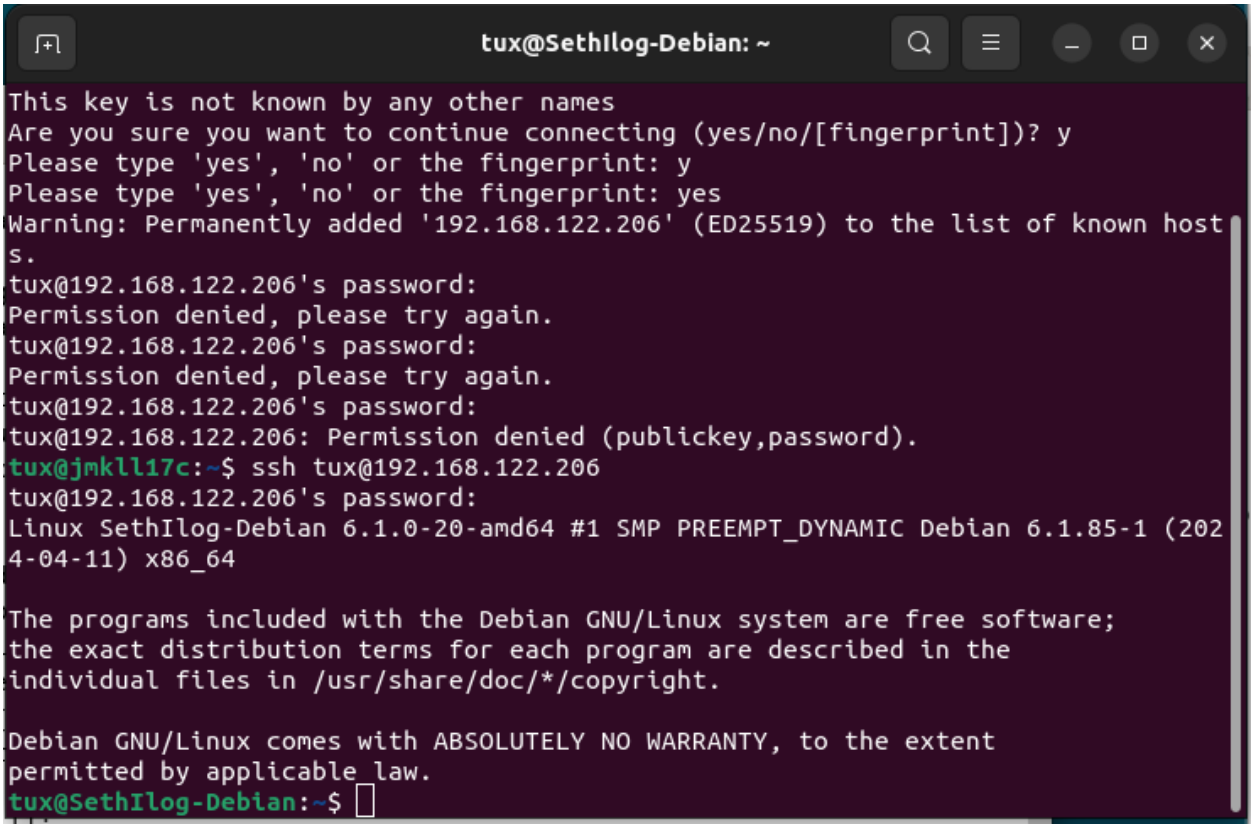
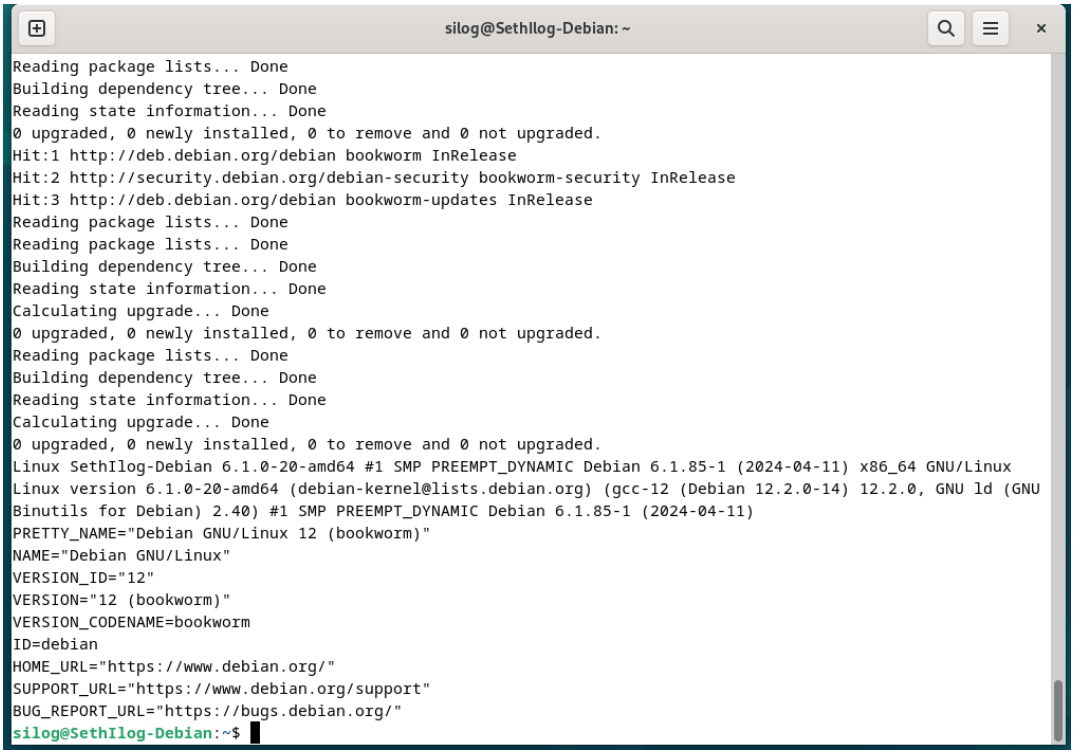
Seth Andrew Ilog

