# Setup Bochs emulator to run Pick R83 3.1M

Rev: 0.3

## Background

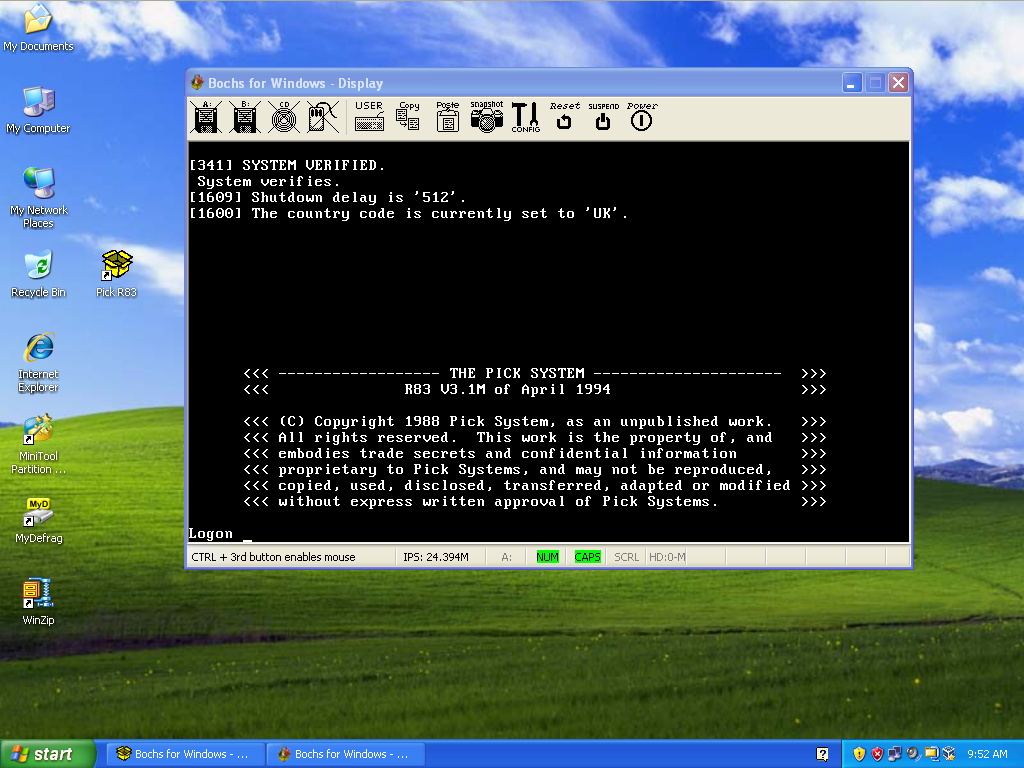
Bochs (http://bochs.sourceforge.net/) is a highly portable open source IA-32 (x86) PC emulator written in C++, that runs on most popular platforms. It includes emulation of the Intel x86 CPU, common I/O devices, and has a custom BIOS.

We can use it to emulate a Pentium 4 system, even though the host platform may be far more powerful (eg: multi-core) system that would not normally permit R83 to load. This enables Pick R83 to be loaded and run on a modern host system, alleviating the requirement for building a separate 386/486 system.

This document shows how to setup Bochs on a Windows host system, and to load R83.

## What is the end result?

The point of all this is to allow you to run Pick R83 within a Windows environment. It is not perfect – no testing has been made to determine if multi-user mode works, nor have attempts been made to connect devices like tapes. The image below shows R83 running within Windows XP.



## Get Bochs emulator

Download the latest copy of the Bochs emulator from:

<http://bochs.sourceforge.net/cgi-bin/topper.pl?name=See+All+Releases&url=http://sourceforge.net/projects/bochs/files>

For this document, it is assumed that file **Bochs-2.5.1.exe** was selected for download. If you download a different version you must be aware that the setup dialog screens may have changed from those presented in this document.

***\*\*NOTE: Only Bochs versions 2.5 and later will allow R83 to load properly.***

## Install Emulator

Double-click on the downloaded file to install it. It is fine to accept the default values presented in the installer. The default install will put the Bochs software into: **C:\Program Files\Bochs-2.5.1**

## Create a working directory for the R83 system

This document assumes the R83 system will be created in a directory called **C:\PickR83** if you choose something else, be sure to substitute your path for all future references to **C:\PickR83** in this document.

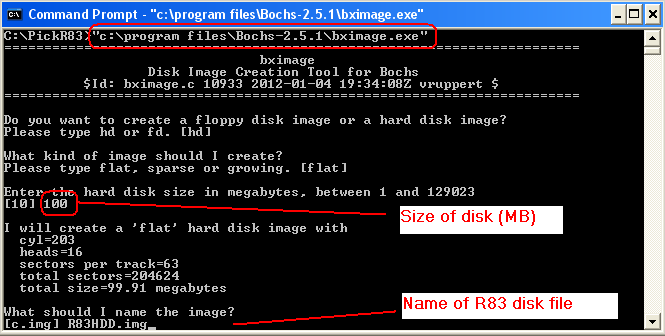
Use Windows explorer, or a DOS command window to create a new directory **C:\PickR83**

## Create a hard disk for R83

Before attempting to setup the emulator, we will create a hard-drive (really just a file) for the R83 system to install on.

