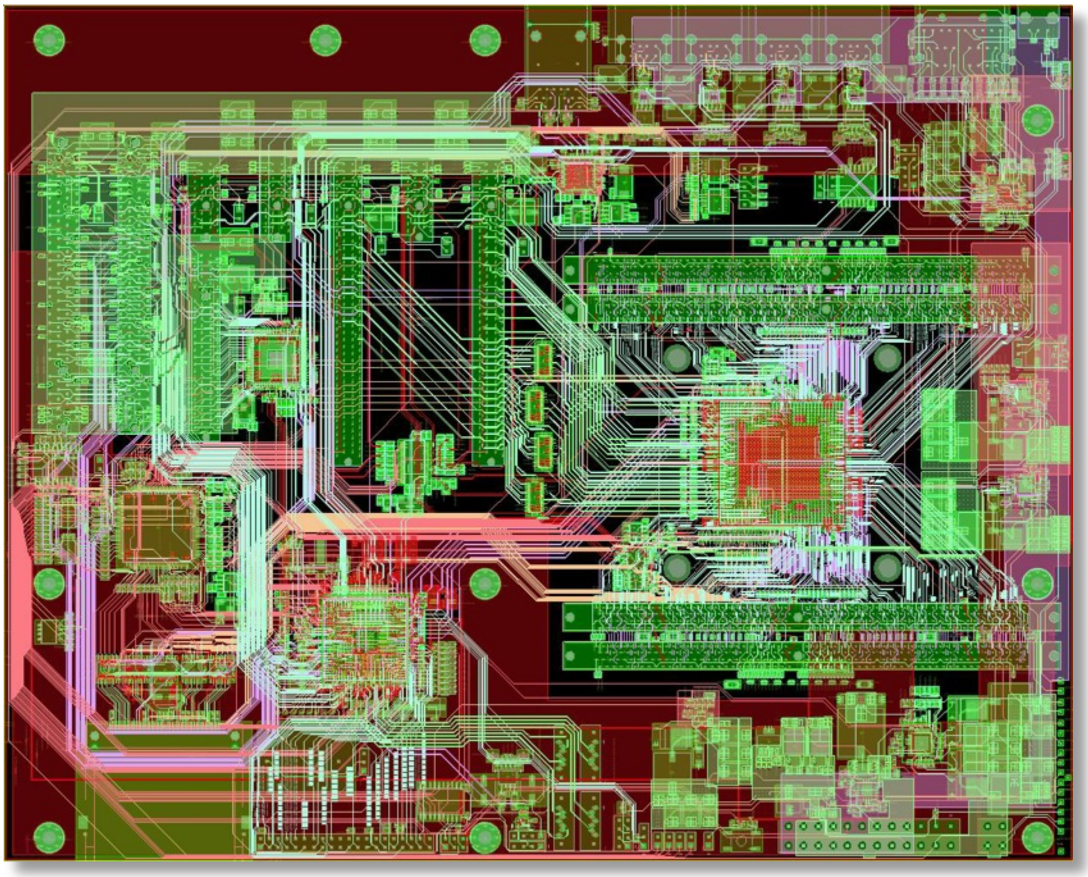


AmigaONE X1000 Firmware and Booting Guide

Version 5.1

July 2012

Table of Contents

1. INTRODUCTION 3

2. THE CFE FIRMWARE 3

CFE Command Line 3
Menu Start-Up 3
Menu Configuration 4
EXAMPLE ONE: Booting AmigaOS 4
More Menu Options 5
EXAMPLE TWO: Booting AmigaOS Three Ways 6
Locations and File Systems 6
More CFE Tips & Tweaks 7
More BOOT Command Options 8
CFE Commands List 8

3. THE AMIGAOS BOOTLOADER & KICKSTARTS 10

Starting AmigaBoot 10
Searching for AmigaOS 10
Basic Amigaboot CFE Variables 11
Amigaboot Menu 11
Kickstart Files & Kicklayout 12
Other CFE Variables for Amigaboot 12

4. AMIGAOS START-UP 13

Startup-Sequence 13
User-Startup 13
Network-Startup 13
Other Initialization 13

5. REBOOTING 13

6. THE DEFAULT CFE CONFIGURATION 14

7. CFE AND SETTING UP MULTIBOOT MENUS 14

AmigaONE X1000 First Contact System
Developed by: A-EON Technology
http://a-eon.com/

AmigaOS 4.1
Developed by: Hyperion Entertainment CVBA
http://hyperion-entertainment.biz

Distribution
AmigaKit LLC
http://amigakit.com

Guide
AmigaOS Development Team &
Paul Sadlik, Darren Eveland, Vicente Gimeno, R. Trevor Dickinson, Carl Moppett, Tejinder Brar

1. Introduction

Welcome to the Firmware and Booting Guide for the AmigaONE X1000 First Contact system. This Guide is intended to assist experienced Amiga users in understanding how the A1-X1000 boots and how they can adjust that process.

As you may have already noticed, starting your AmigaONE X1000 First Contact system is actually a three phase process:

Phase One: Your AmigaONE runs its "firmware", named "CFE" or "Common Firmware Environment". This software initializes the hardware of your X1000. This phase occurs as you usually see the "brushed metal A1-X1000" screen and its "Boot Menu". Underneath that GUI, the CFE command line is where all the work happens.

Phase Two: To start AmigaOS, the "amigaboot" secondary "bootloader" program is run. This software loads the AmigaOS operating system into memory and starts it. This phase usually occurs while you see the red "progress bar" running across the bottom of the grey boot screen, although "amigaboot" might also present a second boot menu, if so configured.

Phase Three: Finally, AmigaOS begins running and its "startup" scripts are executed. This phase begins as the black "AmigaOS" logo screen is displayed and is mostly complete when the Workbench is ready to use.

As with all the next generation AmigaOS computers, when you power up your machine, it will go through all three of these phases. This is typically known as "cold booting".

In the next section, we will examine how you can use CFE in your A1-X1000 to reconfigure the first phase of booting and
control how your AmigaONE starts and what operating systems it uses. In the following sections, we will explore the next two phases of booting your AmigaONE X1000.

2. The CFE Firmware

CFE Command Line

Under most circumstances, your AmigaONE X1000 will already be configured so you will only see CFE's built-in boot menu system on the screen with the brushed metal AmigaONE X1000 background and then it goes on to rest of the boot process. As such, you would normally not need to use this section of the Guide.