Creating a Virtual Machine on Linux

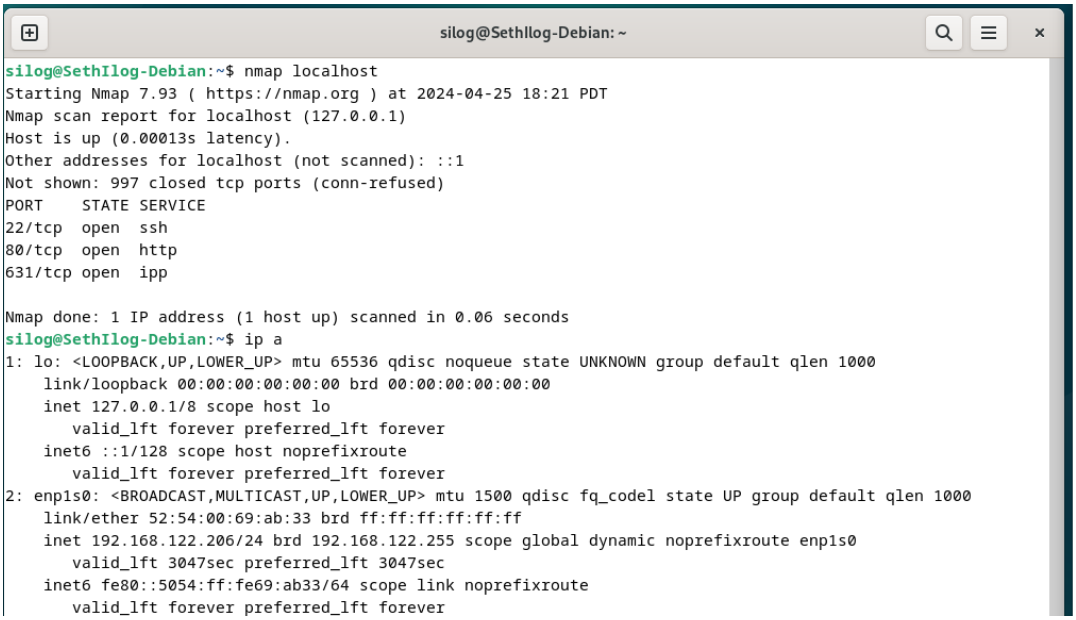
The IP address designated for the newly created virtual machine with the confirmation of being able to connect to said virtual machine from the *tux* host machine.