This is easy to do from the command line. Open a DOS window and navigate to the **C:\PickR83** directory created in the previous step. Run the **bximage** tool to create the file that represents the hard-drive to be used for R83. This tool is located in the Bochs installation director, so you will need to reference it by including the path (see image)

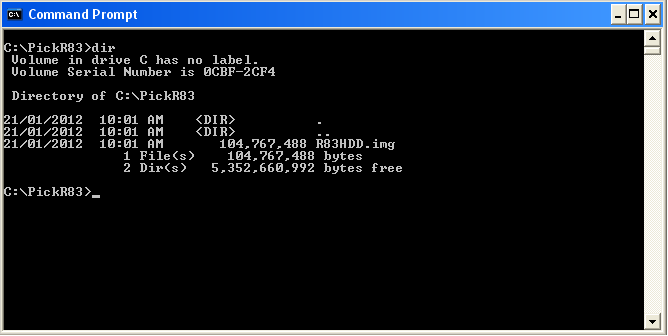
Hit RETURN to accept the defaults for prompts *Please type hd or fd. [hd]* and *Please type flat, sparse or growing. [flat]*



The image shows the **bximage** tool being run, and a 100MB disk (file) being created with a name of **R83HDD.img** – This name is used elsewhere in this document, so if you use another name – be sure to substitute your name as required. Feel free to choose a different size for your disk.

*\*\*IMPORTANT NOTE: Take care to note down the disk geometry values (cylinders, heads, sectors per track), as we will need these numbers later.*

The directory will now contain a single file called **R83HDD.img** – this is the ‘hard disk’ that R83 will be installed onto.

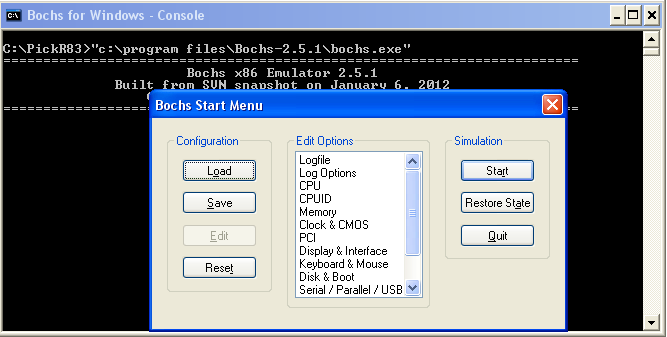


## Setup to Emulator (create a config file)

From the DOS window, in the **C:\PickR83** directory, invoke the Bochs emulator by type the command:

**"c:\program files\Bochs-2.5.1\bochs.exe"**

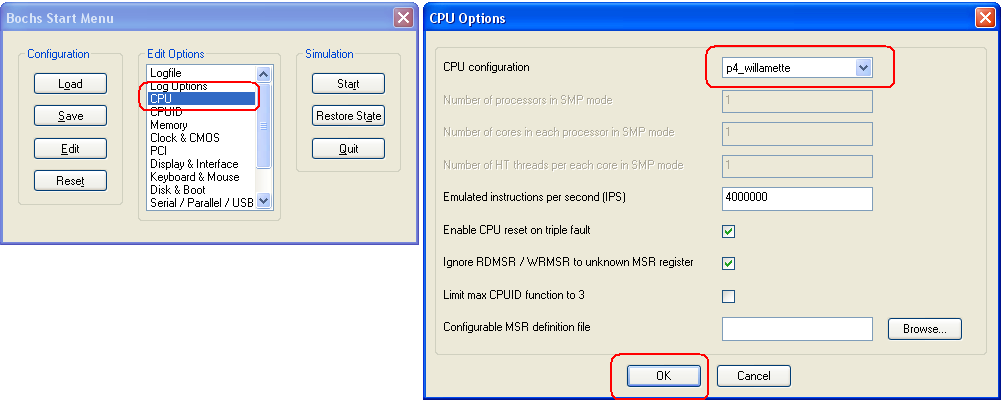
This will invoke the configuration/startup gui.



We must now create a configuration file that will enable R83 to load/run. This file will be stored in the **C:\PickR83** directory. Note that only some of the configuration options need to be changed from their default values.

