

NSF-REU 2008: Using Android to Promote Education in Mobile Device Development

Our experiences with Android on various hardware platforms

Chris Baker, Danny Noler

Auburn University

July 25th, 2008

Outline

Introduction

Android

Obstacles

Android on Hardware

Nokia N800

Sharp Zaurus SL-C3200

Then and Now

Acknowledgements

About Us

Danny Noler

- Senior at Mercer University
- BS in Computer Science, minor in Math and IST

Chris Baker

- Senior at Auburn University
- BS in Software Engineering in the College of Engineering

What is Android?

Software stack for mobile devices that includes:

- Operating System (Linux kernel 2.6)
- Middleware
- Key Applications

Currently being developed by The Open Handset Alliance

- Headed by Google
- Consists of over 30 technology and mobile companies

Current Hardware Requirements

Software derived from code in SDK

- SDK uses QEMU, an open source processor emulator
- QEMU used to create virtual ARM system-on-a-chip called Goldfish
- Goldfish is booted from a 2.6.23 ARM Linux kernel with support for an x86 host
- Since Goldfish has software compiled for the ARMv5 instruction set, only hardware with an ARM926 or higher can be used

Other requirements

- 32MB RAM
- 32MB Flash Memory
- 200Mhz processor

Obstacles

Full source not released

- Android must be "patched" together to work natively
- Provides challenge, but also limits developers

Learning Curve

- Knowledge of Linux kernel and device drivers
- Use of cross compilers
- Android architecture and APIs

Android on Hardware

Ported Android to two devices

- Nokia N800
- Sharp Zaurus SL-C3200

Both devices use Linux kernels with their standard OS

Nokia N800

Specifications

- CPU : OMAP 2420 @ 400Mhz
- OS : Linux Maemo
- RAM : 128 MB
- Flash : 256 MB
- 4.1" touch screen display @ 800 X 480
- WLAN 802.11b/g



Figure: Nokia N800

Nokia N800

Tools and files used

- ARM GNU Toolchain
- Maemo flasher 3.0
- Linux kernel 2.6.21
- Prebuilt Android file system (m3)
- Android kernel patch (v1 for m5)
- Touch screen patch
- MicroSD card formatted using ext2

Nokia N800

Steps Taken

1. Patch kernel with Android patch
2. Apply touch screen patch
3. Cross compile kernel using GNU Toolchain
4. Flash N800 with new kernel using Maemo flasher
5. Copy Android file system to MicroSD card
6. Boot N800 (still boots into OS2008)
7. Start Android using script

N800 Results

Pros

- Applications are functional
- Touch screen is functional
- WiFi support

Cons

- No keyboard

Android on N800



Figure: Chris holding N800 with Android running

Sharp Zaurus SL-C3200

Specifications

- CPU : Xscale @ 416 Mhz
- OS : Linux
- RAM : 64 MB
- Hard Disk : 6 GB
- 3.7" rotating touch screen display @ 640 X 480
- Full QWERTY keyboard
- Only available in Japan



Figure: Sharp Zaurus SL-C3200

Sharp Zaurus SL-C3200

Tools and files used

- Linux kernel image with Android patches applied (m5)
- Android file system image (m3)
- Android file system image (m5)
- OpenEmbedded cross compiler
- Angstrom Linux
- SD card formatted with FAT32 standard

Sharp Zaurus SL-C3200

Initial success with Android

1. Copy image files to SD card
2. Extract image files
3. Update Zaurus using built-in update utility
4. Wait 10 minutes
5. Enjoy your fresh Android Zaurus

Limitations to this approach

- No touch screen support
- Only m3 version of Android available

Sharp Zaurus SL-C3200

Attempt to build Android from scratch

1. Install OpenEmbedded with BitBake
2. Alter config files and add Android patches
3. Compile for Angstrom Distribution
4. Copy resulting image files to SD card
5. Update Zaurus using built-in update utility
6. Copy Android file system image files (m5) to SD card
7. Extract files on Zaurus
8. Create startup script and run
9. Android sticks, and we are left with an eternal loading screen

Sharp Zaurus SL-C3200

Causes of infinite loading time

- Distribution missing Binder module needed for Android support
- Module available, but for Linux kernel 2.6.23
- Angstrom uses Linux kernel 2.6.26
- Resources used for building Android used earlier version of Angstrom (no longer available)

SL-C3200 Results

Pros

- Applications are functional
- Keyboard is responsive

Cons

- No touch screen support
- No built in WiFi, nor CF WiFi card support

Android on SL-C3200



Figure: Danny holding C3200 with Android running

Then and Now

Topics covered in research

- Android architecture
- Android application structure
- Use of toolchains
- Cross compiling
- Basics of Linux kernels
- Linux commands and terminal usage

Acknowledgements

- Dr. Chapman - Mentor
- Dr. Umphress - Mentor
- Google - Information and Research Source

Thank You!

Any Questions?