

# CSc 82010 UNIX Application Development

Thursdays, 11:45 A.M. - 1:45 P.M.

*Professor Stewart Weiss*

## What It Is About, and Why You Should Consider Taking It

**What is it about?** In a nutshell, this course is about how to develop applications, libraries, and systems programs for the UNIX operating system. This means, in essence, that you will learn about the UNIX *application programming interface* (API), which is the programming interface to the UNIX kernel, and you will learn how to write “real” programs that can interact with UNIX. It covers the basic parts of the kernel interface and libraries, including files, processes, terminal control, the NCurses library, signals, and threading. It also includes material on the internal structure of selected kernel components.



Fig. 1: The kernel, revealed

**Why take it?** Most computer scientists have to write programs, and the UNIX platform is the important platform in most disciplines. There are many reasons to know how to write programs for UNIX, including: (1) UNIX is free and it continues to grow in popularity; (2) UNIX is used in the financial sector, the scientific sector, and the academic sector; and (3) it makes you more versatile and is a noteworthy skill to have on your CV. In short, it is a survival skill in computer science.

minating in a final project that will have a research component. The final project will be of your choosing, subject to my approval. There is no required textbook; the content will be based upon my lecture notes, which will be available online, and recommended readings. The lecture notes for a similar undergraduate class given at Hunter College can be found on my website at [http://www.compsci.hunter.cuny.edu/~sweiss/course\\_materials/csci493.66/csci493.66\\_spr08.php](http://www.compsci.hunter.cuny.edu/~sweiss/course_materials/csci493.66/csci493.66_spr08.php).

**What will be expected of you?** *This is a project-based course, with **no exams**.*

There will be several programming projects, cul-

## What Else You Should Know

Application development involves *creative*, *scientific*, and *technological* work. The technological work is the part that involves knowledge of the API. The scientific work is the part that involves an understanding of the structural principles of operating systems and data and algorithm design. Last, but not least, is the creative work. In the process of developing an application, judgments and decisions must be made that are not founded on any principles, but are creative leaps of human ingenuity. Whether large or small, these leaps are human in nature. Each person brings a different mix of these types of intellectual activities into development, and each appreciates it in different ways. Everyone usually finds something that is interesting, exciting, or rewarding by taking this class.