Projects for an Operating Systems Class

This repository holds a number of projects that can be used in an operating systems class aimed at upper-level undergraduates, from either **Southeast University**, or **Efrei Paris**.

Also available are some tests to see if your code works. A specific testing script, found in each project directory, can be used to run the tests against your code.

For example, in the initial utilities project, the relatively simple `seucat` program that you create can be tested by running the `test-seucat.sh` script. This could be accomplished by the following commands:

```
prompt> cd projects/initial-utilities/seucat
prompt> emacs -nw seucat.c
prompt> gcc -o seucat seucat.c -Wall
prompt> sudo chmod 777 test-seucat.sh
prompt> ./test-seucat.sh
test 1: passed
test 2: passed
test 3: passed
test 4: passed
test 5: passed
test 6: passed
test 7: passed
prompt>
```

Of course, this sequence assumes (a) you use `emacs`, (b) your code is written in one shot, and (c) that it works perfectly. Even for simple assignments, it is likely that the compile/run/debug cycle might take a few iterations.

Syllabus of OS Labs

Chapter	**#**	**Project**
Introduction	1	Reverse, Unix Utilities
Operating System Structures	2	Xv6 Syscall (part 1)
Processes	3	Unix Shell
Threads	4	Xv6 Kernel Threads
CPU Scheduling	5	Xv6 Scheduling (Lottery)
Process Synchronization	6	Xv6 Syscall (part 2)
Deadlocks	7	Map Reduce
Main Memory		
Virtual Memory	8	Xv6 Virtual Memory
Mass-Storage Structure		
File-System Interface		
File-System Implementation	9	File System Checker
I/O		

The projects marked as blue are **kernel hacking projects**. They are to be done inside the xv6 kernel based on an early version of Unix and developed at MIT. Unlike the C/Linux projects, these give you direct experience inside a real, working operating system.