# Lab 7 Process

Leonardo Fusser (1946995)

**Objectives**:

- Analyse processes

**Log in as debian or your username.**

1. Implement the following script file to blink usr0 led at 1 HZ.

bashUSR0blink.sh

```
LED0_PATH=/sys/class/leds/beaglebone:green:usr0

while [ "1" = "1" ]; do
  echo "1" >> "$LED0_PATH/brightness"
  sleep 1
  echo "0" >> "$LED0_PATH/brightness"
  sleep 1
done
```

a) To run the file, you must do what?

*Make executable by typing chmod +x bashUSR0blink.sh and to execute, use ./bashUSR0blink.sh command.*

b) To run the file, you must be logged in as what?

*Sudo (super user).*

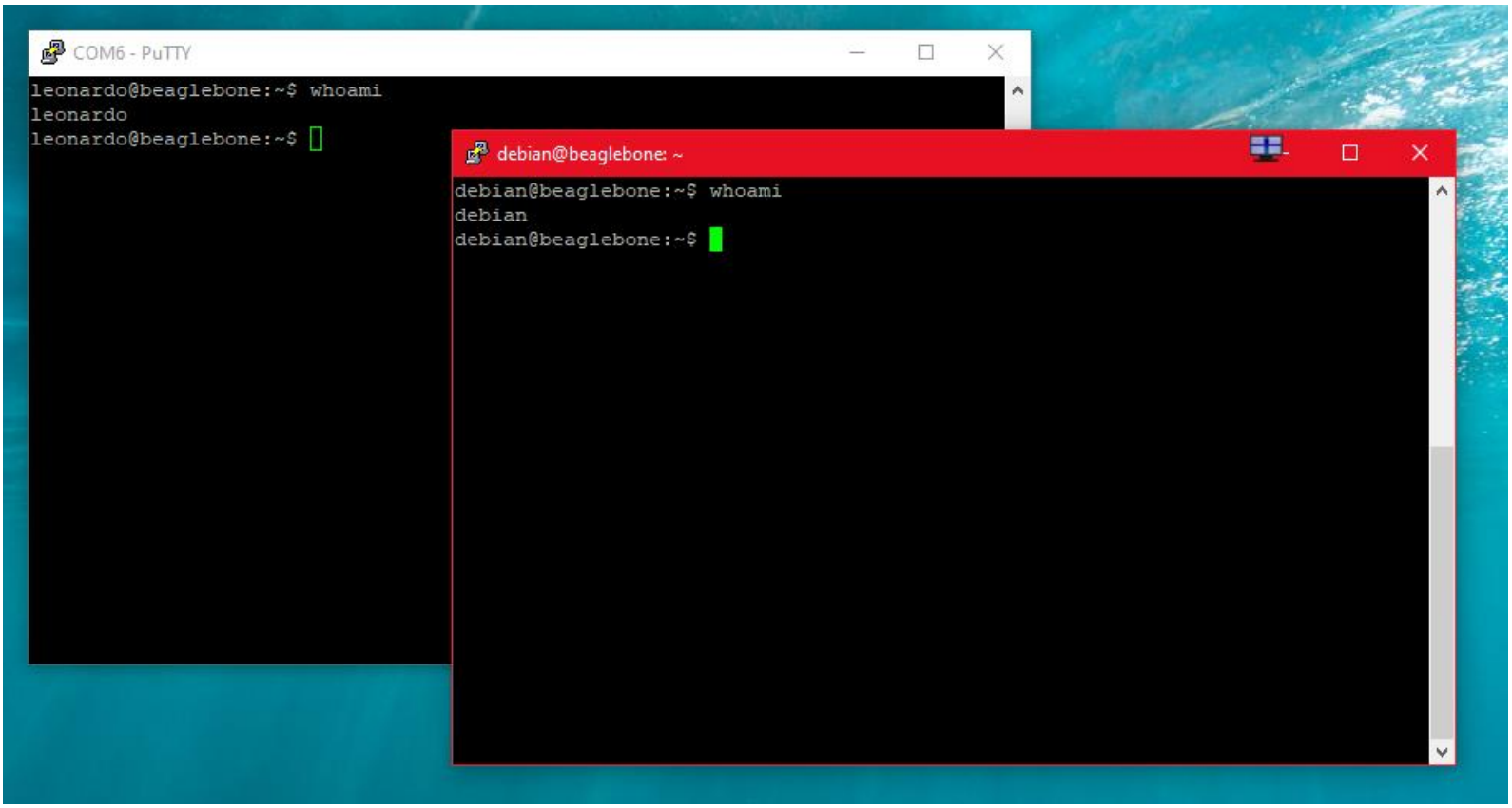
```
  GNU nano 2.2.6                File: bashUSR0blink.sh

#!/bin/bash
#Created by Leonardo Fusser (1946995)

LED0_PATH=/sys/class/leds/beaglebone:green:usr0

while [ "1" = "1" ];do
        echo "1" >> "$LED0_PATH/brightness"
        sleep 1
        echo "0" >> "$LED0_PATH/brightness"
        sleep 1
done
```