Updated software packages for virtual machine and found the home URL used for the default Debian webpage.



Found my local host IP address (192.168.122.206) of my virtual machine to replace the Debian default webpage with my own custom webpage.



Adjusted *index.html* to insert the header, body, and footer to my custom webpage.

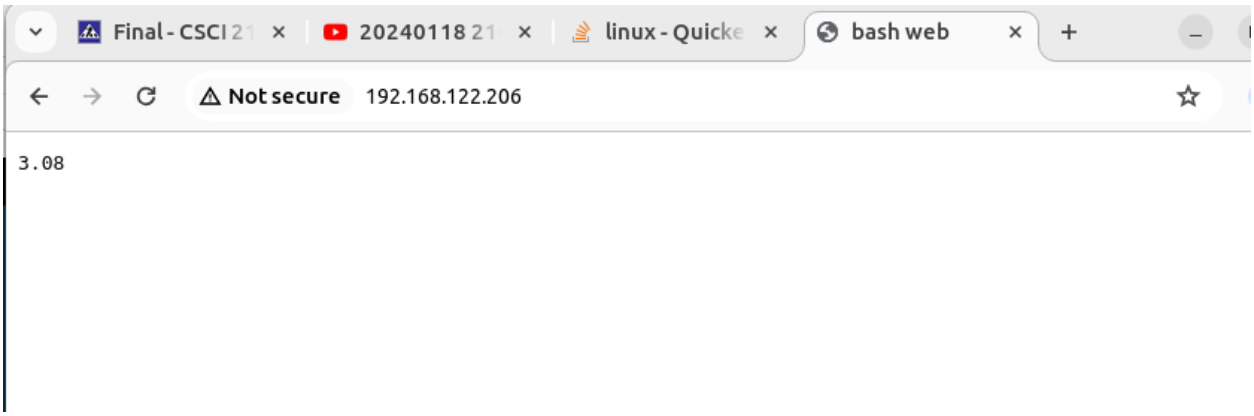
whdr() containing the contents to the header of my webpage.

The results of the **calcit()** command being used for the body of my webpage.

wftr() containing the contents to the footer of my webpage.

```
silog@SethIlog-Debian:~$ whdr >> index.html
silog@SethIlog-Debian:~$ fortune
bash: fortune: command not found
silog@SethIlog-Debian:~$ calcit "355/115"
3.08
silog@SethIlog-Debian:~$ calcit "355/115" >> index.html
silog@SethIlog-Debian:~$ wftr >> index.html
silog@SethIlog-Debian:~$ cat index.html
<!DOCTYPE html><head><title>bash web</title></head><body><pre>
3.08
</pre></body></html>
silog@SethIlog-Debian:~$ sudo mv index.html /var/www/html/
silog@SethIlog-Debian:~$
```

