Server System Management - Linux

Lab 1 - recap

# 

# Lab 01 : Recap

Kiss that VMWARE console goodbye and re-familiarize yourself with some basic stuff

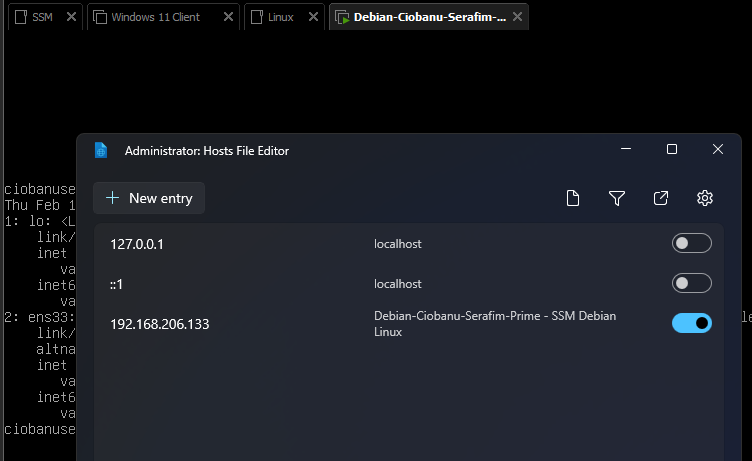
1. Working in the VMware console blows. Resolution issues, copy / paste oddities. It’s much better to connect to your machine using the SSH server you installed while setting up your VM. Lets make it easy on ourselves from now on.

Look at the screenshot you took after finishing your setup. It shows your Prime’s IP ADDRESS. Now, edit “a” special file (remember OSC ?) on your main host machine and add an alias for that ip, called **Debian-YourFamilyNameWithoutSpaces-YourFirstNameWithoutSpaces-Prime**. (tip : you may have to use an elevated editor to modify this special file)

What is the name of that “special” file ? (QUIZ)

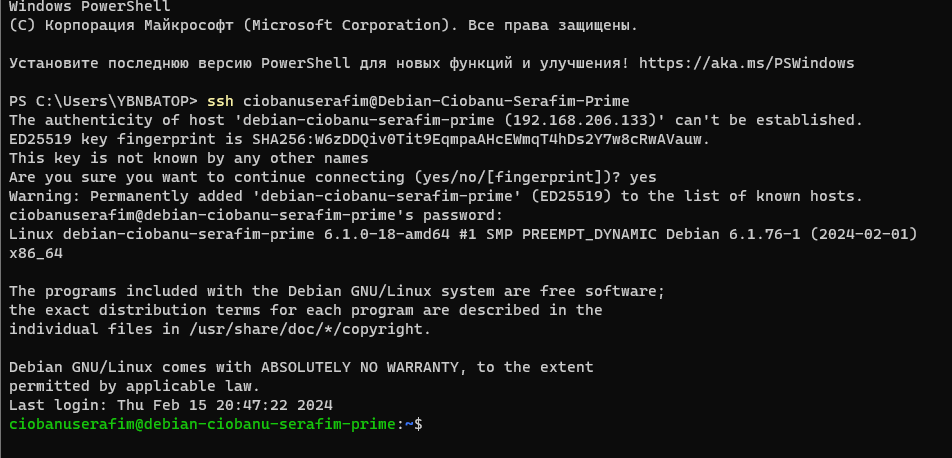
In Linux - /etc/hosts (love it)

In Windows – C:\Windows\System32\drivers\etc\hosts (you need a administrator Notepad to edit it, or just use PowerToys)



1. Now let’s try this out ! Open a new console on your main host machine and type ssh yourfamilynameyourfirstname@Debian-YourFamilyNameWithoutSpaces-YourFirstNameWithoutSpaces-Prime.

Accept the fingerprint it shows you and connect. For the rest of the labs, always connect using ssh this way. Only when specifically asked for (or when ssh won’t budge for some reason, fall back to the vmware console)



Your prompts looks something like this :



What does the $ stand for ? (QUIZ)

It means that we are logged in as a simple user, and not as a super privileged user, i.e. root. For the root it would be #. Apart from that, it is just the end of a prompt.

1. What command do you use to see your current username ? (QUIZ)

whoami – print effefctive user name



1. How do you get more information about the use of ANY command ? Apply that to the ls command to find out how to **never** apply color to files located in the root folder (QUIZ)

**man** gives you more information about the command. You can also use **—help sometimes**  because there might be no manual package for the package itself.

