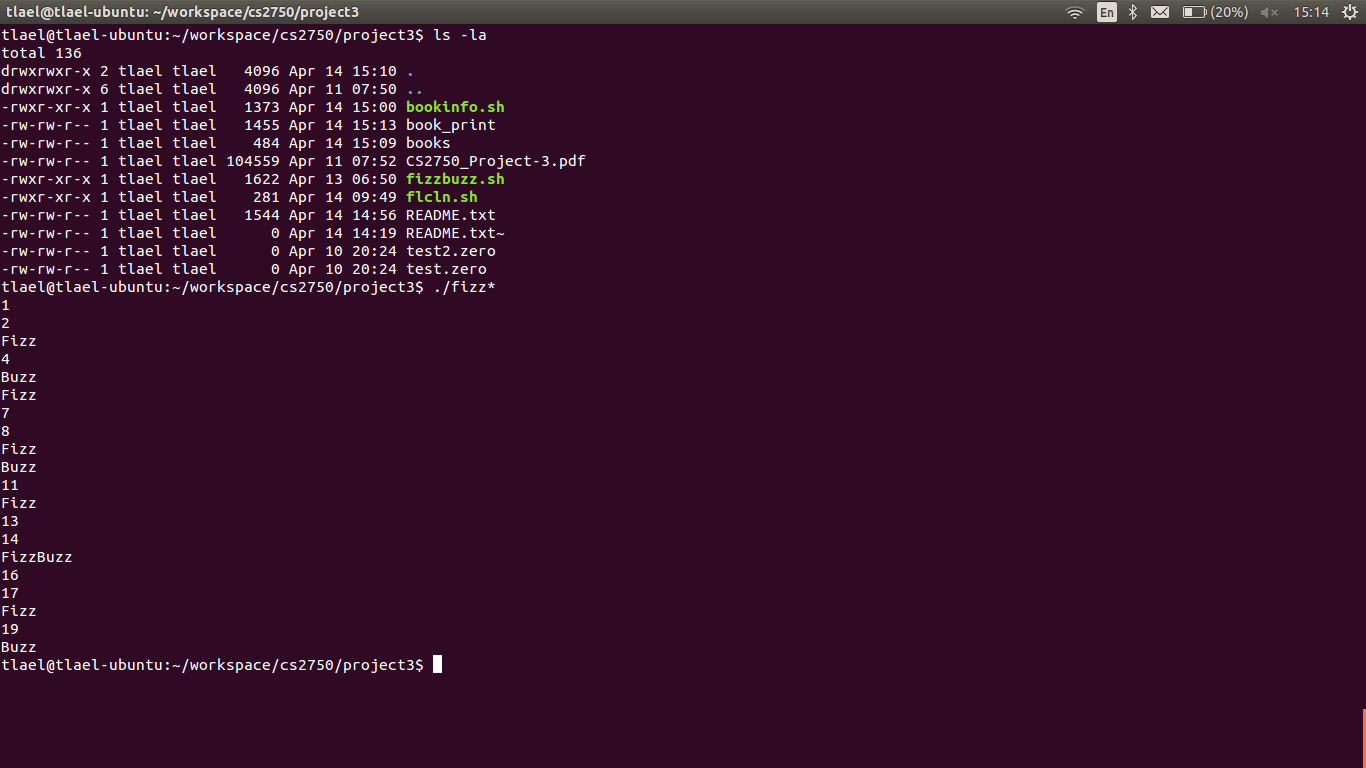
# fizzbuzz.sh

This script will print Fizz if the number evaluated is divisible by 3, Buzz if divisible by 5 and FizzBuzz if divisible by both, 3 and 5. If executed with no argument, the values 1..20 are evaluated and the result for each is printed on its own line. If passed an argument (which must be an integer), the loop will evaluate all numbers from 1..n where n is the passed integer argument.