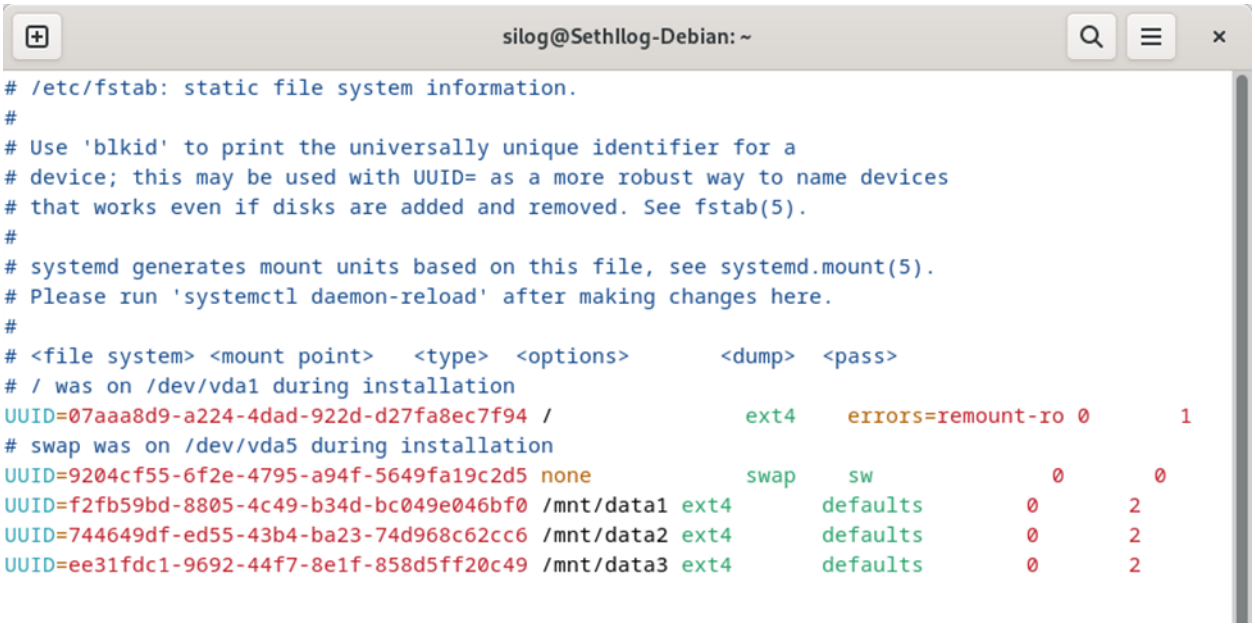
Results of my newly built, custom webpage utilizing my virtual machine' IP address (192.168.122.206):



Created a new vi text file under, */etc/fstab*, which would contain the UUIDs of the created logical volumes of the physical drives manually installed onto the virtual machine via virtual hardware.

Used the **vi /etc/fstab** command to create the default, pre-made text file for */etc/fstab*.

Added the UUIDs for the logical volumes of */mnt/data1*, */mnt/data2*, and */mnt/data3* manually with each volume containing 2GB of usable space.



Used the **pvcreate** command to create physical volumes for the virtual machine.

Created 2 new physical volumes under, *vdb1* and *vdc1*.

Used the **pvdisplay** command to showcase the 2 newly created physical volumes added to the virtual machine.

With the full names of the physical volumes being */dev/vdb1* and */dev/vdc1*.

```
silog@SethIlog-Debian:~$ sudo pvdisplay
[sudo] password for silog:
  --- Physical volume ---
  PV Name               /dev/vdb1
  VG Name               datavg
  PV Size               <5.00 GiB / not usable 3.00 MiB
  Allocatable           yes (but full)
  PE Size               4.00 MiB
  Total PE              1279
  Free PE               0
  Allocated PE          1279
  PV UUID               PMXpv4-cxR0-k60j-atP4-AJQA-GJQG-dGajcn

  --- Physical volume ---
  PV Name               /dev/vdc1
  VG Name               datavg
  PV Size               <5.00 GiB / not usable 3.00 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              1279
  Free PE               254
  Allocated PE          1025
  PV UUID               N5rTyI-0Tba-XY2h-m6O3-kUjP-PUIW-cfP0JZ
```

Used the **vgcreate** command to create a new volume group that contains both of the 2 newly created physical volumes.

The new volume group being named, *datavg*.

The physical volumes being */dev/vdb1* and */dev/vdc1*.

Used the **vgdisplay** command to review the newly created volume group called, *datavg*, which contains the 2 previously created physical volumes.

```
silog@SethIlog-Debian:~$ sudo vgdisplay
  --- Volume group ---
  VG Name               datavg
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  4
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                3
  Open LV               3
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               9.99 GiB
  PE Size               4.00 MiB
  Total PE              2558
  Alloc PE / Size       2304 / 9.00 GiB
  Free  PE / Size       254 / 1016.00 MiB
  VG UUID               7H8VIY-4ITJ-mfXp-S2FY-FZCx-NJnW-RaICM2
```

Used the **lvcreate** command to create 3 logical volumes based upon the previously created volume group named, *datavg*.