*Script file for 1 shown above.*

```
leonardo@beaglebone:~/Lab7$ chmod +x bashUSR0blink.sh
leonardo@beaglebone:~/Lab7$ ls
bashUSR0blink.sh
leonardo@beaglebone:~/Lab7$ ./bashUSR0blink.sh
./bashUSR0blink.sh: line 7: /sys/class/leds/beaglebone:green:usr0/brightness: Pe
rmission denied
./bashUSR0blink.sh: line 9: /sys/class/leds/beaglebone:green:usr0/brightness: Pe
rmission denied
./bashUSR0blink.sh: line 7: /sys/class/leds/beaglebone:green:usr0/brightness: Pe
rmission denied
^C
leonardo@beaglebone:~/Lab7$ sudo ./bashUSR0blink.sh
```

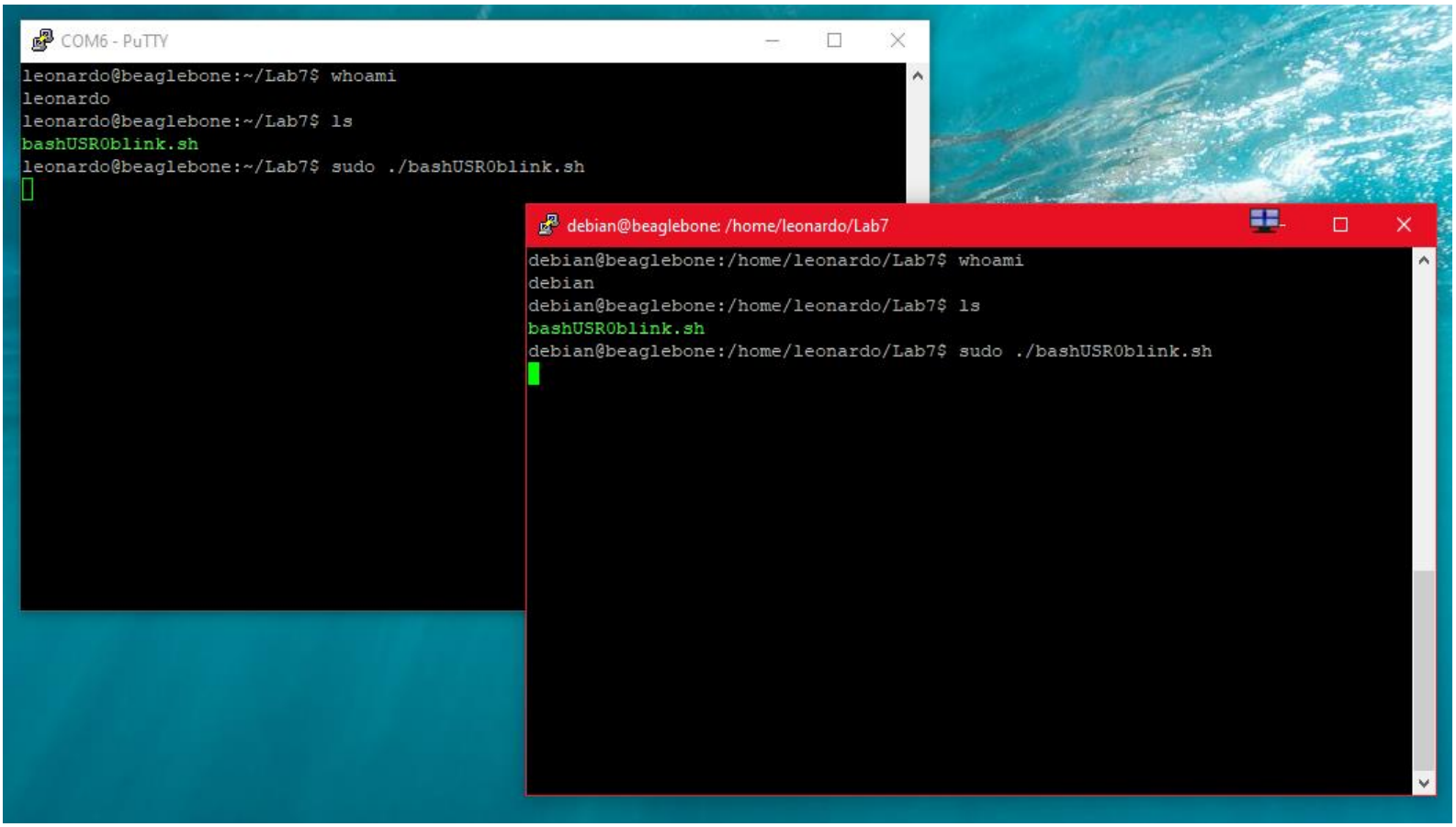
*Making script executable and running scrip as super user and without super user privileges for 1.*

