Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.033 Lab, Spring 1999 | Handout 1– Administrivia | February 2

1 Introduction

The 6.033 lab is an elective 6-credit add-on to 6.033 designed to provide hands-on experience with building computer systems. Three projects, each progressively more challenging, comprise the lab. The first is coding a simple TCP proxy server. This project introduces the tools you will use for the later projects. The second project involves designing and implementing a multi-threaded, caching, web proxy server. The final project is in the area of file systems, though the design will be very open ended. The lab will run on Athena under UNIX, and the programming will be done in C.

A word on terminology: we will refer to the unchanged 12-credit 6.033 subject as "6.033." The add-on lab will be called the "6.033 lab." We will use the word "system project" to refer to the projects in the 6.033 lab. The projects in 6.033 are called "design projects."

2 Who should take this lab?

To take the lab, you must be enrolled in 6.033 this term. We require fluency in C or C++; this is not the right place to learn C. However, UNIX experience is not required. In addition, you should have taken 6.170 so that you have a working knowledge of how to construct large programs. Unlike the programming assignments associated with 6.001 or 6.034, you are required to build the systems from scratch. We do not advise taking this lab concurrently with another lab such as 6.111 or 6.170.

The lab will average around 6 hours of work each week *in addition* to the regular 6.033 course load. Typically, students invest a lot of time near the code deadlines. If your schedule is packed, you should think twice before enrolling. There are easier ways to earn 6 units. But if you have extra time and meet the prerequisites above, you should seriously consider enrolling. However, you do not need to take the lab to do well in or get a lot out of 6.033.

3 The system projects

If you don't understand all the terms in this section, don't worry! We will cover them in class and in special recitation sections for the lab. Subsequent handouts will cover the assignments in much greater detail.

To cut your teeth on UNIX programming and refresh your C skills, the first project is a TCP proxy server. We imagine this initial project will take about ten hours. There are no papers to write for this project, but you must hand in well-documented code for a functional server.

In the second system project, you are to design and implement a multi-threaded, caching web proxy server. Although we will give a basic specification, you are free to choose an implementation and any additional methods you wish to support. Early on, we will ask you to give us a preliminary design document including rationale for the design as well as an anticipated schedule. Then, after the implementation is done, you will hand in a second paper, describing the final design and implementation. This paper can substitute for your design project 1 in 6.033.

The final system project is the creation of a file system. However, the design will be very much up to you. For example, the file system could include security, work across a network, or provide guarantees in case of sudden shutdown. You will may choose what functionality you wish to support within the constraint of basic file system functionality. The last system project is designed to take five weeks for a team of two or three students to complete: one week for writing the initial design document, and four weeks to implement the design, evaluate it, and document it. The final design document for this project can be used to satisfy design project 2 for 6.033.

4 6.033 and the lab

Every student enrolled in the lab should also be enrolled in 6.033. In addition to the 6.033 recitations, special recitations for the lab section will meet on several **Fridays from 2 to 3PM**. Our preliminary schedule of topics is as follows:

February 12	Intro to UNIX network programming, TCP proxy assigned, add cards
February 19	Web proxy assigned, asynchronous communication, TCP proxy Q/A
February 26	TCP proxy post mortem, web proxy Q/A, caching issues
April 2	file systems intro, team forming
April 9	File system robustness, file system Q/A
April 23	Team meeting with lab TAs
April 30	Team meeting with lab TAs

6.033 has two design projects. For each design project you have to write up a paper. If you are enrolled in the lab, you can substitute the system project papers for papers on the design projects – a sweet deal. Note that we consider the time to write these papers as part of 6.033, not 6.033 lab.

5 Due dates

We have made the due dates for the final design documents correspond to those for the 6.033 design projects. In addition, the final code for the second and third projects is due a week earlier to give you time to refine the design documents.

TCP Proxy Server	Code due	Monday, February 22
Web Proxy Server	Preliminary Design	Friday, February 26
	Code due	Friday, March 12
	Final Design Paper	Thursday, March 18
File System	Preliminary Design	Friday, April 9
	Working Code due	Friday, April 23
	Robust Code due	Wednesday, May 5
	Final Design Paper	Friday, May 7

Projects 2 and 3 are large, complicated projects, similar in scale to the fourth and final projects of 6.170. As such, you should start early on the coding (February 27th for the web proxy server and April 9th for the file system). If you start coding only a week before the due date, you will most probably be swamped with work. In addition, there are many aspects of the design which do not become clear until you begin to code. The earlier you begin to code, the more time you have to play with your design and come out with something truly excellent.

6 Grades, credits, and collaboration policy

Successful completion of the 6.033 lab will earn you 6 credits and 6 Engineering Design Points.

Your lab grade will be determined as follows:

$$\text{lab grade} = 1/6 \text{ P1} + 1/3 \text{ P2} + 1/2 \text{ P3}$$

The lab grades will reflect the correctness and creativity of the design as well as the correctness of the implementation. The clarity of your design documents will help us greatly in understanding your design. In addition, if you choose to substitute the system project papers for the design papers in 6.033, they will partially determine your 6.033 grade.

The first two projects must be done individually; the third project can be done in teams of two or three students. For the first two projects you are allowed to discuss the problem with other students, but we expect you to work out the project on your own (i.e., you should do the coding and writing of design documents on your own). Similarly, for the last system project, each team is responsible for its own code and documentation. Borrowing code from previous students of the 6.033 lab is forbidden. Talk to the lab TAs if you have any questions or concerns.

7 Books

There are no required books to purchase for the lab. But you may find these books useful (all written by W. Richard Stevens):

- Advanced Programming in the UNIX Environment (ISBN: 0201563177), 1992.
- UNIX Network Programming Volume 1: Networking APIs: Sockets and Xti (ISBN: 013490012X), 1997.
- UNIX Network Programming Volume 2: Interprocess Communication (ISBN: 0130810819), 1998.

- TCP/IP Illustrated (several volumes).

We asked the Barker Engineering Library to reserve a copy of the first three books. Visit the circulation desk for more information. These books may not leave the library.

8 Where to find help

The 6.033-lab zephyr instance tends to get a lot of traffic from 6.033 lab students and other denizens of the net. Although this is not an official voice of the 6.033 lab, you can often stir up useful discussion with other students. Type `zctl add message 6.033-lab \*` to subscribe to the 6.033-lab zephyr instance. Type `zwrite -i 6.033-lab` to write a message.

Another source of help is 6.033-lab-tas@mit.edu. The lab TAs include Kevin Fu and Jeffrey Hu, both veteran lab students with battle scars to prove it. Email us if you have course administration questions, bug reports, etc. However, do not expect us to reveal solutions to the lab. We may purposefully respond to your design questions with open-ended answers.

9 How do I officially sign up?

We will email you if you signed up during the first lecture. If you forgot to sign up, email 6.033-lab-tas@mit.edu. We will send more information about enrollment soon.