### CPU

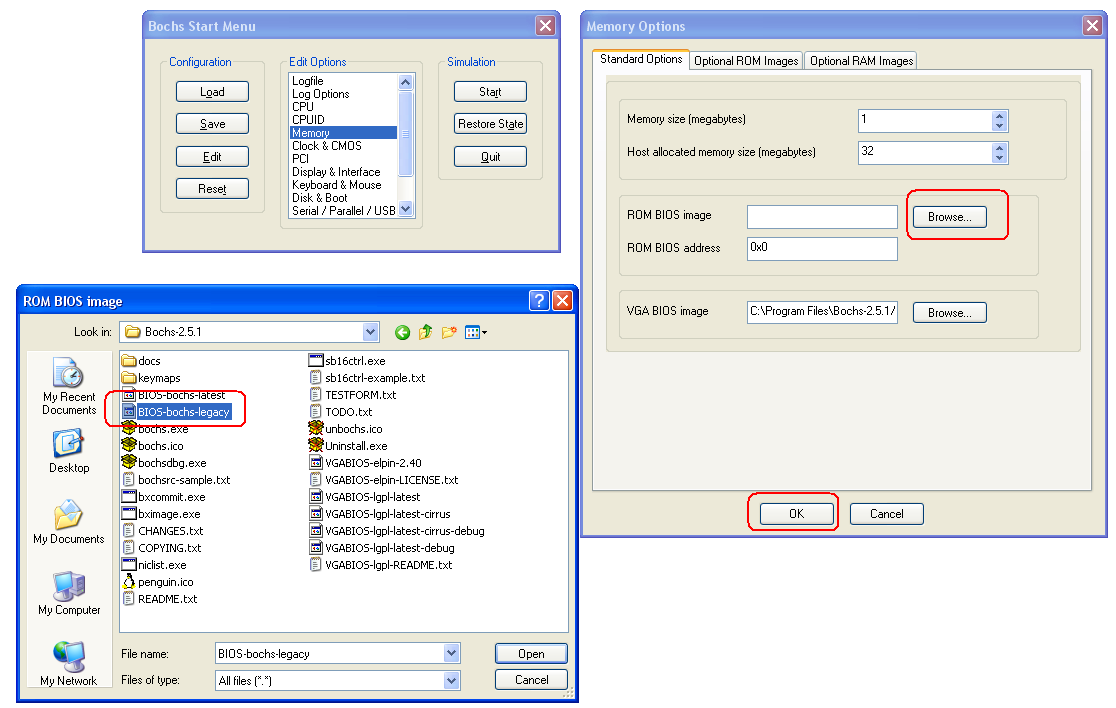
Double-click on the CPU option, and select the Pentium 4 from the list of available options.



Press OK to save your selection.

### Memory

Double-click on the Memory option. We need to change the *ROM BIOS image* value. Clear/delete the existing value from this field (avoids a UI bug) then press the Browse button and select the **BIOS-bochs-legacy** file from the Bochs installation directory **C:\Program Files\Bochs-2.5.1** (Note: Be sure to use the ‘legacy’ BIOS file and not the ‘latest’).



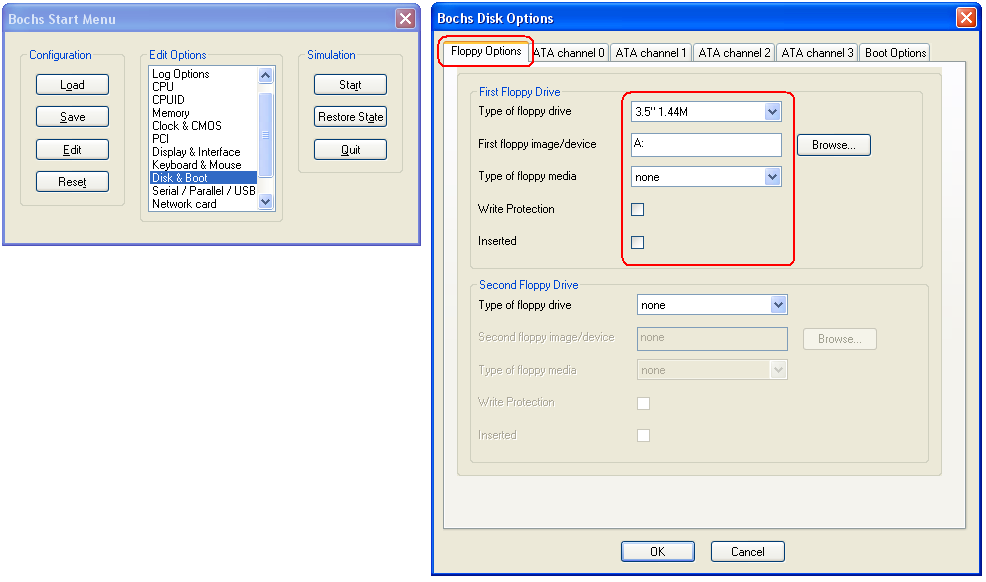
Press OK to save your selection.

### Disk & Boot

Double-click on the **Disk & Boot** option. This is where we tell the emulator where the hard-disk file is, what floppy drives there are, and the ‘boot order’.