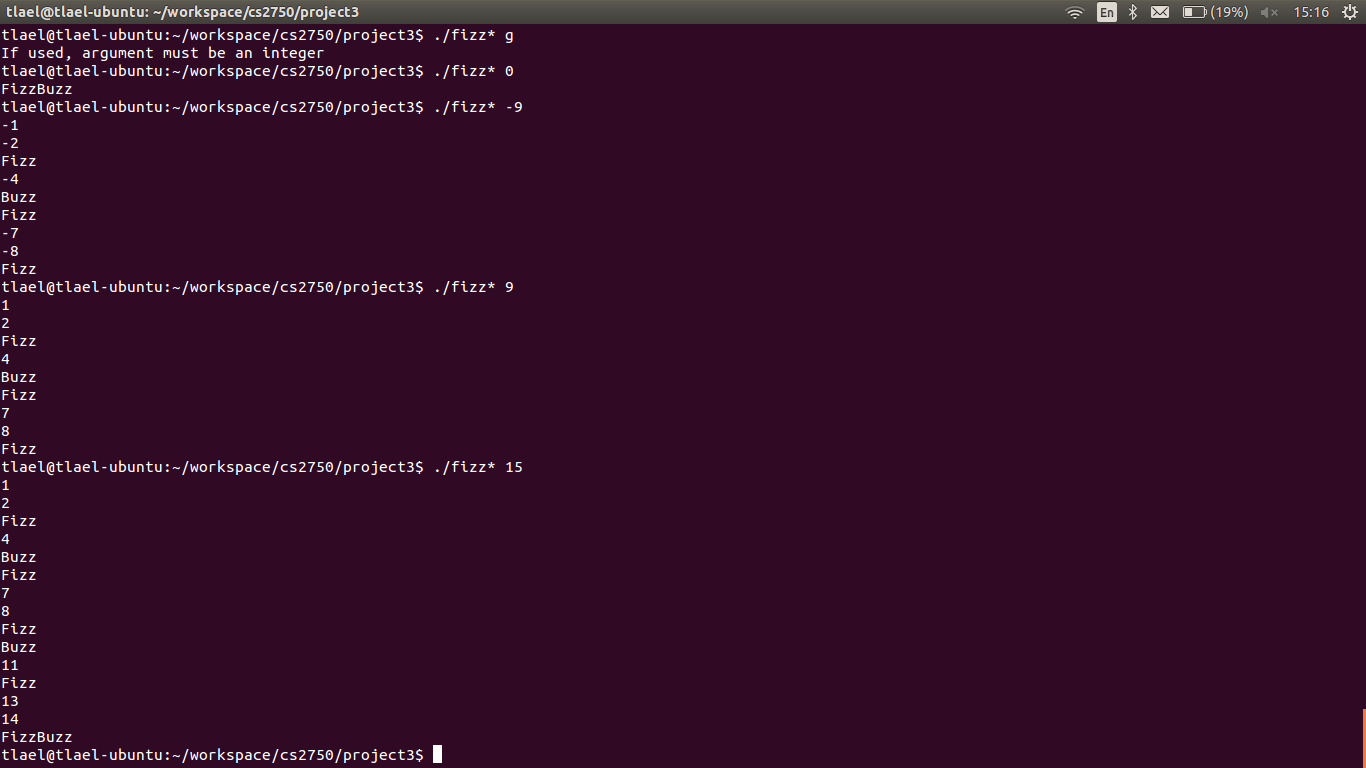
I have added the ability to evaluate 0 and negative numbers also. The evaluation of zero is done statically and results in FizzBuzz and was included since I used regex to check that the passed argument is an integer containing 0-9 and the signs +, -. Zero is a valid combination of 10, so zero must also be allowed by itself. Also, since I allowed for signage, zero is allowed as 0, -0 and +0 – all evaluated to FizzBuzz.

