Lecture # 2 – Introduction to UNIX (Part 2)

- UNIX is case sensitive (lowercase, lowercase, lowercase)

- Logging in (Terminal Method)

  Two basic techniques:

  1. Network login (using SSH client)
  2. Console login

  login: <enter username>
  Password: <enter password>

  <<< some text will appear>>>

  prompt (often $)

  When done, you can logout by typing "exit" at the prompt (or ctrl-D).

- Logging in (GUI Method)

  The Linux machines in the UAH labs are running a GUI interface.

  You might see a list of machines willing to manage your login session.
  If so, choose the machine you want to login to

  You should see a login box.

  login: <enter username>
  Password: <enter password>

  Screen should change, and you should be a windowed desktop.
  You may have to open a terminal window via an icon or menu choice.

  To logout, right click on desktop, and choose Exit or Logout.

- Getting help

  Textbook
  Internet searches (Google, Yahoo, etc.)
  Class notes
  Online manual pages – man <command>

Example:

$ man write

The online manual is broken up into several sections:

1. User commands
2. System calls
3. Subroutines
4. Devices
5. File Formats
6. Games
7. Misc
8. System Administration
9. Local
10. New

Sometimes a command or topic appears in the manual in multiple sections, and you can include the specific section to view a particular page.

$ man 2 write

- PICO editor

  $ pico filename

  Most common editing commands are displayed on screen.

- VI basics

  Check terminal type (echo $TERM)

  $ export TERM=vt100 (for Bourne, Bash, or Korn shell)
  $ setenv TERM vt100 (for Csh or Tcsh)

  To start "vi":

  $ vi <filename>

  Command mode versus input mode (:set showmode)
  Default mode is command mode on startup

<Esc>	Change to command mode
i	Change to input mode (if in command mode)
h	One character Left
j	One character down
k	One character up
l	One character right
x	Character delete
dw	Delete word
dd	Delete line
u	Undo
ZZ	Save and Exit
:wq	Save and Exit
:x	Save and Exit
:q!	Exit without saving

Command review

- File manipulation commands

1. ls – directory listing

   ls [-l] [-s] [-a] [dir or file names…]

   -l long listing
   -s include file sizes
   -a all files

   Display a listing of files in the current directory

   ```
   $ ls
   file1  file2  homework3.txt
   ```

   Display a listing of files in the /tmp directory

   ```
   $ ls /tmp
   homework4.txt  file8.txt
   ```

   Display a long listing of the current directory

   ```
   $ ls –l

   [rjohnson@lancelot testdir]$ ls -ls
   total 4
     0 -rw-r--r--    1 rjohnson rjohnson     0 Dec  5 14:53 file1
     0 -rw-r--r--    1 rjohnson rjohnson     0 Dec  5 14:53 file2
     0 -rw-r--r--    1 rjohnson rjohnson     0 Dec  5 14:53 file3
     0 lrwxrwxrwx    1 rjohnson rjohnson     5 Dec  5 14:53 link1 -> file1
     4 drwxrwxr-x    2 rjohnson rjohnson  4096 Dec  5 14:53 tmpdir
   [rjohnson@lancelot testdir]$
   ```

2. cat and more – displaying the contents of a file

   "cat" display contents of a file to screen
   "more" does the same except the output is presented one page at a time

   cat <filename> (i.e. $ cat test1.txt)
   more <filename> (i.e. $ more test2.txt)

3. cp – copying files

   cp <original file> <new file>

   Example:

   ```
   $ cp file1 file2
   ```

4. mv – rename files

   mv <original filename> <new filename>

   $ ls
   file1 file2 file3

   $ mv file1 file1.txt

   $ ls
   file1.txt file2 file3

5. rm – remove files

   rm <option> <file>

   -f force
   -i interactive (ask before removing)
   -r recursive

   $ rm homework4

6. grep – searching files

   grep <string> <filename>

This is line1. This is line2 for cs390. This is line3.

homework1.txt

   $ grep cs390 homework1.txt
   This is line2 for cs390.

