Multiple Platforms

Author : Andras Tantos

Windows with Visual Studio

The simulator have always worked under Windows, that was my original development platform. The news here is that (after a comment on the site brought my attention to it) I’ve realized that Visual Studio 2012 doesn’t by default generate code that runs on WinXP. In the latest version, I’ve flipped the appropriate switches to support XP as well.

You’ll have to download and build Boost for the sources, the rest (including pdcurses) are part of the distribution.

The projects are set up to use Boost from it’s default install location (C:\Boost), so make sure you either install it there or change the location in common.props

When you run the simulator make sure that the binaries (especially the terminal emulator) is in the path, otherwise the simulator will just hang trying to launch it in a loop.

Windows and MinGW

If you want to use free tools to compile the simulator, your main choice is GCC. Of the two versions that are available under Windows I’ve tested MinGW (the other being Cygwin). There are makefiles provided that should work in a regular command line or under MSys.

Either GCC 4.7.x or 4.8.x should work. Others might work as well, but YMMV.

Just as for Visual Studio, you’ll have to download and build Boost.

Due to the funky way boost names its libraries, you’ll have to potentially update common.mak with the version of boost and GCC you’re using. The defaults are GCC 4.8 and Boost v1.53. To change them, edit these settings in common.mak:

```
GCC_VER=48
BOOST_VER=1_53
```

Alternatively you could build by specifying the right values from the command line. So for example to build with GCC version 4.7.2, you would do:

```
make GCC_VER=47
```

The makefile is set up to use Boost from it’s default install location (C:\Boost), so make sure you either install it there or change the location in common.mak

When you run the simulator make sure that the binaries (especially the terminal emulator) is in the path, otherwise the simulator will just hang trying to launch it in a loop.

Windows and Cygwin

The latest Cygwin release seems to work almost as well as the MinGW build does. The latest version comes with GCC 4.8.0 which should work. Just as with MinGW other versions might work as well, but I haven't tested them, and you'll need decent C++11 support. To build under this environment, you'll have to make sure you've installed the boost and ncurses libraries that come with the Cygwin distribution. Since the build system defaults to MinGW, when building under Cygwin, you'll have to specify your system on the command line:

```
make SYSTEM=cygwin
```

When you run the simulator make sure that the binaries (especially the terminal emulator) is in the path, otherwise the simulator will just hang trying to launch it in a loop.

Linux

I always intended the simulator to work under Linux, but until this latest release some bugs prevented it from working (same with MinGW by the way). With this release, Linux builds are functional as well.

To build you'll need a recent GCC: 4.7 or 4.8. As with MinGW, other versions might work too, but you need decent c++11 support.

Here's the simulator running on Debian Wheezy in it's full "glory":

On Debian, you’ll have to install the following packages:

gcc
gcc-4.7
make
libboost1.49-all-dev
libncurses5-dev

Other distributions should work as well, but you’ll have to make sure that the equivalent of these libraries are installed:

boost
ncurses

When you run the simulator make sure that the binaries (especially the terminal emulator) is in the path, otherwise the simulator will just hang trying to launch it in a loop.

Getting the package

As usual, you can grab the sources and the precompiled Windows binaries from the download page.