



Assignment 1: Shell Programming

Submission: by 5pm, Monday, 10 February 2003

Where: Electronically via this URL: <http://nicku.org/perl2/submit.cgi> You can submit as many times as you like; a later submission will never overwrite an earlier submission.

Format: Must be *plain text*. Submissions in proprietary formats (i.e., Word documents) will be copied straight to `/dev/null` and will receive *ZERO MARKS*.

You can put the shell scripts together as a `.zip` file or as a tarball. Do not submit `.rar` archives.

Add your email address Put your email address in a comment on the second line of your submissions, so that I can return them marked to you. I will not return assignments that do not have an email address on the second line of each file.

Weighting: This assignment shall provide 20% of your Continuous Assessment mark, i.e., 10% of the total marks for this subject.

Marking Criteria: Originality is essential. Efficient implementations get more marks. Elegant and well-structured solutions get more marks. Flexible solutions get more marks (i.e., offer more useful options to the user).

Original submissions that answer the first four of these questions will receive a bonus of 5% if they are submitted one week before the deadline.

Cheating: Your work *must* be original. Copying will be *severely* dealt with.

1 Background

1.1 Example Programs and Online Resources

I have written a number of example shell scripts—please download them, run them, examine them, understand them. Get them from <http://nicku.org/ossi/lectures/shell/bin/>. My colleague Joe Lee, the lecturer for the part time version of this subject, wrote some of these examples too. Thanks, Joe.

Please also remember that there are hundreds of excellent examples in the freely downloadable book *Advanced Bash Scripting Guide* at <http://tldp.org/LDP/abs/html/index.html>.

There is also an excellent book about the `awk` programming language, available from <http://www.ssc.com/ssc/eap/>.

Note: I have put one copy of the book, *Learning the Bash Shell*, 2nd edition, by Cameron Newham and Bill Rosenblat, in the library open reserve. Another copy is on

the shelves for loan. I have also purchased some other books on shell programming for the library, and expect them to be available after Christmas. One book you may find helpful is *UNIX Power Tools*, 3rd edition.

1.2 Set User ID and Set Group ID Permissions

Files that have the “set user ID bit” (SUID) set, and those which have the “set group ID bit” (SGID) set may be security risks, since a program file with the permission SUID will execute as the user that owns the file, while a program with the SGID permission will execute as the group owner of the file. You can set the SUID permission on a file like this:

```
$ sudo chmod u+s <file>
```

You can remove the SUID permission from the file called *<file>* with this command:

```
$ sudo chmod u-s <file>
```

Similarly, you can add the SGID permission with

```
$ sudo chmod g+s <file>
```

Read the documentation for `chmod` with

```
$ info chmod
```

or

```
$ pinfo chmod
```

My preferred way to read `info` documentation is in `emacs` by:

- start `emacs`
- type `(Control-h)(i)` to go to the top `info` menu

I.e., hold down the `(Control)` key, and press the `(h)` key, then release them both, then press the `(i)` key.

- type

```
(m)chmod
```

- Click on hyperlinks with the middle mouse button, and also click on the arrows at the top of `emacs`.

You can use the `-perms` option to `find` to identify files that have any particular permissions.

1.3 Some uses of the RPM Package Manager, `rpm`

Using the RPM Package Manager, `rpm`, you can:

- determine which software package a file belongs to:

```
$ rpm -qf <file>
```


Assignment Requirements

If these requirements are not clear, please come and ask for clarification. I will then update this document with the clarification. Note the version number at the bottom right of each page.

1. Write a shell script that will display all the times that the computer was booted, as recorded in all the files `/var/log/messages*`.
2. Write a shell script that will show the number of times that the user `root` has logged in each day of at least the last four weeks. Do not include days that the `root` user did not log in at all.

Note that the `last` command is useful here. Do `man last` for more information.

Note that the `/var/log/wtmp` log is *rotated* every month so that there is at least four weeks of data available before the old log is deleted. At the start of every month, the following happens automatically:

- `/var/log/wtmp.1` is deleted
- `/var/log/wtmp` is renamed to `/var/log/wtmp.1`
- A new, empty `/var/log/wtmp` log file is created.

The behaviour is controlled by the `logrotate` software package, and in particular, by the following extract from `/etc/logrotate.conf`:

```
# no packages own wtmp -- we'll rotate them here
/var/log/wtmp {
    monthly
    create 0664 root utmp
    rotate 1
}
```

Read the manual page: `man logrotate`

3. Write a shell script that will show all failed attempts to log in as `root` for each day of at least the last four weeks. Provide options to sort by date, or by the number of failed attempts.

I suggest that you investigate the man page for the `lastb` command. You will need to create a file `/var/log/btmp` and process output from the command `lastb`.

4. Write a shell script that will list the size of each directory given on the command line, sorted by size. The size includes disk space used by the directory and all the files and subdirectories inside it. The script should take options to sort with smallest first, and with largest first.
5. Write a shell script that can search for all files below any given directory, and provide a nice report on all ordinary files that have the SUID bit set, and also on files that have the SGID bit set. Refer to section 1.2.
6. Extend the program from question 5 so that the script can look up each file in the RPM package manager database and determine if the original state of the file had the SUID (or SGID) permission set. Refer to section 1.3.