

# CMSI 387-01

## OPERATING SYSTEMS

### Spring 2013

## Assignment 0312

So here's the big thing—kernel building and modification! Best. Spring Break. Ever. Enjoy the ride :)

### Outcomes

This assignment will affect your proficiency measures for outcomes *1a*, *2a*, *2b*, and *4a–4f*.

### Not For Submission

If you have access to the text, read Chapters 1 and 2 in SGG.

### For Submission

Figure out how to modify, compile, install, and demonstrate an Ubuntu 12 kernel. The core concepts behind this activity are generally the same across the board, for any flavor of any operating system. But as with many technical activities, the devil is in the details. And it *is* quite a devil. Put another way, this is perhaps the mother of all “Hello world!” variants.

For this, start with the Ubuntu kernel wiki (<https://wiki.ubuntu.com/Kernel>). Beyond that, turn to the web, but with care: you will find all kinds of information there, ranging from good to bad to ugly. It is your deeper, conceptual understanding that will help you discern the quality of these information sources.

In broad strokes, the tasks you are to perform are:

- Acquire the latest kernel source code.
- Install whatever prerequisites are necessary to modify, compile, install, and demonstrate an Ubuntu 12 kernel.
- Configure, build, and install a new kernel from that code.
- Add a new system call to the kernel source code (for this assignment, a system call that emits “Hello world!” will suffice—but if you can do more, don't stop yourself).
- Write and build a C program that demonstrates the existence and functionality of your new system call.

### New git Tricks

This is not a separate set of tasks, but a list of new git operations that you may find useful while working on this assignment. Look them up:

- `branch`—Manages different code lines
- `checkout`—Changes your working copy to match a commit or branch
- `clean`—Deletes added, untracked files
- `diff`—Compares commits or branches; can also create *patch* files, which capture these differences in a way that can be applied to another git repository
- `tag`—Manages *tags*, or particular marked versions of the repository

### What to Turn In

To show that you did the work (and knew what you were doing), your deliverables are:

- One or more web pages, uploaded to *public\_html/ubuntu-kernel-howto* on *my.cs.lmu.edu*, documenting *precisely* how to perform this task, from a vanilla Ubuntu 12 install to the finished product.

Your web pages should be as “turnkey” as possible, meaning that a user who follows your instructions step by step should end up with the same results that you did. Screenshots or exact command input/output will definitely help here.

Your web pages should also demonstrate your own knowledge of the process: include explanatory sections where necessary, and state what might vary (identifiers, names, labels) based on user preferences or situations.

- When all is said and done, you will have your own customized version of the kernel source code. Learn how to derive a *git patch* from your changed version, and commit that to *homework/kernel*. Remember, this is the *patch*, and not the entire source tree—that would be, um, unwieldy (to say the least).