Class 01
Intro to Unix Operating Systems

Presented by
Instructor: Greg Scott
Phone: (310)660-3203
email: gscott@elcamino.cc.ca.us
Class Topics

1. Logging In
2. The Unix Command Line
3. Command Structure
4. Command Input/Output
5. Commands/Concepts for Class 01
   - cal, cat, cd
   - date
   - echo, exit
   - hostname
   - logname, ls
   - man
   - pwd
   - redirection (>, >>)
Logging In

• Your access to Unix is controlled through a login process. You will be provided with:
  – A user login name, e.g., gscott
  – A password, e.g. class
  – A server IP address, e.g. 10.129.6.50 or a domain name, e.g. m-net.arbornet.org

• Open the TeraTerm telnet client using the icon on the desktop. In the host box, type the IP address of the server.
TeraTerm Login Screen
Logging In

- If you are using Win 2000 or Win XP telnet client,
  Click Start-Run-type Telnet and Click Ok or press Enter.
You get the Telnet Window, now type:
open 10.129.6.50 (or your server's domain name) and Enter.
Win 2000/XP Telnet

Welcome to Microsoft Telnet Client
Escape Character is 'CTRL+1'
Microsoft Telnet> open 10.129.6.50
Logging in to Unix

• Type your login name (first letter of your firstname and first 7 letters of your lastname- lowercase) and press enter. If you have arranged for a “special” login name, then use it. Note that on most Unix login screens, the password keystrokes do not echo to the screen, so that you cannot even count the keystrokes. This is for extra security in password protection.

• Your login should look similar to the next slide. When you see the $ appear, you have the command line prompt for the Bourne Shell.
Logging in to Unix

SCO OpenServer(TM) Release 5 (mcs106.elcamino.cc.ca.us) (ttyp0)

login: gscott
Password:
Last successful login for gscott: Sun Mar 09 18:32:45 2003 on ttyp0
Last unsuccessful login for gscott: Sun Mar 09 18:32:16 2003 on ttyp0

SCO OpenServer(TM) Release 5

(C) 1976-2000 The Santa Cruz Operation, Inc.
(C) 1980-1994 Microsoft Corporation
All rights reserved.

For complete copyright credits,
enter "copyrights" at the command prompt.

NOTICE: Unregistered SCO software is installed on your system. Please refer to SCO’s online help for registration information.

you have mail
TERM = (ansi)
Terminal type is ansi
$
The Command Line

• The **command line** is the most direct way to communicate with Unix. It is primitive, but more efficient, like driving with a stick vs automatic. As you type, your keystrokes are recorded in a command buffer (memory). When you make a mistake and wish to correct it, you must only use the special “**erase**” character authorized by your shell to erase the unwanted characters. On most systems, the erase character is the backspace. Another special character, called the **kill** character will kill the entire line so you can start the command over. You can use the **stty** command to display or set your terminal settings. Type the command **stty  -a** and press enter. Your screen should look similar to the next slide:
Note what the eof key is on your system. Most Unix systems use `<ctrl>-D`
Command Structure

- The commands in Unix have a basic structure:

```
cmd  [options]  {arguments}
```

- `cmd` – a keyword for a process to be performed.
- `options` (if used) alter the normal way a command is processed. Options are specified by placing a minus(−) or plus(+) after the `cmd`, followed immediately by a letter or number.
- `arguments` (if used) define what data is to be processed by the `cmd`. In the previous example, `cmd` was `stty`, the option used was `−a`. 
Command Input/Output

• A command gets its input from the arguments or from system defaults (when no args are given), or from other commands (when pipes are used). The most common default for command input is the keyboard, also called stdin.

• A command sends its output to stdout, which is the terminal screen. The output can be sent elsewhere by using output redirection. This will be discussed a bit later in the class.
The Unix filesystem consists of files and directories (groups of files) which are owned by various users of the system. The directory that each user owns and logs into is called his **home directory**

- The **pwd** command shows what directory you are currently in. Display the name of your home directory.
- The **logname** command displays what your login name is in case you forget who you are.
- The **hostname** command displays the name of the host computer that you are logged into.
- The **date** command displays the system time, day, and date information.
Commands/Concepts

- The **cd** command allows you to change to any directory that you have permission to read and execute. To change back to your home directory, just type `cd` without any argument. Your class has files that have been put into a directory for you to access. Its name is `/usr/cs40`. Change into that directory.

- The **ls** command lists the names of the files and directories in the current directory. If the `–l` (that’s el) option is used, the long (expanded) listing is displayed showing detailed information about the directory entries. List the contents of the cs40 directory.
Commands/Concepts

– The **man** command displays the description of Unix commands and how they can be used. Use the man command to find out what option(s) for the `ls` command can make it print its filenames out in a different order than the default a-z order.

– The **cal** command displays a calendar for the month [and/or] year specified. The default is the current month’s calendar. Find out what day New Year’s Eve will be on this year. Take a look at the calendar for September, 1752. You will find it “very interesting!”

– The **echo** command displays the argument you type back to the screen. Type an echo command to make the terminal say “Hello”.
Commands/Concepts

• Redirection

  – Suppose you wanted to create a file that had the calendar for 09/1752 to keep as an oddity or piece of trivia. The output from any command can be redirected in to a file for safekeeping by using the > symbol, followed by the filename you want to save it in. Make sure you are in your home directory, then type the command:

    `cal 09 1752 > short.sept`

    Verify the file is there with the ls command. Wah Lah!

  – If you want some output to be added on to an existing file’s contents, then use the appending redirection symbol >>. The > erases the contents of the target of the redirection. The >> keeps the original contents intact.
Commands/Concepts

- The cat command displays the contents of file(s) on your terminal screen. If you would like to see the contents of the file short.nov, then type: `cat short.nov`

  If cat is used without an input filename, it expects input from stdin, so you must type each line and it will be displayed on the screen like an echo. Type `cat` and press enter. Notice that the $ prompt did not return on the next line. Type anything and press enter. It will be echoed to the screen. To get out of this mode of cat, press the eof key.

  If cat is used without an input filename and with output redirection, it can be used to create a file from the keyboard. Type: `cat > somelines`
Commands/Concepts

Type a few lines, then press the eof key. Now type:
cat somelines
You should see the lines you typed!

– The **exit** command is used to terminate the login shell. To logoff, type exit. A shortcut to logoff is <ctrl>-d.

You should now be able to do the first lab assignment. In the /usr/cs40 directory, you will find a group of files named unxlab01, unxlab02, . . .

Do the first lab assignment for homework!