

Week 2: Research Report

Student: *Craig Harrison*

Advisor: *Dr. Richard Eason*

This week's development has been focused on completing two major goals. The first task was to set up an out of date computer to run Fedora 9 with newly installed and/or rewritten drivers to work with the old motor controller and IO cards in a Linux-based environment. Due to the much more complex kernel design implemented in Linux than in DOS this presented a large technical difficulty.

The first challenge we experienced was in writing code to interface with the digital IO cards currently in use by the SST. In DOS this involved simply calling the functions `inportb()` and `outportb()` with the correct port number. Linux, however, does not allow such simple access to ports as that would clearly create a potential security issue. By using a function called `ioperm()` before the Linux versions of `inportb()` and `outportb()`, `inb()` and `outb()`, it is possible to request permission from the kernel to have low-level access to the PC's ports. We soon discovered that in order to be granted permission our program must run with root privileges, which involved giving the root user ownership of the file and setting the `setuid` permission bit.

The next, and greatest, challenge faced this week involved interfacing with the currently used motor controller card. One morning was spent compiling a driver and API supplied by the motor controller company only to discover that while it would work for the newly ordered motor controller cards, it would not work with the cards currently in use. As it is necessary to ensure that the newly ported code works with the old hardware system before installing a new hardware system, it was vital that we found a way to communicate with the old cards under Linux. To this end we were able to start with a piece of DCX interfacing code written by UMaine faculty member Andrew Sheaff for version 2.2 of the Linux kernel. Unfortunately, the internal kernel workings have changed to a large degree since version 2.2 and neither the user space code nor the kernel module compiled. The kernel build system had recently changed, so the first step involved writing a new Makefile for the kernel module that would work with the new build system. This new Makefile allowed me to attempt to build the kernel module, which resulted in a large number of errors. Through a painstaking search of online documentation and updates to other open source kernel modules I was eventually able to piece together enough information to rewrite the portions of the kernel module that no longer functioned. At this point it also became necessary to modify the user space portion of Andrew Sheaff's DCX interface as the mapped memory was not being updated before reads and after writes and in order to fix some liberties that had been taken with the compatibility of the API. These new changes were then merged into the ported source code and a Makefile was written to compile and link the entire ported project.

At the same time as this other work was going on we were also setting up an older computer with all the necessary hardware, software, and settings to be installed with a currently used trimming system that has been taken off line and shipped to the Allen Edmonds factory in Lewiston for our development purposes. We hope to communicate with the IT department at the factory so that we can work on the test machine remotely over SSH.

References:

C++ Library Reference. Retrieved June 5, 2008 from <http://cplusplus.com/reference>

Why FreeDOS. Retrieved June 17, 2008 from <http://www.freedos.org>

The Fedora Project. Retrieved June 12, 2008 from <http://fedoraproject.org/>

MultiFlex PCI 1000 Series Documentation. Retrieved June 12, 2008 from <http://www.pmccorp.com/support/mfx1000.php>