

VTSU CIS 2230 Linux System Admin
Lab #4 Assignment

Objective

You will demonstrate the use of piping and text processing. This is a very typical action for system administrators to perform.

Task 1: Unpack the Lab #5 Tarball

Below are some instructions for retrieving real Unix text files from some large servers at IBM. These servers have 24 CPUs, 500 GB of RAM, several TB of hard disk space, and hundreds of users running thousands of jobs.

A. Get the tarball:

Download the file `lab-04.tar` from the class website using the command:

```
$ wget https://www.pchapin.org/VTSU/cis-2230/lab-04.tar
```

Ensure the file is located in your home directory.

B. Extract files from tarball:

1. Get a list of the files in the tarball (`tar tf lab-04.tar`).
2. Extract the files in the tarball to a subdirectory in your home directory. I suggest `~/lab-04`. (Use the 'x' command on tar to extract files. See the man page for more information.)

C. Uncompress the files:

Uncompress the `*.txt` files only (leave the `log` file compressed). Use the `gunzip` command to “unzip” the compression. Now you should see three `*.txt` files. These files came from a busy IBM AIX Unix server. They are typical of what system administrators deal with. They came from these commands:

```
$ ps -ef > psef.txt
$ w > w.txt
$ who > who.txt
```

Here are some notes on these commands. You could also consult the manual pages for them.

- `ps -ef` → each line is a process
- `w` and `who` → each line is a “terminal” for each user.

Task 2: Log file examination

You have just come on duty for a 24/7 Unix Help Center, and there are concerns about the machine's performance. You have been asked to investigate.

- A. How many total terminals are being used by all the users right now? (Hint: How many lines are in `w.txt`?)
- B. How many total processes are running right now? (Hint: How many lines in `psef.txt`?)
- C. Without uncompressing the `saslog.log.gz` file, show the top 2 lines of this file. (Hint: If it is not provided with a file name, the `gunzip` program will read compressed data at its standard input and write decompressed results to standard output.

Task 3: Piping and text processing

Here are typical answers the Help Center will get. We'll answer with specific examples. All these can be answered with a single line command (with pipes). Show the command and the answer.

- A. How many terminals does the user "ruegsegs" have in `who.txt`? Note that the file is one terminal per line. (Hint: Use `grep` to search for lines that start with the text "ruegsegs." Count those lines.)
- B. How many terminals in `w.txt` are running the "sas82" program? OPTIONAL: Out of curiosity, what *percentage* is this of all the processes? (Hint: Count the number of processes mentioned in `psef.txt`. Doing floating-point math in `bash` is impossible since it doesn't support it. Instead, you have to invoke an external tool like `bc` or `awk`.)
- C. How many *unique* users are running on this box? (Recall that user IDs are in `who.txt`.) (Hint: I needed four commands piped together)
- D. Is someone taking too many resources? To find out, let us ask which user has the most terminals open? Show the users with more than four terminals. (Hint: Use `uniq -c` to count similar lines. The command `awk -e '$1 >= 4 { print $2 }'` might also be helpful.)
- E. Notice that `w.txt` has nice fixed-cols (pretty-print). Columns 53-60 are the number of CPU minutes. The longer the time, the more resources the process uses. Use `cut` to display just that column, then `sort` it. What's the longest-running process time?
- F. Now go back to `w.txt` and filter (`grep`) for that number. Which user and process does that belong to?
- G. I usually do the 2-step process of E and F above. *However*, it can be done in one step, where both the longest running time and the owner can be captured in a single command. Do that here.