

Toggle Dark Theme

- 1. [Introduction](#)
- 2. [Part 1: Shell spelunking](#)
- 3. [Part 2: General Questions](#)
- 4. [Part 3: Obtaining your VM](#)
 - 1. [Local Setups:](#)
 - 2. [Cloud Setups:](#)

Lab 1 - Unix, the Shell, OSS

Facilitator: [Ben Torres](#), [Trinity Chung](#)

6 min read

TABLE OF CONTENTS

- 1. [Introduction](#)
- 2. [Part 1: Shell spelunking](#)
- 3. [Part 2: General Questions](#)
- 4. [Part 3: Obtaining your VM](#)
 - a. [Local Setups:](#)
 - b. [Cloud Setups:](#)

Introduction

Welcome to the first lab!

All labs are graded on completeness and effort, so don't worry too much about getting an exact right answer. (We'll release staff solutions after the lab is due!)

Labs are also usually due a week from when they are assigned. Remember to ask for help if you need it on Edstem or Discord/Slack!

It may be convenient to submit your answers to Gradescope as you go.

Pro Tips:

- Here are some commands you might find helpful: `vim`, `ls`, `cd`, `man`, `file`, `grep`, `cat`, `less`, `wget`, `nano`, `tar`, `...`, (and other inferior text editors)
- Google and `man` are your friends!

Part 1: Shell spelunking

Everything should be done via the shell!

The purpose of this lab is to get you comfortable with using the shell for things you might typically use a GUI for. While these tasks may seem simplistic or limited, you'll quickly find that the commands have many different options (flags) to perform tasks that are either impossible or incredibly tedious / difficult to complete using traditional methods.

Don't worry about fully understanding how the commands work just yet- as long as you can gain a sense of familiarity with the tools at hand, we'll be in good shape to explore them further next week!

1. `ssh` into `tsunami.ocf.berkeley.edu` using your OCF account, or login at [ssh.ocf.berkeley.edu](#)
2. Run the following command to download the file we have provided: `wget https://github.com/0xcfdecal-labs/raw/master/b1/b01.tgz`

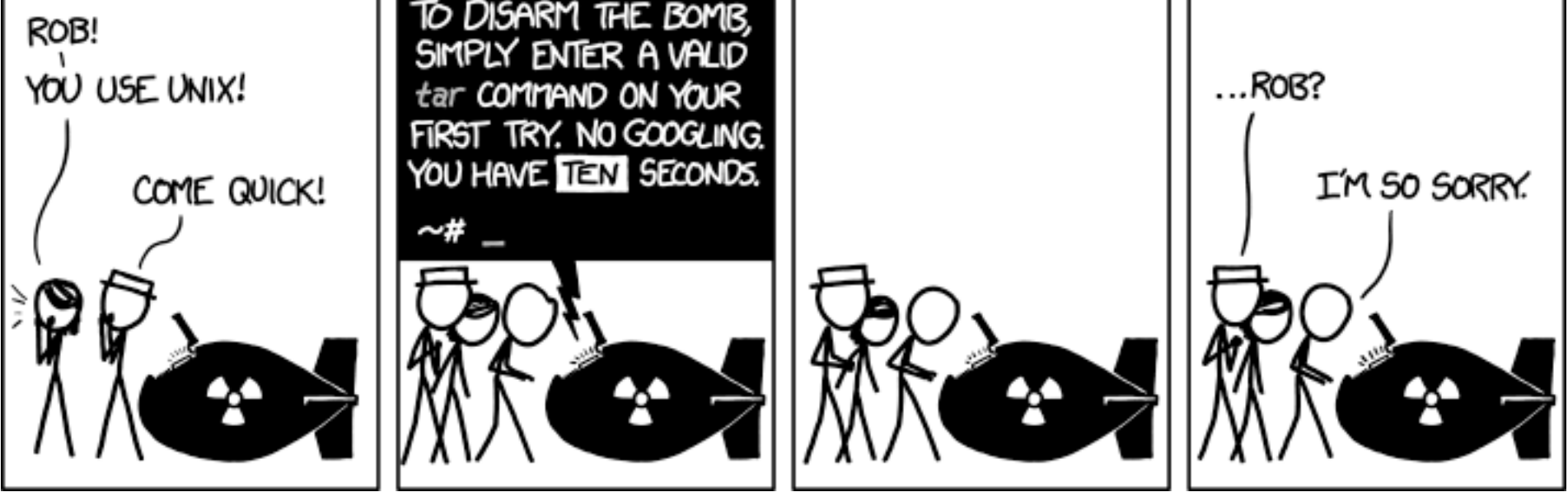
A `.tgz` file is actually a composition of two file formats. Sometimes you'll see these files as `.tar.gz` instead. A common (and old) way of archiving is with magnetic tapes. However, in order to archive the data, it needs to be a single file, and often you want to archive multiple files at once. This is where the `tar` command comes in (`tar` stands for tape archive). `Tar` will group (or ungroup) multiple files into a single one.