To never apply color you will probably use the **—color=never** option

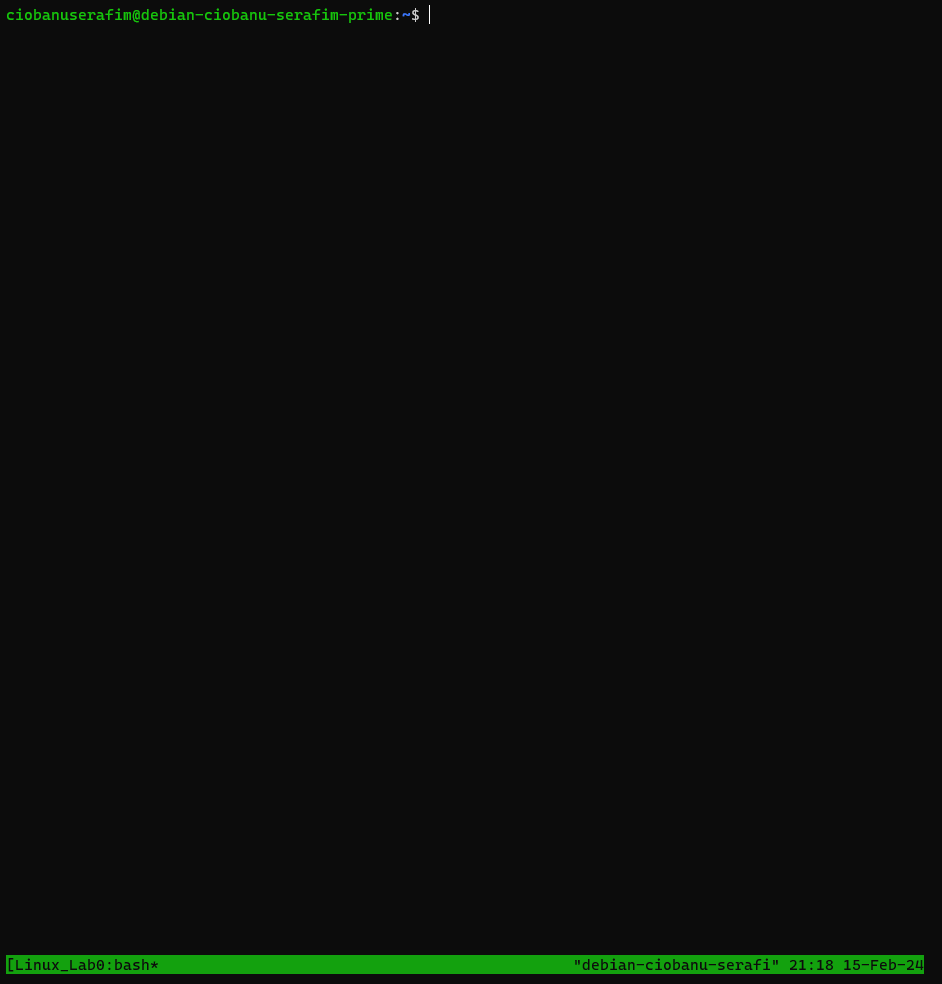
1. You’ll need to become root for the next part. Use the complete *su –login* statement so you’ll become root in a true root login shell

Once you are root, type in *apt install tmux*. Once it is installed, leave the root shell by typing exit. Make sure you have a maximized (or at least pretty large) window for your ssh session, because we’ll be splitting it up next.

TMUX (terminal multiplexer) allows you to keep any session(s) available even if you disconnect from the server (as long as your server isn’t shutdown or you explicitedly terminate a session. Use the helpful command we mentioned above to find out how to (QUIZ)