But if your AmigaONE has lost its configuration (for example, CFE's configuration gets damaged), your machine will likely start in the CFE command line console. Furthermore, the CFE and its boot menu can be configured to boot from multiple partitions and use multiple operating systems directly. To do this, the configuration of CFE needs to be adjusted as described below.

To directly access CFE, you can press the "F" key while the "boot menu" is visible and the boot screen will disappear, uncovering the CFE command line or "console". From the CFE console you can run CFE commands, set variables, run programs, check your system, etc.

Of course, using the CFE console presumes you have a certain familiarity with how a command line environment works. If you are not comfortable with entering text commands, understand commands and arguments and the concepts of variables, it might be better to seek more experienced assistance. Using the wrong commands or creating an invalid configuration could result in your AmigaONE X1000 being unable to boot.

Menu Start-Up

Your AmigaONE X1000 First Contact system should already have CFE and its STARTUP variable configured for automatic use. In case you are landing in the CFE command line at power up, apparently you have lost at least some of your CFE configuration and you will need to tell CFE to present its "boot menu" and configure it.Literally, CFE needs to be told to run the "menu" command every time your A1-X1000 is powered on. This is done with a CFE variable called STARTUP. The STARTUP variable contains a command text that is run every time your A-X1000 powers up and the firmware initializes. Setting this variable is done by entering this command in the CFE console:

```
CFE> setenv -p STARTUP menu
```

(Do not type the "CFE>" part. It is the prompt that you will see in the CFE console.)

After entering the above command, you should see the following response in the CFE firmware console:

```
*** command status = 0
```

This response tells you there were no errors in executing the command to set the firmware variable. Hopefully you will see the same successful response with every command you enter in the CFE console. On the other hand, a positive number indicates the severity of the error.

As you see, the "setenv" command is how you set firmware variables. You will use that command to set the rest of the variables used by CFE. The "-p" option of the "setenv" command tells CFE to make the variable "persistent" - so it is saved in battery backed up memory and it will still be there when you restart your A1-X1000. The final two arguments are the name of the variable to be set ("STARTUP") and its value ("menu").

Once you have defined this variable, enter the "printenv" command in the console to have CFE print all its variables. From the output, you can then verify the printed value of the "STARTUP" variable matches what we set ("menu" - without quotes). If not, you can re-enter the above command to re-write the variable's definition.

Menu Configuration

In case your machine is missing its CFE configuration, when it runs the menu command, you will be presented with the grey boot screen and an empty menu. So you will also need to set a few more CFE variables that control what the menu shows and does. The menu can hold up to ten different entries. These entries are each associated with a number key that can be pressed to use the corresponding command.

In order to configure the CFE boot menu, you need to know how to boot the OS in question. Each line (or configuration) in the boot menu is specified with a pair of variables. The names of these variables have this format:

```
MENU_n_LABEL
MENU_n_COMMAND
```

In each set of these variable names, "n" is replaced with the number of the entry, from 0 to 9. For example, the variables of entry 0 would be called "MENU_0_LABEL" and "MENU_0_COMMAND" respectively.

The "LABEL" variable determines what the user will see in the boot menu. The string has to be short (around 30 characters maximum) and can contain any text. A good example would be to call it "AmigaOS 4.1" or "Linux Debian" (more on this below).

The "COMMAND" variable is not displayed. Rather, it is the command that is executed by CFE when this menu item is selected by the user. So for example, the MENU_0_COMMAND is executed when the "0" key was pressed in the menu (or the selection was highlighted when the timer ran out or the user pressed the Return key).

Such command variables can actually contain more than one CFE command, separated by ";" (semicolon) character. Also, the command need not strictly be a boot command. It can contain anything that the CFE console can run, like setting other variables.

EXAMPLE ONE: Booting AmigaOS

The following is a simple configuration of CFE to automatically boot your AmigaONE X1000 into AmigaOS from the first hard drive partition or an AmigaOS CD. It also provides a fallback to boot directly from your optical drive should something happen to your hard drive.

This configuration presumes your A1-X1000 has the standard drive configuration (hard drive on SATA port 0 and opticaldrive on SATA port 2) and has the "amigaboot.of" bootloader installed on the first hard drive partition (FFS2 formatted).Remembering that computers usually start their numbering at zero, we will set up two menu entries like this:

Choice 0 =	Automatically boot the first AmigaOS hard drive partition (or a Install CD).
Choice 1 =	Boot directly from the AmigaOS Install CD.

To set this configuration, go to the CFE command line and enter the following commands to define a LABEL and COMMAND for these menu items. You will do this the same way as we set STARTUP variable above (remember "CFE>" is the console prompt - don't type that):

```
CFE> setenv -p MENU_0_LABEL "AmigaOS 4.1"
CFE> setenv -p MENU_0_COMMAND "setenv amigaboot_quiet Y ; boot -fs=amigafs ide0.0:amigaboot.of"
CFE> setenv -p MENU_1_LABEL "AmigaOS 4.1 CD"
CFE> setenv -p MENU_1_COMMAND "setenv amigaboot_quiet Y ; boot -fs=iso atapi0.1:amigaboot.of"
```

With these commands we set up our two menu entries (you should have received succesful "command status = 0" responses with each line entered). In these COMMAND variables, we have used a semicolon (";") to string two commands together in each one variable.

The first command sets a non-persistent variable that tells the AmigaOS bootloader to automatically proceed with the first bootable AmigaOS system it can find. The second command ("boot") starts the Amiga bootloader ("amigaboot.of") to begin the second phase of the boot process.

The second CFE command ("boot") is used to load and start the Amiga bootloader directly from an AmigaOS hard drive partition or Install CD. Since the second menu entry refers directly to the optical drive, you will be able to use that entry even if your hard drive is unusable or not ready for booting (AmigaOS has not been installed yet).

In either case, the "boot" CFE command uses the same parameters: specifying the device file system (or method), a location and a "bootloader" file. Specifically, the "bootloader" file is an ELF program that will be loaded into memory and it will run the second phase of the boot process.

In the next section of this Guide, we will discuss the details of using the "amigaboot.of" bootloader. In the text below, additional CFE commands are described for finding the device names CFE uses and how to check on file locations, in case your hardware differs from the above configuration.

Again, you can use the "printenv" command to print out all the CFE variables to check them. If our new MENU variables match, it's time to test. From the CFE prompt, enter "menu" to restart the CFE boot menu. The screen will show the brushed metal CFE boot menu again with the entries you set. You can then select a device, press Return and your machine should boot off your hard drive or the AmigaOS boot CD (if it is loaded).