Created 3 separate logical volumes named, *data1lv*, *data2lv*, and *data3lv*.

Each of the logical volumes contain 2GB of usable space with the exception of *data1lv* which contains a full 5GB of usable space.

data1lv:

```
silog@SethIlog-Debian:~$ sudo lvdisplay
  --- Logical volume ---
  LV Path                /dev/datavg/data1lv
  LV Name                data1lv
  VG Name                datavg
  LV UUID                7Sdfx6-d1zp-Uflj-mJV1-IlYw-gFmD-CiF51d
  LV Write Access        read/write
  LV Creation host, time SethIlog-Debian, 2024-04-18 17:29:12 -0700
  LV Status              available
  # open                 1
  LV Size                5.00 GiB
  Current LE             1280
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:0
```

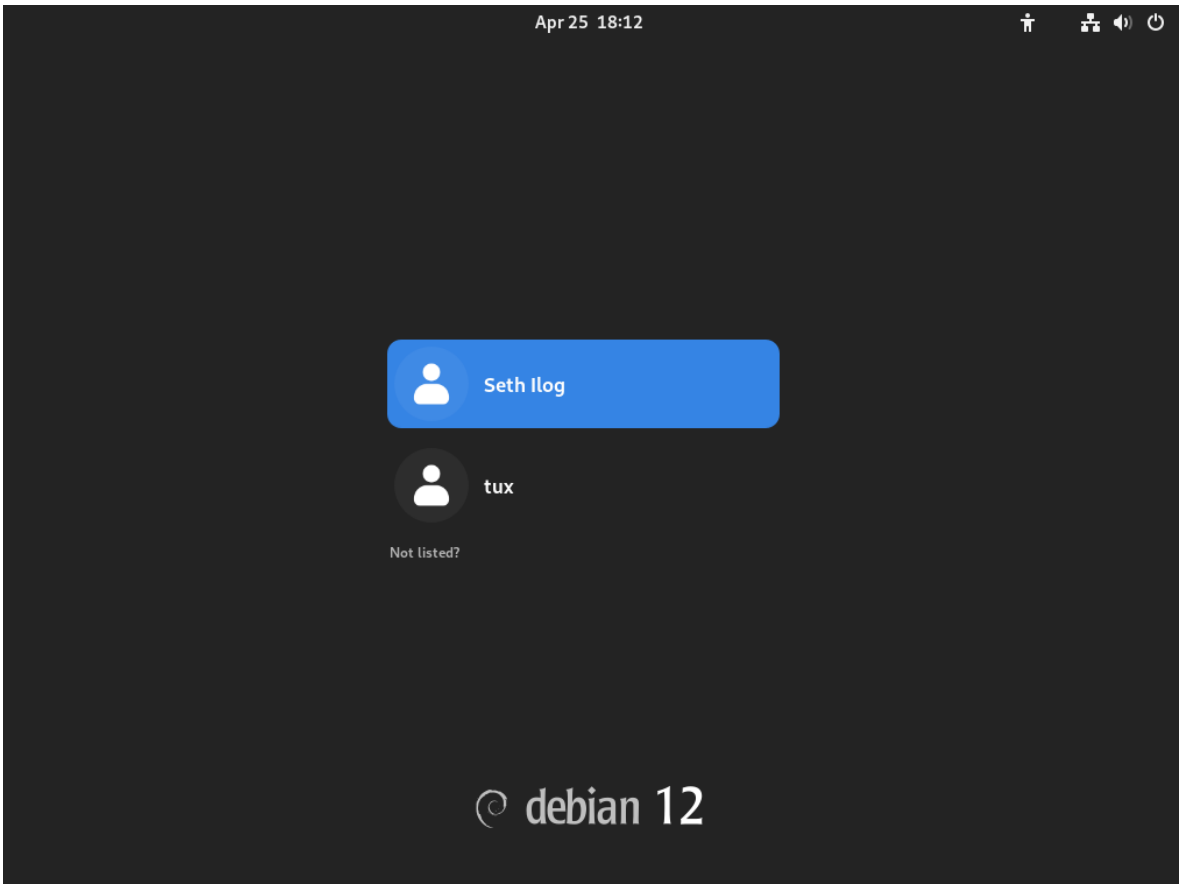
data2lv:

```
  --- Logical volume ---
  LV Path                /dev/datavg/data2lv
  LV Name                data2lv
  VG Name                datavg
  LV UUID                Cork8a-b5Zv-T4JX-GkyH-9nub-pZdD-nxAZG5
  LV Write Access        read/write
  LV Creation host, time SethIlog-Debian, 2024-04-18 17:34:38 -0700
  LV Status              available
  # open                 1
  LV Size                2.00 GiB
  Current LE             512
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           253:1
```