1. Start a named session called Linux\_Lab\_01 from the terminal

tmux new –s Linux\_Lab\_01

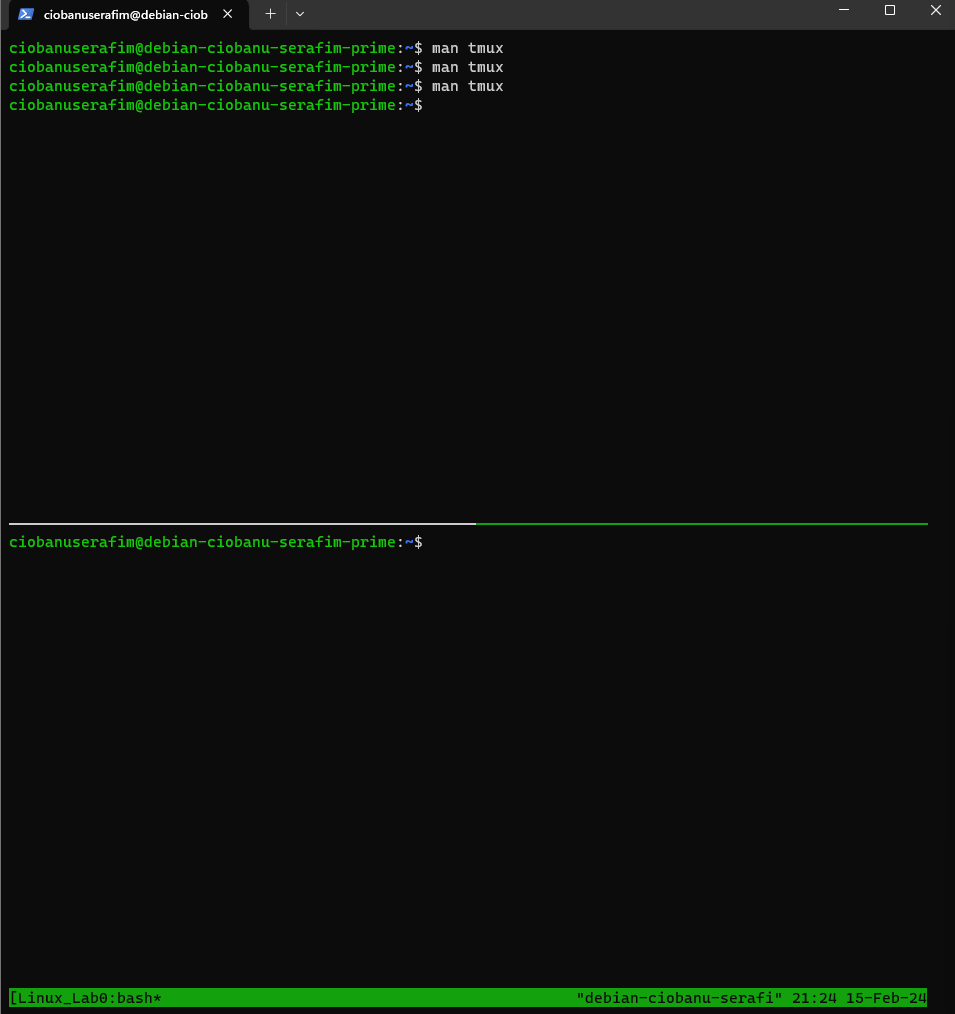


1. Split your tmux session window horizontally (=create a new pane) while in a tmux session

This is absolutely horrible

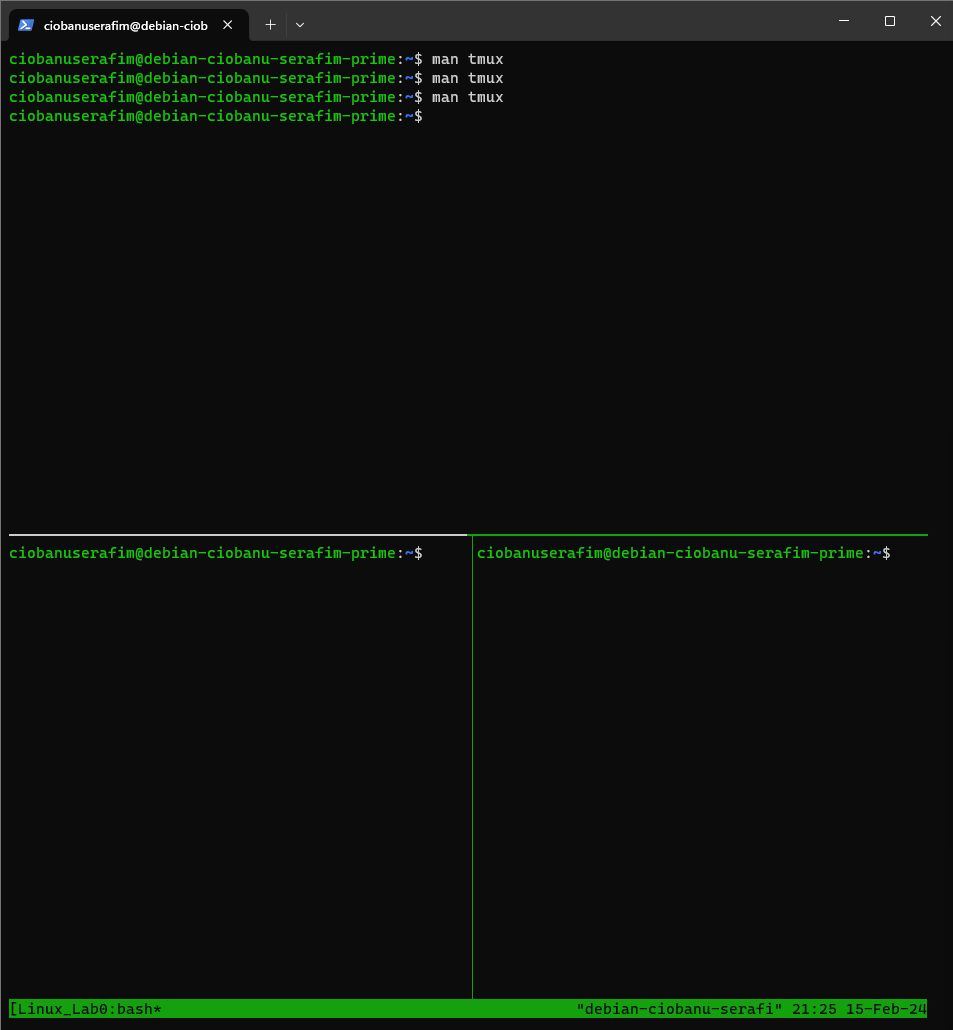
CTRL + b and then “ – split horizontally (top and bottom)

