003917b8-0

Derek Piper

COLLABORATORS

	TITLE : 003917b8-0	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Derek Piper	November 2, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

003917b8-0

1.1 BootController

Boot Controller V2.0 - Copyright D.Piper 1994,1995

DISCLAIMER

Neither I Derek Piper or whoever supplies this program can take any responsibility for any damage caused whether caused by this program or errors in this documentation. If anything untoward should happen then the user can seek no recourse against myself, Derek Piper or the supplier of this software.

The previous version of BootController was OS1.3 and OS2.0+ compatible. Due to the rewrite and subsequent features used the program is now OS2.0+ only. I know this may seem a bit nasty but we can't live in the past forever. I had an A500 too for 4 years but technology marches on.

1.2 Introduction

I wrote this program as an attempt to produce a useful and usable startup protection program for myself. I experimented with passwords and encryption and as I added features I realised "Hey, this isn't too bad" and so I decided to smarten up the program and add more features. The first version of this program was written in AMOS and as such was far from system friendly. I rewrote this program in C as a programming exercise and to improve its usefulness and also remove some bugs.

As you can see this program has quite a lot of features which I will go through in a minute. First of all I have to say that although this program is very difficult to break, it is possible to remove the protection it provides without too much hassle (if you know what you are doing) but it keeps prying eyes from accessing your data too easily. If you use it you are advised to lock away all bootable disks as they can provide an easy means of getting into your system.

The password is a character string of a maximum of 30 characters although I recommend about 8-12 characters for ease of typing !!.

This password controls access to the preferences, the system locking toggle and the change password functions. When the system is 'Locked' the password will have to be input before the program will let you carry on booting. If

you do not know the password the only way to exit is to reset.

1.3 Features

```
                  Note : As explained at the start, BootController 2.0 is now OS2 ↩
                     .0+ only.

       Features BootController 2.0

  Protected password entry      (ie. '*' characters instead of letters)
  Adjustable timeout            (0-255 seconds)
  Encrypted password file
  Encrypted user file
               (registered version)
                 Vector Checking
               (registered version)
                 Last Alert display
               (registered version)
                 Failed Attempts log file
               (registered version)
                 Keyboard shortcuts
  12 / 24 Hour Clock
  Two date display formats
  Memory display
  Attractive GadTools GUI - Designed with
              GadToolsBox
                      Required by BootController 2.0 before running

  SetClock Load to initialise the system clock
  Assigning of Env:
  Diskfont.library (V37+) in Libs:
  WB_Font/8 in Fonts: (essential)

       Recommended to run before BootController

  SetPatch
  RTPatch to improve requesters (ie. centring in screen)
```

Also it is important that no programs before BootController produce any output to the screen. In doing so you bring up a console window which means that the startup may be halted by pressing Ctrl-D.

1.4 Aminet

The Aminet is the largest collection of Amiga Shareware and Public Domain software around. It has its main home in Illinois at (ftp.wustl.edu). There are many mirror sites around the world. As well as this there are CDs of its contents produced regularly.

Some Aminet mirror sites :

```
        UK    - micros.hensa.ac.uk
        UK    - ftp.src.doc.ic.ac.uk
```

1.5 GadToolsBox

The interface was designed and implemented with GadToolsBox, written by Jan van den Baard. GadToolsBox makes interface design relatively painless.

1.6 Installation

Simply run the Install script to copy the files into their correct places. After thst has completed then you need to edit your Startup-Sequence.

For proper usage the first few lines should look like this :

```
SetClock Load                          - Load the current time.

Assign Env: Sys:EnvArc                 - Assign Env: before program start.

C:BootController                       - Or whatever you want to call the
                                         Boot Controller program.

...........                            - The next lines will only be
                                         reached if the user makes it
                                         through the program.
```

The default password for the program can be found in the DestroyMe.txt This password is no longer useful after changing the password from the program.

If you have registered copy your BC.User file into S:

N.B. You should already have the Diskfont.library in Libs: which the program needs in order to run. Version 37 or higher is required (ie. OS 2.04+)

1.7 Controls

The gadgets on the main display are easy to use. But here is a description of them for reference.

```
        About           - Fairly obvious this one, this tells you about the
                          program, its compilation date for example.

        Lock/Unlock     - This toggles the state of the system into being
                          locked (user must enter password before
                          continuing to boot) or unlocked (user can
                          continue boot without inputting password).

        Preferences     - This takes you to the
                preferences
```

screen. This is
not available unless you enter the password.