More Menu Options

Two other common boot menu options can be configured with CFE variables. One is the MENU_TIMEOUT variable. This determines how long the CFE boot menu will count down until it automatically continues with the default option. The value is a number and specifies the number of seconds. It defaults to 10. To set the time-out to 5 seconds, enter:

```
CFE> setenv -p MENU_TIMEOUT 5
```

To restore the default value, delete the variable with the "unsetenv" command:

```
CFE> unsetenv MENU_TIMEOUT
```

Needless to say, if you set the timeout delay to too short a time, it will make it more difficult for you to change your boot selection.

The other option is the MENU_DEFAULT variable. This variable specifies the number of the boot menu entry that should be booted by default. If you had configured the CFE boot menu with multiple entries, you could set the following variable to have CFE use the second entry by default:

```
CFE> setenv -p MENU_DEFAULT 1
```

As with the timeout, the default can be deleted by the "unsetenv" command. If this variable is not set, CFE will boot from the first menu item by default. Of course, we also use the "-p" option to make these variables survive powering down.

EXAMPLE TWO: Booting AmigaOS Three Ways

The following is an example of how you can set up three different ways to boot into AmigaOS. The first two choices would be the "automatic" and CD methods from the first example. The third CFE menu choice will let the user manually select which partition and bootable device they want to run AmigaOS from.

This arrangement is useful if you make a lot of modifications to your AmigaOS system configuration and want to keep a another "safe" or "rescue" AmigaOS system installed on another partition. It is also useful for being able to install and boot from different versions of AmigaOS. Of course, this method will support any number of bootable AmigaOS partitions.

As such, we would expect to see the following setup for this example:

Choice 0 =	Automatically boot the default AmigaOS hard drive partition.
Choice 1 =	Boot from the AmigaOS Install CD if the hard drive is unavailable.
Choice 2 =	Let the user select an AmigaOS partition & kickstart configuration to boot.

In the CFE console, enter the following commands (at the "CFE>" prompt) to define a LABEL and COMMAND variables for each of the three menu items:

```
CFE> setenv -p MENU_0_LABEL "AmigaOS Auto"
CFE> setenv -p MENU_0_COMMAND "setenv amigaboot_quiet Y ; boot -fs=amigafs ide0.0:amigaboot.of"
CFE> setenv -p MENU_1_LABEL "AmigaOS CD Boot"
CFE> setenv -p MENU_1_COMMAND "setenv amigaboot_quiet Y ; boot -fs=iso atapi0.1:amigaboot.of"
CFE> setenv -p MENU_2_LABEL "AmigaOS Menu"
CFE> setenv -p MENU_2_COMMAND 'setenv os4_bootdevice "auto" ;
        <...same line...>                   boot -fs=amigafs ide0.0:amigaboot.of'
```

As you see, the first two sets of MENU variables configure automatic and CD booting as in example one. Naturally, do not type "<...same line...>", just continue the rest of the variable definition in one long command in the CFE console.

The third menu definition ("MENU_2_COMMAND") substitutes the "amigaboot_quiet" variable definition with that of "os4_bootdevice". By omitting the previous variable we will no longer "quietly" boot off the first partition found. By including the new variable, we are telling the "amigaboot.of" bootloader to fully load the configuration from whatever source the user selects.

You can also see from the third menu item, we mix single- (apostrophes) and double-quotes to have two levels of quotation recognized by CFE - single-quotes encapsulate the entire definition of the command variable and the double-quotes encapsulate the definition of the "os4_bootdevice" variable. As long as order & pairing is correctly used, these two means are interchangeable with CFE.

Once you have set these variables and used "printenv" to confirm your settings are correct, you can enter the "menu" command to see how your new menu looks and works. Of course, you can also set other CFE variables, like "MENU_DEFAULT", if you wish.

This configuration should generate a CFE menu and "AmigaOS Auto", "AmigaOS CD Boot" and "AmigaOS Manual" entries. Selecting either of the first two entries will boot as it did in the first example. Selecting the last menu entry will present you with a second "amigaboot" menu that shows all the bootable AmigaOS devices and "kickstarts" that could be found. This second "amigaboot" menu will be described in more detail in a next section of this Guide.

Locations and File Systems

As you have seen, booting from CFE is usually done by running a bootloader program such as "amigaboot" from a disk or other medium. The program is usually a native PowerPC ("PPC") binary (in the "ELF" format) and the normal procedure is to load it from a hard disk or CD. Disks and other storage media are represented as "devices" in CFE.

You can see a listing of all the devices visible to the CFE firmware with the following command:

```
CFE> show devices
```

Bear in mind, you will see a full list of devices that are present, but only some of them can be used for booting.SATA and IDE (PATA) hard drives are usually called "ideM.N:", where "M" is the controller number and "N" is the disk number. Sometimes, you can add a comma followed by a partition name or number afterwards. For example, "ide0.0,dh0:" would access partition "dh0:" on an AmigaOS disk on the A1-1000’s SATA 0 port.

CD- and DVD-Drives are usually called "atapiM.N:" with the same semantics for "M" and "N". They usually are not partitioned.

Please note, to be recognized by CFE (and accessible for loading the bootloader), hard drives or optical drives must be installed on the AmigaONE X1000’s SATA ports numbered 0 or 2 or on a supported SATA/PATA PCI card. The other port motherboard ports will not be recognized by CFE and you will not be able to boot from them.

There are two main ways to boot: From a device or from a network. In order to boot something from a disk, you also have to know the disk’s formatting. What that means is that the file system used on the disk must be specified - there is no code to autodetect that. Typically, file systems are either "amigafs" (AmigaOS FFS2), "ext2", "fat", or "iso" (ISO9660). The file system is specified via the "-fs=<id>" command line option. CFE does not recognize or read from other AmigaOS file systems like SFS and JXFS.

Most CD’s would be using the ISO format, so you would set "-fs=iso" when needed. If you have an AmigaOS disk, the file system is would be "-fs=amigafs". You can check this with the "dir" command that lists all files in a directory on the disk:

```
CFE> dir -fs=amigafs ide0.0:Kickstart
```

That command will list all files in the Kickstart drawer on the "ide0.0" device, if the file system is AmigaOS FFS2.

Booting from a network uses slightly different options. Typically, booting is done by connecting to a TFTP server. To specify this, the option "-tftp" and an IP address of the server must be used with the "boot" command. You can use the CFE "ifconfig" command to try to configure the connection and use the "ping" command make contact with the IP address of the TFTP server you want to connect to. Of course, you can use the "help" command for more information on these commands.