c) Open a new window and login with the right user (keep first window open)




*Two sessions open, one serial – logged in as "leonardo" (left screen) and the other SSH – logged in as "debian" (right screen) shown above.*

d) Run the program in the new window.




*Running the script in both sessions (serial and SSH) based on script file in 1 shown above.*

2. Improve this file to allow you to input turning the blinking on or off with command line parameters.
   Ex: **bashUSR0blink on** … **the light starts blinking.**
   Or: **bashUSR0blink off** … **the light turns off.**

```
COM6 - PuTTY
  GNU nano 2.2.6                File: bashUSR0blink.sh

#!/bin/bash
#Created by Leonardo Fusser (1946995)

LED0_PATH=/sys/class/leds/beaglebone:green:usr0

while [ "1" = "1" ];do

        if [ $1 = "on" ]; then
                echo "1" >> "$LED0_PATH/brightness"
                sleep 1
                echo "0" >> "$LED0_PATH/brightness"
                sleep 1
        else
                echo "0" >> "$LED0_PATH/brightness"

        fi

done
```

*Script file from 1 updated to turn on or off LED blinking depending on user arguments shown above.*

```
COM6 - PuTTY
leonardo@beaglebone:~/Lab7$ chmod +x bashUSR0blink.sh
leonardo@beaglebone:~/Lab7$ ls
bashUSR0blink.sh
leonardo@beaglebone:~/Lab7$ sudo ./bashUSR0blink.sh on
^Cleonardo@beaglebone:~/Lab7$
leonardo@beaglebone:~/Lab7$
leonardo@beaglebone:~/Lab7$ sudo ./bashUSR0blink.sh off
^Cleonardo@beaglebone:~/Lab7$
leonardo@beaglebone:~/Lab7$
leonardo@beaglebone:~/Lab7$
```

*Making updated script executable and running script shown above.*

3. Now run it as a deamon with an "on" parameter (run in the background)
   a) What is the command? *sudo ./bashUSR0blink.sh on &*
   b) What happened? Can you type another command? Explain:
      *The script was moved to run in the background and I was able to keep inputting commands after the script was moved to run in the background.*

4. Display all background running process using `jobs`

```
COM6 - PuTTY
root@beaglebone:/home/leonardo/Lab7# ./bashUSR0blink.sh on &
[1] 4334
root@beaglebone:/home/leonardo/Lab7# jobs
[1]+  Running                 ./bashUSR0blink.sh on &
root@beaglebone:/home/leonardo/Lab7#
```

*Script running in background and PID is assigned to 4334 shown above.*

5. `Go back to (debian/your name window) try jobs there!`

```
COM6 - PuTTY
leonardo@beaglebone:~/Lab7$ jobs
[1]+  Running                 sudo ./bashUSR0blink.sh on &
leonardo@beaglebone:~/Lab7$
```

*Script shown still running in background while logged in using different account shown above.*