Enter Password - In order to access the main program functions, or to boot if the system is locked, you must enter the current password. Initially this is the default password.

Change Password - This allows you to (unsuprisingly) change the current password to anything you like. It requires you enter the original password and then its replacement which you must verify.

Continue Boot - This merely exits the program and allows the Startup-Sequence to continue. This is the basic principle behind the program. If the system is locked then you will have to enter the password to allow this option to work.

The preferences file is written whenever you click 'Save' in the prefs window or exit the program 'Continue Boot'. There are keyboard shortcuts for all of the programs gadgets which are indicated by underscores below the shortcut key.

1.8 Preferences

The preferences window comprises a series of checkboxes which ↩
toggle
various options.

12 Hour Clock format - This controls whether the clock display is in 12 or 24 hour format.

Written Date format - This toggles the date type between a full written date or number date.

Timeout - This selects whether the timeout is active on startup or not. The number gadget sets how many seconds before the program exits. Timeout only functons if the system is unlocked.

Memory Display - Toggles the memory display on and off.

* Vector Checking - This toggles the display of the reset vectors address and value. It also turns the checking of their state on and off.

* Display Last Alert - Toggles the display of the last alert. The display of the last alert is the standard $alertnumber.$taskaddress.

* Failed Attempts Log - Toggles the generation of a log file of failed attempts at password entry. The log is only generated when failing to input

the
correct password in 'Enter Password'.

All those marked * are only available in the
registered
version.

The 'Save' and 'Cancel' options should be obvious. When clicking on 'Save' the options are saved immediately to disk.

1.9 Unregistered

The unregistered version has been crippled, a shame this has to be done.
Three features are disabled, the Vector Checking, Last Alert display and and the Failed Attempts log file.

The unregistered version cannot be used after December 1996 as an added encouragement to register and not to use unregistered software.

So do yourself a favour and
register
now for the paltry sum of £5, sorry,
no other currencies accepted.

1.10 Registration

If you like and use this program on your system then send £5, a disk and an SAE to me and you will receive a registered user file along with the latest version of the program. The registered version has three other (very useful) options and of course has no 'Unregistered' statement at the front.

The 5 pounds can be sent as cash (disguise it well though!!)
or cheque (Payable to 'Derek Piper')

Sorry, but no other currencies accepted.

Please support the shareware ethic as it encourages programmers to try harder in the future to improve and upgrade their software.

Send registration fees with SAE and disk to :

Derek Piper,
63 Hadley Road,
New Barnet,
Herts.
EN5 5QU
LONDON
ENGLAND

1.11 User files

The user file is used for the registration of programs. It is easier to do it this way than have a separate version of the program.

N.B. If the file is damaged then the program will inform you. If no file is present then the program is deemed to be unregistered.

If you register you will receive a customised user file that has your name encrypted within the file and various flags which enable the extra features. See Registration for more details.

Any registered user can just send me an SAE with a disk and ask me for a new user file and the latest version of the program. No more money need be sent as I will have your name on file. I will not change the name in the user file to prevent spreading of unauthorised user files.

1.12 Contact

If there is anything you would like to see changed or added the don't hesitate to contact me, either by e-mail (preferred, due to to quicker response) or snail mail. Any bug reports are of course important so let me know if anything goes awry.

Snail mail : Derek Piper,
63 Hadley Road,
New Barnet,
Herts.
EN5 5QU
LONDON
ENGLAND

E-Mail : se2dp@dmu.ac.uk

Thankyou for taking the trouble to look at BootController, I hope you will use (and register :-)) it.

Derek Piper.

1.13 Future

In the future I may change some options or add features like

* Password controlled partition locking - This was in 1.2, I may reinstate this feature. If anyone likes it, let me know.

* Program menus - To control features of the program, ie flush vectors.

* Definable colours - Customise BootController to suit.

1.14 GCC

The GCC version I used was 2.6.3. This Amiga port of the well known GNU C compiler is available from the Aminet in dev/gcc. A new version 2.6.4 or later may be out by now.

GCC is a common standard on UNIX and other platforms which allows simple programs to be easily ported between machines.

