

Palm OS Emulator HOWTO

Jan Schaumann

06/12/00

Abstract

A brief overview of how to install and use the Palm OS Emulator (POSE) and other tools to write and compile software for the Palm OS on a Linux system.



Contents

1	Introduction	3
2	Installation	3
2.1	How to obtain POSE	3
2.2	Requirements	3
2.3	Compilation and Installation	3
3	Obtaining ROM image files	4
4	Other tools	5
4.1	Obtain the prc-tools	5
4.2	Obtaining the SDK	5
4.3	Obtaining and installing pilrc	6
5	An Example	6
5.1	Hello World Code	6
5.2	Hello World Compilation	7
5.3	Installing “Hello, World” on POSE	8
5.4	Keeping “Hello, World”	8
5.5	Deleting “Hello, World”	8
6	Resources	8

1 Introduction

So you've decided to start developing software for the Palm OS, running on all those nifty little handheld organizers, but you don't actually have a PalmPilot. Or maybe you're sick of always having to synch your Palm device after you wrote an application. That's when the Palm OS Emulator (POSE) comes into play.

The Palm OS Emulator is an application for desktop computers that emulates the operations of Palm Computing Platform hardware devices. With it, users can load ROM images, applications, and databases and use them on their desktop computers.

The procedures described in this document have been successful on a Linux 2.2.14-15mdksecure i686 system, but should not be much different on other systems (especially not other RedHat Linux systems). Most of the information provided in this document is part of the documentation that comes with POSE, other information has been compiled from various websites.

2 Installation

2.1 How to obtain POSE

POSE for Unix can be downloaded from

http://www.palmos.com/dev/tech/tools/emulator/emulator_src_30a5.tar.gz

However, in order to make sure that you get the most recent version of POSE it is probably a good idea to go to <http://www.palmos.com/dev/tech/tools/emulator/> and read carefully through the page and the relevant links.

2.2 Requirements

POSE uses pthreads, so you need GNU glibc2 (aka Linux libc6) C library and a 2.x kernel on a Linux system. The source makes heavy use of C++ templates and exceptions. On Linux, you must have egcs-1.1.1 or better to compile it. egcs-1.1 or any gcc below 2.95 may not work. RedHat Linux 6.0 comes with egcs 1.1.2, so you should be OK there. If you are running RedHat Linux 5.2, you will need to download the latest from <http://egcs.cygnum.com> and build it.

POSE also uses the FLTK X toolkit - you will need to download and install it before you attempt to build POSE on your system. The FLTK X toolkit is available from <http://www.fltk.org>, but if you happen to be on RedHat-Linux system, you might consider downloading the RPM from <http://rpmfind.doc.ic.ac.uk/fltk.html> (or your favorite RPM-mirror). I will not go into the details of installing the FLTK here, just make sure that "fluid" (the FLTK User Interface Design tool) is part of your package and gets installed, too.

2.3 Compilation and Installation

Unpack the tar.gz - on Linux systems this is easily done with the following command:

```
tar zxvf emulator_src_30a5.tar.gz
```

On other Unices, or if you have not installed GNUtar, you might need to unzip the package first and then untar it:

```
gzip -d emulator_src_30a5.tar.gz
tar xvf emulator_src_30a5.tar
```

This will extract the contents of the archive into a new directory “Emulator_Src_3.0a5”. Now cd into “Emulator_Src_3.0a5/Docs/” and read through the relevant documents. You will find that most of what you are reading right now was shamelessly stolen from those documents. After you have made sure that your system meets all the requirements, cd into “Emulator_Src_3.0a5/BuildUnix and execute the following commands:

```
./configure
make
make install
```

The last command will require superuser privileges, but this step is optional. However, I recommend installing POSE systemwide, in order to make your life easier. If you can’t do that, you need to either add the path to “Emulator_Src_3.0a5/BuildUnix/pose” into your \$PATH or you will always have to call POSE with an absolute pathname (ie ~/your/path/to/Emulator_Src_3.0a5/BuildUnix/pose). From now on, it is assumed that POSE has been installed systemwide and can be executed by simply typing:

```
pose
```

3 Obtaining ROM image files

Assuming everything went according to plan, you can now start POSE by simply issuing the command `pose`, and a window will open on your desktop, showing you a Palm Device. Neat. But this doesn’t get us very far yet - in order for POSE to actually work, and in order for you to use POSE to actually emulate a Palm device, you need a ROM image. The ROM image contains all of the code used for a specific version of the Palm OS. You can obtain ROM images in several ways:

- follow the procedure as described under <http://www.palmos.com/dev/tech/tools/emulator/>
- download the ROM from a handheld that has been placed in the device cradle and connected to the desktop computer
- have somebody sent you a ROM image

In order to download a ROM image file from a device, start up POSE and right-click anywhere in the window. A menu will pop up from which you can select “File->Transfer ROM”. This assumes, of course, that you have set up your system to talk to the cradle etc. For information on how to set up your Palm device under Linux, refer to <http://www.orbits.com/Palm/PalmOS-HOWTO.html>

While it certainly is easy to just get the file from somebody else, or to transfer it from your Palm device, I still recommend joining the Palm Solution Provider Program as described under <http://www.palmos.com/dev/tech/tools/emulator/> since you will gain access to a large variety of different ROM images.

At this point I recommend that you set up a directory where you keep all your palm-stuff. Place the ROM image file into this directory so you don’t lose them.

Now start POSE, and right click anywhere in the window. Select “File->New” from the menu and choose the location of the ROM-file and the appropriate Device. For example, if you want to use the ROM image palmvii-dbg-75.rom, you will want to choose “Palm VII” from the device-menu. Voila! Now go and play around with your new Palm device.

4 Other tools

OK - so you’ve got POSE set up and you finally get bored with the applications from your ROM image and you want to develop your own. Great! You will need a few tools to achieve this. On Linux you have two choices:

- buy “CodeWarrior” from Metroworks: <http://www.metroworks.com/>
- install the prc-tools on your system

This document will cover the second option. It should be mentioned, however, that CodeWarrior is the Development tool of choice for many Palm OS Software Developers and probably a decent product.

If you choose to install the necessary tools by hand and not use CodeWarrior, you also need:

- the Palm OS SDK (Software Development Kit) 3.5
- pilrc

4.1 Obtain the prc-tools

PRC-Tools is a complete compiler tool chain for building Palm OS⁶ applications in C or C++. It is free software. The PRC-Tools package includes patched versions of the GNU packages GCC, binutils, and gdb, along with various post-linker tools to produce Palm OS .prc files.

If you want, you can follow the instructions on <http://www.palmos.com/dev/tech/tools/gcc/> and see if you are in luck. However, I had some trouble getting everything set up with “prc-tools-2.0.1”, and eventually reverted to “prc-tools-0.5.0”.

PRC-Tools can be obtained in RPM format from <http://rpmfind.doc.ic.ac.uk/prc-tools.html> (or your favorite mirror). The source-code for prc-tools-2.0.1 can be downloaded from <http://www.palmos.com/dev/tech/tools/gcc/dist/prc-tools-2.0.tar.gz> - follow the instructions on <http://www.palmos.com/dev/tech/tools/gcc/building.html> for building them.

4.2 Obtaining the SDK

Now you need to download the Palm OS Software Development Kit 3.5. To do so, go to <http://www.palmos.com/dev/tech/tools/sdk35.cgi> and agree to the license and download the SDK from the following page. Optionally you might want to download a few programming examples and some documentation. Until now I am unsure if this step is actually necessary. The RPMs of prc-tools and pilrc seem to have provided my system with all the headers and libraries necessary for Palm OS Software development.

4.3 Obtaining and installing pilrc

You also need to download and install “pilrc”. Pilrc is available as source-code from <http://www.ardiri.com/>, or, again, if you prefer, from <http://rpmfind.doc.ic.ac.uk/pilrc.html> (or your favorite mirror). The installation from source-code failed when I tried it, so I can not go into the details of how to do that. Installing the RPM however was no problem, and everything seems to be set up correctly. So just go ahead and get the RPM.

5 An Example

It seems almost required that the first example in any programming tutorial be the infamous “Hello, world.” program. Once you have this program working, you know that you have at least a basic grasp of how to create, compile and install a Palm OS program. One thing to take note of is that Palm OS programs are event-driven. Basically, this means that unlike linear programs, (such as make, more, or pilot-xfer) Palm OS applications mostly wait for events, then process them in turn. In case you are not familiar with them, events can be any occurrence. Pen down & up, graffiti input, and alarms are all different events.

