
csce215 — UNIX/Linux Fundamentals

Fall 2021 — Lab Assignment 6

This assignment is intended to provide some practice and additional content for the material covered in lecture on Monday, October 18. You'll practice using `find`, `grep`, and regular expressions. The assignment is meant to be completed in the lab sessions on Wednesday, October 20, and must be submitted by 11:59pm on Friday, October 22. A total of 40 points are available.

1 Get started

First, a reminder: If you are in the lab during the scheduled time, don't forget to use the QR code or the password to claim credit for attendance.



Record your attendance using the QR code.

The format of this assignment will be the same as the previous weeks. You know the drill by now: make and navigate to a lab6 directory, download lab6.zip from

`https://cse.sc.edu/~jokane/215/lab6.zip`

and then unzip lab6.zip unto your lab6 directory.



Use asciinema to record your terminal session. Create a new directory called lab6 to use for the tasks below. Download and unzip lab6.zip.

2 Exploring a big program: Finding files

The files included in lab6.zip are the source code, mostly in C, for a recent version of a classic drawing program called xfig. This includes about 3500 files spread across over 100

directories. This is a good example of the source code for a ‘real’ piece of software (rather than a small program designed specifically for a class).

Use `cd` to navigate to the `xfig-3.2.8b` directory.

Let’s use this directory to practice searching for certain files. For each of the following criteria, use `find`, with appropriate tests, to print a list of files that match those criteria. *Some of these will require tests that we did not directly discuss in the lecture, so you’ll need to check the documentation for the `find` command to complete these.*

1. Files with an `.xpm` extension. Those are images in a certain format that traditionally worked well with graphical Linux programs. (Hint: There are 2 of them.)
2. Directories whose name contains the word ‘metric’. (Hint: There are 3 of them.)
3. Files whose size is exactly 351 bytes. (Hint: There are 13 of them.)
4. Files with a `.c` extension whose file size is at least 90,000 bytes. (Hint: There are 4 of them.)
5. Files that are executable. (Hint 1: There’s a specific test that asks `find` to check whether something is executable.) (Hint 2: Be sure to exclude directories, which may also be considered ‘executable’ — We’ll discuss this in Chapter 7.) (Hint 3: There are 8 of them.)
6. Files (not directories) that are newer, i.e. have a modification time more recent than, the format documentation file `doc/html/fig-format.html`. (Hint: There are 37 such files.)



From within the `xfig-3.2.8b` directory, give a `find` command that lists exactly the files and/or directories that meet each of the criteria listed above. *18 points*

3 Exploring a big program: Searching for patterns

Now let’s look for some patterns inside the files, to practice searching for various types of code. For each of the patterns below, use `grep -r -E` to find and print lines, from any file in `xfig-3.2.8b`, that match those patterns.

1. Lines that contain the string `ellipse`, in any context. (Hint: There are 995 such lines.)

2. Lines that mention one of xfig's 'drawing initialization' functions. These are functions whose names start with `init_` and end with `_drawing`. (Hint: There are 58 such lines.)
3. Lines that mention one of the data types `uint16`, `uint32`, `uint16_t`, or `uint32_t`. (Hint: There are 22 such lines.)
4. Lines that contain either 'fixme' or 'todo', either of which are sometimes added to comments as a reminder that the code is still a work-in-progress. These can be either upper case or lower case. (Hint: There are 33 such lines.)
5. Lines that *begin with* `4 2`. (Hint: There are 193 such lines.) (Who cares? In the xfig file format, diagrams are stored as text files, in which (more or less) each line defines an object in the diagram. Lines that start with `4` represent text labels; the `2` after that indicates that the label should be right-justified. So you in this case, you are searching for right-justified text labels.)
6. Lines that *end with* a pound sign (`#`) followed by exactly 6 hexadecimal digits (0–9, a–f). (Hint: There are 3464 such lines.) (Who cares? In the xfig file format, these lines represent colors used to represent custom colors to use for other objects in the diagram.)



From within the `xfig-3.2.8b` directory, give a `grep` command that lists exactly the source code lines that meet each of the criteria listed above. *18 points*

Finally, here's one more `grep` exercise. Let's look for lines that contain a copyright notice. Each copyright notice consists of the word 'copyright' (which may or may not be capitalized), possibly followed by a '(c)' (which may or may not be capitalized and might be missing completely), followed by a four-digit year. To give you a sense of how varied these lines might be, here are some examples from this xfig code.

```
# Copyright (C) 2004-2020 Free Software Foundation, Inc.  
> Parts Copyright (c) 1991 by Paul King  
  uses code from xpaint (Copyright 1993 David Koblas (koblas@netcom.com)  
  and Copyright 1995, 1996 Torsten Martinsen (bullestock@dk-online.dk))  
<br>Parts Copyright (c) 1989-1999 by Brian V. Smith  
* Parts Copyright (c) 1991 by Paul King
```

Just like the exercises above, use `grep` with a regular expression to find all of the lines with these kinds of copyright notices. As a hint, there are 884 such copyright notices.



From within the `xfig-3.2.8b` directory, give a `grep` command that finds and lists all of the copyright notices, as described above. *4 points*

4 *It's Done*

When you've completed all of the tasks above, submit as usual. End the asciinema recording using `Ctrl-D` or `exit`. Replay it ensure that it shows all of the steps mentioned in the green boxes. Then upload and submit.



Upload `lab6.cast` and then submit.

