

## CIS 68C1

### Homework 2 Answers

Due: October 8<sup>th</sup>, 2001

Instructions: Answer the following questions in your own words; do not copy words from the book, other sources, or other students – rather, summarize your understanding and comprehension of the material. Remember, the goal here is to explore the system and to *learn!* You may simply copy the questions below to a word processor, and then type your answers below the questions. Bring your printed answers with you to class on the due date above – we will go over the questions and answers then. Remember, your answers must be typewritten (sorry - no email this time).

1. Take a look at the /etc/rc#.d startup directories used to boot Linux. Run `ls -l` on each of these directories. You should find that all the files are symbolic links. Where do these links point (i.e. where are the actual files)?

First of all, you should notice that /etc/rc.d is a directory, but in fact the /etc/rc?.d files are not directories, but are in fact symbolic links pointing to directories elsewhere:

```
$ ls -ld /etc/rc?.d
lrwxrwxrwx 1 root root 10 Jan 1 1990 /etc/rc0.d -> rc.d/rc0.d
lrwxrwxrwx 1 root root 10 Jan 1 1990 /etc/rc1.d -> rc.d/rc1.d
lrwxrwxrwx 1 root root 10 Jan 1 1990 /etc/rc2.d -> rc.d/rc2.d
lrwxrwxrwx 1 root root 10 Jan 1 1990 /etc/rc3.d -> rc.d/rc3.d
lrwxrwxrwx 1 root root 10 Jan 1 1990 /etc/rc4.d -> rc.d/rc4.d
lrwxrwxrwx 1 root root 10 Jan 1 1990 /etc/rc5.d -> rc.d/rc5.d
lrwxrwxrwx 1 root root 10 Jan 1 1990 /etc/rc6.d -> rc.d/rc6.d
```

So, /etc/rc0.d is really /etc/rc.d/rc0.d, etc.

All of the files in the /etc/rc.d/rc0.d, /etc/rc.d/rc1.d, etc. are actually symbolic links. The links all point to files in the directory ../init.d. This allows for a single shell script to be used at various runlevels, without having to copy the shell script itself, thus avoiding duplicate copies.

```
$ ls -l /etc/rc2.d
lrwxrwxrwx 1 root root 10 Jan 1 1990 /etc/rc2.d -> rc.d/rc2.d
```

(notice that `ls -l` does not follow the symlink when you do not add the trailing /. Add the trailing / and it will follow the symlink)

```
$ ls -l /etc/rc2.d/
total 0
lrwxrwxrwx 1 root root 15 Jan 1 1990 K01pppoe -> ../init.d/pppoe
lrwxrwxrwx 1 root root 15 Jan 1 1990 K03rhnsd -> ../init.d/rhnsd
lrwxrwxrwx 1 root root 13 Jan 1 1990 K20nfs -> ../init.d/nfs
```

[ remaining output omitted ]

2. What init levels typically correspond to single user and multi-user mode?

Single-user: 1 or S, Multi-user: 2 or 3. RedHat Linux uses runlevel 5 as the same as runlevel 3, but a graphical login will be used on the console instead of a text based login.

3. What does the “telinit” command do, and what is it used for?

The telinit command is used to change run levels. It does this by sending a signal to init. Telinit is really just a symbolic link to init:

```
lrwxrwxrwx  1 root  root           4 Jan  1  1990 /sbin/telinit -> init
```

4. There is a single command that will output just the currently running init level in RedHat Linux? What is the commands name, and what is its output when run during multi-user mode? Hint: Look at the SEE ALSO section of the man page for telinit.

The command is **runlevel**.

```
$ /sbin/runlevel
N 3
```

The output is N 3 (or N 5 if you are using the graphical console logon). The first field (N) is the previous runlevel (N = no previous level) and the second field is the current runlevel (3).

By the way, the command for Solaris is **who -a**.

5. What is the name of the program used to boot Linux?

**LILO**

6. What are the names and PIDs of the first 5 processes in Linux?

It may vary depending on which version of Linux you have:

```
1: init
2: kflushd
3: kupdate
4: kpiod or kswapd
5: kswapd or keventd
```

7. What do you suppose the [3] means in the CMD field of the ps -ef output for process id 1?

It is the current runlevel.

8. The /etc/rc#.d/\* startup scripts expect an argument – what is this argument and what are two typical values for the argument?

The scripts accept an argument to tell them how to behave; the required arguments are: **start** and **stop**. Other arguments commonly used are: **status**, **restart**, **reload**, **condrestart**. When a script is passed the **start** argument, the script is supposed to start its services. Conversely, when it receives the **stop** argument, it is supposed to kill or stop the services that it controls.

9. What makes the root user, UID 0, so special?

UID is special because the kernel allows UID 0 to perform any operation that normal users cannot. When checking for privileges to perform an operation, if a user's process permissions do not match the file permissions, or if an operation is considered privileged (e.g. setting the system clock), the kernel does not allow the operation, and returns an error to the users. The kernel will, however, grant permission to UID 0.

10. How many different Manual sections are there in RedHat Linux? What are the names and numbers of the sections you as an administrator would use?

There are about 9 to 12 sections, depending upon your version. The sections are: man1 - man9, and sometimes manl, mann, and mans, and others. The section numbers and titles are below; those that are certainly used by average system administrators are in bold:

```
1: User level commands and apps
2: System calls and kernel error codes
3: Library calls
4: Special Files
5: File Formats
6: Games and demonstration
7: Conventions and Miscellany
8: System Administration commands
9: Obscure kernel specs and interfaces
```

11. What are the various ways the system can be safely powered down or rebooted?

Use any of the following commands: **shutdown**, **halt**, **reboot**, **telinit 0**, **telinit 6**. Some systems allow you to send a signal to init (which is what telinit does), but this is not a good practice. Also, never just turn off the power, unless you are certain there is no other choice.

12. Can user “sam” temporarily become user “sally”? If so, how is this accomplished?

Yes. Use the command **su sally** or **su – sally**. The later sources the user sally's startup scripts.

13. Describe why it poor practice to use the root account as a general user account?

It is bad practice for several reasons: a) it decreases your ability to determine *who* created files or who performed some action, b) it is risky because you can accidentally delete or modify critical system files inadvertently, c) you may inadvertently leave a root session running on your system, possibly making it available to others, and d) as root, you have privileges that others do not and the system will behave differently for you that it does for your users. When administrating a system, it is valuable to *be* a normal user of the system just like others are so that you can experience first hand how the system behaves for everyone else.

14. Which runs first in Linux: S80sendmail or S20netfs?

Startup scripts (those that start with an S) run in numeric, then alphabetic order. 20 comes before 80, so the S20netfs script will run before the S80sendmail script.

15. Take a look at the /etc/rc3.d/K20nfs and /etc/rc3.d/S15nfs startup scripts. What do you notice about these two scripts?

S15nfs and K20nfs are really symbolic links, that both point to a single script file:

```
$ ll K20nfs S15nfs
lrwxrwxrwx  1 root    root      13 Jan  1  1990 K20nfs -> ../init.d/nfs
lrwxrwxrwx  1 root    root      13 Jun 20 14:12 S15nfs -> ../init.d/nfs
```

This allows a single script to be written so that it can be used for different purposes. Your system may have different links.