How to configure Pintos on UB CSE servers

Luigi Di Tacchio

1 Introduction

This is a quick guide that will help you configure the environment to work with Pintos on almost any CSE server at UB.
If you want to configure Pintos on your own machine, please follow the instructions at the following URL:
`http://web.stanford.edu/class/cs140/projects/pintos/pintos_12.html`
In such a case, you will need to patch, build and install Bochs first, then install all Pintos scripts and executables. If you need help, please contact the instructor or the TAs.

Premise: The following guide adopts a C shell syntax, used on the majority of tested UB CSE servers. If you are trying to install Pintos on your own machine, you may be using a bash shell, in which case some commands will have a slightly different syntax (although the same semantics).

2 Step by step guide

1. Log into your favorite UB CSE server, enabling trusted X11 forwarding with the option *-Y*:

```
~$ ssh -Y ubitname@timberlake.cse.buffalo.edu
```

   Very important: if you are using PuTTY under Windows, X11 forwarding will not work "out of the box". You need to install XMing first. Just google "PuTTY X11 forwarding" and follow any tutorial to install it.

2. Create a directory for Pintos, where you will be working on (not mandatory but good practice). For example:

```
timberlake {~} > mkdir -p $HOME/projects/cse421/pintos
```

3. Set up the environment variables, opening $HOME/*.cshrc* with vim or nano and appending the following lines:

```
setenv PINTOSDIR ${HOME}/projects/cse421/pintos
setenv PATH /projects/luigidit/pintos/bin:${PATH}
setenv BXSHARE /projects/luigidit/pintos/share/bochs
```

   Replace the string after PINTOSDIR with the path to the directory created in step 2 (if different).
   Please, do not change the other two lines.

   Alternatively, you can execute the following commands:

```
> echo 'setenv PINTOSDIR ${HOME}/projects/cse421/pintos' >> .cshrc
> echo 'setenv PATH /projects/luigidit/pintos/bin:${PATH}' >> .cshrc
> echo 'setenv BXSHARE /projects/luigidit/pintos/share/bochs' >> .cshrc
```

   The effect will be the same: the previous three commands append the lines between single quotation marks at the end of .cshrc.

   In particular, the first command creates the environment variable PINTOSDIR, pointing to the directory created in step 2; the second command modifies PATH in order to include the directory where all Pintos scripts and executables are installed; finally, the third command creates the environment variable BXSHARE, pointing to the location of Bochs shared libraries.

   Attention: When issuing the previous commands, use single quotation marks like printed. DO NOT use double quotation marks unless you want to experience unexpected errors later on. Brief explanation: Using double quotation marks, the environment variables contained in the string (like ${HOME}) will be undesirably replaced with their actual values.

4. (**Necessary only for this ssh session. Next time you ssh you will not need to issue the following command**). Execute .cshrc:

```
> source .cshrc
```

5. cd into PINTOSDIR, then download and extract Pintos:

```
> cd $PINTOSDIR
> wget http://www.stanford.edu/class/cs140/projects/pintos/pintos.tar.gz
> tar --strip-components=1 -xzf pintos.tar.gz
```

   The option *--strip-components=1* discards the top directory (*pintos*) and extracts starting from the first level (directory *src*). You can take a look at the tar structure to understand the effect of the command.

6. Modify *src/Make.config*, forcing gcc to compile for i386. Search for the line *CC = $(CCPROG) -m32* and add *-march=i386* at the end of it. The resulting line should be *CC = $(CCPROG) -m32 -march=i386*.

   Alternatively, you can run the following command, that has the same effect:

```
> sed -i 's/CC = $(CCPROG) -m32/CC = $(CCPROG) -m32 -march=i386/g' src/Make.config
```

7. cd into *src/threads* and compile:

```
> cd src/threads
> make
```

8. Finally, cd into *build* and test Pintos:

```
> cd build
> pintos run alarm-multiple
```

   You should see a new window popping up, with Bochs loading Pintos and then running the alarm-multiple test.

   After the test is completed, you can click on *power* to turn off Bochs or just close the window.