CS 406 – Linux System Administration – Spring 2006

Solutions to Homework #1

Note: the answers here are what I would consider standard and/or best answers, though there may be others that are acceptable as well.

1. Commands to reset current working directory:
   (1) `cd` (which is equivalent to doing `cd $HOME`)
   (2) `cd ~`
   (3) `cd ~carver` (replaced with your username)
   (4) `cd /home/carver` (replaced with your username)

2. Command to determine path of `mv`:
   `which mv` (this is the simplest/best)

3. Two commands to display your current search path:
   (1) `printenv PATH`
   (2) `echo $PATH`

4. Command to show references to "pipe" in the Bash man pages:
   `man bash | grep pipe`

5. Command to delete some files from `/tmp`:
   `rm [AB]*406*{.txt,.text}`

6. Hidden files:
   Files/directories whose names begin with a dot are considered hidden files, and many file listing programs (e.g., `ls`) won't display them *by default.* Most hidden files are configuration files/directories.

7. Command to list all files in the current working directory:
   `ls -A` (also acceptable is `ls -a`, though it lists `.` and `..`)

8. Meaning of output from `ls -l test`:
   `-rwxrw-r--  2 carver faculty 1732 Jan 23 20:39 test`
   (a) Owner is carver, with read, write, and execute permissions.
   (b) Group is faculty, with read and write permissions (not execute).
   (c) Users other than carver or members of the faculty group have read permission only.
   (d) The file is 1732 *bytes.*
   (e) Jan 23 20:39 is the last date/time the file contents was *modified.*
   (f) The 2 before carver means there are two hard links for this disk file.

9. Simplest command for adding execute permissions for owner:
   `chmod u+x test`

10. What set UID (SIUD) files are, and why they are used/needed:
    (1) SUID files are executable files that act as if they were run by the *owner* of the file rather than the user that actually does start them, specifically, the process will have its *effective user ID* (EUID) set to the owner of the file, which may be different from the *real user ID* (RUID).
    (2) SUID files are required to allow users to run programs that can do things that the users would not normally be permitted to do, such as modify the /etc/passwd file.

11. Why SIUD root files are considered a security issue:
    An SUID root file will run with its EUID being root, which means it will have permission to do virtually anything on the system. If the program has bugs it might able to damage the system. For example, if it has a *buffer overflow* vulnerability, a normal user or a hacker might be able to exploit this bug to take control of the system.

12. Command to set a file to be SUID:
    `chmod u+s file` or `chmod 4555 file` (if want normal permissions to be 555 octal).

13. Difference between the `su` and `sudo` commands:
    `su` starts a new shell process for another user, which basically means switching the UID of the shell. Typically it is used to switch to running the CLI as root (UID of 0). `sudo` allows non-root users to run a single command as if they were root, if they have the appropriate permissions to do so in the file `/etc/sudoers`.

14. Command to determine the process ID (PID) of a running SSH server: Typically would pipe `ps` command through `grep`, such as:
    `ps ax | grep sshd`
    (Note that `ps` has many options and accepts styles of arguments, so there are many variations on this, depending on what you want to see—often you want to see more than just the PID to make certain it is the desired process.)

15. Command to best identify which process is using excessive CPU resources:
    `top`

16. Command to definitely terminate the process identified above:
    `kill -9 pid` or `kill -SIGKILL pid`
    (Need the 9 or SIGKILL to guarantee process is killed.)

17. Purpose of the `/home` subdirectory:
    Contains the home directories of users (other than root).

18. Purpose of the `/boot` subdirectory:
    Contains the linux kernal files and related files needed to boot the OS.

19. The superuser's home directory:
    `/root`

20. What a logical partition is:
    It is a partition that is created inside of an *extended partition.*

21. Why it is advantageous to install Linux into logical partitions:
    Only 4 *primary partitions* are allowed on a disk (one of which may be an *extended partition*, while up to 16 logical partitions can be created insides of an extended partition. Since Linux can boot from logical partitions, this gives one more flexibility in partitioning a disk.

22. Two main problems sharing a `/home` partition among multiple Linux distributions:
(1) The UIDs and GIDs for a username and group might not match, so file ownership would not be correct in some distributions, and (2) Configuration files (e.g., for KDE and its applications) may not be the same if different versions of software are being used in each distribution.