**Floppy Drive**

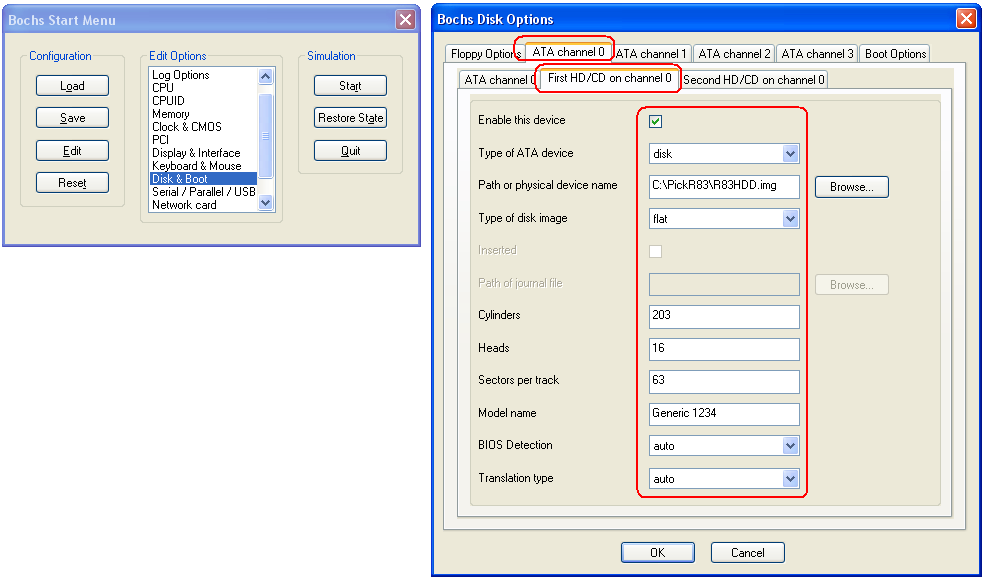
Ensure you can see the **Floppy Options** tab. Define a floppy device.



Define 1 floppy-disk drive. We need this in order to ‘boot’ the R83 system. In this example we are defining a 3.5” 1.44MB drive. It is assumed the host system has such a FDD installed as drive ‘A’. Note that for a physical floppy drive, the *First floppy image/device* input is A: and should not include a ‘\’ (ie enter ‘A:’ and not ‘A:\’)

**Hard Disk**

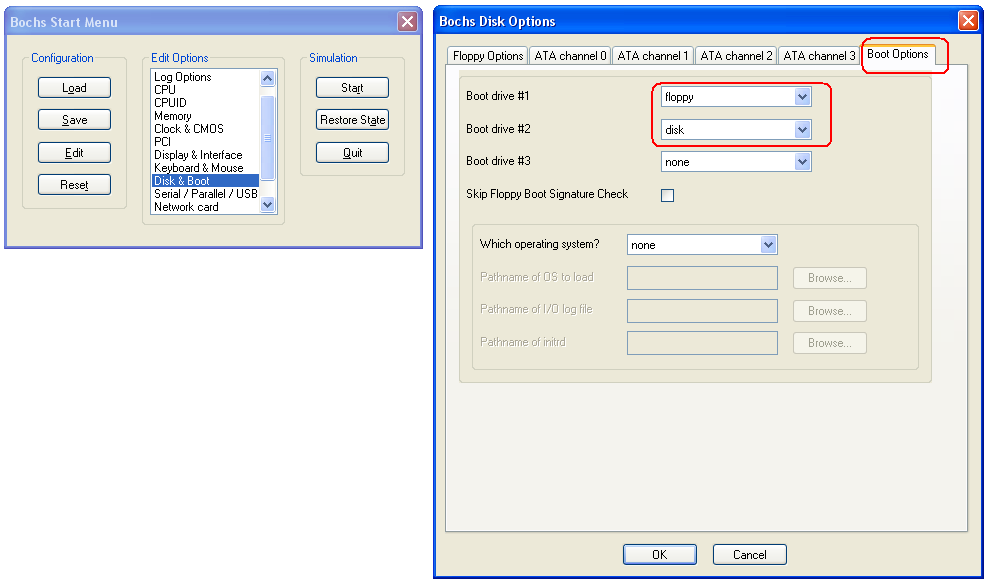
Do not press OK to save, but rather click on the next tab called **ATA Channel 0** – This is where we define the hard drive. Click on the sub tab called **First HD/CD on channel 0**



Check the *Enable this device* checkbox, and set the *Path or physical device name* to be the full path to the hard disk file you created earlier. If you are using the names from this document, this path will be **C:\PickR83\R83HDD.img**

Enter the drive geometry values that you saved when the hard disk file was created.