CTRL + b amd then x – Kill the focused pane



1. Split your tmux session window vertically (=create a new pane) while in a tmux session

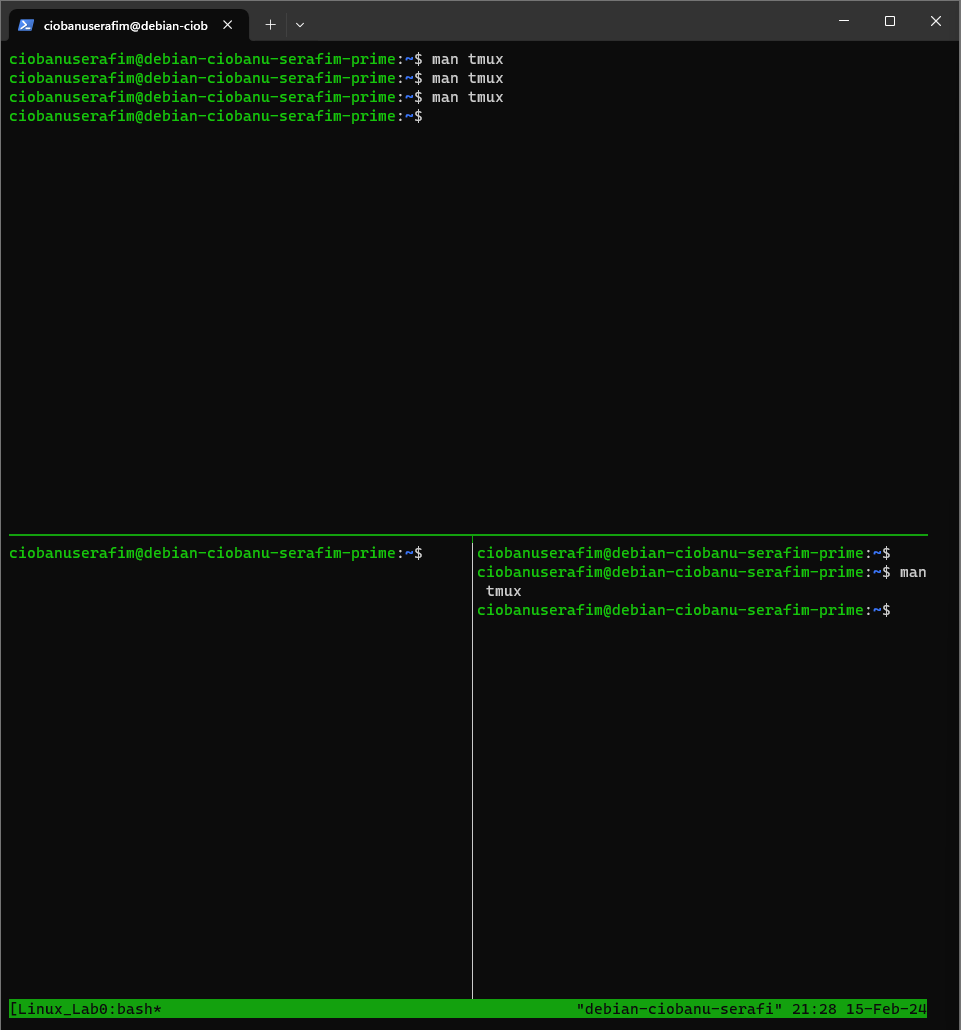
CTRL + b and % – split vertically (left and right)



1. Switch focus between panes while in a tmux session.

Horrible key binds, like Emacs

CTRL + b and arrowkey (left, right, up, down)