data3lv:

```
--- Logical volume ---
LV Path                /dev/datavg/data3lv
LV Name                data3lv
VG Name                datavg
LV UUID                nJZifX-wspu-npXV-tfOg-Hkia-0Ng0-g2NokY
LV Write Access        read/write
LV Creation host, time SethIlog-Debian, 2024-04-18 17:35:41 -0700
LV Status              available
# open                 1
LV Size                2.00 GiB
Current LE             512
Segments               1
Allocation             inherit
Read ahead sectors     auto
- currently set to     256
Block device           253:2
```

Additional user, *Tux*, added onto my virtual machine.



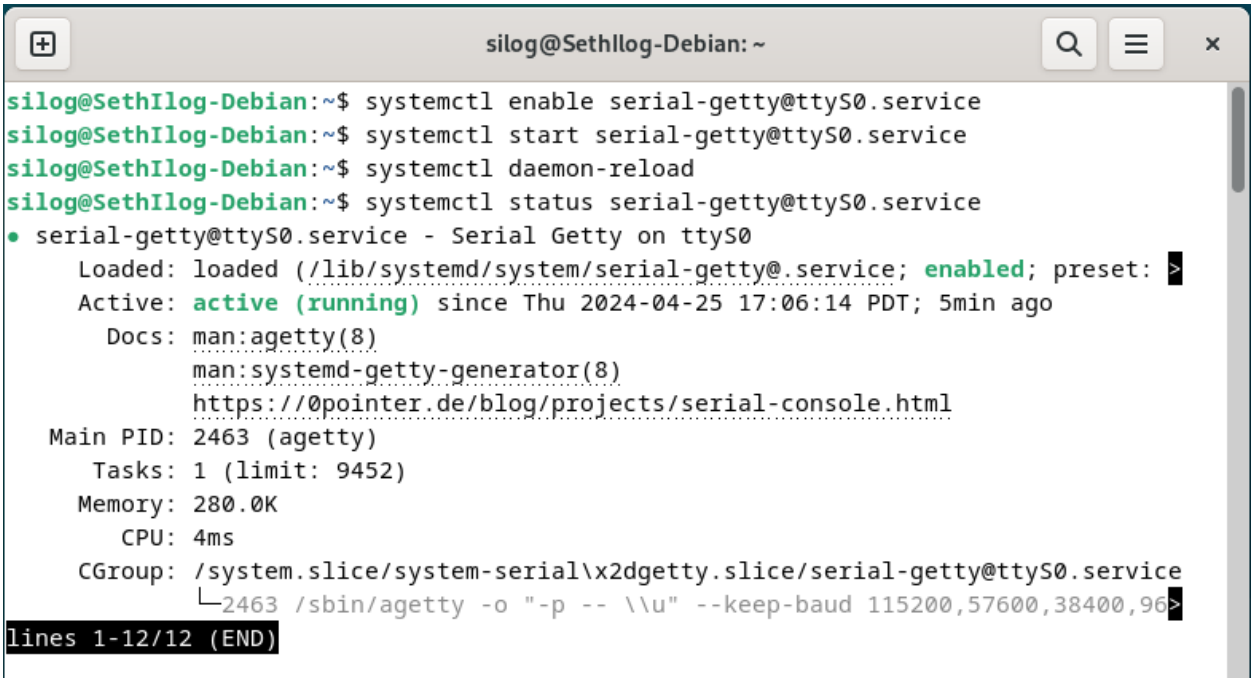
Serial console setup for my virtual machine utilizing **systemctl (…) serial-getty@ttyS0.service** commands as well as the **systemctl daemon-reload** command.

systemctl enable serial-getty@ttyS0.service enables the serial console for the virtual machine so that the system can be directly connected to, and interacted with, by another host's system.

