Introduction to Unix and GNU / Linux Training lab book

Michael Opdenacker
Free Electrons
http://free-electrons.com

About this document

This document is part of an embedded Linux training from Free Electrons.

You will find the whole training materials (slides and lab book)
on http://free-electrons.com/training/intro_unix_linux.

Lab data can be found on http://free-electrons.com/labs/embedded_linux.tar.bz2.

Copying this document

Document updates and translations available on http://free-electrons.com/training/intro_unix_linux

Corrections, suggestions, contributions and translations are welcome!

Training setup

Instructions for the instructor, or for self training people. Otherwise, you can proceed to the next page!

The training session can be performed on any GNU / Linux installation, in particular on a live cdrom like Knoppix.

Download the lab files directory from http://free-electrons.com/labs/embedded_linux.tar.bz2 and extract the archives in a convenient directory. In the labs, we will assume that this directory is `/mnt/labs`.

Disclaimer

Do not take the contents of the training lab files for granted!

Sardars are very nice, friendly and even clever people originating from India. I know some of them in person, and I can certify this! This is probably why other people in India are not afraid of telling kind jokes about them.

Microsoft is a perfectly honorable and trustworthy corporation, which only pursues the best interest of individuals and society as a whole, in a truly altruistic way.

Lab 1 – Basic file handling

Objective: getting familiar with basic file handling.

At the end of this lab, you will be able to

- Display the list of files in a directory, and see their contents.
- Perform simple operations: copying, renaming, removing, creating links.
- Modify access rights to files and directories
- Et even get rid of Microsoft!

Setup

Go to the `/mnt/labs/intro/lab1` directory.

List of files

Display the list of files. How many files and directories are there?

Which is the hidden file?

List the oldest files first, and the most recent last.

Now, list the smallest files first, and the biggest last.

You will find the option that you need in your course, or by consulting the `ls` command manual page.

Accessing file contents

Display the contents of the `answering-machine.txt` file at once.

Now, display them with a tool which stops at the end of each page and waits until you hit a key, to leave you time to read.

Enter your command without typing the whole file name. Just type the beginning of it, and hit the Tab key.

After reading a few pages, directly go to the part of the text containing the `planet` word. If you use one of the 2 most popular tools (suggested in the lecture), you can do this by using the "`/`" command (inside the tool). Then, you can go to the next occurrence of this word with the "`n`" command.

Note that you can find the same commands in the `vi` editor too. Though standard Unix commands are independent, there is some consistency between them!

Once you reach the end of the file, look for the last occurrence of the "`BEEP`" word, then continue upwards to the previous `BEEP` words, in the same way.

Note that only the best of the 2 tools highlights the search word in reverse video. If you don't observe this, it means you are not using the best one!

Display the last 20 lines of the `sardar3.txt` file.

Searching through file contents

Look for `trust` in `Microsoft`.

Look for `money` in all the files in the directory (included in the subdirectory).

Do it by editing the previous command.

Without typing again the full command, now look for `Money`.

Again without typing again the command, look for the same word, whatever its case.

Making changes on file and directory names

Modify the name of the `.lightbulb` file so that it is no longer hidden.

Get into the `sardar/` directory. Check that you are in the right

directory. Move the `sardar3.txt` file from the parent directory to the current directory.

Go back to the parent directory.

Get rid of `Microsoft` once and for all.

Create an `archives` directory and copy all the files in the working directory into it, including the `sardar` subdirectory and all the files it contains.

To check the name of the current directory, the `pwd` ("print working directory") command is your friend.

Symbolic links

Create symbolic links making the files in the `sardar/` directory appear in the current one too.

Once more, list the files in the current directory. Are links easy to identify?

Remove the `sardar/sardar3.txt` file and see the impact on the file list.

File access rights

Try to suppress the `sardar/sardar1.txt` file.

Display the access rights of the various files and try to understand why you are not allowed to do it.

Once you understood, change these rights and remove this file.

Now, try to get into the `safe` directory. Modify access rights to be able to do it.

Once you're inside, your adventure is not over yet. As you are only interested in `fortune`, remove the `-o` which is also there. Good luck!

As these practical labs do not involve several users at the same time, it is not really possible to propose elaborate exercises on file access rights. You will have your own practice on real-life GNU/Linux servers!

Lab 2 – Elaborate commands

Objective: get familiar with redirections, pipes and task control

At the end of these practical labs, you will be able to

- Redirect the output of a command to a file
- Implement pretty complex requests by cascading multiple Unix commands.

Setup

Go to the `/mnt/labs/intro/lab2` directory.

A first redirection

Use the `history` command to show all the commands that you already typed.

It is sometimes useful to keep track of typed commands. The `history` command is definitely useful in this case.