`tar`, unless you ask it to, doesn't compress files itself though. This is where either `gzip` (or `bzip2`) comes in. `gzip` will compress your file, and so, `tar + gzip` is often used in conjunction. It looks something like this: `file --(tar)--> file.tar --(gzip)--> file.tar.gz`.

If you read the `tar` documentation carefully enough, you'll see that you can give the command an option to compress your files using `gzip` as well, saving you a total of one line of shell command!

To unarchive the file we provide you, run the following command: `tar xvfz b01.tgz`. This will provide a `b01` directory for you with some files for the rest of this lab.

`tar` has a reputation for being a bit tricky with its options:



3. Go into the `b01` directory. Make sure you're in there by running `pwd` (Present working directory). **What does `pwd` give you (conceptually)?**
4. There's a hidden file in the `b01` directory. **What is the secret?**
5. A malicious user made its way into my computer and created a message split across all the files in `nonsense/`. What does it say? **How did you find the message?** (Hint: `ls` and/or `xargs` will be helpful here. If you want a challenge, try to do this in a single short command- but it's ok to find it by any means available.)
6. Go ahead and delete everything in `nonsense/` with one command. **How did you do it?**
7. There's a file in `b01` called `big_data.txt`. It's 80 megabytes worth of random text. For reference, Leo Tolstoy's "War and Peace", the novel with a whopping 57,287 words depicting the French invasion of Russia and the impact of the Napoleonic era on Tsarist society through the stories of five Russian aristocratic families with several chapters solely dedicated to philosophical prose, is only 3.2 megabytes large.

For that reason, I don't recommend using `cat` to print the file. You can try it, but you'll be sitting there for a while. There's some text you need to find in there! Go find it without actually opening up the file itself!

Two lines above the only URL in the file is a secret solution. **What is that solution?**

Hints: What makes up a URL (https...)? What is Context Line Control?

8. Try executing `./a_script`. You should get something back that says `permission denied: ./a_script`. This is because files have three different permissions: read, write, and execute. **Which one does `a_script` need?** Change the file permissions so that you can run the script. **How did you do it?**
9. Finally, there's an empty file called `hello_world` in the directory. Write your name in it! **How did you do it?**

Part 2: General Questions

Feel free to use Google and work in a terminal (where applicable) to verify your conjectures.

1. What differentiates Linux/OSX from operating systems like Windows?
2. What are some differences between the command line and normal (graphical) usage of an OS?
3. What is the root directory in Linux filesystems? Answer conceptually, as in depth as you would like,
4. `ls` has a lot of cool arguments. Try using them to get extra information such as file permissions, owner name, owner group, file size, and late date edited. In addition, I want to be able to see the size and have the files ordered by last date edited, with the oldest files on top. How would I do this?
5. Instead of showing the first 10 lines of the file `big_data.txt`, I want to use the `head` command to show the first 4. How would I do that?
6. What's the difference between `cat foo > out.txt` and `cat foo >> out.txt`?
7. Briefly, what is the difference between permissive and copyleft licenses?
8. Give an example of a permissive license.
9. Give an example of (a) open-source software and (b) free, but closed-source software that you use.

Part 3: Obtaining your VM

Using the public OCF login server will allow you to do basic things like the lab above, but if you want root permission (which lets you do basically whatever you want), you'll need your own use/destroy!

We will be providing a virtual machine (VM) to all DeCal students with a Berkeley CalNet account. Check your `@berkeley.edu` email!

If you are not a Berkeley student, then you will have to obtain your own machine. Of course, you can also set up your own VM if you are just curious. Generally, there are two ways to obtain a VM: use your own computer (local) or use someone else's computer (cloud).

Local Setups:

- [Virtual Box](#)
- [UTM](#) - Only for MacOS

Cloud Setups:

These are usually paid services, but if you have a student email, they provide more than enough free credits for the purposes of this course.

- [Digital Ocean](#)
- [GCP](#)
- [Azure](#)



Thanks to [Nvidia](#) for sponsoring GPUs to the OCF.



Huge thanks to [Linode](#) for sponsoring the equipment used to record digital lectures for the Decal



Hosted by the OCF

Copyright © 2017-2022 [Open Computing Facility](#) and [eXperimental Computing Facility](#)