



Assignment: Shell Programming

Submission: by 8 pm, Sunday, 16 May 2004

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.

If you are inside the Campus, you can use SSL and submit via <https://nicku.org/perl2/submit.cgi>

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 25% of your Continuous Assessment mark.

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).

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/>.

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. One book you may find helpful is *UNIX Power Tools*, 3rd edition.

2 Requirements

A number of shell scripts are required. A number of simple programs are required that exercise techniques that you will use in later programs. A number of these programs require commands that you may not have used before. Use the `man` command to investigate how to use them.

1. Use the command `basename` and *command substitution* to write a simple program that prints its own name. This should still work, even if the program is renamed. Here is an example of running it:

```
$ ./myname.sh
This is myname.sh
$ mv myname.sh what-is-my-name
$ ./what-is-my-name
This is what-is-my-name
```

2. Use the `id` program to determine the *effective user ID number* of the user that runs the program. Print the message “I am root” if the user ID number is 0; print a different message otherwise.

Note: you should check the user ID number, not the user ID name.

3. Write a script that prints an error message and exits if a *lockfile* exists. A *lockfile* is a file in the `/tmp` directory, with a name equal to the *basename* of the script, followed by the string, “.lock” For example, if the name of the script is `lockit`, then the name of the lockfile shall be `/tmp/lockit.lock`
4. Modify your answer to question 3 so that the lockfile will be `/var/lock/subsys/⟨basename of script⟩.lock` if the user executing it is root, `/tmp/⟨basename of script⟩.lock` otherwise.
5. Modify your answer to question 4 so that:
 - if the lockfile exists, the program will print an error message and terminate;
 - if the lockfile doesn’t exist, then the program creates it, and makes sure that the lockfile was successfully created;
 - when the program exits normally, it will remove the lockfile.
6. Download the script <http://nicku.org/ossi/lectures/shell/bin/trapall>. Make it executable and run it. Send the executing process all signals in the range 1..31 as described in the existing comments in the file. Modify it so that it will still print the message, but it will exit when it receives any signal. In the modified program, add comments that:
 - explain how you could terminate the original program;
 - list all signals in the range 1..31 that the original program could not trap;
 - explain *why* the signals you list cannot be trapped by the original program.

7. Modify your answer to question 5 so that if the program receives any of the following signals: INT, TERM, HUP or QUIT, the program will do all of the following:
- print a message;
 - remove the lockfile;
 - exit with a non-zero exit status.

8. Configure your system with `keychain`, as described at the web site: <http://www.gentoo.org/proj/en/keychain.xml>. Does this work for you? Write a text file `keychain-setup.txt` which lists the steps you used to set up `keychain` in 30 lines or less. Use your own words.

9. Modify your answer to question 7 so that your program will:

- source the `keychain` information about where your agent is;
- use the `rsync` program to copy the contents of the directory `~/public_html` to the directory `~/public_html-backup` in your own account on another computer.

For example, if your local account contains the file `~/public_html/important.html`, then the file `~/public_html-backup/important.html` will be created on the remote computer if it does not exist, and the content will be identical on both machines.

10. Modify your answer to question 9 so that your program will take a list of names in the form `<computer>:<destination>` on the command line, and will make each location contain a copy of `~/public_html` on the local machine.

For example, if the program is called `rsync-to-websites`, then if it is executed like this:

```
$ rsync-to-websites ictlab:/var/www/html/mystuff \  
172.19.64.50:/var/www/backup 172.19.64.51:/var/www/html
```

and if the local machine has a file `~/public_html/important.html`, then on the computer `ictlab`, a file `/var/www/html/mystuff/important.html` will be created, and on the computer `172.19.64.50`, the file `/var/www/backup/important.html` will be created, and on the computer `172.19.64.51`, the file `/var/www/html/important.html` will be created.

11. Write a text file `problems-with-locking.txt` in which you answer the following question: What are the limitations of the file locking system that you have implemented in question 10? Explain how it can go wrong. What is the technical term for this limitation?
12. Write a text file `crontab-setup.txt` in which you explain how you can set up the `cron` program to execute the program you wrote as your answer to question 10 every morning at 3.24 am. It should be executed with a command line shown in the previous question.

See `man 5 crontab` and `man crontab`.

3 Notes

- You need to submit 12 text files, most of which contain a shell script, but some of which contain plain text answering the question.
- One file, your answer to question 6, will be a shell script that contains some text answering the questions. That file will still be executable as a shell script; the text you write in that file should be comments; each line of text that is not the actual program will begin with the comment character, '#’.
- The twelve files should be zipped or tarred together into one compressed file, and be submitted online as explained on page 1.
- Your program that answers question 10 on the previous page should provide the features required for questions 7 and 4, and should use `keychain` to access your SSH agent.
- Your programs should all provide proper handling of error conditions and should all print a suitable error message and exit if some error condition prevents the program completing normal execution.
- You may assume that the programs execute on a system that complies with the current POSIX standards.
- You may ask any questions by email, to the address Nick Urbanik <nicku@nicku.org>. The subject line should contain the text, “OSSSI ASSIGNMENT”. If I answer your question, I will send the answer to *all* students, but I will not reveal the identity of the person who asked the question.
- Although I am giving you a long time to hand in this assignment, I strongly advise you to begin it immediately. I am not responsible for those who leave it to the deadline and who then disturb their own preparation for the examinations.