6. Open a new SSH console. Display all running process using `ps -ef` and `pstree -p` commands, Take screenshots.

```
leonardo@beaglebone: ~
leonardo@beaglebone:~$ ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 18:59 ?        00:00:01 /lib/systemd/systemd
root         2     0  0 18:59 ?        00:00:00 [kthreadd]
root         3     2  0 18:59 ?        00:00:00 [ksoftirqd/0]
root         5     2  0 18:59 ?        00:00:00 [kworker/0:0H]
root         7     2  0 18:59 ?        00:00:00 [kworker/u:0H]
root         8     2  0 18:59 ?        00:00:00 [migration/0]
root         9     2  0 18:59 ?        00:00:00 [rcu_bh]
root        10     2  0 18:59 ?        00:00:09 [rcu_sched]
root        11     2  0 18:59 ?        00:00:00 [watchdog/0]
root        12     2  0 18:59 ?        00:00:00 [khelper]
root        13     2  0 18:59 ?        00:00:00 [kdevtmpfs]
root        14     2  0 18:59 ?        00:00:00 [netns]
root        16     2  0 18:59 ?        00:00:00 [bdi-default]
root        17     2  0 18:59 ?        00:00:00 [kintegrityd]
root        18     2  0 18:59 ?        00:00:00 [kblockd]
root        19     2  0 18:59 ?        00:00:00 [khubd]
root        20     2  0 18:59 ?        00:00:00 [irq/70-44e0b000]
root        21     2  0 18:59 ?        00:00:00 [kworker/u:1]
root        24     2  0 18:59 ?        00:00:00 [irq/7-tps65217]
root        27     2  0 18:59 ?        00:00:00 [irq/30-4819c000]
root        36     2  0 18:59 ?        00:00:00 [rpciod]
root        38     2  0 18:59 ?        00:00:00 [khungtaskd]
```

*Process shown using "ps ef" command while logged in as "leonardo" shown above.*

```
leonardo@beaglebone: ~
leonardo@beaglebone:~$ pstree -p
systemd(1)-+-acpid(533)
           |-agetty(561)
           |-agetty(1095)
           |-apache2(807)-+-apache2(819)
           |              |-apache2(829)-+-{apache2}(835)
           |              |              |-{apache2}(836)
           |              |              |-{apache2}(837)
           |              |              |-{apache2}(838)
           |              |              |-{apache2}(839)
           |              |              |-{apache2}(840)
           |              |              |-{apache2}(841)
           |              |              |-{apache2}(842)
           |              |              |-{apache2}(843)
           |              |              |-{apache2}(844)
           |              |              |-{apache2}(845)
           |              |              |-{apache2}(846)
           |              |              |-{apache2}(847)
           |              |              |-{apache2}(848)
           |              |              |-{apache2}(849)
           |              |              |-{apache2}(850)
           |              |              |-{apache2}(869)
           |              |              |-{apache2}(870)
           |              |              |-{apache2}(871)
|-sshd(791)---sshd(4795)---sshd(4808)---bash(4809)---pstree(4991)
```

*Process shown using "pstree -p" command while logged in as "leonardo" shown above.*

7. Try different options of ps in the different login windows to really understand the command (ps, ps -aux, ps -ax, ps -ef)

```
COM6 - PuTTY
root@beaglebone:/home/leonardo/Lab7# ps
  PID TTY          TIME CMD
 4715 ttyGS0   00:00:00 login
 4737 ttyGS0   00:00:00 bash
 5330 ttyGS0   00:00:00 ps
root@beaglebone:/home/leonardo/Lab7#
```

*Process shown using "ps" command while logged in as "root" shown above.*

```
COM6 - PuTTY
root@beaglebone:/home/leonardo/Lab7# ps -aux
warning: bad ps syntax, perhaps a bogus '-'?
See http://gitorious.org/procps/procps/blobs/master/Documentation/FAQ
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.5   4464  2616 ?        Ss   18:59   0:01 /lib/systemd/sy
root         2  0.0  0.0      0     0 ?        S    18:59   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    18:59   0:00 [ksoftirqd/0]
root         5  0.0  0.0      0     0 ?        S<   18:59   0:00 [kworker/0:0H]
root         7  0.0  0.0      0     0 ?        S<   18:59   0:00 [kworker/u:0H]
root         8  0.0  0.0      0     0 ?        S    18:59   0:00 [migration/0]
root         9  0.0  0.0      0     0 ?        S    18:59   0:00 [rcu_bh]
root        10  0.1  0.0      0     0 ?        S    18:59   0:10 [rcu_sched]
root        11  0.0  0.0      0     0 ?        S    18:59   0:00 [watchdog/0]
root        12  0.0  0.0      0     0 ?        S<   18:59   0:00 [khelper]
root        13  0.0  0.0      0     0 ?        S    18:59   0:00 [kdevtmpfs]
root        14  0.0  0.0      0     0 ?        S<   18:59   0:00 [netns]
root        16  0.0  0.0      0     0 ?        S    18:59   0:00 [bdi-default]
root        17  0.0  0.0      0     0 ?        S<   18:59   0:00 [kintegrityd]
root        18  0.0  0.0      0     0 ?        S<   18:59   0:00 [kblockd]
root        19  0.0  0.0      0     0 ?        S    18:59   0:00 [khubd]
root        20  0.0  0.0      0     0 ?        S    18:59   0:00 [irq/70-44e0b00
root        21  0.0  0.0      0     0 ?        S    18:59   0:00 [kworker/u:1]
root        24  0.0  0.0      0     0 ?        S    18:59   0:00 [irq/7-tps65217
root        27  0.0  0.0      0     0 ?        S    18:59   0:00 [irq/30-4819c00
```