1. Detach from your named session Linux\_Lab\_01 while in a tmux session

CTRL + b and d (detaches from session, but it is still alive)

**tmux attach** will attach back to it

1. Re-attach to your existing named session Linux\_Lab\_01 from the terminal

**tmux attach**

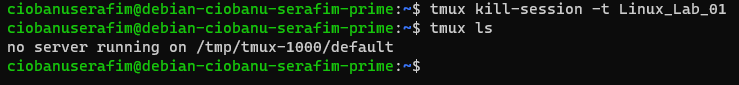
You can run **tmux ls** and see the sessions, and then you can do tmux attach –d –t <session\_name> (stack overflow also knows something useful)

1. List all tmux sessions in the terminal

**tmux ls**

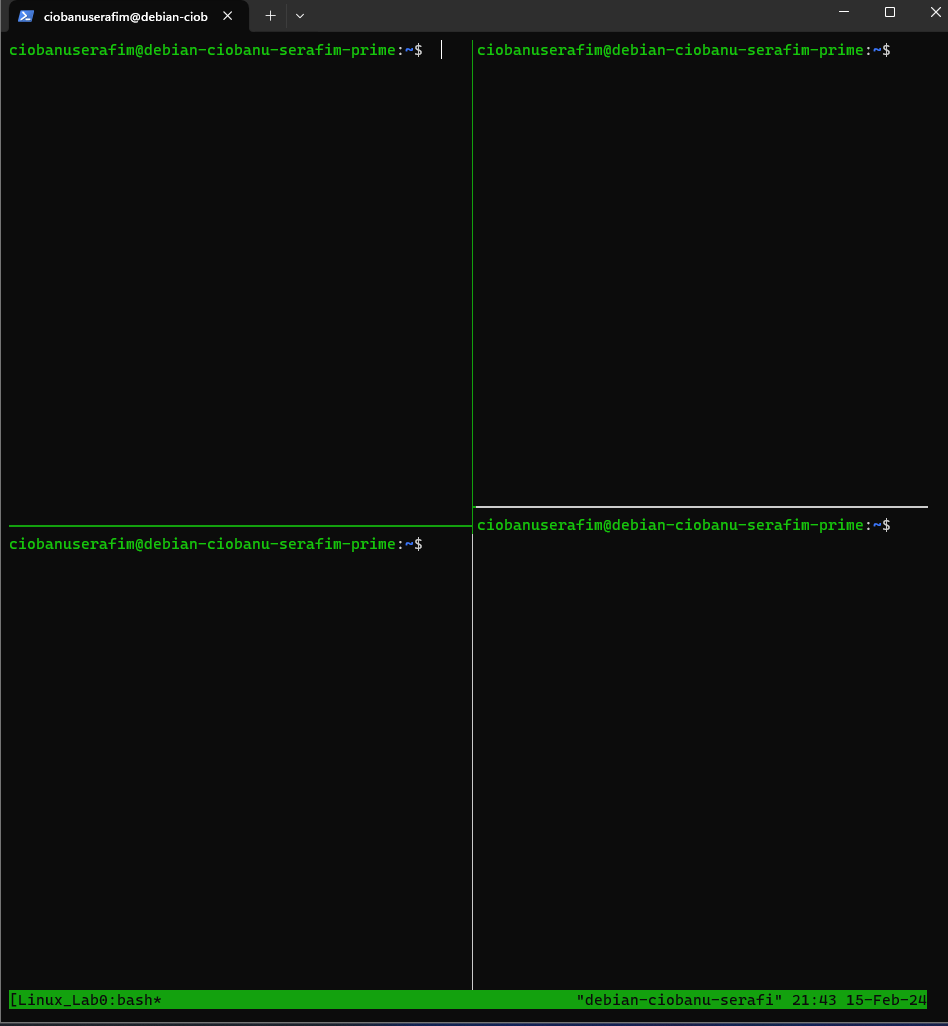
1. Destroy the named session Linux\_Lab\_01 from the terminal (don’t actually execute this, keep the named session alive till the end of the lab)

**tmux kill-session –t Linux\_Lab\_01**



I created a new one, **tmux new-session –s Linux\_Lab\_01**

1. You should now have 4 terminals running in the same tmux session Linux\_Lab\_01 session. Make sure to connect to the tmux session if you hadn’t in the previous question. You can now continue the lab by using any and all panes your created before.



