

**Familiar Linux for the iPAQ H3975
(XScale Processor)
CSC 714 Real Time Computing Systems Term Project**

Adinarayanan Venkatachalam (avenkat2@unity.ncsu.edu)
Srivatsa Venkata Chivukula (svchivuk@unity.ncsu.edu)

Table of Contents:

1. Introduction

2. System Description

- 2.1. Processor Overview
 - 2.1.1. Overview
 - 2.1.2. XScale Architecture
 - 2.1.3. ARM ISA versions
 - 2.1.4. XScale Architecture Extensions
- 2.2. Hardware Overview
- 2.3. Bootloader Overview
 - 2.3.1. Parrot Bootloader
 - 2.3.2. BootBlaster
 - 2.3.3. Compaq Bootloader
- 2.4. Filesystem Overview
 - 2.4.1. WinCE filesystem
 - 2.4.2. JFFS2

3. Solved Issues

- 3.1. Problems Faced

4. Future Work

5. Conclusion

- 5.1. Skills Developed

6. References and Links

- 6.1. Project Web Page

1. INTRODUCTION:

Linux is an open source operating system with a monolithic kernel and an active developer community. The iPAQ H3975 is the latest in a range of PDAs (Personal Digital Assistant) from Hewlett Packard. The Familiar project has been undertaken by the Handhelds.org people to port Linux to the iPAQs. Linux has been ported to the Intel StrongARM processor. But it has not been fully ported to the Intel XScale processor. This motivated us to take up this challenging project.

2. SYSTEM DESCRIPTION:

2.1. Processor Overview:

2.1.1. Overview:

The Intel XScale core is an ARM V5TE compliant microprocessor. The core is not a stand alone product but a building block for an ASSP (Application Specific Standard Product) with embedded markets such as handheld devices, networking, storage, remote access servers, etc. The PXA250 and PXA210 applications processors are an example of such an ASSP designed primarily for handheld devices.

2.1.2. XScale Architecture:

The Intel XScale core has a 7-stage pipeline. It implements a 32-Kbyte, 32-way set associative instruction cache with a line size of 32 bytes and a 32-Kbyte, 32-way set associative data cache and a 2-Kbyte, 2-way set associative mini-data cache. Each cache has a line size of 32 bytes, supporting write-through or write-back caching. The Intel XScale core supports both big and little endian data representation.

2.1.3. ARM ISA versions:

There are a lot of versions of the ARM Instruction Set Architecture (ISA). ARM Version 5 (V5) Architecture added new features to ARM* Version 4:

- tiny pages of 1 Kbyte each
- a new instruction (CLZ) that counts the leading zeroes in a data value
- enhanced ARM-Thumb transfer instructions
- new breakpoint instructions (BKPT)
- a modification of the system control coprocessor, CP15
- floating point instructions

ARM Version 5TE is an enhancement to the ARM V5. T refers to the Thumb instruction set and E refers to the DSP-Enhanced instruction set. The Thumb instruction set are 16-bit ARM instructions that implement similar functions to the ARM 32-bit instruction set, but offer advantages in reducing code size. The DSP-enhanced instruction

set are new multiply instructions that operate on 16-bit data values and new saturation instructions.

2.1.4. XScale Architecture Extensions:

The Intel XScale core implements the integer instruction set architecture specified in ARM Version 5TE. It implements the integer instruction set of ARM V5, but does not provide hardware support for any of the floating point instructions. The Intel XScale core made a few extensions to the ARM Version 5 architecture to meet the needs of various markets and design requirements:

- A DSP coprocessor (CP0) has been added that contains a 40-bit accumulator and 8 new operations in coprocessor space, hereafter referred to as new instructions.
- New page attributes were added to the page table descriptors. The C and B page attribute encoding was extended by one more bit to allow for more encodings: write allocate and minidata cache.
- Additional functionality has been added to coprocessor 15. Coprocessor 14 was created.
- Enhancements were made to the Event Architecture, instruction cache and data cache parity error exceptions, breakpoint events, and imprecise external data aborts.

The Intel XScale core provides the ARM V5T Thumb instruction set and the ARM V5E DSP extensions. To further enhance multimedia applications, the Intel XScale core includes additional Multiply-Accumulate functionality as the first instantiation of Intel Media Processing Technology. These new operations from Intel are mapped into ARM coprocessor space. Backward compatibility with StrongARM products is maintained for user-mode applications. Operating systems may require modifications to match the specific hardware features of the Intel XScale core and to take advantage of the performance enhancements added.

