of Nic Leymann's A/UX ports. Nics ports form an essential part of this guide. Without them this guide would be a whole lot longer.

ftp://ftp.geo.tu-freiberg.de/pub/aux/ is a live (at the time of writing) Jagubox mirror.

http://www.xianshield.org/guides/apache1.3guide.html is an essential resource for anyone considering placing their Apache installation on a live network.

About the Author

Steve is the founder of SnakeOil Labs, an independent research organisation created to counter the web of jargon and FUD spun by many in the IT industry. Steve is also the author of Athena, a Search Engine testing tool and has written code, articles and reports on a number of subjects for a variety of organisations. When Steve isn't coding, writing or speaking he enjoys spending time with his wife, Ozge.

About SnakeOil Labs

SnakeOil Labs is an independent research organisation created to counter the web of jargon and FUD spun by many in the IT industry. Working in the spirit of similar research organisations of the past, SnakeOil Labs' goal is to identify the questions and present the answers where the IT industry has yet to develop a 'scalable enterprise solution'. We aim to achieve this through the release of innovative tools, informative white papers and educational presentations and workshops. Unlike other research organisations, SnakeOil Labs does not believe in 'handles' or 'hats'.

If you have a requirement for a research project but lack the resources, need technical documentation written in an informative yet entertaining manner or need approachable technology speakers capable of communicating from the boardroom to the basement, contact us at info@snakeoillabs.com.