1. More or less ? use one (and the pipe | symbol) to paginate up and down through a long list of all files in the /etc folder

ls /etc | more

ls /etc | less

1. Get a long list of all files recursive from the root folder. Pause it. Resume it. Stop it. Bring it back to the foreground. Stop it again. Destroy it without bringing it to the foreground.

ls –lR /

CTRL + Z – Pauses it

fg (command) – Resumes it

CTRL + C – Stops it

fg (command) – to get it back to foreground

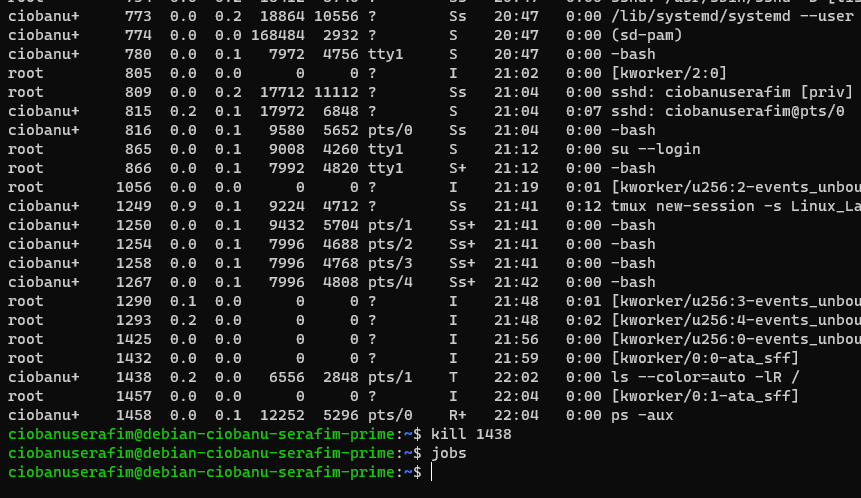
CTRL + C – Stops it again

jobs (command) – will show current running jobs

ps –aux – shows the processes with pid’s

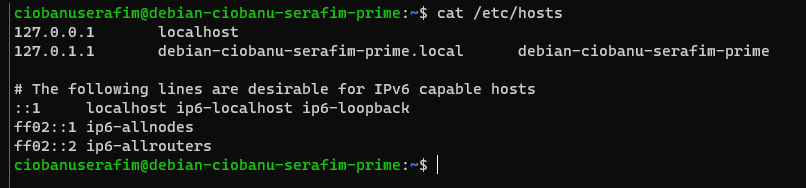
kill <pid> – will kill that process

some googling also showed the command **kill %1, that will kill the last job?**



1. Give a command that shows the content of the /etc/hosts file (QUIZ)

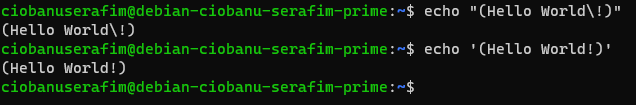
cat /etc/hosts



1. Use the echo command to show this on the session input window (including brackets) :

(Hello World!)

echo ‘(Hello World!)’

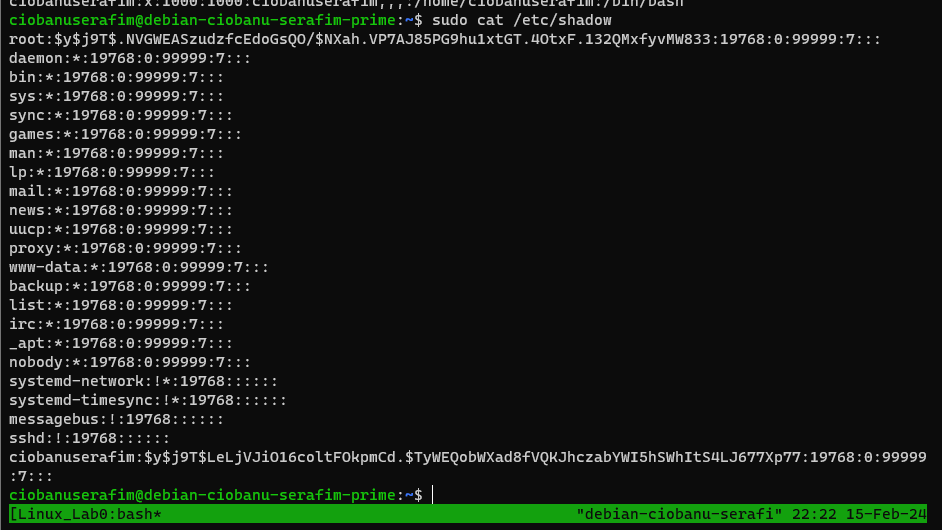


1. Will sudo (remember OSC?) work out of the box in a vanilla debian setup like we just did ?

No, it will not. You will need to **su —login, apt install sudo, usermod –aG sudo <user>, exit**, and it should suffice. If not – reboot, or relog.