1.15 AMOS

AMOS is now being discontinued but it was, and still is a very good implementation of the BASIC programming language. AMOS is of most use to games programmers as it can produce very good programs quickly and simply with the minimum of effort. AMOS was originally written by Mandarin Software but got bought out by Europress Software. I used AMOS Professional to write BootController 1.0 and 1.2. I also wrote two games which are on the Aminet.
They are :

game/wb/mimic.lha - A Simon Says type game.

game/board/codefinder.lha - A Mastermind type game.

1.16 Programmers

BootController 2.0 was written in C. I decided to do the rewrite as the original AMOS version was not very system friendly and had some large bugs in it. BootController 2.0 was compiled with GCC. The interface was designed with

GadToolsBox. The program length after compilation was 31k, a saving of around 80k from the original AMOS version.

When BootController loads it will try to open a custom screen with a font WB_Font 8. If it fails to locate this or encounters any other errors then it will bring up a requester to warn of its failure. As this is a password program the program cannot exit. You will have to reboot your machine with your Workbench disk and install the font. Any font can be used as long as it is a fixed-width 8-point font.

The preferences file and the user file are encrypted in a special way and appear completely different and random every time they are saved. The

preferences file is rewritten every time the program exits. If someone does manage to crack the password files I would be interested in their method (it would have to be very clever).

I cannot distribute the source with this archive but if anyone wants to take a look at a stripped down program (without password related functions) then let me know.

The two versions included are for 68000 and 68020 or better processors. There is not much difference in speed but the size of the executables is different.

The program has been tested well during development and causes no memory loss after running and exiting.

I learnt C as I programmed BootController, the two books I used were very useful. They were :

The C Programming Language - Brian W. Kernighan & Dennis M. Ritchie
Published by Prentice Hall ISBN 0-13-110362-8

Amiga ROM Kernal Reference Manual, Libraries - Commodore-Amiga, Inc.
Published by Addison Wesley ISBN 0-201-56774-1

I highly recommend these books to anyone wishing to learn about C and programming the Amiga.

The program was written and tested on a 6MB A1200, 120MB HD. The program is 2.0+ only now as explained before.

1.17 Bugs

I have listed here any problems I have with BootController. If anyone has got any good solutions to any problem then I would be grateful to know how to fix them.

Bug #1 - Display Last Alert (Registered versions only)

After running BootController the Last Alert feature seems to clear the address $100 which holds the alert type number. The down side to this is that programs like
LastAlert2
cannot read the alert number. I have altered it so that if you disable the 'Display Last Alert' feature then this interference does not occur. I use
MCP
s 'gurulist' feature to record the alerts anyway so this does not affect me.

1.18 LastAlert2

LastAlert2 is a program for giving extra information about the last alert
or guru suffered by the system. It is written by Jorgen Da Larsen and is available from the
Aminet
in util/moni.

1.19 MCP

MCP is a multi-function program with many good features. It performs many
operations that usually need a multitude of other programs to do. It is written by Stefan Sommerfeld and Michael Knoke and is available from the

Aminet
in util/wb.

1.20 Troubleshooting

Q : When I run the program it just crashes the machine. What's wrong ?

A : Are you using the correct version of the program for your processor ?
You can tell which version you have by typing :

version C:BootController

from the Shell. You must be using OS2.0+ to run BootController.

Q : When I run the program all I get is a requester telling me something went wrong. I have to reset from this. What's wrong ?

A : If an error occurs then the program has to halt the startup as it could be the result of hacking that the program went wrong. Possible errors at startup are :

Couldn't open the font	- The font WB_Font/8 HAS to be in Fonts: The installer copies this font to the correct place.
Low Chip Ram	- Not enough memory to open the screen. This shouldn't happen when booting as that's when most of the ram is available.

Q : I have registered but I lost my user file, what do I do ?

A : Just write to me and send a blank disk and an SAE and as I will have your name on file I will send you a new user file.

Q : Where do I get updates of the program ?

A : Updates will be posted to
Aminet
in util/boot. Registered users will
be able to use their user files to unlock subsequent versions of the program.

Q : The program reports that it is 'past its run-time' what does this mean ?

A : It means that the shareware demo is over and you will have to register in order to enable it. This happens after December 1996.

Q : How do I register ?

A : See the
registration
part of this guide.

1.21 History

Version	Release Date	Notes
1.0	October 1994	First version. Written in AMOS. Password and preference file encryption. Lock partitions. User files. Lock/Unlock. 12/24 hour clock, 2 Date formats.
1.2	January 1995	Added password verification.
2.0	October 1995	Complete rewrite in C. Compiled with GCC. All features of previous versions except partition locking.