The bootloader file location is always specified as a two-parter, separated by a colon (":"). Part one (before the colon) is the device or network address. Part two (after the colon) is the file name/path. This is best illustrated by examples, so here are a few:

ide0.1:amigaboot.of	= load amigaboot.of from a SATA/IDE device.
usbdisk0:vmlinux	= load a file "vmlinux" from an usb device.
192.168.0.1:amigaboot.of	= load amigaboot.of from a network TFTP server at IP address 192.168.0.1.
cf0:amigaboot.of	= load amigaboot.of from a card in the X1000 compact flash slot.

More CFE Tips & Tweaks

As we saw at the beginning of this Guide section, we set the "STARTUP" CFE variable to run commands at start-up of your machine. We also saw we can use a semicolon (";") to string multiple commands together in a single variable. Using semicolons, we can add more commands to the "STARTUP" variable.

A popular command is to have CFE set your AmigaONE X1000 to run at its maximum speed as soon as it powers up (by default it only runs around one quarter of its normal speed when booting). This can be done by changing the STARTUP variable to this extended value:

```
CFE> setenv -p STARTUP "set pmu -astate=A4 ; menu"
```

Finally, it is possible to define CFE variables of your own (using the "-p" option to make them permanent). Then you can simply enter those variable names in the CFE command line or in another command variable and your variable's contents will be used as though you'd entered the commands again.

For example, you can set:

```
CFE> setenv -p speed "set pmu -astate=A4"
```

And then just enter "speed" in the command line and the machine speed will be set to full speed. Alternatively, you could then use this variable in another one, like the STARTUP variable (this will keep your variable definition's size manageable):

`CFE> setenv -p STARTUP "speed ; menu"`Bear in mind that CFE variable & command names are case sensitive.

So if you defined a variable called "speed", subsequently calling "SPEED" will not work. Also entering the command "MENU" in the CFE console will not bring up the boot menu, only the lower case "menu" will.

In case you would like to know how warmly your AmigaONE X1000 is running, your machine is equipped with a number of internal temperature sensors. These can be read from the CFE console by entering the "show temp" command. It will provide you with the following readings (in centigrade):

local 1	= ambient temperature in the case.
remote 1	= CPU core 1
remote 2	= CPU core 2
remote 3	= Southbridge chip

As you have seen the CFE console runs on the on the screen of your X1000 and is accessed by the keyboard plugged into the first USB port. You can alternately connect a serial "null modem" cable to the X1000's first serial port and access the same CFE command line by a terminal emulator program on another Amiga by entering this CFE command:

```
CFE> set console uart0
```

You can revert back to the "vga console" by entering this command:

```
CFE> set console pcconsole0
```

More BOOT Command Options

A few additional command line options exist that can be given to the CFE "boot" command. These options include:

-nousb	= will disable USB while booting. Amigaboot.of does that automatically.
-noints	= will disable interrupts when booting. This option is usually used when booting a linux kernel
-elf	= specifies that the file type is "elf". Usually the default, it may be specified when booting Linux.

CFE Commands List

The following is a comprehensive list of all the possible CFE commands. Please note, in some cases these commands may not work, depending on your system, the version of the firmware, etc.

In the case of each command, you can enter "help <command name>" in the CFE console and get a description of that command's function and usage. Simply entering "help" will give you the following commands list. Use with care.

altcpu	Start a program on another CPU
arp	Display or modify the ARP Table
autoboot	Automatic system bootstrap.
batch	Load a batch file into memory and execute it
boot	Load an executable file into memory and execute it
copydisk	Copy a remote disk image to a local disk device via TFTP
cpu zero	Zero a memory region using CPU
d	Dump memory
dush	This command flushes data out of chip to memory
dir	Display directory listing for a device
dramcfg	Update MMC/SDRAM configuration
e	Modify contents of memory
f	Fill contents of memory
flash	Update a flash memory device
go	Start a previously loaded program
help	Obtain help for CFE commands
ifconfig	Configure the Ethernet interface
int handler	Display interrupts an acknowledge
int mask	Mask interrupt by vector
int unmask	Unmask interrupt by vectorload Load an executable file into memory without executing it
loop	Loop a command
mccfg	This command changes the run time MC configuration
mcdll	This command displays MMC slave DLL settings
memcmp	Compare two memory blocks
menu	Show the boot menu
pci cfg	PCI read/write 32-bit register
pci dump	Dump PCI configuration space
ping	Ping a remote IP host
printenv	Display the environment variables
r	Read MSR or SPR register
ramdisk	Load a ramdisk image into memory
randmemtest	Tests a 256MB memory region using random access pattern
reserve	Mark a region of memory as reserved
reset	Causes a system reset
save	Save a region of memory to a remote file via TFTP
set astate	Modify an active state register
set console	Change the active console device
set hid	Set HID values based on a named profile
set pmu	Set PMU information, including CPU speed
setenv	Set an environment variable
show agents	Display list of SOC agents or blocks in an agent
show boot	Display boot block from device
show devices	Display information about the installed devices.
show devtree	Dump Open Firmware device tree
show dramcfg	Display SDRAM configuration
show ecc	Display memory controller ECC counters
show heap	Display information about CFEs heap
show hid	Display current HID profile
show memory	Display the system physical memory map
show pci	Display information about PCI buses and devices
show pmu	Show PMU information, including CPU speed
show soc	Display SOC register contents
show temp	Display temperature sensor readings
show usb	Display devices connected to USB bus
show vga	List available VGA adapters
sleep	Wait for some period of time
spi block erase	Erase a 64KB page in SPI flash
spi compare	Compare a byte string in the SPI flash
spi copy	Write a byte string to the SPI flash
spi erase	Erase the entire SPI flash
spi id	Show the id of the SPI flash
spi read	Read a byte from SPI flash

CFE COMMANDS LIST (cont.)

spi status	Show the status of the SPI flash
spi write	Write a byte to the SPI flash
test disk	Do a disk test, read/write sectors on the disk
test filesys	Do a file system test
testdram	Test sdram memory
unsetenv	Delete an environment variable
usb init	Start USB and set debugging options
vga dumpbios	Dump the indexed VGA BIOS to the console
vga inb	Input a byte from an I/O port
vga init	Initialize the VGA adapter
vga inl	Input a word from an I/O port
vga modelist	Display list of supported VGA modes
vga outb	Output a byte to an IO port
vga outl	Output a 32 bit integer to an IO port
vga select	Choose VGA adapter to enable
vga setmode	Set VGA adapter VESA mode
w	Write MSR or SPR register

3. The AmigaOS Bootloader & Kickstarts

As we discussed in the previous section of this Guide, after the CFE initializes the hardware and lets the user select a bootable drive, the second phase of booting your AmigaONE X1000 begins. Specifically, CFE loads a secondary "bootloader" program into memory and it takes over. As we saw above, the bootloader can be loaded from a variety of sources: a hard drive, optical disk, USB drive, compact flash card or network server.