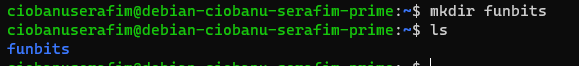
1. Show the content of the file where the hashed passwords are stored.

sudo cat /etc/shadow



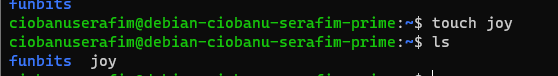
1. Create a folder called funbits in your normal user’s homedir

mkdir funbits



1. Create an empty file called joy in your home dir without using an editor

touch joy



1. Use vi to edit the content of joy. Enter these lines :

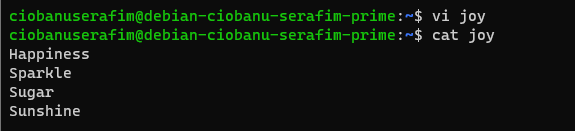
Or just install Vim

Happiness

Sparkle

Sugar

Sunshine



1. Was the vi you used the REAL vi ? what was the real underlying command ? where did it reside ?

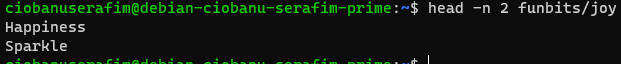
Most of the binaries are in **/usr/bin, hence if you try /usr/bin/vi joy – it will also work!**

1. Move the joy to funbits

mv joy funbits/

1. Show the first two lines of funbits/joy

head -n 2 funbits/joy



1. Show the last line of funbits/joy

tail -n 1 funbits/joy

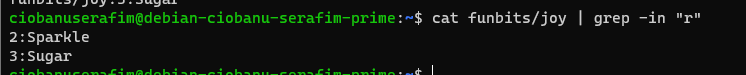


1. Show all lines in funbits/joy containing the letter “r” and their line number

cat funbits/joy | grep -in "r"

grep –in “r” will apparently also work (I am more used to piping)

grrp –n “r” is the correct option for the quiz



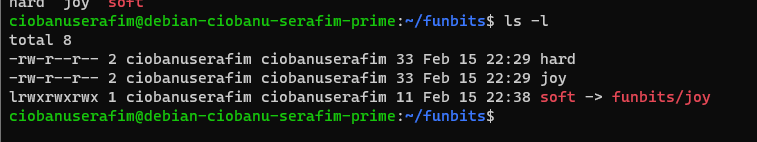
1. Create a hard link called funbits/hard and a soft link called funbits/soft to funbits/joy

What’s difference between the two ?

ln -s funbits/joy funbits/soft

ln funbits/joy funbits/hard

Hard link is like a twin of the original file, while the soft link is more like an alias or link to the file itself, or maybe even a directory.



1. Show the file permission (and only that!) in rwx format of funbits/joy

stat –c %A funbits/joy



1. What is 664 in rwx format ?

If I understand the question correctly, it will be this:

**rw-rw-r—**

1. Change funbits/joy so that only root will be able to read it (you may need several steps)

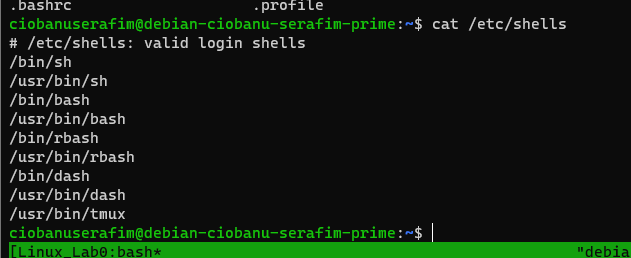
sudo chown root funbits/hoy

sudo chmod 400 funbits/joy



1. Use a statement to find out which shells debian supports out of the box

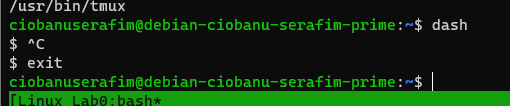
cat /etc/shells



1. Start a dash shell. How do you go back to your previous shell ?

dash – to start a dash shell

exit – to exit the **shell**



1. How can you make variables be persistent in subshells ?

export them

for example **export VAR=”funbits”** should work

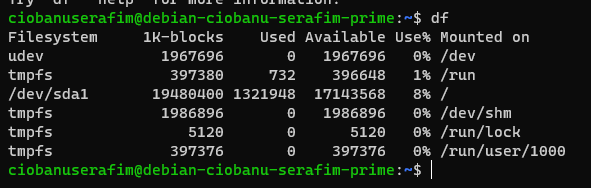
This should also work likewise the paths in .bashrc