2.2. Hardware Overview:

The H3975 has the following hardware features among others:

- Xscale 400MHz CPU
- 48MB of ROM
- "transflective" 65,536 (64K, 16-bit) color screen
- SD expansion cards – Supports SD Memory cards and SDIO
- 64-MB SDRAM, 48MB Flash ROM Memory

2.3. Bootloader Overview:

The bootldr is firmware. It is the first code executed by the processor when it comes out of reset, whether reset was induced by turning on power to the device or by

hitting the reset switch. The first word of the bootldr image is loaded at the beginning of Flash so that it will be executed when the processor comes out of reset. Usermode programs and even the kernel have some other program that interprets the executable and loads the code into memory at the appropriate point. These often have executable file headers that describe where to load them into memory. The bootldr does not have this luxury - it is the one that loads the kernel.

2.3.1. Parrot Bootloader:

This is the original bootloader that ships with the iPAQ H3975. This is responsible for booting WinCE.

2.3.2. BootBlaster:

This is a WinCE application that helps save the bootloader for Linux onto the initial portion of the Flash. It deletes the Parrot Bootloader and saves the Compaq Bootloader. It also saves the WinCE partition and prevents accidental erasure of WinCE.

2.3.3. Compaq Bootloader:

This is the bootloader developed by Compaq. As the Parrot Bootloader does not load any other operating system other than WinCE, this bootloader satisfies the requirement of loading other OSes. It loads the Linux kernel.

2.4. Filesystem Overview:

In Flash memory, a word must be erased before it can be programmed. In both types of Flash, the storage array is organized into erase blocks. All words of an erase block must be erased simultaneously. Sectors tend to be fairly large: 256KB on the iPAQ. Each erase block can only be erased about 100000 times before the sector fails. These characteristics of Flash mean that Flash is not a direct replacement for DRAM. Trying to use it like DRAM would result in very low performance and early failure of the Flash memory. Flash is quite useful, however, for file repositories. It has similar performance characteristics to disk, on average. With suitable “wear leveling”, the lifetime of the Flash sectors is likely to match the lifetime of the device.

2.4.1. WinCE filesystem:

The WinCE filesystem is the original filesystem. It provides a layer of abstraction over the FAT filesystem so as to “wear level” the Flash memory.

2.4.2. JFFS2:

A standard filesystem such as VFAT or the Linux Ext2/Ext3 filesystem can be used over Flash if a block remapping layer is used. For the root filesystem on the iPAQ, a

filesystem has been designed expressly with the characteristics of Flash in mind: Journaling Flash Filesystem Version Two (JFFS2). JFFS2 is a log-structured filesystem. That is, all the data and meta data is written sequentially to a journal in Flash. The complete filesystem directory structure is maintained in DRAM. It is reconstituted from the journal when the filesystem is mounted.

3. Solved issues:

- Copy BootBlaster onto iPAQ (Adinarayanan):
The Bootblaster3900 was copied onto the iPAQ using Microsoft ActiveSync software from the desktop (running Windows 2000) to the iPAQ. The Bootblaster3900 was downloaded from <http://handhelds.org/~pb/3900/>
- Save the WinCE image (Adinarayanan & Srivatsa):
The WinCE image was backed up so that in case something goes wrong WinCE can be loaded back. The original bootloader was also backed up. The details of the instructions can be accessed at the project web page.
- Install the Bootldr (Srivatsa):
The Bootldr-2.19.33 was installed after erasing the Parrot Bootloader in the Flash. The Bootldr-2.19.33 was downloaded from <http://handhelds.org/~pb/3900/>. The installation instructions were the ones in the site - <http://www.handhelds.org/feeds/bootldr/install.html>
- Checkout Linux Kernel sources from Handhelds.org site (Adinarayanan & Srivatsa):
The Linux kernel sources version 2.4.19-rmk4-pxa2 were checked out from the CVS repository at - <http://cvs.handhelds.org/>.
- Build the cross compiler for the arm-linux/xscale-*-elf platform (Adinarayanan):
This was a pretty challenging task. The host platform was i386-unknown-linux and the target was xscale-*-elf. The host compiler was gcc-2.96. The installation log has been posted at the project web page.
- Setup serial communication with the iPAQ (Adinarayanan):
The serial communication has to be setup with the iPAQ in order to issue commands to the bootloader. The kernel image is transferred through this serial communication. Hyperterminal on the Windows desktop is used towards this.
- Architecture Support for XScale in Linux kernel (Adinarayanan):
The kernel needs to be specifically configured for the XScale architecture using make menuconfig. Also, optimizations as given in Operating System Developer's Manual [4] can be followed for better performance. The co-processor feature exploited in the XScale architecture (XScale Microarchitecture Manual [4]) also needs to be supported. Device support for the XScale architecture needs to be

added. This includes touchscreen driver, handwriting recognition, PCMCIA driver support.

