

CS 486: Assignment 3

Input and Time

Jon A. Solworth

Due: 9 Mar 2015

Due: **9 Mar 2015**
Demo: Working code

1 Overview

For this assignment, you will use time, console, and scheduling routines to build an interactive timing system.

2 Details

After starting up, your kernel should display a prompt and accept any of the following commands from the user via the console:

st *p* Set a timer to go off in *p* seconds. *p* should be a decimal integer. Multiple timers can be active at any given time.

sm *s* Set the message to be printed when a timer goes off. The string *s* may contain any printable character and is ended by the new line. The string may or may not be enclosed in quotes.

ct Cancel the last timer that was set.

sd Shutdown the kernel.

After each command other than shutdown, your kernel should return to the prompt. Since your kernel must accept input, you will need to use your own console driver (the binary driver from Assignment 1 will not work). Timers should be managed using the appropriate Xen hypercalls along with any bookkeeping necessary to keep multiple timers running at once. To avoid wasting CPU time, your kernel should block when idle (i.e. when there is no input or timer to process).

3 Background

In order to complete this assignment, you should be (or become) familiar with the following concepts:

1. Making hypercalls
2. Sending and receiving events
3. Correctly copying data to and from ring buffers

These topics will be/have been covered in class; further information may be found in the text, on the course webpage, and in the Xen headers.

4 Notes

You must link the library you created for the last assignment and your new console driver with your kernel. You may do so by copying any necessary source files into the `src` directory and adding the appropriate object names to the Makefile. Without your library, the kernel will not build.