

Assignment

1 Introduction

The purpose of this assignment is to familiarize you with OS/161 and Sys/161.

OS/161: OS/161 is a simple operating system kernel, which is made available to you along with a small set of user-level libraries and programs that can be used for testing. The baseline OS/161 that we distribute to you has very limited functionality. Each of the CS350 assignments will ask you to improve OS/161 in some way to add additional functionality to the baseline.

Sys/161: Sys/161 is a machine simulator. It emulates the physical hardware on which OS/161 runs. Apart from floating point support and certain issues relating to cache management, it provides an accurate emulation of a server with a MIPS R3000 processor. You will use Sys/161 each time you want to run OS/161. However, you are neither expected nor permitted to make any changes to Sys/161.

Before you can complete this assignment, you will need to obtain and build a copy of OS/161. If you are planning to work on your own machine, you will need to obtain and install both Sys/161 and the toolchain before you will be able to build, run, or debug OS/161 code.

2 Preliminaries

Before you start working on this assignment, you must complete the following steps:

1. Install a copy of OS/161. If you plan to work on your own machine, install OS/161, Sys/161, and the toolchain in your machine instead.
2. Once you have installed OS/161, you will need to learn how to modify it, build it, run it, and debug it. Read and understand the guides before you start working on this assignment.

3 Assignment Requirements

For this assignment, you are expected to make two minor changes to the OS/161 kernel:

3.1 Customize the Kernel Boot Output

When OS/161 boots, it produces output that looks similar to this:

```
dasiyqu@ubuntu:~/cs350-os161/root$ sys161 kernel
sys161: System/161 release 1.99.06, compiled May 20 2018 05:03:48

OS/161 base system version 1.99.05
Copyright (c) 2000, 2001, 2002, 2003, 2004, 2005, 2008, 2009
    President and Fellows of Harvard College. All rights reserved.

Put-your-group-name-here's system version 0 (ASST0 #1)

304k physical memory available
Device probe...
lamebus0 (system main bus)
  emu0 at lamebus0
  ltrace0 at lamebus0
  ltimer0 at lamebus0
  beep0 at ltimer0
  rtclock0 at ltimer0
  lrandom0 at lamebus0
  random0 at lrandom0
  lhd0 at lamebus0
  lhd1 at lamebus0
  lser0 at lamebus0
  con0 at lser0

cpu0: MIPS r3000
OS/161 kernel [? for menu]:
```

Note the line that says “Put-your-group-name-here's system”. Your first assignment is to change OS/161 such that the kernel identifies itself as your kernel when it boots. For example, if your name were Liberty Valance, your kernel should say “Liberty Valance's system”.

Once you have done this, make sure that you can re-build and run OS/161 with your customized boot output.

3.2 Add a Kernel Menu Command

The OS/161 kernel includes a simple system that allows debugging messages to be displayed when the kernel runs. There can be different types of debug messages, and the kernel can be told to display only messages of certain types. For example, the file `kern/thread/thread.c` includes the statement

```
DEBUG (DB_THREADS, "Forking thread: %s\n", name);
```

This defines a debugging message of type `DB_THREADS`.

The debugging mechanism is implemented in the file `kern/include/lib.h`. This file also includes definitions of all of the pre-defined debugging message types, such as `DB_THREADS`. There is a kernel global variable, `dbflags`, which defines which types of debugging messages should be displayed when the kernel runs (see `kern/lib/kprintf.c`). In the baseline code, `dbflags` is set to zero, meaning that no debugging messages are displayed.

After the OS/161 kernel boots, it displays a command prompt and waits for a command, like this:

```
OS/161 kernel [? for menu]:
```

If you type `?`, you should get a list of available commands and sub-menus, one of which is the operations sub-menu. For this assignment, your second requirement is to add a new command to the kernel's operations sub-menu. The new command, which must be called `dth`, should enable the output of debugging messages of type `DB_THREADS`. (If such messages are already enabled, the command should have no effect). Thus, any kernel commands that are run after `dth` should run with `DB_THREADS` debugging messages enabled.

To do this, you will need to understand how the debug message mechanism works and how the kernel menu system works. The latter is implemented in the file `kern/startup/menu.c`. You should be able to complete this assignment by changing only that single file.

To test your new kernel option, we will use it to run one or more of the kernel's built-in thread tests with `DB_THREADS` debugging enabled using your new `dth` command. The kernel has several simple thread tests (e.g., `tt1`, `tt2`, `tt3`) that can be run from the kernel menu prompt.

For example, without `DB_THREADS` debugging enabled, thread test 2 (`tt2`) produces output like this:

```
OS/161 kernel [? for menu]: tt2
Starting thread test 2...
0123456701234675
Thread test 2 done.
Operation took 0.662667880 seconds
OS/161 kernel [? for menu]:
```

However, if `DB_THREADS` debugging has been enabled by running your new `dth` command, this test should instead produce output similar to this:

```
OS/161 kernel [? for menu]: dth
Operation took 0.000020800 seconds
OS/161 kernel [? for menu]: tt2
Starting thread test 2...
Forking thread: threadtest0
F0orking th0read: threadtest1
F1orking t1hread: threadtest2
F2orking t2hread: threadtest3
F3orking t3hread: threadtest4
F4orking t4hread: threadtest5
F5orking th5read: threadtest6
F6orking th6read: threadtest7
77
Thread test 2 done.
Operation took 0.717869840 seconds
OS/161 kernel [? for menu]:
```

Notice the messages produced by the `DEBUG` statement from `threads.c`.

4 Submitting Your Work

Deadline:

Submission: 2020-6-28 (You need to submit the files modified with your experiment report to sh727422908@163.com.)

5 Reference

You can refer to the following course websites for more information about OS/161 and other assignments:

Uwaterloo: <https://www.student.cs.uwaterloo.ca/~cs350/S18/assignments/>

Harvard: <http://os161.eecs.harvard.edu/>