- Installation of root filesystem (Srivatsa):
This involved transferring the file bootstrap-pb0-3900.jffs2 to the iPAQ via the serial communication. When load root was typed in the boot prompt the root filesystem was loaded and the basic networking utilities were setup.
- Kernel Installation (Adinarayanan):
This involved transferring the file zImage-2.4.19-rmk4-pxa1.hh3 to the iPAQ by typing boot at the boot prompt. Thus the compressed image of the linux kernel was transferred and installed.

3.1. Problems Faced:

Each stage of the project was a hurdle. To start with we did not have kernel/device driver programming experience. We also had no experience on building cross-compilers and low-level programming experience.

After having installed the Bootloader, communication with the iPAQ was tried using Hyperterminal in Windows. The communication was configured to 115200 baud rate, no flow control. But there was apparently no communication with the iPAQ. The iPAQ was reset according to instruction#15 in <http://www.handhelds.org/feeds/bootldr/install.html>.

After that Win CE could not be booted into.

What we understand from this:

- Bootloader erased the original bootloader (also called parrot bootloader) and saved the Win CE partition. Thus the Win CE partition was unaffected.
- Thus it should be possible to reinstall the original bootloader and boot into Win CE.

Approach to solving the problem:

- Find out if original bootloader can be reinstalled.
- Try to communicate using Hyperterminal and install the saved bootloader.
- Hope that the iPAQ has not been turned into a brick.

Solution:

The new bootloader scans all of the Flash and doesn't turn on the LCD. The Flash scanning code seems to take a long time. So we had to wait about 5 minutes before serial communication could be started.

Trying to build the cross-compiler was largely unsuccessful. One reason for this might have been the later version of the cross-compiler (gcc-3.2) as compared to the host compiler (gcc-2.96). But then, the binaries downloaded from <http://handhelds.org/> also did not work.

Even setting up the serial communication was difficult as the bootloader does not respond immediately and as the Xmodem protocol is used, the communication is prone to errors. This is because the Xmodem protocol uses just a simple checksum.

Without the ability to cross-compile for the XScale platform, we could not proceed much in terms of fully porting the linux kernel to H3975.

As Linux is a large, monolithic kernel, it was very difficult to identify specific functions to modify to suit the XScale processor.

4. Future Work:

This project laid the foundation for the porting of Linux to the iPAQ H3975. We gained invaluable insight into the working of the compiler, assembler, linker, linux kernel and varied other system features. We started from scratch (not knowing anything) and ended up learning a lot. Future work involves building a stable cross-compilation tool chain from the latest sources of binutils, glibc and gcc. The Linux kernel-2.4 should be fully ported to the H3975 and should have full device support. The Linux kernel-2.5 should also be ported to the H3975. Filesystem support should be improved.

5. Conclusion:

In this project we have tried porting the Linux operating system to the iPAQ H3975 (XScale processor). Towards this goal we built a cross-compilation toolchain, studied the XScale architecture, analyzed the various low-level features and interactions of the Linux kernel with the hardware and installed linux kernel.

5.1. Skills Developed (as a result of this project):

- Building a cross compiler.
- Familiarity with a handheld device and a totally different architecture (from what we are already comfortable with).
- Familiarity with a handheld OS (WinCE and Linux)
- Linux Kernel Programming (this includes compiling a kernel, getting familiar with the architecture for low-level assembler directives, working without user-level libraries like glibc).
- Basic familiarity for hacking Bfd (backend), GNU as, ld and gcc.
- Understanding Processor Manuals and developing code based on those specifications.
- Device Driver Programming (learnt to work with real hardware... have built the platform for further development).
- Using newsgroups, mailing lists and IRC.

6. References and Links:

[1] HP Handhelds
<http://www.handhelds.org>

[2] Porting Minix to the iPAQ
<http://www.cs.uno.edu/~venkata/www/index.htm>

[3] Assembly Programming Guide
<http://linuxassembly.org/>

[4] Developer Manuals for PXA250 chip
<http://www.intel.com/design/pca/applicationsprocessors/manuals/index.htm>

[5] ARM Architecture Reference Manual
http://www.altera.com/literature/third-party/ddi0100e_arm_arm.pdf

[6] ARMv5E ISA Quick Reference Card
[http://www.arm.com/techdocs/5AEDLK/\\$File/QRC-0001D.pdf?OpenElement](http://www.arm.com/techdocs/5AEDLK/$File/QRC-0001D.pdf?OpenElement)

6.1. Project Web Page:

<http://www4.ncsu.edu/~avenkat2/csc714project.html>