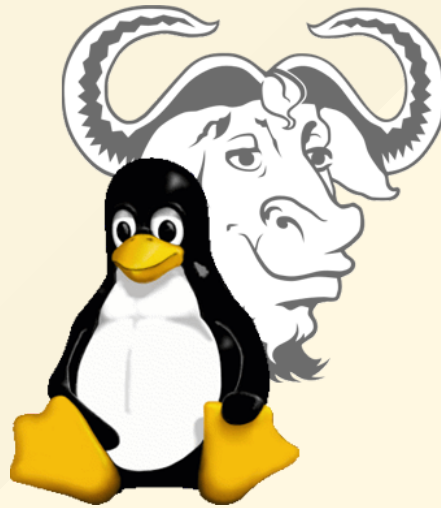
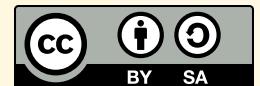


grok GNU/Linux



Task-based learning materials by Sean O'Brien

sean@webio.me | sean.obrien@yale.edu



CLI Basics & System Administration

These materials were written for a [Debian GNU/Linux](#) system or its popular derivative [Ubuntu](#) but should be broadly applicable to other [distros](#).

It's best to have an environment you don't mind messing up, such as a [Live USB](#), a [virtual machine](#), or some old hardware you can wipe easily.

The goal is to finish the tasks without having to consult the Web or exit the [command line](#) (CLI).

Before You Begin

- Grab the files from the [git repository](#).
- The tasks require superuser access ([sudo](#) or root).
- Try to go through the tasks in order, without skipping around. They build upon each other.
- The first time you go through, feel free to look things up. However, you need to [learn by doing](#).
- Your username is "GROK" in the examples.
- The tasks use files in *sandbox* or *sandbox.tar.gz*. Start the tasks inside the *sandbox* dir, which is best placed in your [home](#) ~/ or /home/GROK

Tips & Tricks

- Make sure you know how to interrupt a command in your terminal emulator (usually CTRL+C).
- Use the up arrow to pull up previous commands from your command history and TAB to autocomplete command, file, and dir names.
- Need help? Try *--help* after the command in question or *man* before it. Learn to [RTFM](#).
- Keyboard shortcuts like paste (SHIFT+INSERT) are different in the terminal emulator. Learn them, but don't become Pete the Repeat Parrot.

“ I think the major good idea in Unix was its clean and simple interface: open, close, read, and write.

— [Ken Thompson](#)

”



“ ...the Linux philosophy is 'laugh in the face of danger'.
Oops. Wrong one. 'Do it yourself'. That's it.

— [Linus Torvalds](#)

”

Ready, Steady, Go!

- Clear your terminal emulator's command history.

```
$ history -c
```

- Read your OS version and write it to a file named *distro-version* in the directory *current-distro*.

```
$ mkdir current-distro  
$ cat /etc/lsb-release > current-distro/distro-version  
  <!-- or, alternatively: -->  
$ lsb_release -d > current-distro/distro-version
```

- Change into *current-distro*. What's in it?

```
$ cd current-distro  
$ ls
```

- Create a hidden file *.hidden* in *current-distro*.
Write a detailed list of *current-distro* to *.hidden*

```
$ touch .hidden  
$ ls -lah > .hidden
```

- View a detailed list of the filesystem root.

```
$ ls -lah /
```

- Append a list of your home directory to *.hidden*

```
$ ls -lah ~/ >> .hidden  
<!-- or, alternatively: -->  
$ ls -lah /home/GROK >> .hidden
```


- Create a user named *rms*. Give *rms* the password *FreeAsInFreedom*.

```
$ useradd -D  
$ sudo useradd rms  
$ sudo passwd rms
```

- Create a user named *linus* with no home directory and the password *MonolithicKernelOrDie*.

```
$ sudo useradd -M linus  
$ sudo passwd linus
```

- Create the group *sflc*. Create the directory */home/sflc*. Give the group *sflc* read/write permissions on */home/sflc*.

```
$ sudo groupadd sflc
$ sudo mkdir /home/sflc
$ chown -R :sflc /home/sflc
$ chmod -R g+rw /home/sflc
```

- Create the file */home/sflc/README*. Give the owner read/write/execute permissions and both the group and public read/execute permissions.

```
$ touch /home/sflc/README
$ chmod 755 /home/sflc/README
```

- Create a user named *eben* with home directory */home/sflc*. A group named *eben* should not be created along with the user. The user *eben* should expire on New Year's Eve, 2021.

```
$ sudo useradd -e 2021-12-31 -N -d /home/sflc eben
```

- Add *eben* to the group *sflc*. Give *eben* the password *GPLorGTFO*.

```
$ sudo usermod -a -G sflc eben  
$ sudo passwd eben
```

- Set inactive user accounts to expire in 30 days.

```
$ sudo nano /etc/default/useradd  
<!-- or, alternatively: -->  
$ sudo vi /etc/default/useradd  
<!-- set value -->  
INACTIVE=30
```

- Log in as *linus*, say something, then log out.

```
$ su linus  
$ echo "I like kernels"  
$ exit
```

- Disable the account *linus*.

```
$ sudo passwd -l linus
```

- Enable the *root* account. Set the *root* password to *DangerZone*.

```
$ sudo passwd -l root  
$ sudo passwd root
```

- Log in as *root*, say something, then log out.

```
$ su root  
<!-- or, alternatively: -->  
$ sudo -i  
$ echo "wow I just pwned the system"  
$ exit
```

- Disable the *root* account.

```
$ sudo passwd -l root
```

- Send a message to the user named *rms*

```
$ write rms
```

- Send a message to all users on the system that contains the current operating system version.

```
$ wall < current_distro/distro-version
```

- View the command history, then save it to the file *recent_commands*.

```
$ history  
$ history > recent_commands
```

Warning!



Potential Trouble Ahead

These tasks alter fundamental parts of the system, require the machine to shut down or restart, or could change the system in ways that might make it inoperable or otherwise annoy you.

However, it's really important stuff you should know how to do.

- Shut down the system

```
$ sudo poweroff  
<!-- or, alternatively: -->  
$ sudo shutdown -h now
```

- Shut down the system in 10 mins and send the users a warning message.

```
$ sudo shutdown -h +10 "Giving HAL 10 more mins to live."
```


- Reboot the system.

```
$ sudo reboot  
<!-- or, alternatively: -->  
$ sudo shutdown -r
```

- Reboot in five minutes and warn the users. Then decide to cancel shutdown.

```
$ sudo shutdown -r +5 "Ship is going down in 5, pls save."  
$ sudo shutdown -c "Nev mind, sorry for the scare."
```

- Reset the system forcibly.

```
$ sudo reboot -f
```