*Process shown using "ps aux" command while logged in as "root" shown above.*

```
leonardo@beaglebone:~$ ps -ax
warning: bad ps syntax, perhaps a bogus '-'?
See http://gitorious.org/procps/procps/blobs/master/Documentation/FAQ
  PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:01 /lib/systemd/systemd
    2 ?        S      0:00 [kthreadd]
    3 ?        S      0:00 [ksoftirqd/0]
    5 ?        S<     0:00 [kworker/0:0H]
    7 ?        S<     0:00 [kworker/u:0H]
    8 ?        S      0:00 [migration/0]
    9 ?        S      0:00 [rcu_bh]
   10 ?        S      0:10 [rcu_sched]
   11 ?        S      0:00 [watchdog/0]
   12 ?        S<     0:00 [khelper]
   13 ?        S      0:00 [kdevtmpfs]
   14 ?        S<     0:00 [netns]
   16 ?        S      0:00 [bdi-default]
   17 ?        S<     0:00 [kintegrityd]
   18 ?        S<     0:00 [kblockd]
   19 ?        S      0:00 [khubd]
   20 ?        S      0:00 [irq/70-44e0b000]
   21 ?        S      0:00 [kworker/u:1]
   24 ?        S      0:00 [irq/7-tps65217]
   27 ?        S      0:00 [irq/30-4819c000]
```

*Process shown using "ps ax" command while logged in as "leonardo" shown above.*

```
leonardo@beaglebone: ~
leonardo@beaglebone:~$ ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 18:59 ?        00:00:01 /lib/systemd/systemd
root         2     0  0 18:59 ?        00:00:00 [kthreadd]
root         3     2  0 18:59 ?        00:00:00 [ksoftirqd/0]
root         5     2  0 18:59 ?        00:00:00 [kworker/0:0H]
root         7     2  0 18:59 ?        00:00:00 [kworker/u:0H]
root         8     2  0 18:59 ?        00:00:00 [migration/0]
root         9     2  0 18:59 ?        00:00:00 [rcu_bh]
root        10     2  0 18:59 ?        00:00:10 [rcu_sched]
root        11     2  0 18:59 ?        00:00:00 [watchdog/0]
root        12     2  0 18:59 ?        00:00:00 [khelper]
root        13     2  0 18:59 ?        00:00:00 [kdevtmpfs]
root        14     2  0 18:59 ?        00:00:00 [netns]
root        16     2  0 18:59 ?        00:00:00 [bdi-default]
root        17     2  0 18:59 ?        00:00:00 [kintegrityd]
root        18     2  0 18:59 ?        00:00:00 [kblockd]
root        19     2  0 18:59 ?        00:00:00 [khubd]
root        20     2  0 18:59 ?        00:00:00 [irq/70-44e0b000]
root        21     2  0 18:59 ?        00:00:00 [kworker/u:1]
root        24     2  0 18:59 ?        00:00:00 [irq/7-tps65217]
root        27     2  0 18:59 ?        00:00:00 [irq/30-4819c000]
root        36     2  0 18:59 ?        00:00:00 [rpciod]
root        38     2  0 18:59 ?        00:00:00 [khungtaskd]
```

*Process shown using "ps ef" command while logged in as "leonardo" shown above.*

8. Try the pstree –p a few times, what changes?

```
├─login(4715)───bash(4737)───pstree(5741)
```

```
├─login(4715)───bash(4737)───pstree(5788)
```

```
├─login(4715)───bash(4737)───pstree(5925)
```

*After each time, pstree gets a new PID shown above.*

9. Who is the parent process of the current -bash? Make a table and write all the parents increasing the age.
*4715 is the parent process of current -bash.*

| |
|---|
| *0* |
| *1* |
| *2* |
| *238* |
| *520* |
| *646* |
| *791* |
| *807* |
| *1002* |
| *1042* |
| *1077* |
| *4715* |
| *4737* |
| *4795* |
| *4808* |

