

## Linux Assignment 2

This assignment assumes you have read Chapters 2, 3, 4 and 5 of Sobell. Some specific cross-references may also be given in the questions below.

Prepare your answers to the questions below in a single plain ASCII text file.<sup>1</sup> Submissions in other formats will be ignored. If you work with a partner, make sure the submitted file contains a properly-completed copy of the Partners Form posted on the assignments page. Failure to do that will result in at least one of you not receiving credit for the assignment.

Submit your file to the Curator system, under the heading L02, by the posted deadline for this assignment. No late submissions will be graded.

---

For questions 1-8 use the provided submissions.tar file.

1. **[8 Points]** On your CentoVM, create the following directory in your home (~) directory:

```
2505/L02/submissions
```

You should have already created the 2505/L02 directories per the L01 assignment.

What command single did you use to create the submission directory and what was your working directory when you issued the command to create the `submissions` directory?

---

<sup>1</sup>How can you tell if you've prepared a plain ASCII text file? The simplest way is to use a Linux text editor to create the file you are going to submit (e.g., gedit, geany, etc). Alternatively, use the file command in a Linux shell:

```
centOS > file Tolstoy.txt
Tolstoy.txt: ASCII text
```

It is possible you may get a different response; for example:

```
ASCII text, with very long lines
ASCII text, with CRLF line terminators
ASCII text, with very long lines, with CRLF line terminators
UTF-8 Unicode text
UTF-8 Unicode text, with very long lines
UTF-8 Unicode text, with CRLF line terminators
UTF-8 Unicode text, with very long lines, with CRLF line terminators
Non-ISO extended-ASCII text
Non-ISO extended-ASCII text, with very long lines, with CRLF line. . .
Pascal source, ASCII text
C source, ASCII text
```

Those are probably all acceptable. However:

- long lines indicate your file may be difficult to view in certain applications; try inserting line breaks around column 80 when you type your files
- CRLF line terminators indicate your file was produced on a Windows environment; that may also cause issues if your file is opened in certain applications
- Non-ISO extended ASCII text indicates you've used non-standard characters in your file; this may be due to the inclusion of garbage text you don't intend to submit; such characters may cause some applications to interpret the file incorrectly, or even to refuse to open it

On the other hand, responses like these from the file command would indicate your file is certainly not acceptable:

```
Microsoft Word 2007+
PDF document, version 1.5
Rich Text Format data, version 1, ANSI
POSIX tar archive
data
gzip compressed data, from Unix. . .
```

You can also use the `cat` or `less` command to display your file to a Linux terminal window. If this displays anything other than the simple text of your answers, it's not a plain text file.

Apple users beware: the standard text editors on Apple computers seem to save files in RTF format, even if they indicate they are saving as plain text. We will NOT grade RTF files. You must convert the file to Plain Text before you submit. This is on the Format Menu -> Make Plain Text, or Shift-Command-T or you may either install a text editor that can save files in the correct format, or prepare your file using a Linux editor.

If you aren't sure, ask a TA to look at your file before you submit it.

2. **[8 Points]** From the `submissions` directory you made in the previous question, move the `submissions.tar` file to the `submissions` directory. What single command did you use to do this?
3. **[8 Points]** Untar the `submissions.tar` file in your `submissions` directory. What single command did you use?

Just a note about the files. These are meant to simulate a directory of student submissions. The letters are meant to mimic a VT pid, the C02 is the assignment name, the number after that is the number for this (made up) student's submission, and the `.txt` is the file extension for this assignment.

4. **[8 Points]** Write a single `ls` command that will show all of the files that start with the letters `ab`.
5. **[8 Points]** Write a single `ls` command that will show all of the files that start with an `e` and have a `9` for the submission number.
6. **[10 Points]** Now move to the `L02` directory. Write a single command to tar all of the files that have an `8` as the submission number so that the file is created in the `2505` directory. Use relative paths in your answer. Call this file `eights.tar`. Do not just use the `safer.tar.sh` script.
7. **[8 Points]** Repeat the question above, but make a zip file instead of a tar file. Call this file `eights.zip`.
8. **[10 Points]** Finally, copy both `eights.tar` and `eights.zip` to your home directory on login using secure copy. Your working directory is still `L02` and you may not change your directory.

Now suppose your working directory is `Vector` as shown below, which is rooted in the `PFiles` directory somewhere in the system's file system.

```
PFiles
`-- Vector <----- You are here
    |-- catch.hpp
    |-- CMakeLists.txt
    |-- grader
    |   |-- catch.hpp
    |   |-- CMakeLists.txt
    |   |-- instructor_tests.cpp
    |   |-- parse-tests.py
    |   |-- parse-valgrind.py
    |   `-- run
    |-- grader_sieve
    |   |-- catch.hpp
    |   |-- CMakeLists.txt
    |   |-- InstructorSieve.cpp
    |   |-- InstructorSieve.hpp
    |   |-- instructor_tests.cpp
    |   `-- InstructorVector.hpp
    |-- instructor_tests.cpp
    |-- main.cpp
    |-- Sieve.cpp
    |-- Sieve.hpp
    |-- sieve.ini
    |-- SieveIterative.cpp
    |-- SieveIterative.hpp
    |-- sieve.txt
    |-- Vector.cpp
    |-- Vector.hpp
    |-- vector.ini
    `-- vector_test.cpp7
```

9. **[8 Points]** Write a single command that would copy `vector_test.cpp` to the user's Documents directory, `~/Documents`.
10. **[8 Points]** Write a command that would list the files in the `grader_sieve` directory in sorted order ascending by file size (smallest files first).
11. **[8 Points]** Write a command that would copy the entire directory tree rooted at `grader`, including all of the files within `grader`, to the user's `~/backup` directory.
12. **[8 Points]** Write a command that would remove the entire directory tree rooted at `grader_seive`, including the directory `grader_sieve` as well.