Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.033 Lab, Spring 1995-96 Handout 1 — 6.033 Lab February 6

1 Introduction

The 6.033 lab is an elective 6-credit lab that should be taken in conjunction with 6.033. The lab is designed to provide you with hands-on experience with concepts taught in 6.033. The lab consists of a small programming exercise to get familiar with UNIX programming and two programming projects: the implementation of a reliable network protocol and the implementation of a secure network file system. The lab will be executed on top of UNIX using Athena machines. It is assumed you are fluent with the C programming language and you have a working knowledge of UNIX tools. This year is the first year that the lab will be offered, and will therefore have limited enrollment and run in pilot mode.

A word on terminology: we will refer to the unchanged 12-credit 6.033 subject as "6.033." We will call the new lab-based subject "6.033 lab." We will call the 6.033 lab projects "system projects," as opposed to the 6.033 projects, which are "design projects."

The rest of this handout details the organization and the structure of the lab. Subsequent handouts describe the individual system projects in details.

2 The lab

To successfully pass the 6.033 lab you must complete two system projects. These system projects are different from the design projects in 6.033 in that they require an implementation in addition to a paper design. The first system project explores networking issues in detail. In the second system project you are asked to create a complete system, building on the first project.

The two programming projects are preceded by a simple UNIX programming exercise. In this exercise you will be asked to modify a very simple distributed client/server program using UNIX sockets. Later in the lab you will end up using this code. This exercise is purely intended to give you some experience in UNIX network hacking.

In the first system project you are asked to implement the core set of the Internet Transmission Control Protocol (TCP), a reliable connection-oriented transmission protocol with window-based flow control. We will call this core set *simple TCP* (STCP). This project will teach you how to write reliable networking code using state machines and window-based flow control. We will provide you with a library to send and receive network packets. To help you develop your implementation we will also provide a version of the library that sends these packets over a UNIX pipe. You are asked to submit the sources and the documentation for your STCP library. In addition, we expect a short document explaining the performance of your STCP library.

In the second system project you are asked to implement a secure distributed file system (SDFS). This project is more open-ended and requires creativity on your part. You have to design a distributed file system based on the client-server model that provides storage of files on the server, caching of files on the clients, and secure communication between the client and the server. To help you design the system, this project is split in two parts. In the first part, you write a design document and you have to convince your TA that your design is reasonable and doable. In the second part, you will implement your design and analyze your implementation.

The first system project is designed to take 6 weeks for a student to complete. The last system project is designed to take 6 weeks for a team of two students to complete: two weeks for writing the design document, and five weeks to implement the design, evaluate it, and document it.

3 Requirements

You must enrolled in 6.033 this term. If you are not taking 6.033 in the same term as the lab, you cannot enroll in the 6.033 lab.

You should be fluent in C (or C++) and have working knowledge of UNIX tools. In addition, you should have taken 6.170 so that you understand how to construct large programs. To help you to get started with UNIX programming, we will assign a simple UNIX programming exercise in the first week.

4 Who should take this lab?

The 6.033 lab is an elective 6-credit addition to 6.033.

If you like to understand computer systems in detail, like to acquire more practical knowledge in building computer systems, have knowledge of C and UNIX, have time to earn another 6 credits, have Tuesdays and Thursdays free from 1 to 2pm, and are not afraid of being a guinea pig for a new lab, you should enroll in the lab. If you are taking 6.033 only because of its reading and writing components, you should not enroll. You do not need to take the lab to do well in 6.033.

Note that the lab will not satisfy the Institute or department lab requirements.

5 6.033 and the lab

We will reserve one recitation section for all students enrolled in the lab. The recitation instructor of this section, Jerry Saltzer, will assume that you are in the lab and may discuss topics related to the lab. Dawson Engler, the lab TA, is also the TA for this section. This section will meet from 1pm to 2pm on Tuesday and Thursday; if you cannot attend section at this time, you cannot take the lab.

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 use your system project papers to satisfy the design project requirements, a sweet deal.

6 Lab due dates

System project 1 is due on the same date as design project 1 in 6.033. System project 2 is due on the same date as design project 2 in 6.033. See the handout 6.033 At a Glance for more details.

7 How to enroll for the lab

Sign up for the lab during the first recitation on the first day of classes. The subject number for the lab will be 6.917, but please wait before registering for this subject, particularly because we may need to limit enrollment.

8 Grades, credits, and collaboration policy

Your grade for the 6.033 lab is independent of your 6.033 grade. You will receive 6 additional credits if you successfully complete the lab. (It is also likely you will some receive engineering design points; we are working that out with the administration.)

Your lab grade will be calculated as follows:

lab grade = 1/2 SP1 + 1/2 SP2.

Your system project will be judged on creativity, the completeness of the solutions, and the quality of the code and documentation.

The first project must be executed on an individual basis; the second project is a team project, which will be executed in teams of two students. For the first project 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 documentation on your own). Similarly, for the last system project, each team is responsible for its own code and documentation.