Operating Systems – Practical Lab 12

Introduction to PalmOS, POSE, and PalmOS Software Development.

Assigned: **June 2, 2023**
Due: **June 9, 2023**

In this lab we will learn how to use **PalmOS**, the **PalmOS Software Development Kit (SDK)**, and the **PalmOS Emulator (POSE)**.

PalmOS is an operating system designed to run on a Personal Digital Assistant (PDA) using the **Motorola 68000** Microprocessor. These devices include the **PalmPilot**, and the **Handspring Visor**. PalmOS based devices were popular between approximately 1995 and 2006. The devices are not manufactured anymore, but you can buy them on websites such as http://ebay.com. You can also run PalmOS Applications in the **PalmOS Emulator (POSE)**.

Today several websites continue to be maintained to support development on PalmOS. Some of these websites include:

PalmOS C Compiler: http://prc-tools.sourceforge.net/
Setting up PalmOS Development on Linux: https://shallowsky.com/linux/palmlinuxdev.html
Downloading the PalmOS Software Development Kit: https://github.com/jichu4n/palm-os-sdk
Downloading the PalmOS Developer Suite: https://palmdb.net/app/palm-os-developer-suite
Downloading the POSE Emulator: https://www.netmeister.org/palm/POSE/POSE-HOWTO.html
Software Development on PalmOS: https://www.netmeister.org/palm/PalmMisc/PalmMisc.html
PalmOS Programming in a Nutshell: https://www.fuw.edu.pl/~michalj/palmos/Nutshell.html
PalmOS System Programming: https://www.fuw.edu.pl/~michalj/palmos/SystemFeatures.html

Information can be found in Korean here: https://namu.wiki/w/Palm%20OS

Submission Instructions:

After finishing this lab, you will have created **helloPalm.c** and **myPalm.s** as well as a **screenshot** of your name. Additionally you should edit this document directly. Please create a zip file containing these three files. Do not include other files, such as executable files.

Use the filename: **OS2023_Lab12_IDNumber_Name.zip**
For Example: **OS2023_Lab12_2021012345_DavidWagner.zip**

Please upload the zip file to our class website**.**

Software Development in PalmOS

It is possible to write PalmOS Applications in Windows, MacOS, or Linux. In this lab we will build PalmOS Applications in Ubuntu. Then we will examine the Operating System features that are used by our applications.

Instructions

Start **Ubuntu** in VirtualBox. To setup the PalmOS Development Environment, we need to install both the PalmOS Software Development Kit and the PalmOS Emulator.

First install the PalmOS Software Development Kit (Palm SDK):

1. Download palmsdk.zip from the class website.
2. Extract palmsdk.zip to a directory
3. Open a Terminal, and change into the palmsdk directory where the files were extracted.
4. Type the following command:

   ./installPalmSDK.sh

5. The PalmOS Software Development Kit should be installed. Test that it was installed by typing the following command:

   m68k-palmos-gcc

6. If the SDK was successfully installed, then you should see this message:

   m68k-palmos-gcc: No input files

Next install the PalmOS Emulatore (POSE):

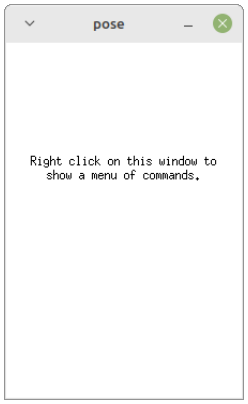
1. Download pose.zip from the class website.
2. Extract pose.zip to a directory
3. Open a Terminal and change into the pose directory where the files were extracted.
4. Type the command:

   ./installPose.sh

5. The PalmOS Emulator should be installed. Test that it was installed by typing:

   pose

6. If the emulator was successfully installed, you should see a window such as this one:



7. Right-Click on the emulator window and choose “Quit” to close the emulator window for now.

Next, let's create a simple app for PalmOS

1. Open a Terminal, create a new directory for this Lab, and change into that directory

2. Create a new file helloPalm.c with the following code:

```
// helloPalm.c

#include <PalmOS.h>

// ------------------------------------------------------------------
// PilotMain is called by the startup code and implements a simple event
// handling loop.
// ------------------------------------------------------------------
UInt32 PilotMain( UInt16 cmd, void *cmdPBP, UInt16 launchFlags )
{
    EventType event;


    if (cmd == sysAppLaunchCmdNormalLaunch) {

        //  Display a string.
        WinDrawChars( "Hello Palm!", 11, 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 0;
}
```

3. Compile the C program to an executable:

    m68k-palmos-gcc -o helloPalm.x helloPalm.c

4. List the files in the directory to confirm that helloPalm.x was created. If you received an error message or if the file is not in the directory then go back and check your source code.

5. Create the Object Resource Files:

    m68k-palmos-obj-res helloPalm.x

6. List the files in the directory. You should see several files with the extention ".grc". These are the resouce files which will be used to create the app.