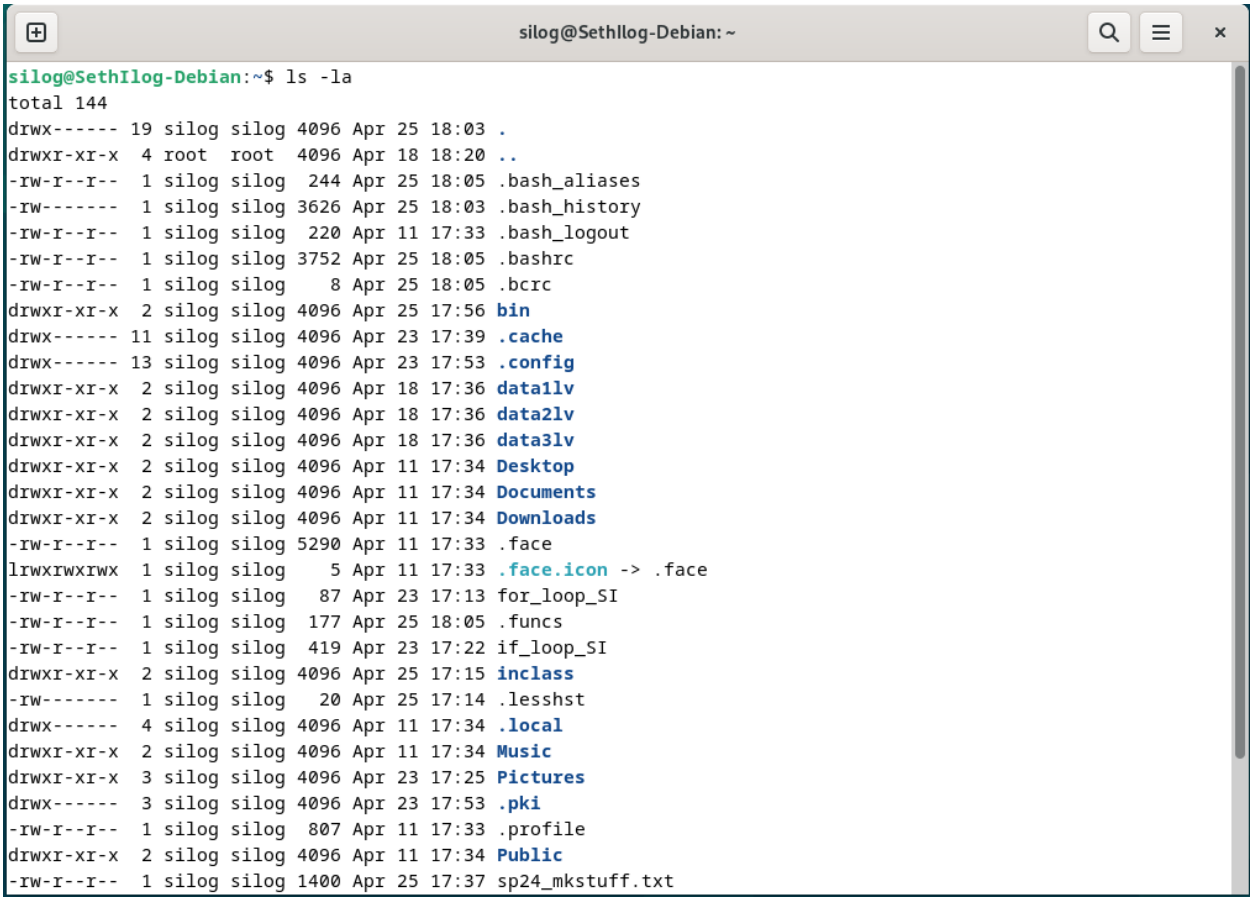
systemctl start serial-getty@ttyS0.service starts the process of the serial console up so that the virtual machine can be interacted with by another host's system.

systemctl daemon-reload reloads the virtual machine's system configuration files without fully restarting the system or any of its services.

systemctl status serial-getty@ttyS0.service checks on the status of the serial console for the virtual machine; in essence, checking to see if it's enabled, disabled, active, inactive, etc.

