G. Installing Pintos

This chapter explains how to install a Pintos development environment on your own machine. If you are using a Pintos development environment that has been set up by someone else, you do not need to read this chapter or follow these instructions.

The Pintos development environment is targeted at Unix-like systems. It has been most extensively tested on GNU/Linux, in particular the Debian and Ubuntu distributions, and Solaris. It is not designed to install under any form of Windows.

Prerequisites for installing a Pintos development environment include the following, on top of standard Unix utilities:

- Required: GCC. Version 4.0 or later is preferred. Version 3.3 or later should work. If the host machine has an 80*x*86 processor, then GCC should be available as gcc; otherwise, an 80*x*86 cross-compiler should be available as i386-elf-gcc. A sample set of commands for installing GCC 3.3.6 as a cross-compiler are included in "src/misc/gcc-3.3.6-cross-howto".
- Required: GNU binutils. Pintos uses addr2line, ar, ld, objcopy, and ranlib. If the host machine is not an 80*x*86, versions targeting 80*x*86 should be available with an "`i386-elf-`" prefix.
- Required: Perl. Version 5.8.0 or later is preferred. Version 5.6.1 or later should work.
- Required: GNU make, version 3.80 or later.
- Recommended: QEMU, version 0.8.0 or later. If QEMU is not available, Bochs can be used, but its slowness is frustrating.
- Recommended: GDB. GDB is helpful in debugging (see section E.5 GDB). If the host machine is not an 80*x*86, a version of GDB targeting 80*x*86 should be available as "`i386-elf-gdb`".
- Recommended: X. Being able to use an X server makes the virtual machine feel more like a physical machine, but it is not strictly necessary.
- Optional: Texinfo, version 4.5 or later. Texinfo is required to build the PDF version of the documentation.
- Optional: TeX. Also required to build the PDF version of the documentation.
- Optional: VMware Player. This is a third platform that can also be used to test Pintos.

Once these prerequisites are available, follow these instructions to install Pintos:

1. Install Bochs, version 2.2.6, as described below (see section G.1 Building Bochs for Pintos).
2. Install scripts from "src/utils". Copy "backtrace", "pintos", "pintos-gdb", "pintos-mkdisk", "pintos-set-cmdline", and "Pintos.pm" into the default PATH.
3. Install "src/misc/gdb-macros" in a public location. Then use a text editor to edit the installed copy of "pintos-gdb", changing the definition of GDBMACROS to point to where you installed "gdb-macros". Test the installation by running pintos-gdb without any arguments. If it does not complain about missing "gdb-macros", it is installed correctly.
4. Compile the remaining Pintos utilities by typing make in "src/utils". Install "squish-pty" somewhere in PATH. To support VMware Player, install "squish-unix". If your Perl is older than version 5.8.0, also install "setitimer-helper"; otherwise, it is unneeded.
5. Pintos should now be ready for use. If you have the Pintos reference solutions, which are provided only to faculty and their teaching assistants, then you may test your installation by running make check in the top-level "tests" directory. The tests take between 20 minutes and 1 hour to run, depending on the speed of your hardware.
6. Optional: Build the documentation, by running make dist in the top-level "doc" directory. This creates a "WWW" subdirectory within "doc" that contains HTML and PDF versions of the documentation, plus the design document templates and various hardware specifications referenced by the documentation. Building the PDF version of the manual requires Texinfo and TeX (see above). You may install "WWW" wherever you find most useful.

The "doc" directory is not included in the ".tar.gz" distributed for Pintos. It is in the Pintos CVS tree available via :pserver:anonymous@footstool.stanford.edu:/var/lib/cvs, in the pintos module. The CVS tree is *not* the authoritative source for Stanford course materials, which should be obtained from the course website.

G.1 Building Bochs for Pintos

Upstream Bochs has bugs and warts that should be fixed when used with Pintos. Thus, Bochs should be installed manually for use with Pintos, instead of using the packaged version of Bochs included with an operating system distribution.

Two different Bochs binaries should be installed. One, named simply bochs, should have the GDB stub enabled, by passing "`--enable-gdb-stub`" to the Bochs configure script. The other, named bochs-dbg, should have the internal debugger enabled, by passing "`--enable-debugger`" to configure. (The pintos script selects a binary based on the options passed to it.) In each case, the X, terminal, and "no GUI" interfaces should be configured, by passing "`--with-x --with-x11 --with-term --with-nogui`" to configure.

This version of Pintos is designed for use with Bochs 2.2.6. A number of patches for this version of Bochs are included in "src/misc":

"bochs-2.2.6-big-endian.patch"

> Makes the GDB stubs work on big-endian systems such as Solaris/Sparc, by doing proper byteswapping. It should be harmless elsewhere.

"bochs-2.2.6-jitter.patch"

> Adds the "jitter" feature, in which timer interrupts are delivered at random intervals (see section 1.1.4 Debugging versus Testing).

"bochs-2.2.6-triple-fault.patch"

> Causes Bochs to break to GDB when a triple fault occurs and the GDB stub is active (see section E.6 Triple Faults).

"bochs-2.2.6-ms-extensions.patch"

> Needed for Bochs to compile with GCC on some hosts. Probably harmless elsewhere.

"bochs-2.2.6-solaris-tty.patch"

> Needed for Bochs to compile in terminal support on Solaris hosts. Probably harmless elsewhere.

"bochs-2.2.6-page-fault-segv.patch"

> Makes the GDB stub report a SIGSEGV to the debugger when a page-fault exception occurs, instead of "signal 0." The former can be ignored with handle SIGSEGV nostop but the latter cannot.

"bochs-2.2.6-paranoia.patch"

> Fixes compile error with modern versions of GCC.

"bochs-2.2.6-solaris-link.patch"

> Needed on Solaris hosts. Do not apply it elsewhere.

To apply all the patches, cd into the Bochs directory, then type:

```
patch -p1 < $PINTOSDIR/src/misc/bochs-2.2.6-big-endian.patch
patch -p1 < $PINTOSDIR/src/misc/bochs-2.2.6-jitter.patch
patch -p1 < $PINTOSDIR/src/misc/bochs-2.2.6-triple-fault.patch
patch -p1 < $PINTOSDIR/src/misc/bochs-2.2.6-ms-extensions.patch
patch -p1 < $PINTOSDIR/src/misc/bochs-2.2.6-solaris-tty.patch
patch -p1 < $PINTOSDIR/src/misc/bochs-2.2.6-page-fault-segv.patch
patch -p1 < $PINTOSDIR/src/misc/bochs-2.2.6-paranoia.patch
patch -p1 < $PINTOSDIR/src/misc/bochs-2.2.6-solaris-link.patch
```

You will have to supply the proper $PINTOSDIR, of course. You can use patch's "`--dry-run`" option if you want to test whether the patches would apply cleanly before trying to apply them.
Sample commands to build and install Bochs for Pintos are supplied in "src/misc/bochs-2.2.6-build.sh".

This document was generated by on *December, 29 2009* using *texi2html*