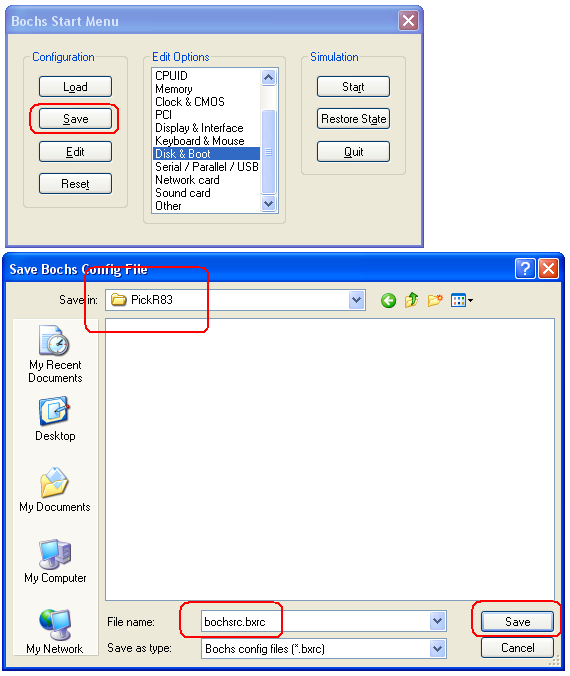
Do not press OK to save, but rather click on the tab called **Boot Options**. Here we say that we want to boot first from floppy, then (if no bootable floppy present) from hard disk.



Press OK to save your selections.

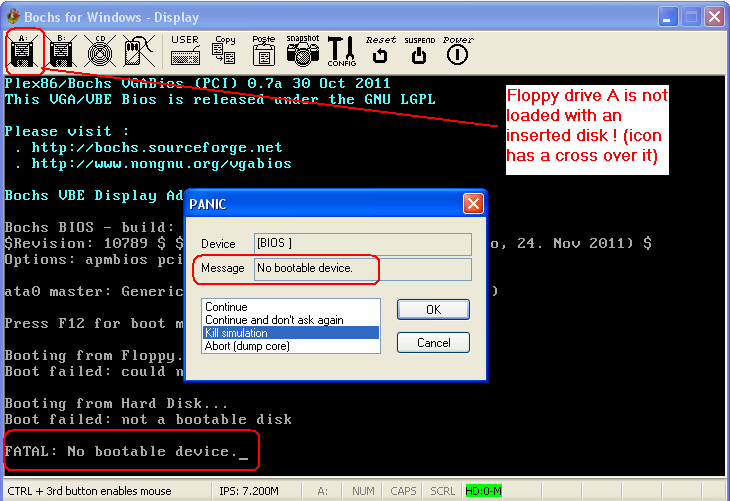
### Save the configuration file.

Having defined a configuration for the emulator, you should save this to the **C:\PickR83** directory – to avoid confusing it with any other configuration file you may define in the future. To do this, press Save and browse to the **C:\PickR83** directory. Save the configuration file using the default name.



## Boot R83

Having saved the configuration file, you can now press Start to run the emulator. However, it is likely you will see a screen like:

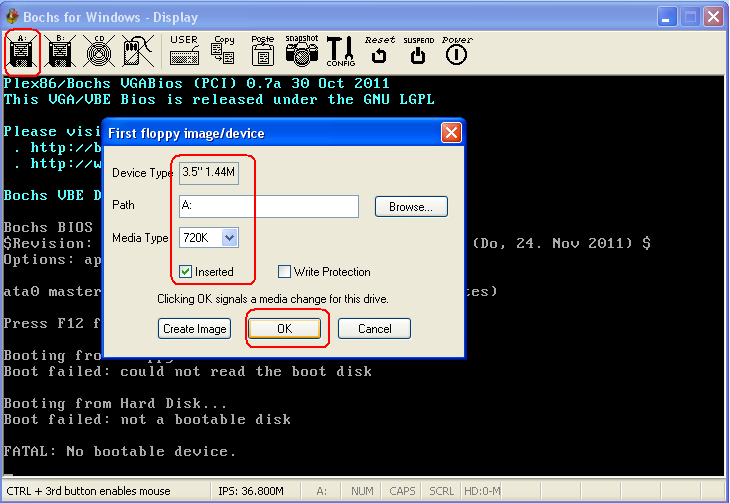


This screen is showing the emulator failing because there is no bootable floppy inserted, and because the hard drive has not yet had an operating system loaded, it too is not bootable. So, emulator crashes!

To fix this, insert your first Pick R83 system disk into your floppy drive.

