MA194 – Using Unix

Instructor: Dan Chaney (chaney@ms.uky.edu)
Office hours: Mon 2-3, 606-257-2150

Class #3 - Tuesday, January 23, 1996 – Week 2

In-class hotline: (606) 323–3681

Community Colleges: Tues 9:00-9:50 & 10:00-10:50 PM EST
UK/Lexington: Wed & Fri 9:00 PM EST, TCI Channel 16

URL: `http://www.ms.uky.edu/~ma194`
Mailing list: ma194-class@ms.uky.edu
To join the mailing list, send a note to ma194@ms.uky.edu

Tonight's Topics

- Boring announcements
- Unix as a multiuser/multiprocess operating system
- Logging In
- Changing your password
- A note about security
- Using the ELM Mail reader

Fun and thrilling announcements!

- *Extended registration! January 31st!*

If you are interested in taking this course for college credit, contact:

Bill Verble or the Distance Learning/Extended Campus Dept.

1-800-432-0963, ext. 73377

- *Account requests*

If you have requested an account but it's not available, check back by the end of the week. Then e-mail me or call me at 606-257-2150.

- The Account request form on the web page

Even if you already have an account, please fill out this form at: http://www.ms.uky.edu/~ma194

A multiuser operating system

An operating system that has several users (Gee!)

The operating system's responsibilities include:

- Identifying one user from another
- Identifying which files belong to which users
- Determining which resources can be 'shared'
- Keeping track (accounting) of users and resources

A multiprocessor operating system

An operating system that can do several things at once.

Here, the OS's responsibilities are:

- Managing resources
- Coordinating information flow from one 'area' to the next
- Tracking which jobs need which resources
- Keeping everything in check

A multiuser/multiprocess operating system

A multiuser operating system, as it's name implies, can have several users, each with their own files, configurations, etc.

Examples:

DOS was a single-user, single-process operating system

Windows 3.1 was a single-user, multiprocess operating system.

Windows '95 is a multiuser, multiprocess operating system.

Unix is a multiuser, multiprocess operating system – with a twist. Unix allows multiple users to acess the system *simultaneously*. Whereas the other 'multiuser' OSs have the facility for multiple users, but necessarily concurrently.

Using Unix

So, to access a Unix system, you must be able to uniquely identify yourself to this all-singing OS, in order for it to know who you are and how to treat you.

The *login ID* (or login name or just login) is how the OS will recognize you from all the other users.

The *password* is how the OS insures it is, in fact, you and not someone attempting to be you. (Why would someone do that? Wait for the **Security** slide :-)

Logging In

Here's a sample login screen:

```
Linux 1.2.8 (pigpen.ate.uky.edu) (ttyp0)


pigpen login:
Password:
```

The first line tells you the name of the system you are accessing. In this case, it also tells you the name of the operating system, the version of the OS and what port you are coming in on. This information may change from system to system.

Logging In - Successfully!

Here's a sample login screen:

```
Linux 1.2.8 (pigpen.ate.uky.edu) (ttyp0)


pigpen login:
Password:
Last login: Thu Jan 18 21:15:10 from t2
Linux 1.2.8. (POSIX).
You have mail.

pigpen:~$
```

The initial screen

Once you have correctly entered your login ID and password, the system will actually ‘log you in’ or register you. It may tell you various things, depending on the system’s configuration, administrator, etc. In this case, we are told the last time we logged in (foreshadow the **Security** slide!) and whether we have any email.

Finally, we are given a *system prompt*. This will be what we use for the text-based interface to Unix. (We’ll also be going over the Graphical User Interface or GUI, called X-windows.)

Notice that the system prompt has the name of the machine we are logged onto.

Logging In – Access methods

There are many ways to login to a Unix system, depending on the way you are accessig the system:

- network - **telnet** or **rlogin**
- console - The 'main screen' of a Unix system
- terminal - A terminal connected via a serial port
- modem - A modem connected via a serial port

Types of accounts under Unix

There are basically two kinds of accounts under Unix, the user account (which is what you have) and the administrators account (which is what the person(s) running the Unix system uses to fix/break things.)

The **root** or **superuser** account is used for maintaining a Unix system. Adding software, adding hardware, adjusting the way certain things work (mail, printers, networking, etc.) are duties and abilities that not everyone should have, but that someone clearly needs to be able to do.

All other accounts are considered *user* accounts, even though some of them may not be used by any one person.

