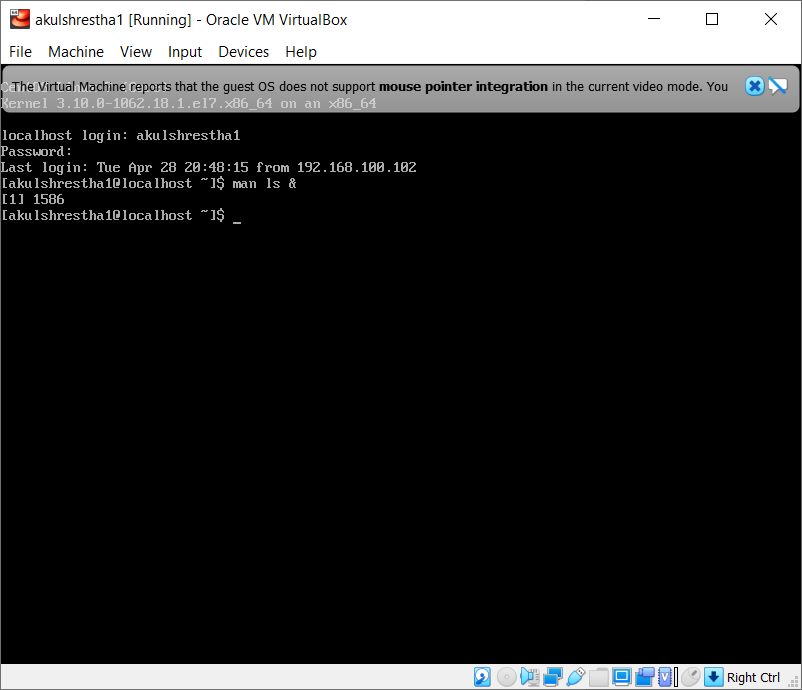
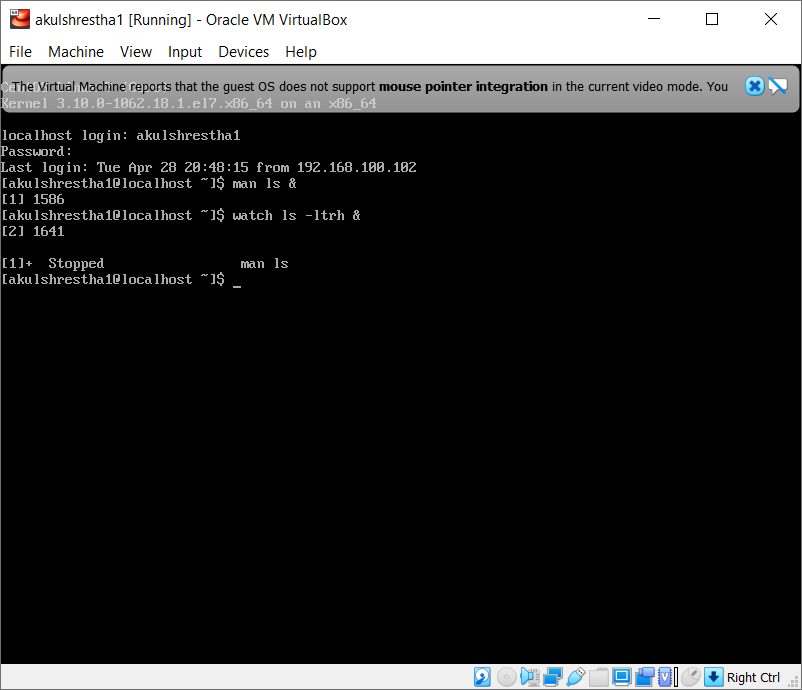
PART I