*All parent process increasing from 0 to 4808 shown above.*

```
root      4334     1  0 21:14 ?        00:00:02 /bin/bash ./bashUSR0blink.sh on
root      4503     1  0 21:16 ?        00:00:00 sudo ./bashUSR0blink.sh on
root      4505  4503  0 21:16 ?        00:00:02 /bin/bash ./bashUSR0blink.sh on
root      4715     1  0 21:18 ttyGS0   00:00:00 /bin/login --
root      4737  4715  0 21:18 ttyGS0   00:00:00 -bash
root      4795   791  0 21:18 ?        00:00:00 sshd: leonardo [priv]
leonardo  4808  4795  0 21:18 ?        00:00:00 sshd: leonardo@pts/1
leonardo  4809  4808  0 21:18 pts/1    00:00:00 -bash
root      5193     2  0 21:21 ?        00:00:00 [kworker/0:1]
root      5924     2  0 21:27 ?        00:00:00 [kworker/0:0]
```

*Parent process of current bash shown above.*

10. Who is the parent process of your daemon?

```
root      7275  4737  0 21:43 ttyGS0   00:00:00 /bin/bash ./bashUSR0blink.sh on
```

*Parent process is 4737 (PID = 7275) shown above.*

11. Stop the script by sending a SIGSTOP signal. The LED should stop blinking

Is the process stopped? *Yes.*

```
COM6 - PuTTY
root@beaglebone:/home/leonardo/Lab7# jobs
[1]+  Running                 ./bashUSR0blink.sh on &
root@beaglebone:/home/leonardo/Lab7# kill -STOP 7275
root@beaglebone:/home/leonardo/Lab7# jobs
[1]+  Stopped                 ./bashUSR0blink.sh on
root@beaglebone:/home/leonardo/Lab7#
```

*Result of sending SIGTOP signal to script running in background shown above.*

12. Restart the script by sending a SIGCONT signal. The LED should blink.

    Is the process running? *Yes.*

```
COM6 - PuTTY
root@beaglebone:/home/leonardo/Lab7# jobs
[1]+  Stopped                 ./bashUSR0blink.sh on
root@beaglebone:/home/leonardo/Lab7# kill -CONT 7275
root@beaglebone:/home/leonardo/Lab7# jobs
[1]+  Running                 ./bashUSR0blink.sh on &
root@beaglebone:/home/leonardo/Lab7#
```

*Result of sending SIGCONT signal to script running in background shown above.*

13. Kill or abort the script by sending a SIGABRT or SIGKILL or SIGTERM signal. The LED should stop blinking

    Is the process aborted? *Yes.*

```
COM6 - PuTTY
root@beaglebone:/home/leonardo/Lab7# jobs
[1]+  Running                 ./bashUSR0blink.sh on &
root@beaglebone:/home/leonardo/Lab7# kill -KILL 7275
[1]+  Killed                  ./bashUSR0blink.sh on
root@beaglebone:/home/leonardo/Lab7# jobs
root@beaglebone:/home/leonardo/Lab7#
```

*Result of sending SIGKILL signal to script running in background shown above.*

14. Continue killing the children increasing in ages. Kill the other bash console (not the one you are currently on)

```
├─login(1126)───bash(1275)─┬─bashUSR0blink.s(1794)───sleep(1969)
                          └─pstree(1970)
```

*"pstree -p" output for current bash console.*

```
root@beaglebone:/home/leonardo/Lab7# kill 1794
[1]+  Terminated              ./bashUSR0blink.sh on
root@beaglebone:/home/leonardo/Lab7# kill 1970
-bash: kill: (1970) - No such process
root@beaglebone:/home/leonardo/Lab7# kill 1969
-bash: kill: (1969) - No such process
root@beaglebone:/home/leonardo/Lab7#
```

*Killing the children for current bash console.*

```
├─login(565)───bash(1395)
```

*"pstree -p" output for the other bash console.*

COM6 - PuTTY

```
root@beaglebone:/home/leonardo/Lab7# kill 1395
root@beaglebone:/home/leonardo/Lab7# kill 565
root@beaglebone:/home/leonardo/Lab7#
```