7. head and tail - display the top or bottom section of a file.

   head <line count> <filename>
   tail <line count> <filename>

   Display the top 10 lines of text from the file “testing”
   $ head testing
   Display the last 2 line of text from the file “testing”
   $ tail –2 testing

8. sort – sort the contents of a file

   sort <options> <filename>

   Sort the file "test.txt" based on the entire line
   $ sort test.txt
   Sort numerically
   $ sort –n test.txt
   Sort by the 3rd field, and then by the 2nd field
   $ sort –k3n,3 –k2b,2 test.txt

9. uniq – display the unique lines of a file

   uniq <filename>

   Note: The file must be sorted first to work properly.

   $ uniq testing.txt

   This same functionality is available as an option on sort (-u)

   $ sort –u file1.txt

10. diff - display differences between 2 files

    diff <file1> <file2>

    Shows three things:
    1. lines in file1 not in file2 (denoted with "d")
    2. lines in file2 not in file1 (denoted with "a")
    3. lines that are different in file1 and file2 (denoted with "c")

    $ diff homework1.txt homework2.txt

11. file – display file type information

    file <filename>

    $ file file1.dat
    file1: data

- Other utilities

1. echo – display a string

   echo <string>

   $ echo “this is a test”

   Used in shell script programming much like printf in C.

2. date – display the date and time

   $ date
   Mon Jan 12 08:05:03 CST 2004

   $ date +”%m-%d-%Y”
   01-12-2004

   Note: See man page for full listing of the formatting commands

3. script – record a shell session

   script <filename>

   Record the current session into a file called session.txt

   ```
   $ script session.txt
   Script started, file is session.txt
   $ ls
   file1  file3  file4
   $ exit
   Script done, file is session.txt
   $
   ```

4. gzip – compress files

   gzip [-d] <filename>

   Compress a file called “letter”

   ```
   $ ls –l
   -rw-rw-r--    1    alex    speedy 584000   Jul 31 06:07  letter
   $  gzip letter
   $ ls –l
   -rw-rw-r--    1    alex    speedy 2030    Jul  31  06:07  letter.gz
   ```

   Uncompress this file

   ```
   $ gzip –d letter.gz
   ```

5. tar – pack or unpack file

   tar [c|t|x] [v] [f] <archive file> <file list…>

   Create a tar file containing file "g", "b", and "d"
   ```
   $ tar –cvf file.tar g b d
   ```
   List the contents of the tar file "file.tar"
   ```
   $ tar –tvf file.tar
   ```
   Extract the files from the tar file "file.tar"
   ```
   $ tar –xvf file.tar
   ```

6. which and whereis – locating commands

   which <filename>
   whereis <filename>

   ```
   $ which tar
   /bin/tar

   $ whereis tar
   tar: /bin/tar /usr/include/tar.h /usr/share/man/man1/tar.1.gz
   ```

- User information commands

1. who – list users on the system

   ```
   $ who
   root   console    Mar 27 05:00
   alex   pts/4      Mar 27 12:23
   alex   pts/5      Mar 27 12:33
   jenny  pts/7      Mar 26 08:45
   ```

2. finger – list users on the system

   finger [user]

   ```
   $ finger
   Login  Name         Tty   Idle   Login Time          Office
   root   root         1     1:35   May 24 08:38
   alex   Alex Watson  /0           Jun  7 12:46  (:0)
   jenny  Jenny Chen   /2    2:24   Jun  2 05:33  (bravo.tcorp.com)

   $ finger alex
   <<< information about user alex >>>
   ```

3. w – list users on the system

```
$ w
        8:20 a.m.       up 4 days,   2:28,  3 users,  load average:  0.04, 0.04, 0.00
USER TTY    FROM        LOGIN@      IDLE  JCPU   PCPU  WHAT
alex pts/4  :0          5:55 a.m.   13:45 0.15s  0.07s w
scott pts/12 bravo      7:17 p.m.         1.00s  0.14s run_bdgt
```

- User communication commands

1. write – send a message to another user

   write <user> [terminal]

```
$ write alex
Hi Alex, are you thre?
```

2. talk – 2 way communication with another user

```
$ talk jenny
```

3. mesg – denies or accepts messages

```
$ mesg n
```

- Email

1. pine or elm – menu driven mail utilities
2. mail – read or send email

```
$ mail richj@hiwaay.net
This is a test email.
.
```

- Special Characters

  & ; | * ? ‘ “ `[ ] ( ) $ < > { } ^ # / \ % ! ~ +

- Quoting Special Characters

  You backslash (\) to “escape” special characters