### Screen captures:

Default/no parameter passed



various parameters passed

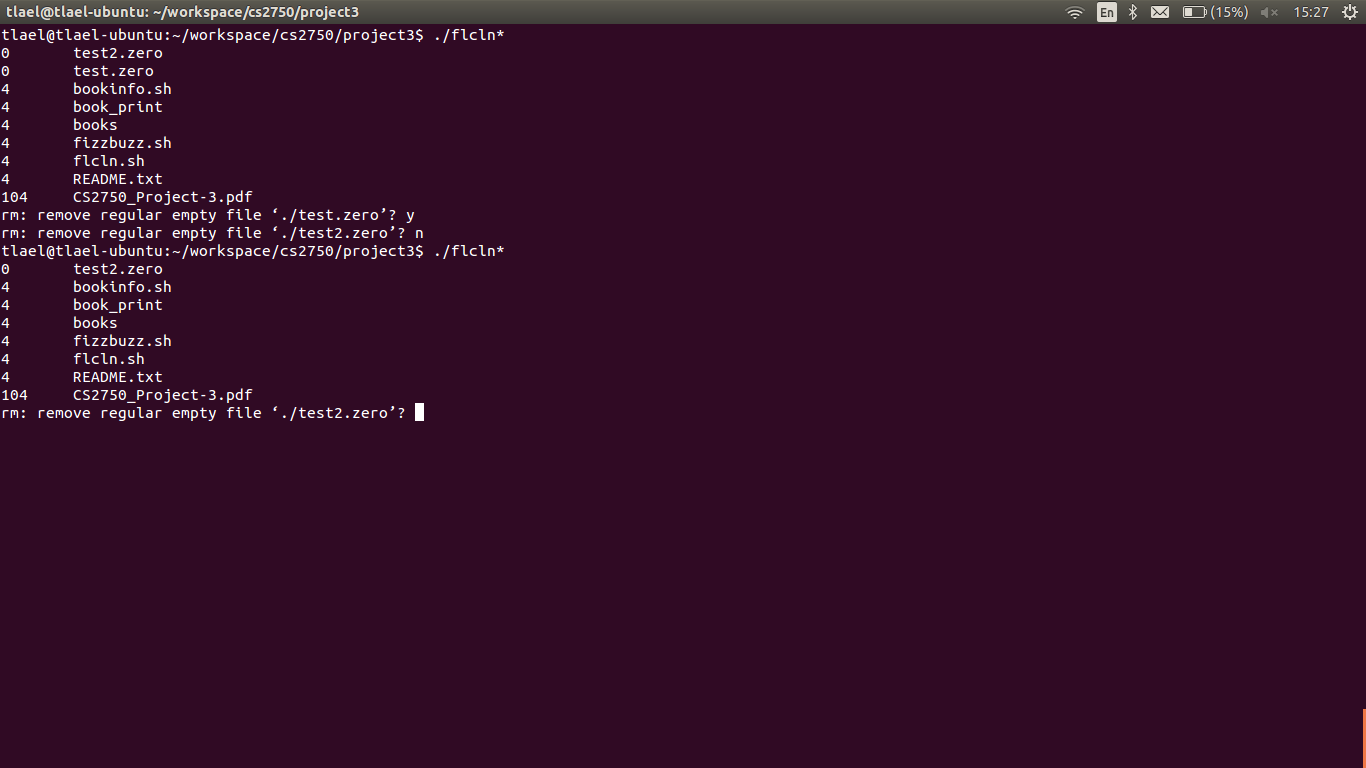


# flcln.sh

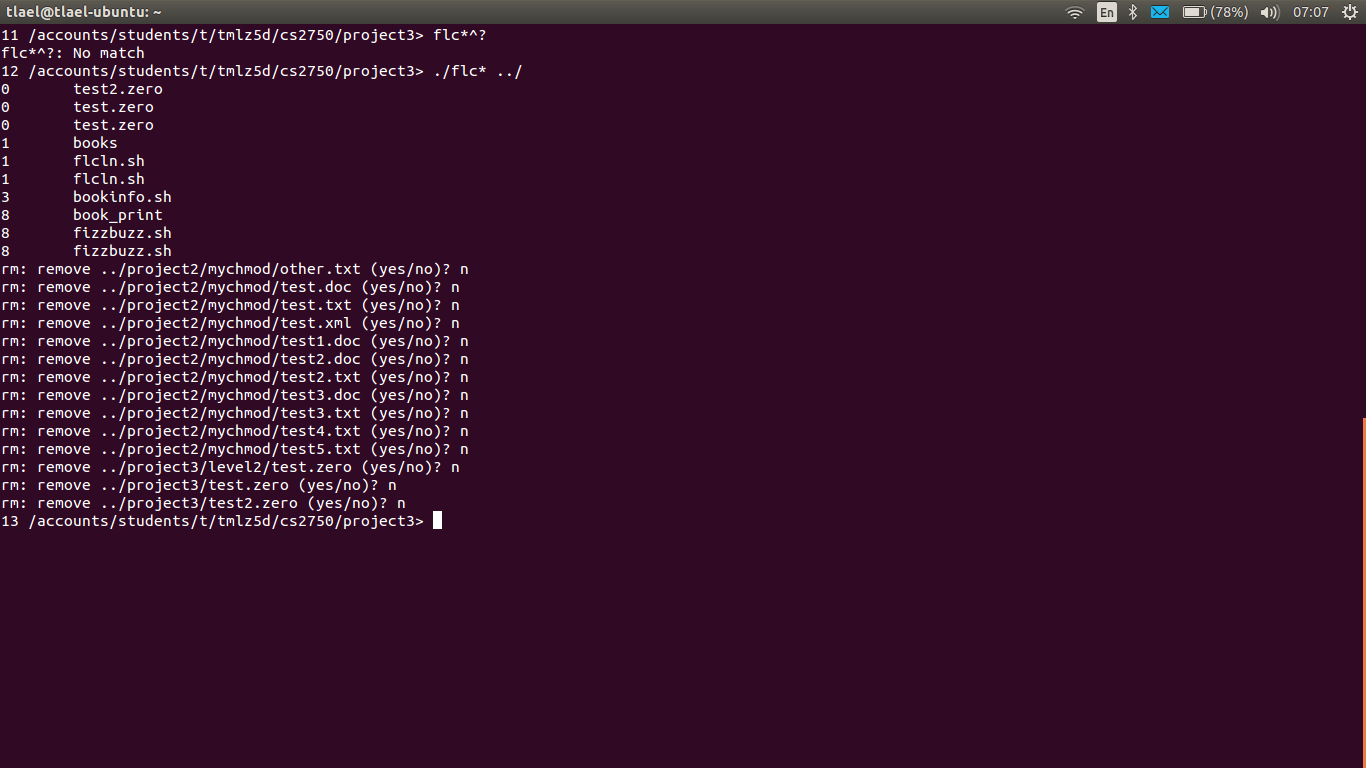
This script will list all files in a directory and its subdirectories and sort them based on size. When the result is displayed, only the size and file name is displayed. Any files with 0 size will cause the user to be prompted if they would like to delete such a file. Since this script spans subdirectories, the delete prompt will provide path for clarity of the location of the file to be deleted. If no argument is passed, the CWD is the directory where the search/sort/removal of empty files begins. One can also pass a path as an argument and that becomes the starting point for the search/sort/removal of empty files.