COM6 - PuTTY

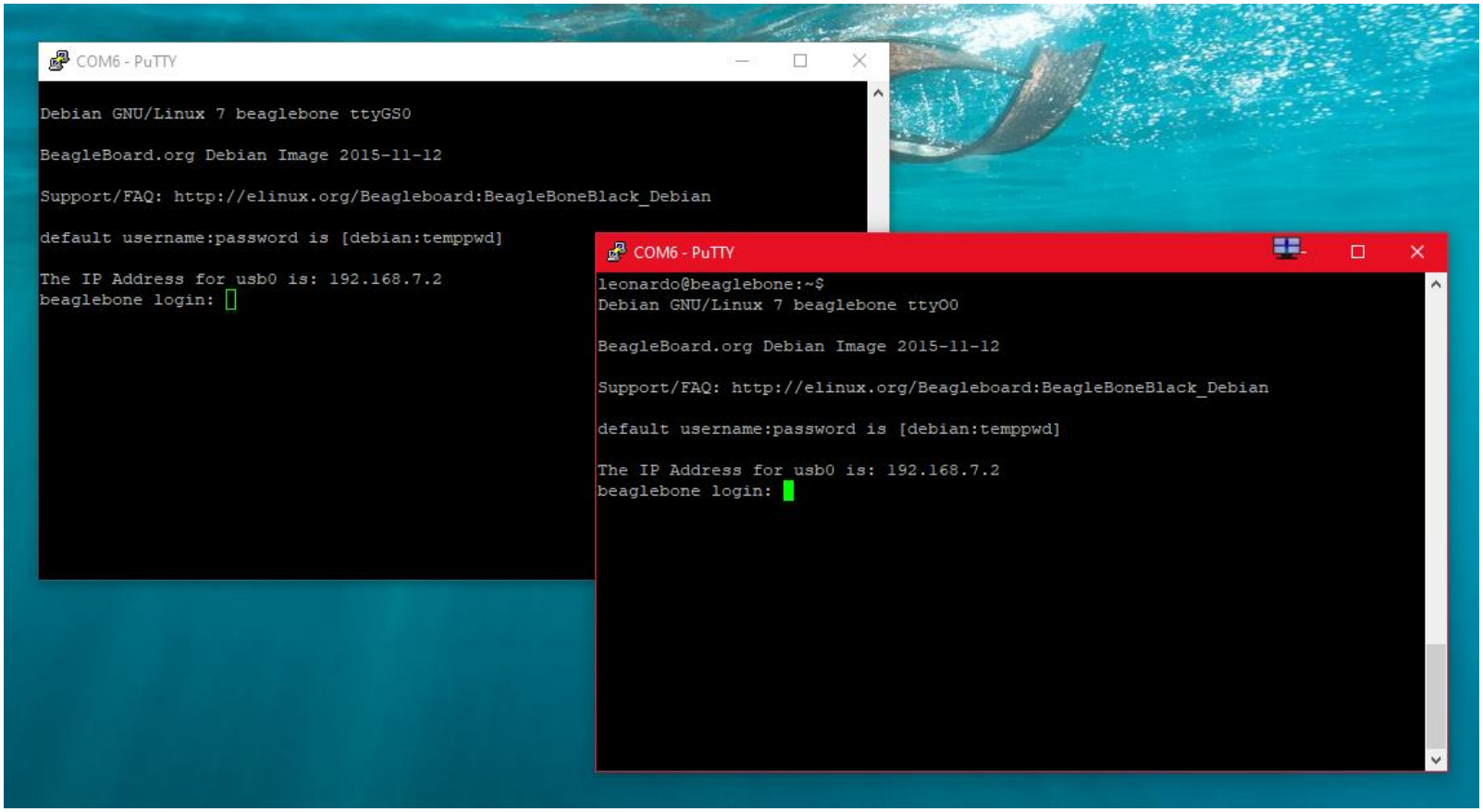
```
leonardo@beaglebone:~/Lab7$ whoami
leonardo
leonardo@beaglebone:~/Lab7$
leonardo@beaglebone:~/Lab7$
leonardo@beaglebone:~/Lab7$
leonardo@beaglebone:~/Lab7$
leonardo@beaglebone:~/Lab7$
leonardo@beaglebone:~/Lab7$
Debian GNU/Linux 7 beaglebone ttyO0

BeagleBoard.org Debian Image 2015-11-12

Support/FAQ: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian

default username:password is [debian:temppwd]

The IP Address for usb0 is: 192.168.7.2
beaglebone login:
```

*Killing the other bash console (located on right screen). Bash console logged out afterwards.*

15. Finally, kill the parent of both bash processes.

```
root      2823  2783  0 20:06 ttyGS0    00:00:00 -bash
leonardo  2843  2807  0 20:06 ttyO0     00:00:00 -bash
```