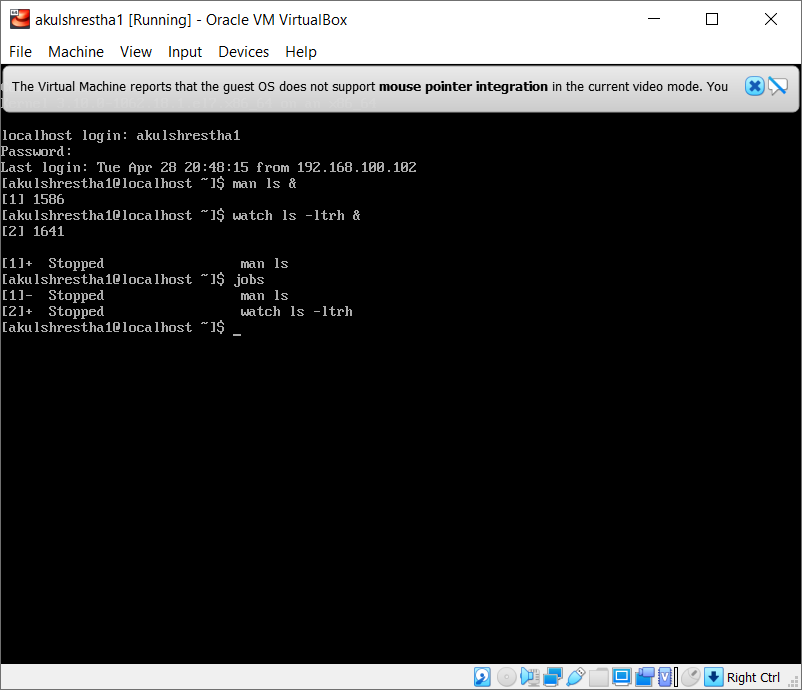
1. Watch [Linux Basic Process Control Command video](https://www.youtube.com/watch?v=swiHsaD8BGI)(22 mins)  
   (It will help you for the next steps)
2. Login to one of your Linux vm as a regular user
3. Type **man ls &**
4. Type **watch "ls -ltrh" &**
5. Type **jobs**
6. Bring job 2 to the foreground. Type **fg 2** or just **fg**.
7. Put job 2 commands to the foreground with **CTRL+z**
8. Type **jobs**
9. Bring job 1 to the foreground. Type **fg 1**.
10. Finish foreground job with **CTRL+z** or **Q** to quit
11. Type **jobs**
12. How many **jobs** do you see?
13. Type **ps**.
14. What is the watch command process id ?
15. What are the other processes you see?
16. Run **ps -e | more** to see a complete list of processes, not only yours.
17. Run **ps -eH | more** to see the process hierarchy (forest).
18. Run **ps -elH | more** to also see the long format.
19. Use the **kill** command to kill the watch process.

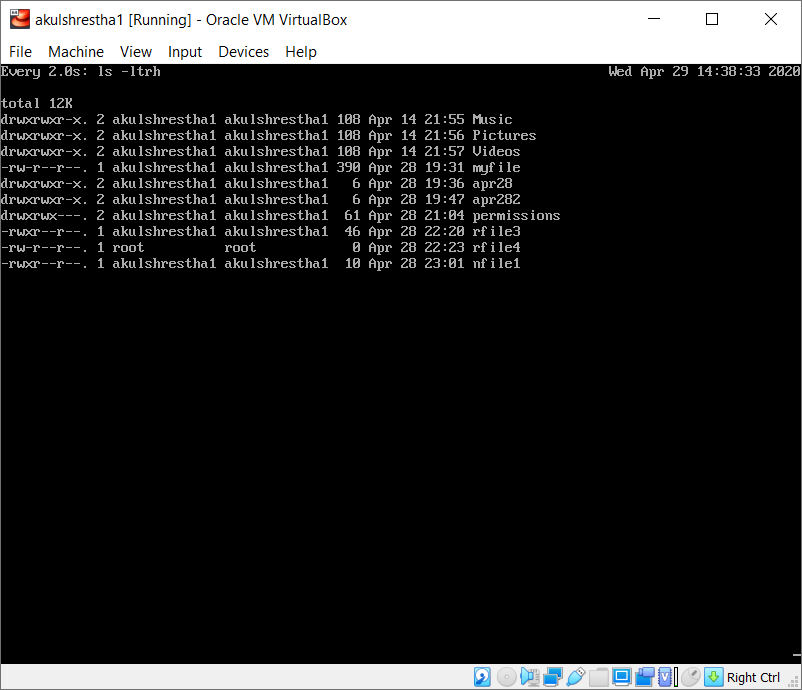
**--------------------------------------------------------------------------------------------------------------  
PART II ( Optional)**