### Screen captures:

Default/no parameter passed



Path passed

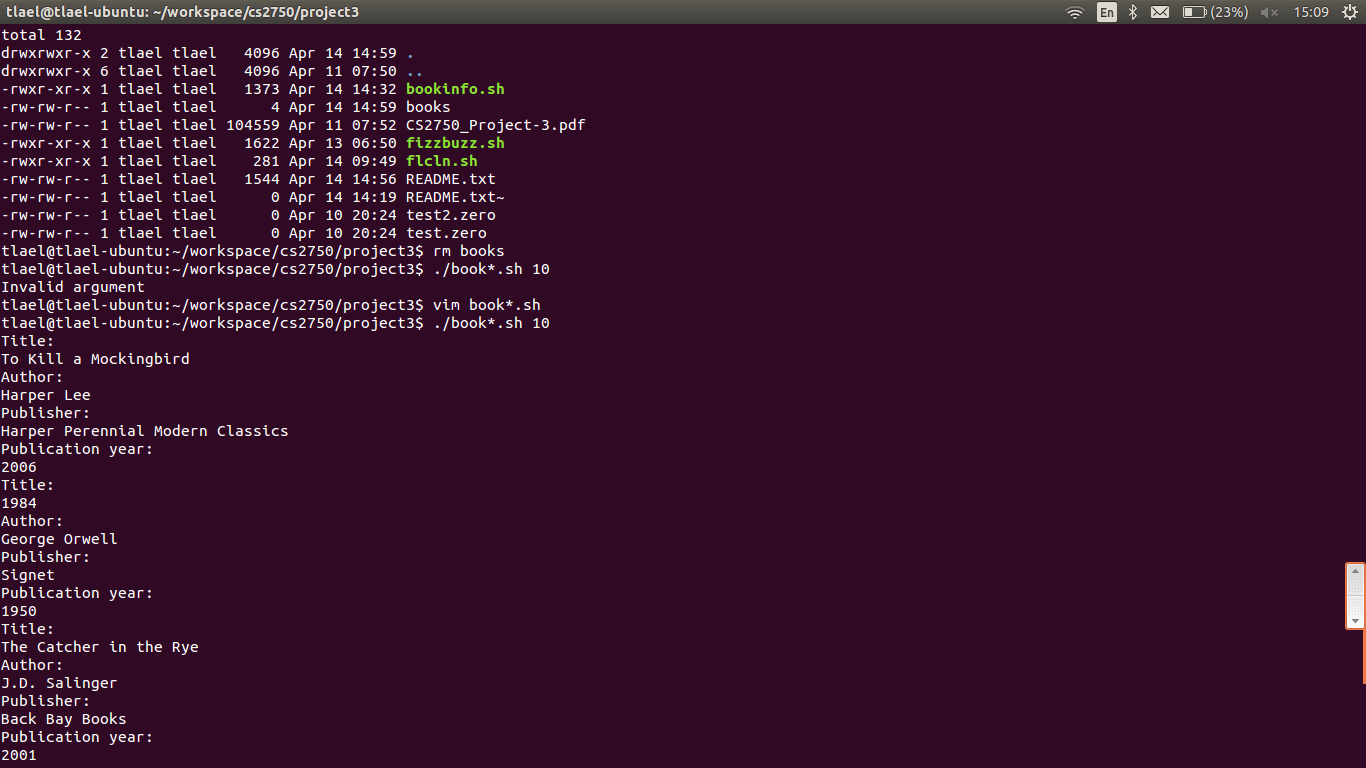


# bookinfo.sh

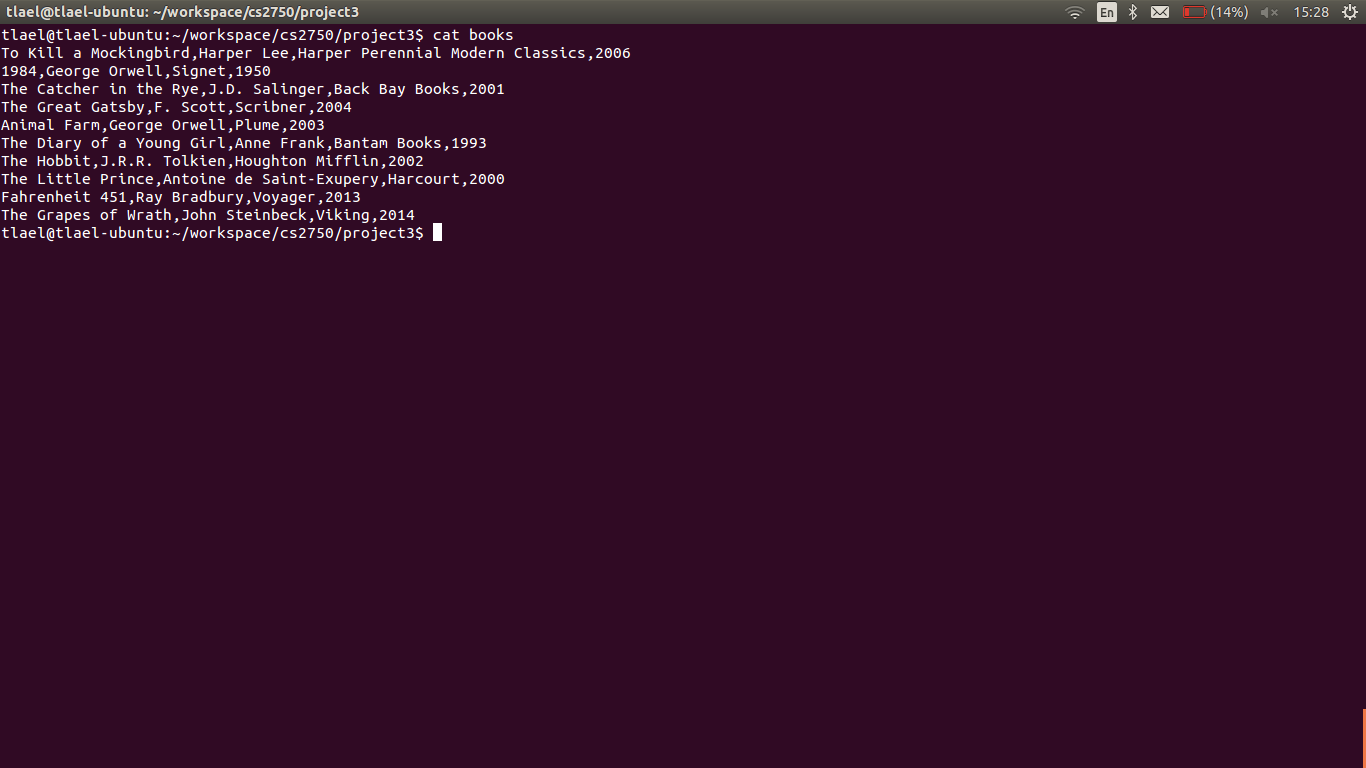
This script allows the user to enter books into a comma delimited file, search that file and print formatted output of either the search results, or the entire contents. If the script is run with no argument, the user is prompted to enter one book's information. The user can also pass an integer which represents the number of books that they wish to enter. These entries are stored in a file named books. If the user passes print as the argument, the entire list is printed to a file called book\_print in a column formatted layout. One can also search on specific keywords and those results will be printed to the same book\_print file. Every time a print or search/print is performed, the same file is appended with the new results and the column header is printed at the top of the latest results for clarity. The search can use regex and the -i flag to perform case insensitive searches.