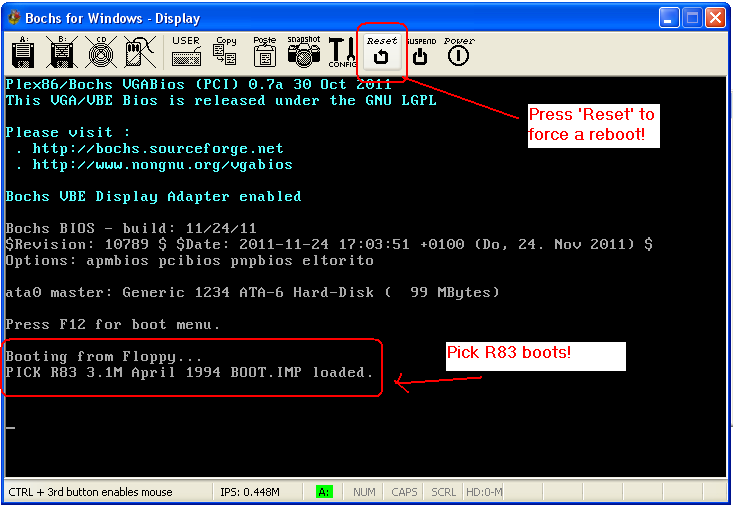
Select the ‘Continue’ option from the PANIC dialog and press the OK button. Now click on the A: drive icon at top left (the left-most floppy drive icon with a cross over it).

Ensure the correct floppy parameters are in the dialog that is presented (update as required) then press OK. Should look like:

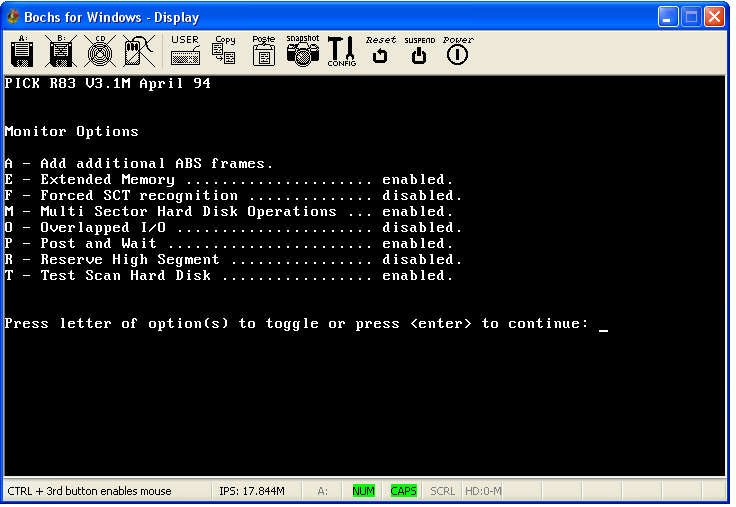


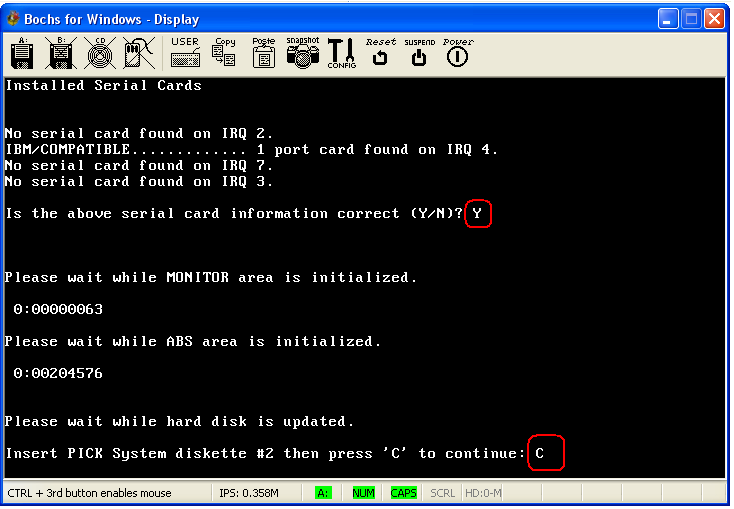
*\*\*NOTE: We have indicated that the media type is 720K which matches the Pick R83 boot media. Note that for a physical device, the path is ‘A:’ and not ‘A:\’*

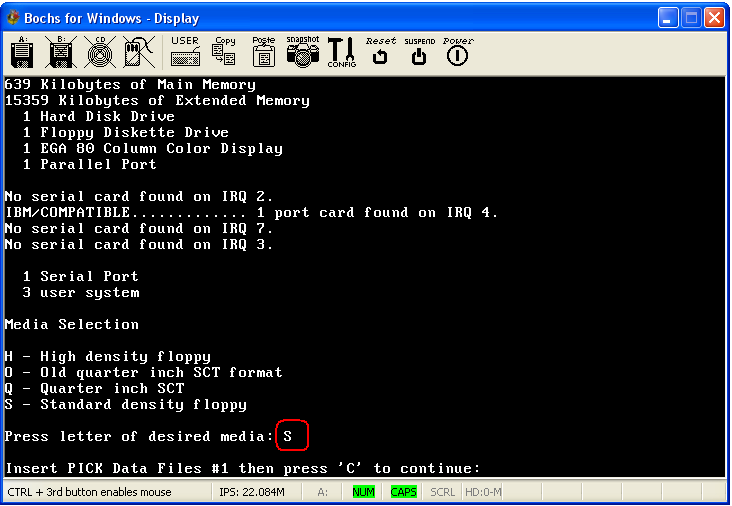
After pressing OK to the floppy setup, push the *Reset* icon to force the emulator to reboot – and hopefully boot from the inserted floppy!!!

