

# Unix Lab III

Tom Kelliher, CS 245

Sept. 23, 2002

## 1 Administrivia

### **Announcements**

Class diagrams and use case diagrams due Wednesday.

### **Assignment**

Read SE Chapter 4.

### **From Last Time**

Unix Lab II.

### **Outline**

1. Discussion of Unix characteristics.
2. Unix Lab III.

### **Coming Up**

Product design.

## 2 Unix

When starting out, the most important command: `man`. As in:

- `man man`
- `man -k g++`

### 2.1 Common Trip-Ups

1. The Delete key doesn't work.
2. There are suspended jobs.
3. The path component separator: `/`, not `\`.
4. The arrow keys and other cursor positioning keys.
5. Those “funny” keystroke sequences in `emacs`.
6. Toggling between `emacs` and the shell.
7. Forgetting your Unix text when you're working.

### 2.2 Anatomy of a Command

```
ls -aCF
```

```
ls -a -C -F ~kelliher/pub
```

```
ls -l ~
```

```
cat Class/Cs245/Exams/evil.tex
```

```
w | grep sabbott
```

```
kill 'ps gaxuw | grep sabbott | awk '{ print $2 }''
```

```
javac Prog.java  
  
java Prog < input > output  
  
rm *  
  
rm -i *  
  
alias rm 'rm -i'  
  
less .cshrc .login
```

## 2.3 Unix Concepts

1. Your userid and group.
2. Your home directory.
3. The filesystem; navigation: `.`, `..`
4. Relative, absolute pathnames.
5. Directory commands: `cd`, `mkdir`, `rmdir`, `pwd`, `ls`.
6. Filenames; wildcards, abbreviations.
7. File commands: `rm`, `less`, `cp`, `mv`.
8. File permissions.
9. The superuser.
10. Processes.
11. Environment and shell variables: `PRINTER`, `PAGER`, `EDITOR`.
12. The shell.
13. Getting help: `man`, your Unix book.
14. `.cshrc`, `.login`, `.twmrc`, and `.xinitrc`.

## 2.4 Lab

Write a Java application to solve the following problem.

The sieve of Eratosthenes is one of the earliest algorithms for generating prime numbers between 2 and  $n$ . It works as follows. Using an array with a maximum index of  $n$ , mark all numbers as being prime (`array[i]` is the indication that the number  $i$  is prime or non-prime). Mark all numbers which are multiples of two as being non-prime. Repeat for three, four,  $\dots$ ,  $(n + 1)/2$ . Finally, print all numbers which are still marked as prime.

Write a program which prints the primes less than or equal to 100.