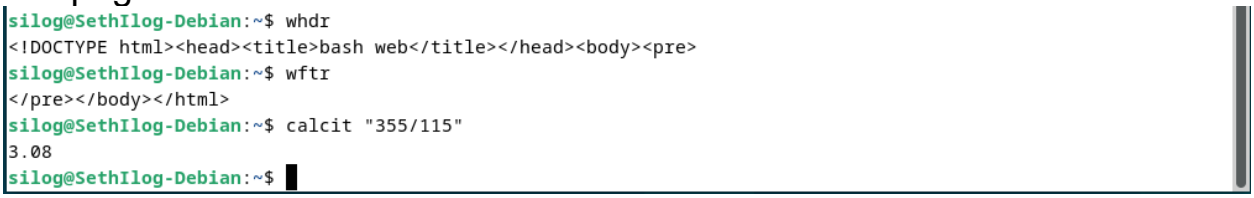


bin and *inclass* directories added into home directory.



bin and *inclass* directories added to $PATH.

Added *.funcs* via *calcit()*, *whdr()*, and *wftr()* to use to create custom webpage.



Created a simple **for** loop which results in my first and last name being listed 10 times with each string containing its own full line of text.

```
#!/bin/bash
#for_loop_SI

LST=$(seq 1 10)
for i in $LST
do
        echo "Seth Ilog"
done

#--
```

Results of **for** loop utilizing the **bash** command:

```
silog@SethIlog-Debian:~$ bash for_loop_SI
Seth Ilog
Seth Ilog
Seth Ilog
Seth Ilog
Seth Ilog
Seth Ilog
Seth Ilog
Seth Ilog
Seth Ilog
Seth Ilog
```

Created an **if** loop which results in an output that corresponds to the user's input.

If the user enters any of the 3 colors of a stoplight, in ALL lowercase letters, the output of the **if** loop will respond to the input with a specifically written response that relates to what a driver should do during a green, yellow, or red stoplight.

green = "Green stoplight = GO"

yellow = "Yellow stoplight = YIELD or SLOW DOWN"

red = "Red stoplight = STOP"

If the user enters anything else other than the 3 stoplight colors in ALL lowercase letters, the if loop will output a response appropriately telling the user it "Cannot.".

If the user enters the letter "q", the **if** loop will stop and the program will cease to continue.

```
#!/bin/bash
# stoplight
# simple if test

display_light() {
 case $1 in
  "red") echo "Red stoplight = STOP";;
  "yellow") echo "Yellow stoplight = YIELD or SLOW DOWN";;
  "green") echo "Green stoplight = GO";;
  *)    echo "Cannot.";;
 esac
}
# main
while true
 do
  echo -n "Enter a stoplight color with ALL lowercase letters. "
  read light
   if [ $light == "q" ]
    then break
   fi
  display_light $light
 done
#--
"if_loop_SI" 23L, 419B                                        7,38          All
```

Results of **if** loop utilizing the **bash** command:

```
silog@SethIlog-Debian:~$ bash if_loop_SI
Enter a stoplight color with ALL lowercase letters. red
Red stoplight = STOP
Enter a stoplight color with ALL lowercase letters. yellow
Yellow stoplight = YIELD or SLOW DOWN
Enter a stoplight color with ALL lowercase letters. green
Green stoplight = GO
Enter a stoplight color with ALL lowercase letters. blah
Cannot.
Enter a stoplight color with ALL lowercase letters. q
silog@SethIlog-Debian:~$
```

The **if** loop will always respond to the correct and incorrect inputs accordingly.

Correct inputs: green, yellow, red, q

Incorrect inputs: anything OTHER than what is listed as a correct input.