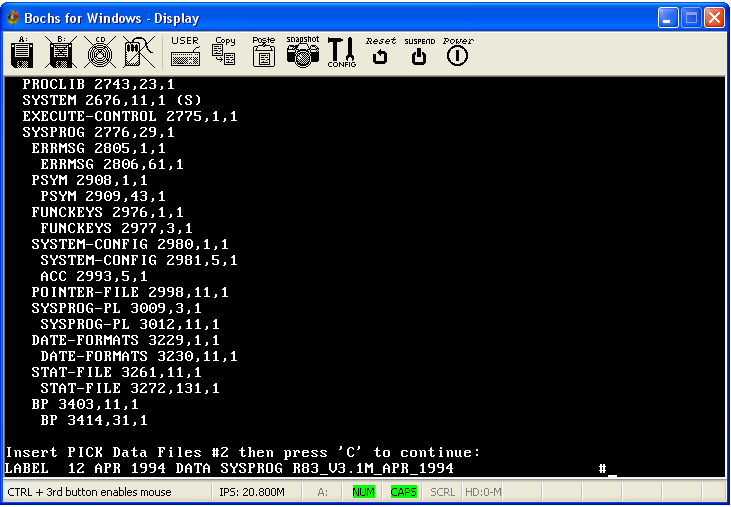


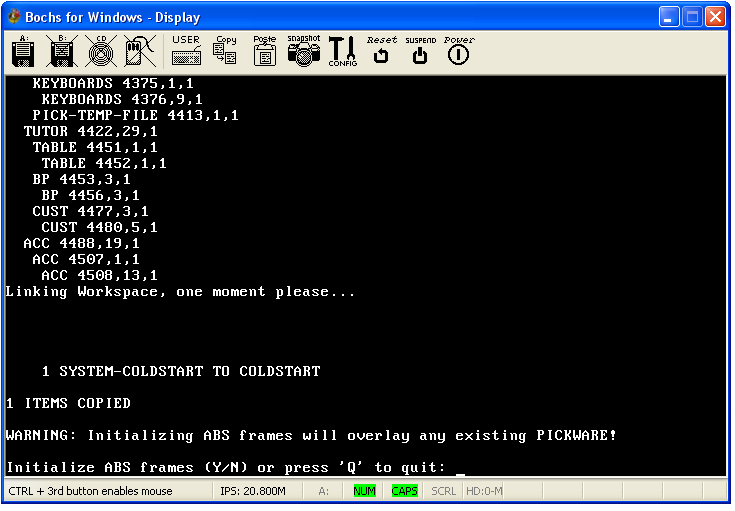
Boot process should continue as normal for R83



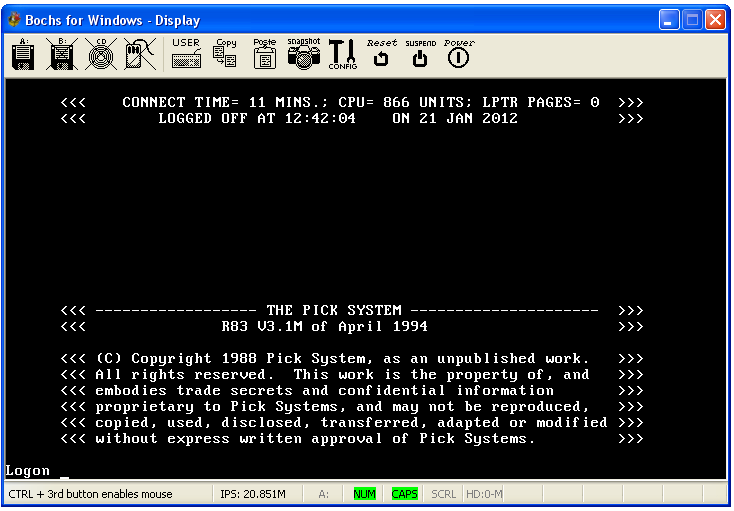








Answer all the remaining questions to complete the boot process and finally see the login prompt.



R83 is now loaded! To shutdown, login into the SYSPROG account and run the POWER-OFF verb. Then close the emulator window.