23. *This question was supposed to be:*
Why it is common to use a separate partition for the `/var` subdirectory:
Because `/var` contains variable files like log files, print spools, etc. which may grow rapidly under certain circumstances, so using a separate partition would prevent the system from crashing due the root partition becoming full.

24. Filesystem pathname for the first IDE harddrive:
`/dev/hda`

25. Filesystem pathname for the second SCSI/SATA harddrive:
`/dev/sdb`

26. Filesystem pathname for the first logical partition on the third IDE harddrive:
`/dev/hdc5`

27. Default configuration files for GRUB and LILO:
GRUB: `/boot/menu.lst`
LILO: `/etc/lilo.conf`

28. Main advantage that GRUB has over LILO:
The main advantage is that GRUB can understand filesystems when it runs, so can find kernels to boot at boot time, and does not require a running/bootable Linux to make changes to the boot options. LILO, by contrast, requires that it configuration changes must be enabled/installed from a running/booable Linux.

29. The two runlevels Linux systems commonly boot to, and how they differ:
3 is the multiuser but non-GUI runlevel, while 5 is the multiuser GUI runlevel.

30. How the default runlevel for a Linux system is determined:
The file `/etc/inittab` contains a line like this that specifies the default runlevel:
`id:5:initdefault:`

31. Command to change runlevel to runlevel i:
`telinit i` or else `init i`
(on Linux systems, `telinit` is just a symbolic link to `init`).

32. How does one configure a Linux system so that single user mode is not a security problem:
In the file `/etc/inittab`, one puts a line like: `su:S:wait:/sbin/sulogin`

33. Where the startup scripts on a RedHat-family Linux are stored:
In the directory `/etc/rc.d/init.d` (which is symlinked from `/etc/init.d`).

34. Two commands that will shutdown the network interfaces on a RedHat-family Linux:
(1) `service network stop`
(2) `/etc/rc.d/init.d/network stop`

35. Command to keep the network service from starting at boot:
`chkconfig --del network`

36. File that contains the list of valid user ID's and usernames:
`/etc/passwd`

37. File that contains passwords on a Linux system:
`/etc/shadow`

38. Configuration file that contains the partitions along with their associated mount points:
`/etc/fstab`

39. Two main generic access control files for servers:
`/etc/hosts.allow` and `/etc/hosts.deny`

40. Main configuration file for the SSH server:
`/etc/ssh/sshd_config`

41. Why it is good to disallow direct SSH login as root, and how accomplished:
Every Linux system will have a root username, so attackers will try brute force password cracking attacks against root. Can eliminate this threat by disallowing root logins via SSH, by having the following in `/etc/ssh/sshd_config`:
`PermitRootLogin no`

42. Command(s) to tell you the listening servers on a system:
Typically the `netstat` command is used with some options, such as `netstat -lpt` to list listening TCP sockets/processes and `netstat -lpu` to list listening UDP sockets/processes.

43. Command to diagnose "no route to host" message:
This message indicates that packets cannot be routed to the specified host (IP address). Assuming that this is not an external network problem, this would indicate a problem with the routing tables of your system, so one should start with the `route` command and verify that: (1) the correct gateway is specified, and (2) the correct local network (IP address and netmask) is specified.

44. Command to diagnose "name not known" message:
This indicates that the hostname cannot be converted to an IP address. Assuming that either the local hosts file or DNS are being used (as opposed to NIS, etc.), this indicates that the host is not in the `/etc/hosts` file and that the name cannot be found in the DNS system. One can use the `host` and `dig` commands to try DNS lookups, and one could simply list out the `/etc/hosts` using `more`. Potentially the hostname could be added to the `/etc/hosts` file if it is not in the DNS system for some reason. (Note: if there is a DNS failure due to incorrect DNS servers being set up in `/etc/resolv.conf`, you get a message like: "temporary failure in name resolution.") (The file `/etc/nsswitch.conf` specifies the name lookup sources and the order in which they should be consulted.)

45. Checking that your Ethernet card (NIC) has been properly set up:
Use the `ifconfig` command, and make certain that three entries are correct for the NIC interface (e.g., eth0): inet addr, Bcast, and Mask.