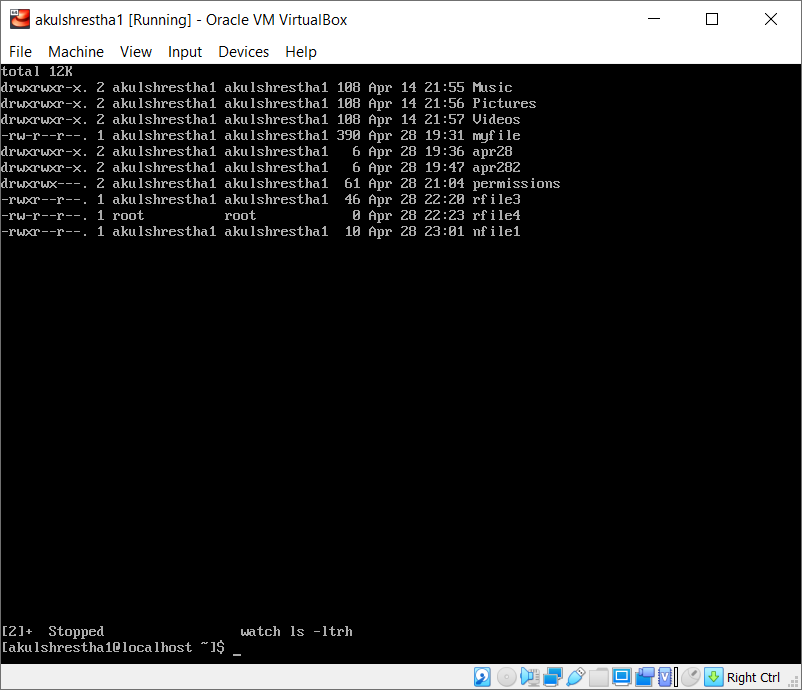
1. Open 2 putty terminals side by side and login as a regular user.
2. On **terminalA**, create a script called **pload1**, which will generate an artificial CPU load. Create the script in your home directory and make it executable.  
   #!/bin/bash  
   while true; do  
     var=1  
     while [[ var -lt 50000 ]]; do  
       var=$(($var+1))  
     done  
     sleep 1  
   done
3. On **terminalB**, run the top utility.
4. On terminalA, check the number of logical CPUs and run the **pload1** script in the background.  
   $grep "model name" /proc/cpuinfo | wc -l  
   $pload1 &
5. In the **terminalB**, observe the top display. Toggle between load, threads and memory. Note the process ID (PID) for **pload1**. View the CPU percentage. It should hover around 10% to 15%. Ensure that top is showing CPU usage once you have viewed load, threads, and memory.
6. Press shift+m, m, t, shift+p
7. Copy the **pload1** script to a new file called **pload2**. Edit the script to create more artificial CPU load. Increase the load from fifty thousand to one hundred thousand. Start the **pload2** process in the background.
8. Start the **pload2** process in the background.
9. Use the jobs command to confirm that both processes are running in the background.
10. In the **terminalB** shell, confirm that the process is running and using the most CPU resources. The load should be hovering between 25% and 35%.  
      
    Note: If you do not see **pload1** and **pload2** at the top of the process list, press Shift + p to ensure that top is sorted by CPU utilization.
11. The load average is still below 1. Copy **pload1** to a new script called pload3. Increase the addition count to eight hundred thousand. Start **pload3**in the background.
12. Confirm that the load average is above 1. It may take a few minutes for the load average to change.
13. In the **terminalA** shell, become root. Suspend the **pload1** process. List the remaining jobs. Observe that the process state for **pload1** is now **T**.  
    #pkill -SIGSTOP pload1
14. Resume the **pload1** process.  
    #pkill -SIGCONT pload1
15. Terminate **pload1**, **pload2**, and **pload3**using the command line. Confirm that the processes are no longer displayed in top.  
    #pkill pload1  
    #pkill pload2  
    #pkill pload3
16. In the **terminalB** shell, confirm that the processes no longer appear in**top**.