1. What is the command to find out much free space (in K) you have left on your partitions ?

df – simple command should suffice?

another option seems to be:

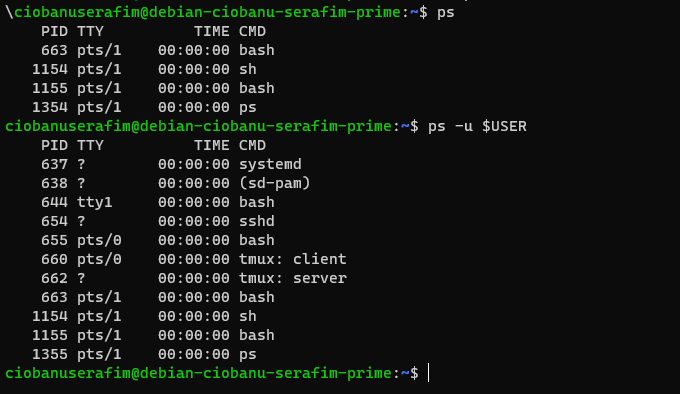
df | awk ‘{print $4}’ (will print only the 4th column



1. Use a statement to find out what processes the current user has started

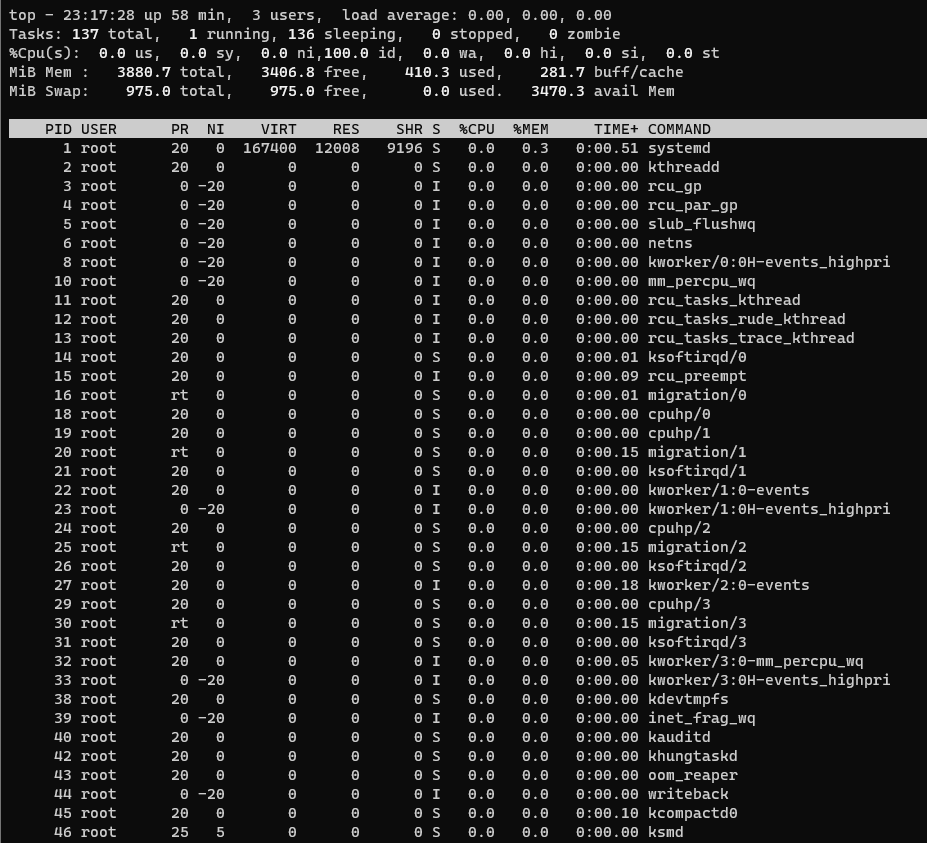
ps – this looks like it should suffice

but another option I found is **ps –u $USER** (for the currently logged in user)

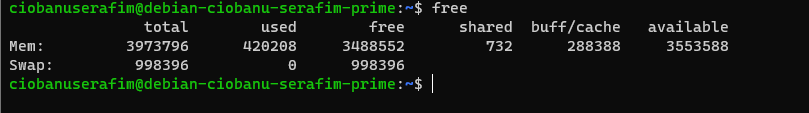


1. Show current memory and cpu usage

top – for cpu usage



**free – for memory**



1. Use a single line statement that shows the line from the /etc/passwd file that contains information about the current user (QUIZ)

sudo cat /etc/passwd | grep -i $USER



1. Power down your linux instance

sudo shutdown now