In the case of AmigaOS, the bootloader program is called "amigaboot.of" and it usually resides on the root of the first AmigaOS FFS2 partition on your hard drive. It performs a number of tasks that occur before the actual AmigaOS is up and running: it finds all available AmigaOS systems to boot from, it can solicit which system to boot from, it loads the "kickstart" components of AmigaOS from that source into memory and then it starts the AmigaOS kernel.

When "amigaboot.of" is finished, the black AmigaOS splash screen will appear, AmigaOS has been started and the third phase of the booting process begins in the background. The third phase of the booting process is described in the next section of this Guide.

Starting AmigaBoot

The only requirement is that the "amigaboot.of" bootloader (actually a CFE program) must be located on a device CFE can access and its location must be specified in the CFE "boot" command. Remember the CFE menu's COMMAND variable defined like this above? :

```
CFE> setenv -p MENU_0_COMMAND "boot -fs=amigafs ide0.1:amigaboot.of"
```

In this case, the "amigaboot.of" was going to be loaded from an AmigaOS FFS2 partition. As discussed above, the bootloader program could have been loaded from a CD, hard drive, USB drive, a compact flash card in the X1000's CF slot or from a network server.

Searching for AmigaOS

Once the AmigaOS bootloader program is running, the CFE environment is no longer used and the "amigaboot.of" program transitions your AmigaONE to running AmigaOS. By default this transition is configured to only appear as a red progress bar along the bottom of the "brushed metal" CFE screen. By changing CFE variables, you can also configure the AmigaOS bootloader to present you with a choice of partitions and configurations to run AmigaOS from.

Typically, an AmigaOS machine has a hard drive and that is divided into a number of partitions. While "amigaboot.of" has to be on a partition with a file system CFE can access (like FFS2), it does not have to be on the same partition as the AmigaOS installation. In the interest of speed, by default AmigaOS is installed on yet another partition using the faster SFS file system (SFS is visible to the "amigaboot.of" bootloader but not to CFE).

While the default system may only have one system partition, many users create multiple system partitions. Additionally, each system partition there can also be more than one "kickstart" configuration that can be used. Accumulated, there may be a range of AmigaOS systems to boot from.

This multiplicity can be an invaluable feature since it allows users to keep a "safe" or backup system to boot from, it allows for easier debugging and it allows users to keep multiple versions of AmigaOS installed on their computer at one time.

As soon as the "amigaboot.of" loader begins, it scans all the available AmigaOS partitions it can find. Specifically, all SATA and IDE devices are scanned for partitions which are set as AmigaOS bootable, using the FFS2 or SFS (not SFS2) file systems and containing an AmigaOS system.

The drive type, order of the partitions and what their AmigaOS "boot priority" is will determine which source is used for booting AmigaOS by default. First, amigaboot will use an AmigaOS CD if it finds one. Next, amigaboot will boot from the highest priority partition it finds on the hard drive (as set in Media Toolbox when configuring the partitions). Finally, amigaboot will boot with the first partition it finds.Depending on how the "amigaboot_quiet" CFE variable is set, the "amigaboot.of" bootloader will either continue automatically booting from the first, highest priority AmigaOS partition and kicklayout it can find. Without the "amigaboot_quiet" variable, amigaboot will generate a menu of all the available AmigaOS boot sources and let the user choose how to proceed.

Basic Amigaboot CFE Variables

In the CFE section of this Guide, we used CFE variables to tell "amigaboot.of" bootloader to automatically use and boot from the first eligible AmigaOS boot device it could find. Setting it was done in CFE console like this:

```
CFE> setenv amigaboot_quiet Y
```

As we saw above, we typically set this variable as part of the CFE menu's COMMAND definition and we did not use the "-p" option to make it permanent. This allowed the user to vary this setting each time the AmigaONE X1000 was turned on. Without this variable, the "amigaboot.of" bootloader will present a menu of all the AmigaOS bootable partitions and configurations it can find. This is discussed further below.

In the case where we wanted to use the amigaboot menu to select a bootable AmigaOS source, we used another CFE variable to tell "amigaboot.of" to load everything ("kickstart" modules and the operating system) from the partition selected by the user.

```
CFE> setenv os4_bootdevice "auto"
```

Under normal circumstances this setting should always be used when booting from multiple partitions. Without this setting, "amigaboot.of" will typically load the "kickstart" modules from the choice selected by the user, but it will continue booting from the highest priority AmigaOS partition found. This unusual behavior is only useful for certain OS testing and will lead to unpredictable results for normal use.

Presumably, these variables will be set as desired when the CFE firmware is configured. All of these variables can either be set to be persistent (in place all the time) or on a case by case basis by setting them as part of the CFE boot menu COMMAND variable. Please see the examples in the CFE section of this Guide on how these CFE variables were used. Below there are more examples of CFE variables that can be set to configure how AmigaOS starts.

Amigaboot Menu

If the "amigaboot_quiet" CFE variable is not set (or is not set to "Y"), then the AmigaOS bootloader program will present a simple text menu which shows the names of all the available boot devices as well as the configurations of kickstart files on each of bootable partition. Here is an example of two bootable partitions with normal/debug kickstart configurations on each:

```
AmigaOS 4.x OpenFirmware Bootloader V1.0

Choose a configuration to boot (Default choice in green):
1= AmigaOS4.1                (Volume DH0)
2= AmigaOS4.1 Debug          (Volume DH0)
3= AmigaOS4.1                (Volume DH1)
4= AmigaOS4.1 Debug          (Volume DH1)

[RETURN] Boot default      [SPACE] Stop Countdown
[CTRL-U] Move default up   [CTRL-D] Move Default Down
[1234] Boot appropriate config
```

From that menu the user can select a partition & kickstart configuration and the AmigaOS bootloader will proceed to load the kickstart files from the selected source accordingly. Depending on the settings of the "os4_bootdevice" CFE variable, "amigaboot.of" should also boot the entire AmigaOS from the chosen partition.

Kickstart Files & Kicklayout

The next task the "amigaboot.of" bootloader performs is the loading of "kickstart" files into memory from the chosen source.

These "kickstart" files are a number of low level operating system components that are fundamental to the operation of AmigaOS, including the kernel, device drivers, interface drivers, file systems, etc. The components are usually kept in the "Kickstart" directory of an AmigaOS system partition and have a ".kmod" suffix on each file.