Now, save the output of this command in a new `history.txt` file.

Concatenating files

Concatenate all the files in the `sardar/` directory into the `sardar_power.txt` file, still without leaving the current directory.

How many lines, words and characters are there in this new file?

Display all the lines in this file containing the `singh` keyword (case insensitive)

Remove the `sardar_power.txt` file.

Using pipes

In only one command line, display again all the lines in the files in the `sardar/` directory which contain the `singh` keyword (case insensitive).

We will get the same results, but without using an intermediate file. This shows the power of pipes!

Now, count the number of lines this represents, still in a single command line.

Modify this command to only count the lines containing both `santa` and `singh`, still in a case insensitive way.

Improve once more the command to count only lines containing `santa` and `singh`, but not `banta`.

You have just discovered all the power of Unix pipes! These pipes let you do exactly what you need to, from very simple basic commands.

That's all for the moment! Go on practicing commands introduced in the lectures, on files available your GNU / Linux system.

Lab 3 – Text editors

Objective: get more familiar with text editors

At the end of these practical labs, you will be able to

- Move a whole block of text, creating or removing a margin easily.
- Edit text in a text only console, through the `vi` editor.

Setup

Go to the `/mnt/labs/intro/lab3` directory.

nedit features at a glance

`nedit` is a user friendly yet powerful text editor.

First open the `msreligion.txt` file with `nedit`.

Review the contents of the various menus, and don't hesitate to try the corresponding options and commands.

For example, toggle `Incremental Search Line` in the `Preferences` menu. Use the new dialog to search for all the occurrences of the `Microsoft` word (whatever its case). Easy, isn't it?

Use the `Return` key to go from one occurrence to the next.

Now try `Windows -> Split Window` (no hidden meaning this time). You can work on 2 parts of the text at the same time. Get back to the single pane mode.

Convert the whole first paragraph into uppercase.

Now, a very nice feature: holding down the `[Ctrl]` key, select the contents of the first paragraph, but not the spaces on the left and right hand sides.

Now press the middle mouse button (or both buttons if you don't have any), and move your mouse! You see you can easily remove the space margin on the left of the selection. You can even move the selection over another part of the text and cover it!

Unlike traditional Unix text editors, `nedit` implements most of the now universal keyboard shortcuts:
`[Ctrl][c]`: Copy
`[Ctrl][v]`: Paste
`[Ctrl][x]`: Cut
`[Ctrl][z]`: Undo
`[Ctrl][a]`: Select all

Another way of achieving the same thing: in the same way, select the space box on the left of the 5 next paragraphs. Hit `[Ctrl][x]`, and the empty space is gone!

Editing text with `vi`

First, make sure you have the `vi` command summary page from the training slides in front of you!

Open the declarations.h file with `vi`.

Moving the cursor

Note that you can use the arrow keys to move the cursor in the text. Now, do the same thing without using the arrow keys.

Similarly, move one page forward and backwards without using the `Page Up` and `Page Down` keys.

It takes time to get familiar with this way of moving the cursor. However, as you don't have to move your fingers away from the center of the keyboard, it is a very efficient way!

Inserting text

Go to the first line of the text.

Enter insertion mode and add `/*` at the beginning of the line. Exit

insertion mode when you are done.

Go to the end of the line by using a single command (and not by moving the cursor character by character!)

You can now see that there are 2 modes in `vi`: insertion mode and command mode.

Insert a space at the end of the line, followed by `*/`. Exit insertion mode again. Just as before with the end of the line, go back to the first character of the line by using a single command.

Insert the below line before the first line:

```
long horn;
```

Exit insertion mode and place the cursor anywhere on the second line. Start editing a new line right after the current line. Invent your own funny C declaration!

Save your changes and exit `vi`.

Finding words

Open the `pastacode.txt` file with `vi`.

Use a command to go to the first occurrence of the `code` word.

Unlike traditional `vi`, `vim` (the `vi` implementation you are using), `vim` highlights all the instances of the search words.

Go to the next occurrence by using the `n` command. Once you reach the end of the file, make a search backwards for the `software` word. Similarly, go to one occurrence to the next.

Replacing words

Look for a particular word using the search commands. Modify the entire word by using a single command.

This "`.`" command is definitely one of the most useful commands in vi!

Now, modify the next occurrences of this word by using the "`.`", command, which repeats the latest edition.

Also remove entire words, or just a given character.

Find how to replace a single character by several characters.

Find how to delete one line in a single command. Find how to delete 10 lines, still in a single command.

Now, how would you replace the 5 next characters starting from the cursor?

That's all for today! You can continue to try more `vi` features on your own from your command summary sheet. The instructor can also show you more.