*"ps -ef" output for the two running bash processes.*




*After killing parent of both bash processes, both consoles are logged out.*

16. Now try to open a new ssh console

17. Explain the effect of killing the parent bash.

    *Killing both parent bash processes caused the two sessions to log out.*

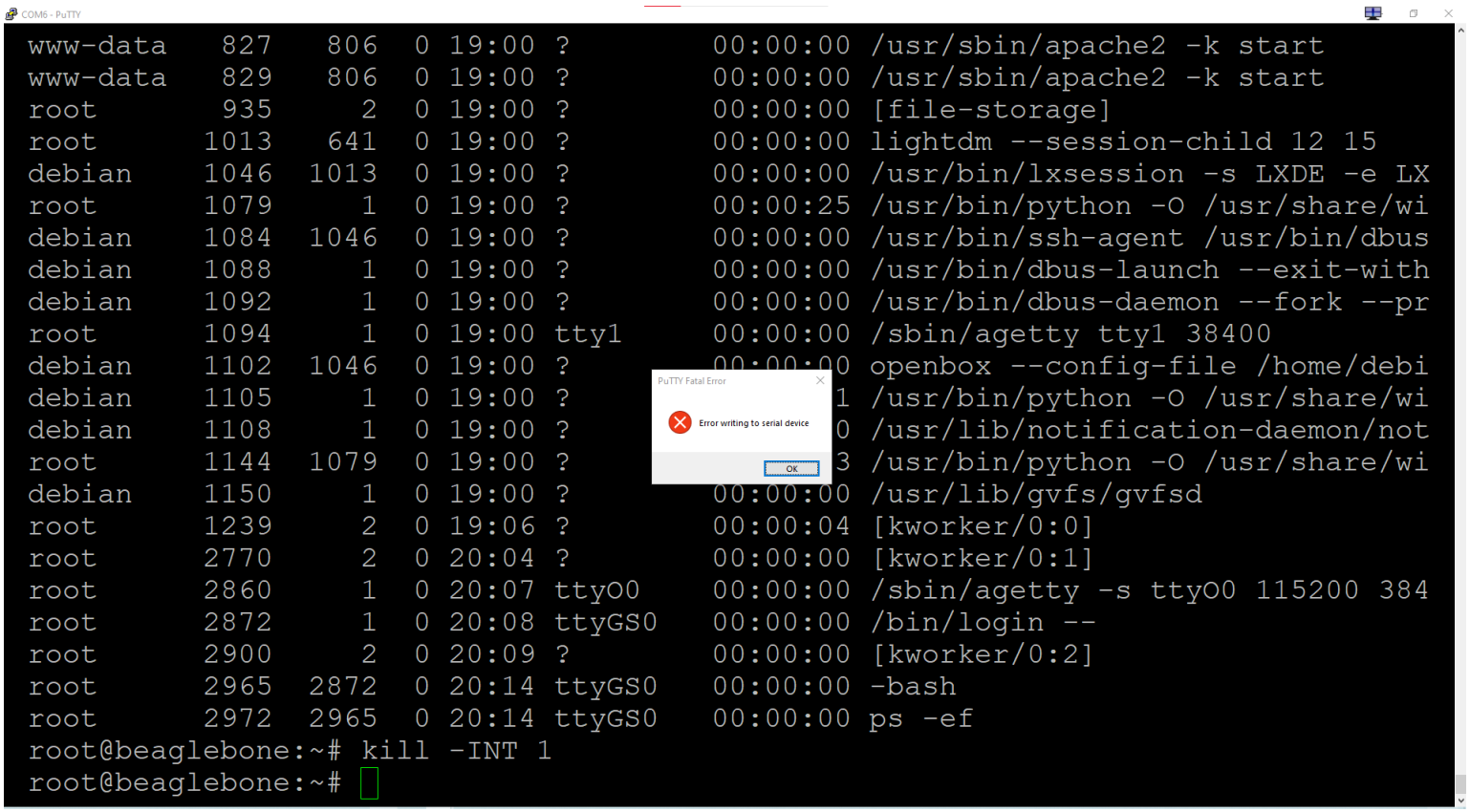
kill init does not work on Debian 8

Finally, kill `init (PID 1)` by typing:

```
sudo kill -ABRT 1
```

OR

```
sudo kill -INT 1
```




*Result after killing "init" (PID 1) shown above.*

Explain what happened when you killed the PID 1:

*After killing PID 1, the beaglebone became unresponsive and the system crashed. After it crashed, the system restarted automatically and I was able to log back in afterwards.*