Along with these files there is one special file called the "Kicklayout". This text file contains one or more lists of kickstart files that should be loaded under various configuration names. Typically AmigaOS comes with a standard configuration and a "debug" configuration that uses the debug version of the kernel. Under certain circumstances there might be more named configurations for testing, etc. Normally, a user will never see these variations.

When the AmigaOS bootloader scans for partitions, it collects and presents these named kickstart configurations for each partition. Depending on configuration and/or the user, "amigaboot.of" will then load the kickstart modules (from the named configuration in kicklayout) into memory and then start the AmigaOS kernel. At that point, AmigaOS has technically begun to run in its most basic form. Please see the next section of this Guide for how booting continues.

Other CFE Variables for Amigaboot

If you are using a PC keyboard, it has a "Windows" keys (on either side & closest to the space bar) that should be recognized as "Amiga" keys in OS4. If the keys closest to the space bar are acting like "ALT" keys, you can set a persistent CFE variable that will attempt to reverse the condition:

```
CFE> setenv -p keymap_swapaltamiga "off"
```

In case this does not work, you can either set the value to "off" or use the "unsetenv" command to remove it.

Other PC keyboards may not have a right hand "Windows" key. In that case, you can set another CFE variable that will use the right hand "menu" key as the Right-Amiga key instead. This is accomplished by setting the following CFE variable as such:

```
CFE> setenv -p kbddev_norwin 1
```

Again, if you don't need this, you can either set the value to 0 or use the "unsetenv" command to remove it.

In case you wish to run the "debug" version of the AmigaOS kernel, there is another CFE variable that configures how debug kernel output will be sent. In that case, you can set this variable:

```
CFE> setenv -p os4_commandline "SERIAL DEBUGLEVEL=5"
```

With this setting, when you boot from a kickstart layout using the debug version of the kernel, debug information will be dumped out your X1000's serial port (of course you'll also need a "null modem" cable, terminal program and another Amiga to capture this data). To be able to run AmigaOS with a debug kernel, you will have to boot with the amigaboot menu and select a "debug" kickstart configuration.

Of course, this variable can also be set at boot time by configuring a CFE boot menu COMMAND variable as such:

```
CFE> setenv -p MENU_2_COMMAND 'os4_commandline "SERIAL DEBUGLEVEL=5" ;
          <...same line...>                    boot -fs=amigafs ide0.0:amigaboot.of'
```

Also please bear in mind, the value "DEBUGLEVEL" sets how much detail is sent out the serial port. The lower the number and the less information is sent. Set the number much above 5-10 and you will get so much data that it will slow your X1000 down.

4. AmigaOS Start-Up

As we mentioned, AmigaOS nominally starts operation once all the kickstart components have been loaded into memory and the AmigaOS kernel has started running. Once the kernel has finished a process of initializing each of those components, a series of Amiga CLI scripts are run.

All of these script text files are kept in the "S" directory of the AmigaOS system. The first of these scripts is the "startup-sequence" file. It calls two additional scripts, the "user-startup" and "network-startup" scripts. While the first file is meant to be standard and not require editing by the user, the other two files are meant to be edited by users and applications.

All of these scripts contain AmigaOS CLI commands, so you can test and explore these commands in your Amiga Shell. By entering "help <command name>" you can get a description of each command.

Startup-Sequence

This standard script file performs some of the last preparations of the operating system before the user starts to use AmigaOS. These include system path name assignments, loading of user preferences, final mounting of disk drivers, and lastly, starting the Workbench program.

Nothing in this file should be edited by the user. All user adjustments should be made in the "user-startup" and "network-startup" files.

User-Startup

The "user-startup" script is run by the "startup-sequence" script just before "network-startup" and the Workbench.

This script file is intended to contain all the unique scripting a user might want to perform at the startup of their machine. This script file is also where applications (installed by the user) typically add their own scripting to be able to run.

Typically this script file will end up containing a number of pathname assignments the user might want to use, pathname assignments to application directories, and similar adjustments to the operating system.

Network-Startup

This script is a newer addition to AmigaOS. It is also started by "startup-sequence", after "user-startup", but before the Workbench. This script is also started "asynchronously", so it does not delay starting the Workbench.

The first thing "network-startup" does is run the "AddNetInterface" command that initiates the Internet connectivity from this AmigaOS machine. In the following lines, the user or applications can add scripting that might be dependent on a network connection.

Because network startup can occasionally take a some time, by using this separate asynchronous script, any delay in network startup will not delay the startup of the rest of the machine and AmigaOS. This also means network startup may not be finished by the time the Workbench is up and running.

Other Initialization

Aside from these scripts, there are a few other areas where AmigaOS start-up operations can occur. While describing those is beyond the scope of this document, you can explore "WBStartUp Prefs" (where applications are started by the Workbench) and "AmiDock" (where active dockies may be loaded).

5. Rebooting

If you need to reboot your AmigaOS machine, you can either perform a "cold reboot", or "warm" or "fast" reboot.

A full "cold reboot" can be initiated by either pressing the "reset" switch on your computer case or pressing the left control, left alt and right alt keys - all three - at once. In either case, this will cause your AmigaONE to repeat the three phase booting process.

On the other hand, if the AmigaOS is still in memory and has not been compromised (i.e., not corrupted by a big crash, etc.), then you have the option of doing a "warm reboot" or "fast reboot" (two names for the same thing). This can be done by pressing the left control, left Amiga and right Amiga keys - all three - at once. In this case, AmigaOS skips the first two phases of the booting process (above) and just restarts AmigaOS and runs the startup scripts again.

In addition to key combinations and the case reset switch, these reboots can be initiated within AmigaOS by using the "reboot" command. To cause a full "cold reboot", enter the following in the AmigaOS CLI command line:

```
1> reboot
```

To cause a "fast reboot" or "warm reboot", you enter the same "reboot" command with the "fast" option

```
1> reboot fast
```

In either case, AmigaOS will cause your AmigaONE X1000 to immediately reboot. Naturally, anything in memory will be erased, any unsaved files will be lost and the RAM Disk will be emptied.

Please see the documentation of the "reboot" command for more information on its options and operation. From the AmigaOS Shell, you can see this documentation by simply entering "help reboot".

