Introduction to UNIX

Recitation for ECES 338
January 25, 1999

I. Logging in

A. UNIX is a multi-user operating system, as such, the concept of a "user" is very tightly defined.
   1. Each "user" in UNIX typically has their own unique access to the system. And all users are separated from one another.

B. Thus, in order to start using a UNIX machine, you must first input your username, followed by your password.

C. If you do not yet have an account on the OLIN UNIX machines, you should get to a console and login using the special name "newuser".

II. "In UNIX, everything is a file."

A. `ls`
   1. This utility displays a list of the files in the current directory.
      a) **`ls -l`** – gives a *long* listing.
      b) **`ls -a`** – shows *all* files (in UNIX, a hidden file starts with a period).
      c) **NOTE:** Options can be combined.

B. `cd` / `mkdir`
   1. Use **`mkdir <name>`** in order to create a directory.
   2. Use **`cd <name>`** in order to change into a directory.

C. `rm`
   1. This program unlinks a given filename from the filesystem directory, effectively deleting it.
   2. **PLEASE BEWARE:** Once you have removed a file, it **CANNOT** be recovered!
      a) **YOU HAVE BEEN WARNED**!
   3. Usage: **`rm <options> <name>`**
      a) Where `<options>` can include:
         (1) **`-r`** – remove directories
         (2) **`-f`** – force removal (you've been warned…)

D. Security
   1. Users and Groups
      a) Every file has both a *userid* and a *grpid* associated with it.
      b) When combined with the permission bits, the system decides whether-or-not you can access a file.
      c) The following programs manipulate these fields:

(1) **chown <userid> <name>**.
(2) **chgrp <grpid> <name>**.

2. Permissions
   a) File permissions control what users have access to a file.
   b) Basically, there are three sets of three bits.
      (1) The major categories are for *user*, *group*, and *other* (world) access.
      (2) The fields that can be set per category are *read*, *write*, and *execute*.
   c) `chmod` – sets the permission bits on a file.
      (1) You can either use octal, or the symbolic notation. Consult the man page or ask me for more information.
      (2) Usage: **chmod <mode> <name>**
   d) `umask` – sets the default permission bits for all newly-created files.
      (1) Use **umask 0077** to be secure.
3. To execute a program, simply type the name of the file at the command prompt.
   a) If the shell complains that the file *isn't found*, then check your path.
   b) If the file is in your path (or in the current directory), then maybe the current directory isn't in your path.
      (1) You should see a single period (".") somewhere in your path (typically at the end).
      (2) If you don't see this, then try telling the shell *exactly* where your program is:
         (a) **./<name>**.

E. The following is a list of *important* directories:
1. `/tmp` – A machine-local repository for garbage.
2. `/etc` – Where most of the configuration files for the system are stored. Only for the curious.
3. `/usr/bin` – Where most of the Sun-provided Solaris binaries reside.
4. `/usr/local/bin` – Where the non-Sun binaries reside.
5. `/usr/openwin/bin` – Where X-Windows resides.

III. Getting help

A. UNIX man pages
1. This is the standard format for UNIX documentation.
   a) There are man pages for programs, daemons, and even standard C functions.
2. Typing **man <name>** will bring up the appropriate set of documentation.
3. If you don't know exactly what you're looking for, try the `apropos` **<name-close-to-what-you're-looking-for>**. It will `grep` through a special index, searching for possible man page matches.

B. GNU info
   1. This is the preferred method to get the documentation for GNU utilities.
   2. It provides more flexibility than man pages (allows links and navigation), but is only used by the *hardcore* GNU people.
   3. Simply type **`info`** in order to be presented with a list of topics, or **`info <topic name>`** if you know what you want.
      a) Example: **`info gcc`**

C. WWW
   1. A list of World Wide Web sites with additional information about UNIX, utilities, editors, etc. will be posted to my webpage for this course:
      a) http://home.cwru.edu/~ajr9/eces338

IV. Editors

A. vi
   1. The inappropriately named "*Visual Editor*". This editor ships with every known UNIX variant, and is still popular with some people.

B. Emacs
   1. This is considered by many to be the ultimate editor on the face of the planet.

C. Pico
   1. This is the visual editor provided by the **Pine** mail system. It is fairly modern and easy-to-use.

V. User environment

A. Startup files
   1. If your shell is **`csh`**, then it will parse through the **`.cshrc`** file upon login.
   2. If your shell is **`bash`**, then the **`.bashrc`** file will be processed.
      a) I will post further information about shells (and their environments) to my course web page.

B. When you login, you will start off in your "*home*" directory.
   1. This is **your** place to put all of your files, etc.
   2. NOTE: There is a *quota* in place, that will limit you to only *ten (10) megabytes* worth of data in your home directory, so please mind your disk space usage.
      a) Use the **`du -k`** command in order to see how much space you're using.
      b) Typing **`quota`** (with no parameters) will tell you how much you need to remove, if any.

C. Environment Variables

1. Each shell provides variables that are accessible by all programs, and govern things such as the *path*, *display*, and *manpath*, among other things.

VI. Printing

A. There are two HP laserjets in the Olin lab.
   1. *olin405* is in the NT lab.
   2. *olin404* is in the UNIX lab.

B. You can print to either printer printer with the following commands:
   1. The print a file, use:
      a) lpr –P<printername> <filename>
   2. To check the status of the print queue, use:
      a) lpq –P<printername>
   3. To remove a job from the queue, use:
      a) lprm –P<printername> <jobid>

C. **NOTE:** The notes for the textbook **do not** print properly on these printers! Please use the Kelvin Smith Library printers.

VII. Compiling

A. The GNU compiler (better known as **`gcc`**) is the compiler of choice, these days.
   1. It has many, **many** parameters, so I recommend browsing the **info** page.
   2. To get you started, try:
      a) gcc <filename.c>
      b) If you gave **`gcc`** a valid C program, it will produce an executable entitled **`a.out`**, which you can run.
   3. GCC calls **`gas`** and **`ld`** in order to assemble and link, respectively.
      a) Thus, when encountering errors, it is important to recognize exactly *what* program is complaining.
         (1) If you see something that says *syntax error*, and lists a line number in your source file, then **`gcc`** itself is complaining, and you need to fix your code.
         (2) If you see something like *unreferenced symbol*, then the problem is in the linking stage, and you must make sure that you're including the proper libraries or object files.