7. Create the app with the following command:

    build-prc helloPalm.prc “Hello, Palm” WRLD *.helloPalm.x.grc

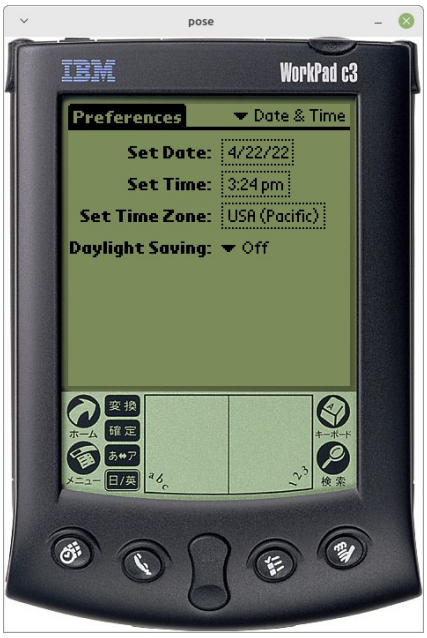
8. List the files in the directory. You should see a file “helloPalm.prc”. This is the app file which will be installed onto PalmOS.

Now we can choose a ROM file for the device:

1. Start the Emulator:

    pose

2. Right-Click on the pose window
3. Click “New…”
4. Select “ROM file:” → “Other…”
5. Select the file “Palm-Vx-4.1-en.rom” from the pose directory
6. “Device:” should automatically become “Palm Vx”
7. “RAM size:” should automatically become “8192K”
8. Click OK
9. You should see a window like this:



10. At this point you can setup the device preferences, and try various things with the device

Next install the Application onto the device. A PalmOS App is stored in a file ending with ".prc".

1. Right Click in the pose window
2. Click "Install Database/Application" → "Other…"
3. Find the file "helloPalm.prc" which you created previously and select it.
4. Click OK
5. Click the Home button which looks like this: 
6. You should see an icon for your new app which looks like this: 

7. Click the icon to open your app
8. You should see your message "Hello Palm!"

System Calls in PalmOS

System Calls in PalmOS are made with the "trap" instruction, of the Motorola 68000 Assembly Language. The trap instruction is called with one parameter, which is the System Call number. Additional System Call parameters are pushed onto the stack before the trap instruction is called. This is different from Linux, CP/M, and MS-DOS System Calls which take additional parameters from the registers.

Let's find out which System Calls our app is using

1. Open a Terminal Window and navigate to the directory where your "helloPalm.c" file is located
2. Compile this to assembly language instead of an executable by typing:

   m68k-palmos-gcc -S helloPalm.c

3. List the files. This should have created a new file "helloPalm.s". This file contains the assembly language of our program.
4. Look at this assembly language file. Find the "trap" instructions. These are the System Calls. Answer the question:

   Which trap number is used to make System Calls? _____

5. Look for the line ".ascii "Hello Palm!\0". This is the data where your message is stored. Immediately before this message is a label which indicates where your message is stored in memory, such as:

   .Label:
   .ascii "Hello Palm!\0"

   What is the "Label" indicating where your message is stored? _____

6. The system call number is indicated in the following format:

   dc.w 0xA000

   In this example the System Call Number is 0xA000.

   List all the System Call Numbers which can be found in the program:

   _____, _____, _____

7. The name of each System Call can be found in the file:

   /usr/share/prc-tools/trapnumbers

   What are the names of the System Calls used in the program?

   _____

   _____

   _____

8. The Motorola System Call parameters a given to the Operating System by pushing them onto the Stack. This is different from Intel x86 System Call parameters. Do you remember where Intel x86 System Call parameters are stored?

   _____

9. The system call parameters are pushed on the stack before the "trap" instruction. To push the number 60 onto the stack, the following instruction would be used:

   move.w #60, -(%sp)

   This copies the number 60 to the location in memory pointed to by the stack pointer (%sp), and subtracts 1 from the stack pointer. These steps have the effect of pushing the number 60 onto the Stack.

   In our program HelloPalm.c, what numbers are pushed onto the stack before the WinDrawChars System Call?

   _____, _____, _____

10. These three numbers are the x-coordinate (horizontal), y-coordinate (vertical), and length of the message. What is each value?

    X-coordinate = _____, Y-coordinate = _____, Length = _____,

11. The message itself is also pushed on to the stack before the WinDrawChars System Call. This must be done by pushing the "Label" of the message. The Label is pushed after the X-Coordinate, Y-Coordinate, and Message Length are pushed onto the stack.

    The label cannot be directly pushed onto the stack with a single instruction, so two instructions are necessary. These instructions first copy the Label its location into register %a0, and then push register %a0 onto the stack.

    What are the two instructions used to push the Label onto the Stack?

    _____

    _____

12. Copy the assembly language program to a new name:

        cp helloPalm.s myPalm.s

13. Edit the myPalm.s using vim or gedit.

14. Replace "Hello Palm" with "Hello, Name!", where "Name" is your own name.

15. Change the Length of the message

16. Change the location of the message so it appears in the lower right corner of the screen.

17. Compile and create a new app which prints your name:

        m68k-palmos-gcc -o myPalm.x myPalm.s

18. Create the Resources and the prc file:

        m68k-palmos-obj-res myPalm.x
        build-prc myPalm.prc "Hello, David" WRLD *.myPalm.x.grc

19. Change "David" to your own name

20. Install this onto the POSE emulator and run it.

21. Take a screenshot of your name printed in **lower-right** corner of the POSE Emulator

22. When you close the POSE Emulator, it will ask you if you want to save the session. If you save this, then you can see your apps the next time you run pose. Otherwise, you need to install them again.