6. The Default CFE Configuration

As discussed above, the AmigaONE X1000 First Contact systems will be shipped with a hard drive connected to SATA port 0 and the optical drive connected to SATA port 2. The configuration of CFE variables reflect that, as follows in this list:

MENU_0_LABEL	"AmigaOS 4.1"
MENU_0_COMMAND	"setenv amigaboot_quiet Y ; boot -fs=amigafs ide0.0:amigaboot.of"
MENU_1_LABEL	"AmigaOS 4.1 CD"
MENU_1_COMMAND	"setenv amigaboot_quiet Y ; boot -fs=iso atapi0.1:amigaboot.of"
MENU_DEFAULT	0
MENU_TIMEOUT	3
os4_commandline	"DEBUGLEVEL=5 SERIAL"
speed	"set pmu -astate=A4"
STARTUP	"speed ; menu"

Please see the CFE and Amigaboot Sections above for a description of these commands and what these settings mean. This configuration is derived from the examples in the CFE section with the addition of CPU speed up CFE command and debugging command line for AmigaOS4 (used if the debug kernel is used).

7. CFE and Setting up Multiboot Menus

The CFE firmware on the AmigaONE X1000 supports booting multiple operating systems in addition to AmigaOS. Please see the GNU/Linux Installation Guide for specific information on the preparation for and installation of linux on the AmigaONE X1000 First Contact system.

As described in the Linux Installation Guide, GNU/Linux can be installed on either a single hard drive shared with AmigaOS or it can be installed on a separate, dedicated drive in your A1-X1000. The GNU/Linux Guide describes the steps to prepare the drives in either case and how to install Linux.

Please refer to the GNU/Linux Installation Guide for more detail on the CFE variables that need to be configured to boot Linux.

The AmigaONE X1000 firmware contains a built-in boot menu that can be used to boot multiple operating systems directly from the firmware without the need for further programs like yaboot. The menu is fully user configurable.

Requirements

In order to perform these steps, you need to know how to boot the OS in question. You also need to know how to persistently set a firmware variable. We'll cover these topics briefly.

Partition Setup

AmigaOS and Linux can both be installed on the same disk on your AmigaONE X1000 if you use an AmigaOS partition setup (a.k.a RDSK partition table). The firmware understands both an RDSK as well as a DOS partition setup, but AmigaOS requires an RDSK partition setup since the DOS setup cannot hold FFS partitions.

If you are using separate disks for AmigaOS and Linux, there is no special treatment required, and you can skip the reminder of this section. If you want to use the same disk, the recommended practice is to install AmigaOS first, and create a separate Linux root and swap partition while doing the initial setup for AmigaOS. You can then install Linux afterwards.

You need to have a kernel that supports Amiga partition tables. If you built your own kernel, make sure to compile this into the kernel; do not use a module for this, or booting will not work.

Menu Setup

The menu is controlled via a few firmware variables. It can hold up to ten different OS/Boot configurations. The configurations are each associated with a number key that can be pressed to boot this configuration. Furthermore, there is a setting for the default configuration to boot, as well as a timeout value that will automatically confirm the default configuration after a certain amount of inactivity.

Each line (or configuration) in the menu is specified via two variables. These variables have the format

```
MENU_n_LABEL
MENU_n_COMMAND
```

with n being the number of the entry from 0 to 9. For example, the variables of entry 0 would be called `MENU_0_LABEL` and `MENU_0_COMMAND` respectively.

The 'label' part determines what is displayed in the menu. The string has to be short (around 30 characters maximum) and can contain anything. A good example would be to call it “AmigaOS 4.1” or “Linux Debian”.

The 'command' part is not displayed. Rather, it is the command that is executed when this option was selected. So for example, the `MENU_0_COMMAND` is executed when the '0' key was pressed in the menu (or the selection was highlighted when the timer ran out, or the user pressed the Return key). The command can actually contain more than one command, separated by ';' (semicolon) character. Also, the command need not strictly be a boot command. It can contain anything that the CFE console can run.

Finally, to make the menu appear at boot-up, we need to instruct `CFE` to run it every time the machine is powered on. This is done with a `CFE` variable called `STARTUP`. The `STARTUP` variable contains a string that is run every time the firmware initializes. For our purposes, we just assign it the `menu` command:

```
CFE> setenv -p STARTUP menu
```

(Do not type the `CFE>` part, it is the prompt that appears on the `CFE` console)

An Example

Time to put this to action. We'll assume the following setup for this example:

1. Disk 0 (ide0.0) is an AmigaOS partition setup which contains only AmigaOS.
2. Disk 1 (ide1.0) is a DOS partition setup with Linux on it.
3. You want option 0 to boot AmigaOS, and Option 1 to boot Linux.

From the CFE prompt, enter the following commands:

```
CFE> setenv -p MENU_0_LABEL “AmigaOS 4.1”
CFE> setenv -p MENU_0_COMMAND “boot -fs=amigafs ide0.0:amigaboot.of”
CFE> setenv -p MENU_1_LABEL “Linux Debian”
CFE> setenv -p MENU_1_COMMAND “boot -fs=ext2 -noints ide1.0:vmlinux”
```

Once you have typed these (do not type the “CFE>” at the beginning, that is the firmware prompt), enter the 'printenv' command and verify that the variables are there.

Time to test. From the CFE prompt, enter “menu”. The screen should show the bootscreen again, and a menu should appear with the above options.

Note the bar at the bottom. Once it reaches zero, the default selection (highlighted above) will be executed. If you press the SPACE key, the countdown will stop. The cursor keys control the highlighted option. Press up and down to move the bar around. Pressing the Return key, or the number in front of the option, will execute the option.

Pressing the 'F' key will get you back into the firmware prompt.

More Options

Two more options exist in the form of variables. One is the `MENU_TIMEOUT` variable. This determines how long the menu will wait until it automatically continues with the default option. The value is a number and specifies the number of seconds. It defaults to 10. To set the timeout to 5 seconds, enter

```
CFE> setenv -p MENU_TIMEOUT 5
```

To get back the default, delete the variable with the unsetenv command:

```
CFE> unsetenv MENU_TIMEOUT
```

The other option is the `MENU_DEFAULT` variable. This variable specifies the number of the selection that should be booted by default. In the example above, if you wanted to boot Linux instead of AmigaOS by default, you would have to type

```
CFE> setenv -p MENU_DEFAULT 1
```

As with the timeout, the default can be deleted by the `unsetenv` command.

More on Booting

Booting on CFE is done (usually) by running a program from a disk or other medium. The program is usually an ELF binary, and the normal procedure is to load it form a disk.

Disks and other storage media are represented by devices in CFE. Typically, the command `show devices` will give you a list of devices that are present, although only some of them can be used for booting.

IDE and SATA disks are usually called `ideM.N:`, where `M` is the controller number and `N` is the disk number. Sometimes, you can add a comma followed by a number or name after this. For example, `ideM.N,dh0:` would access partition dh0: on an AmigaOS disk by name.