### Screen captures:

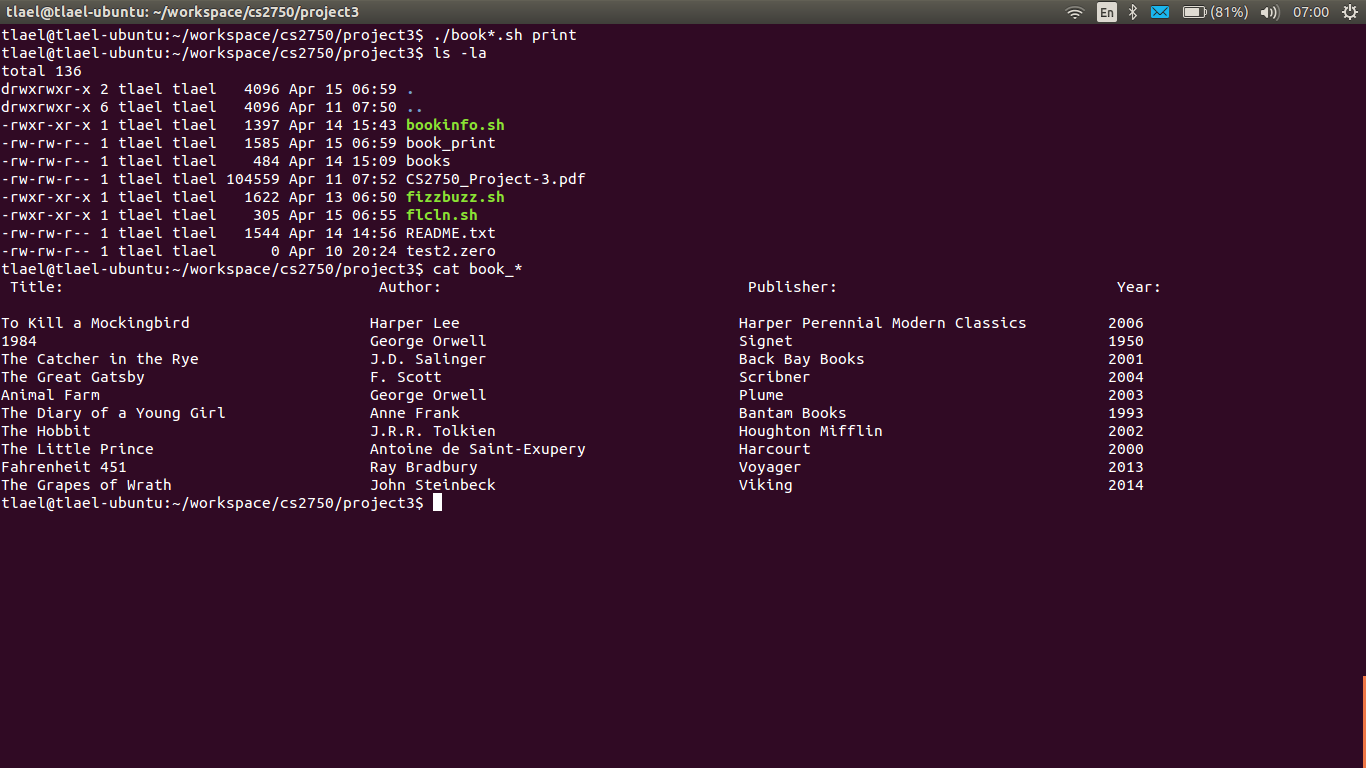
10 parameter passed to enter 10 books as requested