The **Security** slide

Why should I worry about the security of my account?

1. You are responsible for the contents of your files.
   Copyrighted software? Test answers? Too much disk usage?

2. You are responsible for the email from your account.
   Threatening letters? Inappropriate language? Annoying solicitations?

3. You are responsible for this particular tool.
   Test answers? Homework? Grade information?

4. You are responsible for the activity on your account. It's a multiuser system. If someone else online is being boethered by a user with your login ID, clearly, that person must be you.

5. If a hacker can get to your account, (s)he can get to others. The weakest link of security on a Unix system is the person that treats security the lightest.

End of today's sermon :-)

The **Security** slide – How-to!

How can you maintain the security of your account?

Keep your password protected, hidden and:

1. DO NOT Write it down

2. DO NOT Make it trivial to guess:
   First name, spouse's name, *account name*, class name, children's name, birthday or any common words.

3. DO NOT use any combination of the above, reversed or cutely misspelled (3=e, 0=o, 1=i, 2l instead of ll, etc)

Rule of Thumb:
If it's easy to remember at first, it's easy to break at first.

Changing your password

This is a very important practice to begin early. Change your password often and make sure it is not a trivial password.

Good passwords are usually 6-8 characters long and a mixture of characters and numbers.

```
chaney:~$ passwd
franklin:~$ passwd
Changing password for chaney
Enter new password:
Re-type new password:
Password changed.
```

Sending E-mail via ELM

Elm is a mail *user agent*, a program used to both read and send mail. To run the elm program, you need to issue the command:

elm

If you are running X-windows, you will need to 'bring up' an xterm, by clicking the left mouse button on the root window, selecting the shells submenu and choosing one of the entries from that list.

Running Elm for the first time

The first time you run elm, the program will need to create some files and directories in order to have working room in the future. The questions are listed below. You should answer ‘y’ to both of them.

```
Notice:
This version of ELM requires the use of a .elm directory in your
home dir to store your elmrc and alias files. Shall I create the
directory .elm for you and set it up (y/n/q)? Yes.
Great! I'll do it now.

Notice:
ELM requires the use of a folders dir to store your mail folders.
Shall I create the directory /home/chaney/Mail for you (y/n/q)? y
```

The *index screen* of the Elm mail reader

```
 Mailbox is '/var/spool/mail/chaney' with 2 messages [ELM 2.4 PL24]


1   Aug 29 Dan Chaney      (25)   Test of my email
2   Aug 23 root            (702)  no subject (file transmission)


   You can use any of the commands by pressing the first character;
 d)elete or u)ndelete mail,  m)ail a message,  r)eply or f)orward
  To read a message, press <return>.  j = move down, k = move up

Command:
```

The *options screen* of the Elm mail reader

```
                     -- ELM Options Editor --

D)isplay mail using  : builtin+
E)ditor (primary)    : /usr/bin/vi
F)older directory    : /home/chaney/Mail
S)orting criteria    : Reverse-Sent
O)utbound mail saved : =sent
P)rint mail using    : /bin/cat %s | /usr/bin/lpr
Y)our full name      : Dan Chaney
V)isual Editor (~v)  : /usr/bin/vi

Select letter of option, '>' to save, or 'i' to return to index.
Command:
```

Sending mail – the ‘m’ option

```
Command: Mail
Send the message to: ma194@ms.uky.edu
```

```
Command: Mail                                  To: ma194@ms.uky.edu
Subject of message: Hello Dan!
```

Sending mail – Entering the message

```
Command: Mail                                    To: ma194@ms.uky.edu
Subject of message: Hello Dan!
Copies to:

Enter message.  Type Elm commands on lines by themselves.
Commands include:  ^D or '.' to end, ~p to list, ~? for help.
```

Dear Dan,
This is just a sample note to show that I understand how to send email from with the Elm program. What now?

G. Philpott

Sending mail – Sending the message

Finally, after you have entered the message, a final menu appears:

```
Please choose one of the following options by parenthesized letter:
            e)dit message, edit h)eaders, s)end it, or f)orget it.
```

Choose **s** to send your message and Voila! You've sent email!

Suggested readings

For logging in

- Absolute Beginner's Guide to Unix
  Chapter 3, p 40-50

- The command: **man login**

For e-mail and *Elm*

- Absolute Beginner's Guide to Unix
  Chapter 7

- The command: **man elm**