CD- and DVD-Drives are usually called `atapiM.N:` with the same semantics for `M` and `N`. They usually are not partitioned.

In order to boot something from a disk, you have to know the disk's formatting. What that means is that the file system used on the disk must be known – there is no code to autodetect that. If you have an AmigaOS disk, the file system is usually `amigafs`. You can verify this with the `dir` command that lists all files in a directory on the disk:

```
CFE> dir -fs=amigafs ide0.1:Kickstart
```

will list all files in the `Kickstart` drawer on `ide0.1`.

Booting is done via the `boot` command. This usually points the firmware to an ELF file that should be loaded and will continue the boot process. For AmigaOS, this is the `amigaboot.of` program. For Linux, it usually is the Linux kernel `vmlinux`.

A booting command line usually consists of a file system or method, a location and a file.

Method/File System

There's two principle different ways to boot: From a device, or from a network. Booting from a device requires knowledge of the file system. Typically, file systems are either `amigafs, ext2, fat,` or `iso` (ISO9660). The file system is specified via the `-fs=<id>` command line option.

Booting from a network is similar, but different options are used. Typically, booting is done from a TFTP server. To specify this, the option `-tftp` is used.

Location

The file location is always specified as a two-parter. Part one (before the colon) is the device or network address. Part two (after the colon) is the file name/path. This is best illustrated by examples, so here are a few:

`ide0.1:amigaboot.of` – load from device `ide0.1` a file called `amigaboot.of`

`cf0:vmlinux -` load a file `vmlinux` from ancmpavt flash card.

`usbdisk0:vmlinux` – load a file `vmlinux` from an usbstick

`192.168.0.1:amigaboot.of` – load `amigaboot.of` from a network TFTP server.

More command line options

A few additional command line options exist that can be given to boot. These options include:

- `-nousb` will disable USB while booting. `Amigaboot.of` does that automatically.
- `-noints` will disable interrupts when booting. This option is usually used when booting a linux kernel
- `-elf` specifies that the file type is elf. This is usually the default, but should be specified when booting Linux.

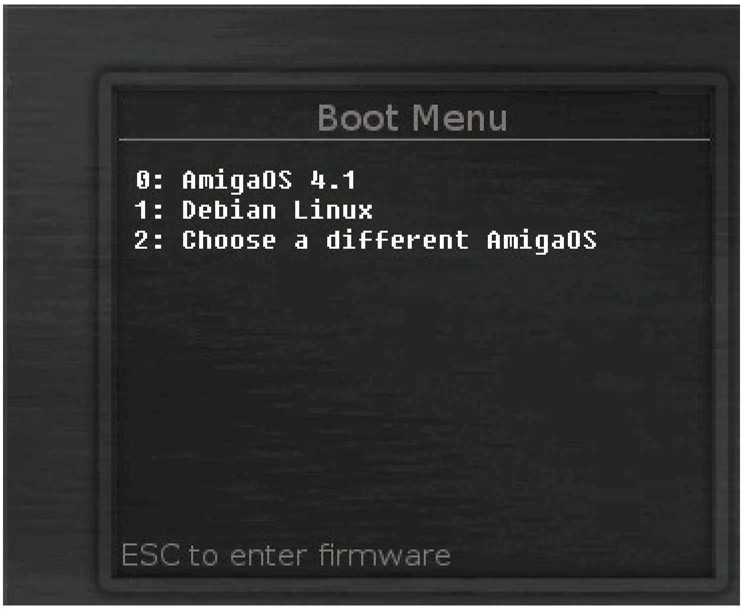
Example Setup

Here's a complete example setup of a menu . By now, you should know what most of these lines mean and what they do. For this example, we assume the following set-up:

We have a single SATA disk connected to port #0. The disk is partitioned into an AmigaOS setup (Workbench: and Work:) and a Linux setup (a / and /swap partition). Both AmigaBoot and the Linux kernel are stored on the Workbench: partition.

The following commands set up a menu with two entries: 0 will boot AmigaOS, while 1 will boot Linux. We want to boot silently (i.e. amigaboot should not ask for a config). We also want a third entry that will present us with all AmigaOS configs via the normal amigaboot menu.

Our menu will look like this:



The 0 option will boot into AmigaOS 4.1 without any further menu being displayed from AmigaOS. We can achieve this by setting the amigaboot_quiet variable to 'Y'.

For option 1, we need to have the bootargs variable already set up. This can be done outside the menu since the bootargs variable is exclusively used for booting linux.

Finally, for option 2, we want the normal amigaboot menu to appear. Since we will not set the amigaboot_quiet variable “persistently” (i.e. store it in the flash memory), this will be a simple matter.

Enter the following on the CFE command line, one line at a time. Do not enter the “CFE>” since that is the command prompt. It is merely there to make it clear where individual commands starts, since some can get longer than a single line in the formatting of this document.

```
CFE> setenv -p MENU_0_LABEL “AmigaOS 4.1”
CFE> setenv -p MENU_1_LABEL “Debian Linux”
CFE> setenv -p MENU_2_LABEL “Choose a different AmigaOS”
CFE> setenv -p MENU_0_COMMAND “setenv amigaboot_quiet Y ; boot -fs=amigafs
ide0.0:amigaboot.of”
```

```
CFE> setenv -p MENU_1_COMMAND “boot -elf -noints -fatfs cf0:vmlinux”
CFE> setenv -p MENU_2_COMMAND “boot -fs=amigafs ide0.0:amigaboot.of”
CFE> setenv -p STARTUP menu
CFE> setenv -p bootargs “root=/dev/sdd1”
```

As has been said, most of the stuff should be clear already. The '`setenv`' command is used with and without the `-p` option. The 'p' stands for 'persistent' and means that the setting is actually stored in the flash memory of the AmigaONE X1000 board. This means that the setting will be there after the machine has reset or was switched off. Without the option, the setting will be “forgotten” after a reset or powerdown.

The `STARTUP` variable is a CFE feature. It specifies a command, or series of commands, that are executed upon power-on. In our case, that means the `menu` command is executed right after the initial bring-up of the machine is finished.

Finally, the `bootargs` variable is used for Linux. The provided example may need alteration on your system, and depends on the specific set up of the harddisk.

Conclusion

The multiboot menu provides a user-friendly way of booting your machine. Even if you only use AmigaOS or Linux exclusively, it can be used to boot multiple configurations. In essence, it can be used for any purpose since it merely executes commands that are specified via variables,