## Issues/Problems/Ideas

* I have seen situations where the configuration file seems to ‘forget’ some settings. If something is not working, use the *Load* button on the Bochs gui to force a particular config file to be loaded, make changes and then be sure to *Save* the configuration back to a file before pressing Start to initiate the emulator.
* The steps shown in this document are not the only way to get Pick R83 loaded, but represent one way that has been shown to work.
* Pick System/Data disks can/do wear out. Bochs is quite happy to use disk *image files* rather than ‘real’ floppy disks. These are both faster to load from, and safer – as your Pick disks can be put away. Converting Pick disks to disk image files is not covered here, and is left as an ‘exercise for the reader’ <g> - Products like ‘Floppy Image’ (<http://www.towodo.com/products/floppyimage/>) are useful for making disk image files. There are numerous other products like this.
* It is possible to copy from the Bochs install directory the 4 files **BIOS-bochs-legacy, bochs.exe, bximage.exe** and **VGABIOS-lgpl-latest** to the **C:\PickR83** directory. After updating the **bochsrc.bxrc** file to reference the local BIOS file this makes this directory a self-contained R83 system that can be saved to CD/DVD. Just need to double-click on **bochs.exe** to run up the system. An alternative might be to just create a ‘shortcut’ on your desktop that points to the **bochs.exe** file. This directory can be moved to another machine without having to download/install the emulator on that new box.
* If using a later version of Bochs, take care to note that the setup screens/dialogs will almost certainly have changed. However, there should be enough information here for you to complete the setup task.

## FAQ

**Q:** Why bother with R83 at all?

**A:** Just for fun really – to see if it was possible. Does not interest you? No problem – just move on.

**Q:** Why use Bochs? Why not VirtualBox or [name inserted] emulator?

**A:** Fair question. A couple of reasons. The first is that Bochs allows you to emulate a particular PC cpu architecture (eg: 386, 486 ..etc), rather than a generic PC virtual-machine. This is handy for running R83, as we may need to emulate a slower/older PC while running on a modern fast PC. Secondly, the BIOS in Bochs includes code for the interrupt 13 al=17h /18h calls which are required for R83 to install. At time of writing, VirtualBox BIOS (itself based on Bochs BIOS code) does not. This is not to say that no other PC emulator will work with R83. Feel free to try some others!

**Q:** Why are we using the *legacy* BIOS in Bochs, and not the *latest* BIOS?

**A:** As of Boch release 2.5.1, only the legacy BIOS has been amended to include the int13 al=17h /18h calls used by the R83 installer.

## Example bochsrc.bxrc configuration file

Contents of the bochs 2.5.1 configuration file created via the steps in this document, opened as a text file.

-------------------------------start------------------------------------

# configuration file generated by Bochs

plugin\_ctrl: unmapped=1, biosdev=1, speaker=1, extfpuirq=1, gameport=1

config\_interface: win32config

display\_library: win32

memory: host=32, guest=32

romimage: file="C:\Program Files\Bochs-2.5.1\BIOS-bochs-legacy"

vgaromimage: file="C:\Program Files\Bochs-2.5.1\VGABIOS-lgpl-latest"

boot: floppy, disk

floppy\_bootsig\_check: disabled=0

floppya: type=1\_44

# no floppyb

ata0: enabled=1, ioaddr1=0x1f0, ioaddr2=0x3f0, irq=14

ata0-master: type=disk, mode=flat, translation=auto, path="C:\PickR83\R83HDD.img", cylinders=203, heads=16, spt=63, biosdetect=auto, model="Generic 1234"

ata1: enabled=1, ioaddr1=0x170, ioaddr2=0x370, irq=15

ata2: enabled=0

ata3: enabled=0

parport1: enabled=1, file=""

parport2: enabled=0

com1: enabled=1, mode=null, dev=""

com2: enabled=0

com3: enabled=0

com4: enabled=0

usb\_uhci: enabled=0

usb\_ohci: enabled=0

usb\_xhci: enabled=0

pci: enabled=1, chipset=i440fx

vga: extension=vbe, update\_freq=5

cpu: count=1, ips=4000000, model=p4\_willamette, reset\_on\_triple\_fault=1, cpuid\_limit\_winnt=0, ignore\_bad\_msrs=1

cpuid: family=6, model=0x03, stepping=3, mmx=1, apic=xapic, sse=sse2, sse4a=0, sep=1, aes=0, xsave=0, xsaveopt=0, movbe=0, smep=0, x86\_64=1, 1g\_pages=0, pcid=0, fsgsbase=0, mwait=1, mwait\_is\_nop=0

cpuid: vendor\_string="GenuineIntel"

cpuid: brand\_string=" Intel(R) Pentium(R) 4 CPU "

print\_timestamps: enabled=0

port\_e9\_hack: enabled=0

private\_colormap: enabled=0

clock: sync=none, time0=local

# no cmosimage

ne2k: enabled=0

pnic: enabled=0

sb16: enabled=0

es1370: enabled=0

# no loader

log: -

logprefix: %t%e%d

panic: action=ask

error: action=report

info: action=report

debug: action=ignore

keyboard\_type: mf

keyboard\_serial\_delay: 250

keyboard\_paste\_delay: 100000

keyboard\_mapping: enabled=0, map=

user\_shortcut: keys=none

mouse: enabled=0, type=ps2, toggle=ctrl+mbutton

-------------------------------end-------------------------------------