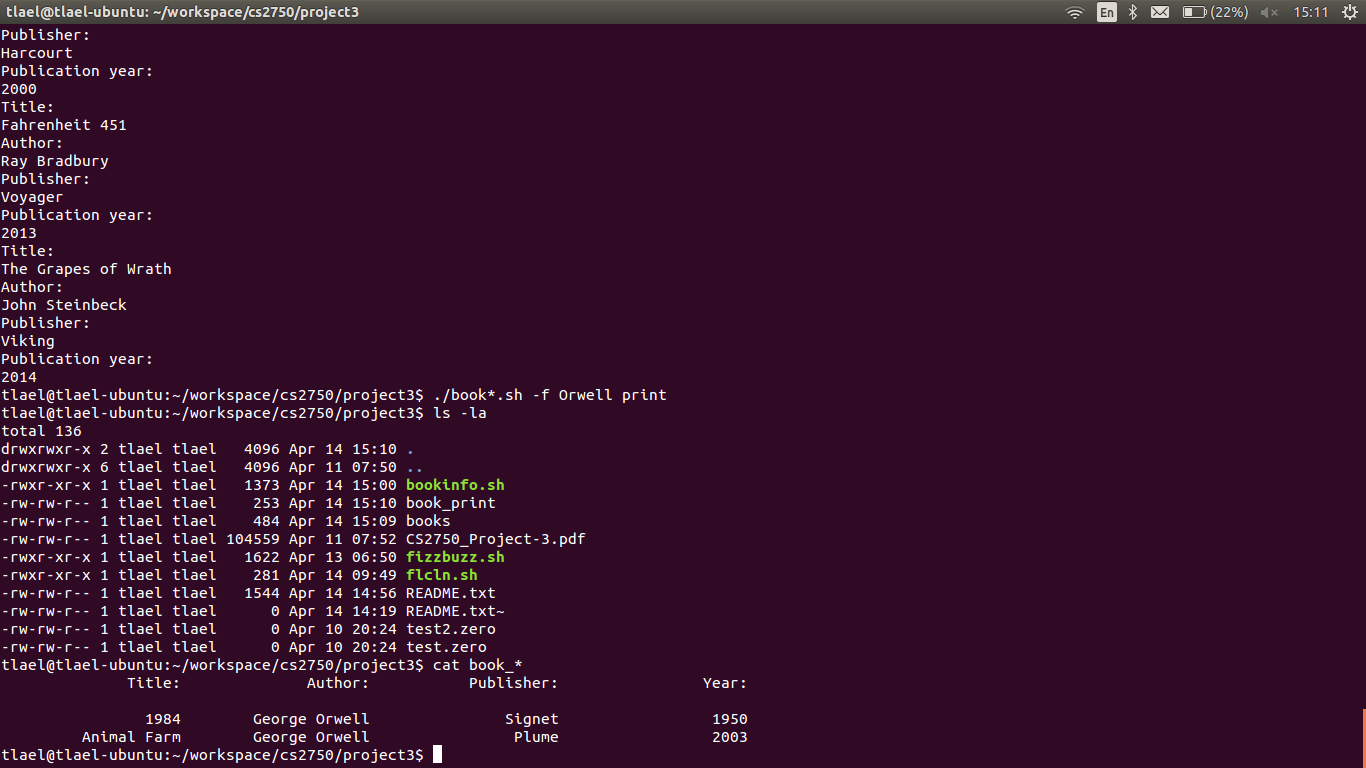
File contents after 10 books entered:



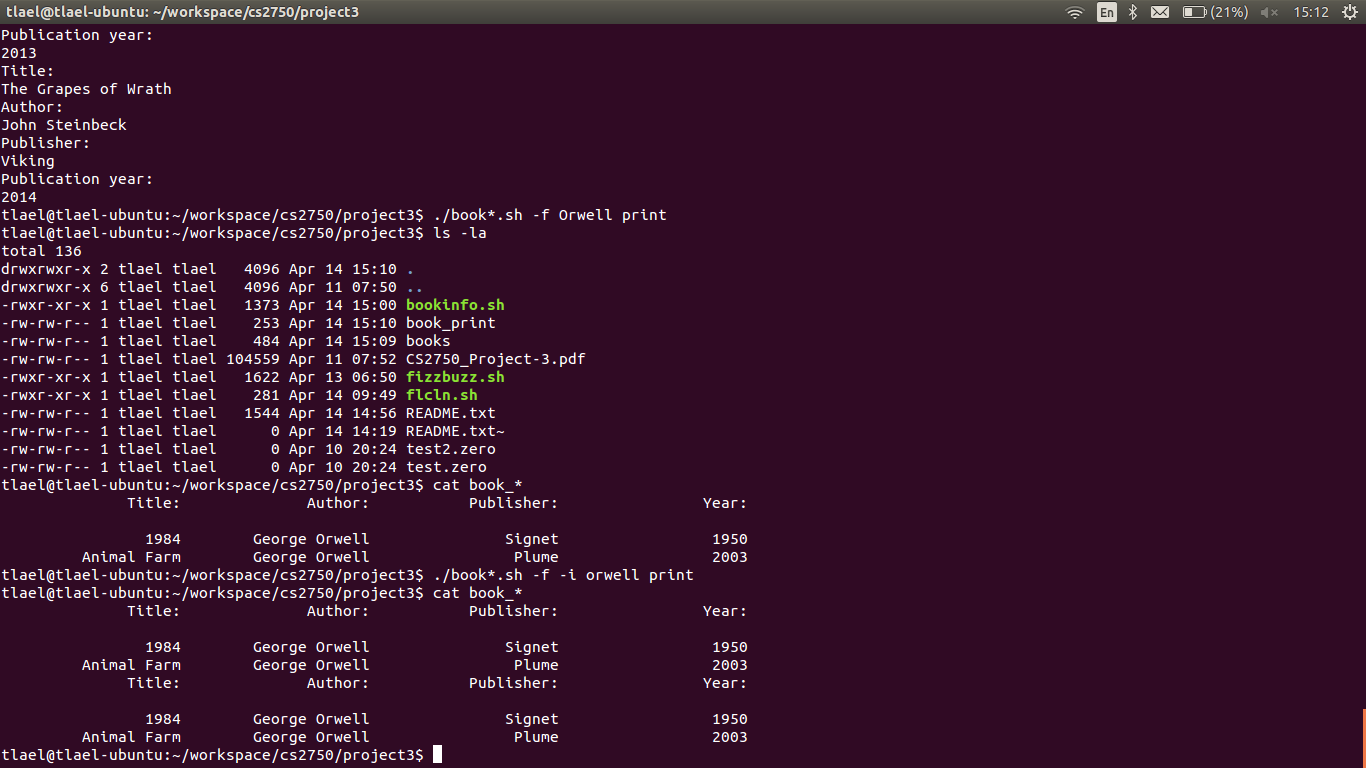
Print all:



Print/Search:



Print/Search -i:



# Conclusions

Both, fizzbuzz.sh and flcln.sh were simple scripts, but I did expand fizzbuzz.sh quite a bit to check the passed argument and to allow for negative numbers. Bookinfo was the most difficult task and required that I spend quite a bit of time searching for the best solution. I did this using man pages and various bash scripting books and online resources. The column formatted text was the most difficult aspect to do correctly in the shell, but I was able to use grep piped to awk to provide a result I was happy with. Similar to the third part of project 2, this definitely was a challenge and provided a great learning experience.