5.1 Hello World Code

So here’s the code to our beloved “Hello World!”:

```
#include <System/SysAll.h>
#include <UI/UIAll.h>

// -----
-
// PilotMain is called by the startup code and implements
// a simple event handling loop.
// -----
-
DWord PilotMain(Word cmd, Ptr cmdPBP, Word launchFlags)
{
    EventType event;

    if (cmd == sysAppLaunchCmdNormalLaunch) {

        // Display a string.
        WinDrawChars( "Hello, world!", 13, 55, 60 );

        // Main event loop:
        do {
            // Doze until an event arrives.
            EvtGetEvent( &event, evtWaitForever );
```

```

        // System gets first chance to handle the event.
        SysHandleEvent( &event );

        // Normally, we would do other
        // event processing here.

        // Return from PilotMain when an
        // appStopEvent is received.
    } while (event.eType != appStopEvent);
}
    return;
}

```

Just save this code in a file named “hello.c”.

5.2 Hello World Compilation

Assuming that the bin directory used by the prc-tools is in your default path, you have only to enter the following:

```

m68k-palmos-coff-gcc -O2 -g hello.c -o hello
m68k-palmos-coff-obj-res hello
build-prc hello.prc "Hello, World" WRLD *.hello.grc

```

m68k-palmos-coff-gcc is the GCC cross-compiler built by prc-tools. m68k-palmos-coff-obj-res takes the program generated by m68k-palmos-coff-gcc and splits it up into the various sections that Palm OS expects an application to have. build-prc assembles the program sections, plus the information provided on its command-line to build the final Palm OS program.

With any other program you will surely want to create a Makefile, so you don’t have to tediously enter the commands by hand every time you compile it. I will not go into the details of Makefiles, see <http://www.tw.gnu.org/software/make/make.html> for further information.

The Makefile for our simple “Hello, World!” would look like this:

```

hello.prc: code0001.hello.grc
    build-prc hello.prc "Hello, World" WRLD *hello.grc

code0001.hello.grc: hello.c
    m68k-palmos-coff-gcc -O2 hello.c -o hello
    m68k-palmos-coff-obj-res hello

clean:
    @rm -f *.grc *.o *.prc hello

```

Note: If you installed prc-tool-2.0 instead of prc-tools-0.5.0, your compiler and linker will most likely be named “m68k-palmos-gcc” and “m68k-palmos-obj”. You should keep this difference between the two versions in mind when you want to compile other examples or tutorials you might have downloaded.

5.3 Installing “Hello, World” on POSE

Now we want to test out application in its native environment. To do so, start POSE and right-click anywhere on the window. Then select “File->Install Application/Database” from the menu. Search through the directories to find our little application “hello.prc”. Now tap (palmspeak for click) the Applications-Icon and you should see a new icon labeled “Hello, World”. (Note that if you already were in the applications-menu (ie all the application icons were displayed on your screen), you need to switch to another screen (for example, tap “calculator”) and switch back (tap “Applications”) for the new icon to show up.) Tap this icon and..... WOW! You will see the words “Hello, World!” displayed in your POSE-window.

5.4 Keeping “Hello, World”

So you are so proud of your very first program, that you decide to keep it on POSE. No problem. The good thing about POSE is, that you can save as many profiles as you want. To do so, right-click on the window and choose “File->Save as...”. Choose a directory and enter the name for this profile (POSE usually saves profiles with a “.psf”-ending). The next time you start POSE, you can choose “File->Open” from the menu and load this profile, and you will find your “Hello, World”-application right where it was.

5.5 Deleting “Hello, World”

Well, “Hello, World” is not all that exiting, so do we really want to keep this useless little application? Sure we do - we have to show off our achievements to everybody else! OK, but after that, do we *still* need it? No. So in order to delete “Hello, World” from POSE, tap the “Menu” button and choose “Delete”. (If you want to get fancy, you can try to invoke the command by drawing a diagonal line followed by a “D” with your mouse in the little “Graffiti”-area). Choose “Hello, World” and hit “Delete”. Confirm with “Yes” and go back to the main screen with “Done”.

6 Resources

The information provided in this document was compiled mostly from:

- <http://www.orbits.com/Palm/PalmDevelopment-HOWTO.html>
- <http://www.palmos.com/dev/>
- and the documentation which comes with POSE

If you have any questions or find that the information provided in this document is wrong or ambiguous, please email me at jschauman@zentropypartners.com