UNIX Tutorial #1

Mike Goldsmith

March 9th, 2004

~ 1 hr

UNIX Shells

- UNIX machines support multiple simultaneous users
- Every user on a UNIX machine executes applications within a **shell**
- Isolates effects of one user on another user
- Can run a *local* shell or a *remote* shell (on another machine)

UNIX Shells

- Types of shells:
  - **csh** – The C-Shell
  - **bash** – The Bourne Again Shell
  - **rsh** / **remsh** – Remote Shell
  - **ssh** – Secure Remote Shell

UNIX Shells

- Shell registry control files (.rc files)
  - .cshrc
  - .bashrc
- These files are scripts that get executed when you start a new shell
  - When you type ‘bash’ at a command prompt, it first runs the .bashrc file when starting up the shell, which allows you to set system variables, etc.

System/Environment Variables

- These are user-specific variables that are used by several programs
  - TERM -> what terminal type are you using, eg ‘vt100’
  - CVSROOT -> the location of your CVS repository
  - HOME -> the user’s personal folder in /home

System/Environment Variables

- Can set and unset these variables using commands (different for each shell)
  - csh:
    - **setenv** CVSROOT /home/magoldsm/foo
    - **unsetenv** CVSROOT
  - bash:
    - **export** CVSROOT=/home/magoldsm/foo
- Use with no parameters to see all set variables

RTFM

- Most programs have ‘manual’ pages, which can be accessed using the ‘man’ program
  - **man** chgrp -> shows the manual page for ‘chgrp’
- Most programs also support command-line help using the ‘-h’ flag
  - chgrp –h -> brings up the inline help

Who What Where?

- **who** – tells you who else is on the machine you're on.
- **whoami** – tells you who you're logged in as
- **finger** *username* – gives you information about a particular user
- **which** *programname* – tells you the location from root of a particular program. Will also return if the program does not exist

Directory Tools

- **ls** – shows the current contents of the directory. –l and –A flags are very useful
- **mkdir** *dirname* – make a directory
- **rmdir** *dirname* – remove a directory, but only if empty

File Tools

- **cp** *filename location* – copies a file from one point to another, or duplicates file
- **mv** *filename location* – moves a file to a particular location, or renames a file
- **rm** *filename* – removes a file. Can use rm –Rf *dirname* to remove (with force) recursively an entire directory tree

File Ownership

- UNIX has strong security built in for files. Each file or directory can be given read, write and execute permissions to the owner-user, group and ‘everyone else’
- Use **chmod** *octet filename* to set these permissions, where ‘octet’ is a three or four-octal number, corresponding to owner-group-else
  - Eg: chmod 744 foo.sh ->gives rwxr--r– permissions to the file ‘foo.sh’, which means rwx for owner, r for group and r for else

File Ownership

- Use **chown** *username filename* to change the owner of a file to the new user
- Use **chgrp** *groupname filename* to change the group a file belongs to

Group Membership

- Each user is part of at least one group, known as their default group, and possibly many ancillary groups.
- Use **groups** *username* to see which groups a user belongs to; if no username given will show your groups.
- Use **newgrp -** *groupname* to switch your current active group – will also start a new shell

User Maintainance

- Use **passwd** to set the current user’s password
- Use **id** to see which user and group you’re currently active as

Application Tools

- Use **ps** to see which processes you’re running (or with –a to see all processes)
- Use **top** to see a real-time consumption view of processes on the system
- Use **kill** *–level process_id* to terminate an active process –9 means **immediately**
- Can redirect output of an application using the pipe (|) command
  - Eg: ls –l | more -> pipes the output of the list command to the more command, which pauses the scrolling display at each page