XinuPi: Completion of the Hardware Conversion of Xinu to the Raspberry Pi

Jason Arnold, Omokolade Hunpatin, Luke Mivshek, Casey O'Hare, and Dr. Dennis Brylow

Motivation

Embedded Xinu at Marquette is a project that allows the Xinu operating system to be run on multiple platforms, largely MIPS based architecture. It is what allows Marquette students to be able to compile and run their own working kernel each time they work on a project. The current implementation runs on older, less reliable hardware. Our work was to finish porting Xinu to a new, more accessible platform: the Raspberry Pi.

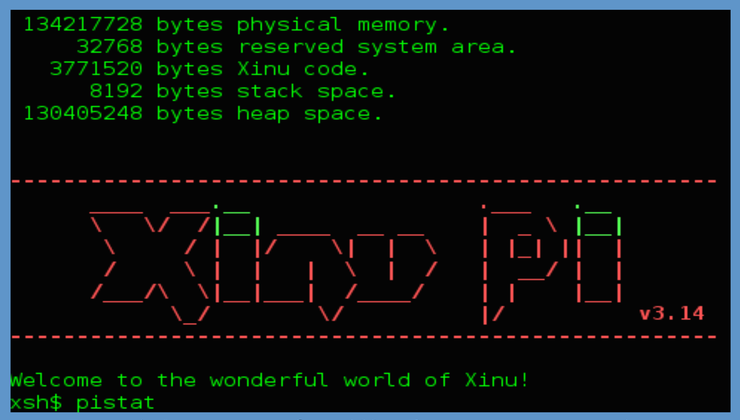

Figure 1: An example of what XinuPi looks like upon bootup.

Previous Work

The research portion of Xinu Is continually worked on by senior design and REU students. The Raspberry Pi version of Xinu, XinuPi, has been a work in progress by many teams and includes:

- Preemptive Multitasking
- Interrupt Handling
- USB support
- Ethernet Connectivity

The Raspberry Pi

The Raspberry Pi is a credit-card sized embedded system that can run and act like a normal computer by uploading a kernel from it's SD card. It runs on the ARM architecture unlike most implementations of Xinu. The Raspberry Pi was chosen as a replacement for the current Linksys Routers for a number of reasons:

- At 35 dollars, it is cheaper than previous implementations.
- There is a well-supported community using the Raspberry Pi products.
- The Raspberry Pi is a highly used and bought item, allowing it to be reliably provided for anyone looking to use XinuPi.

The kexec() Function

Among the most focused on of previous research is the function kexec(). The use of Xinu in curricula at Marquette is to allow students to compile their own kernel. In order for this to be done safely (without the ability for the student to render the default kernel useless) the kernel must be switched over in RAM, while the old kernel is still running. This allows the built-in Xinu kernel on the SD card to stay untouched in the case of a malicious kernel being uploaded.

This work was supported in part by the National Science Foundation, grants CNS-1339392, and ACI 1461264. Special acknowldgements to researchers that have worked on both Xinu and XinuPi.

Arm-Console

In order to run a user kernel on a Raspberry Pi in the Marquette Systems Lab a user must be able to boot up a Raspberry Pi in the pool, send their kernel to the lab's main server, and execute kexec(). Arm-Console is the ARM based version of Mips-Console, which runs an expect script that completes these steps for the user, making the upload process more student-friendly.

Future Use

The purpose of XinuPi is to replace the current use of Embedded Xinu in a laboratory environment. The XinuPi implementation is planned to substitute the current Embedded Xinu setup for this coming fall curriculum in Hardware Systems. A full set of 16 Raspberry Pis is planned to support this course as well as larger ones such as Operating Systems and Networks here at Marquette. Research will continue on both Xinu and its XinuPi flavor in order to make future upgrades to better systems like this one possible.

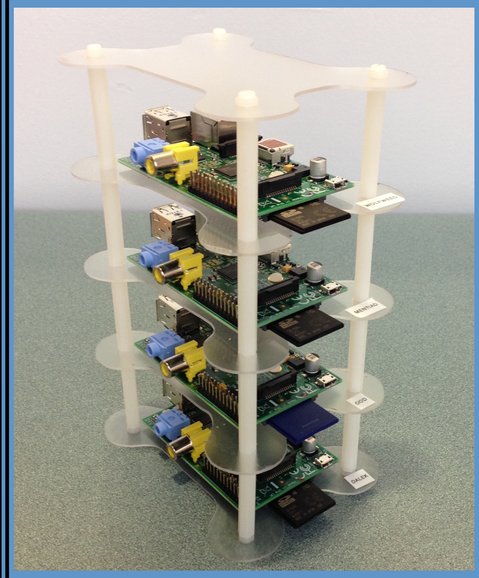

Figure 2: An example of a pool of Raspberry Pis to be used in the Systems Lab.