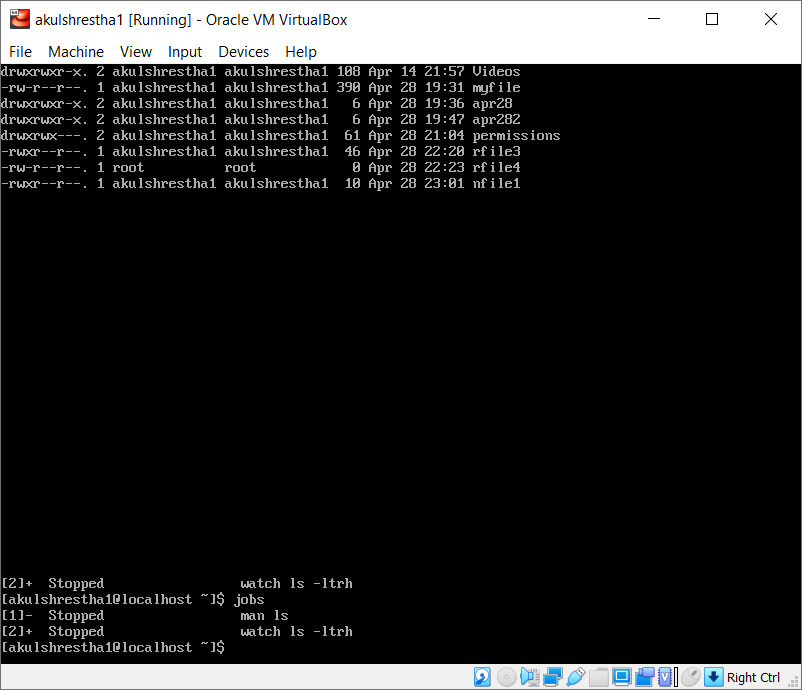


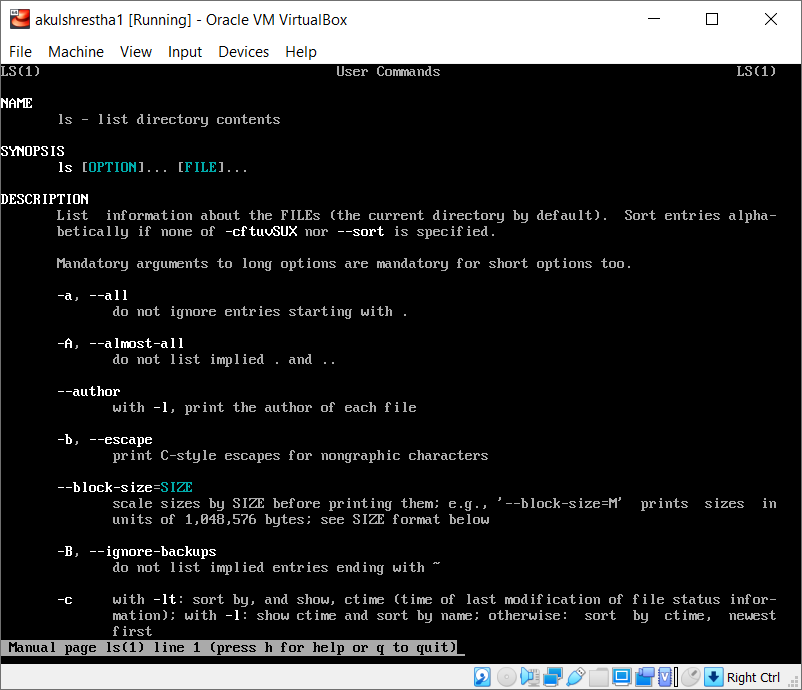


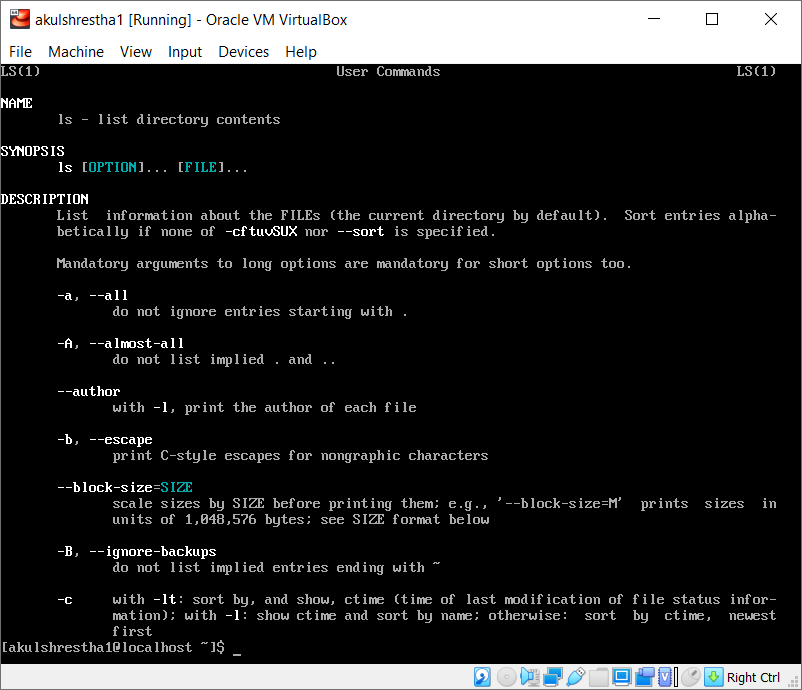


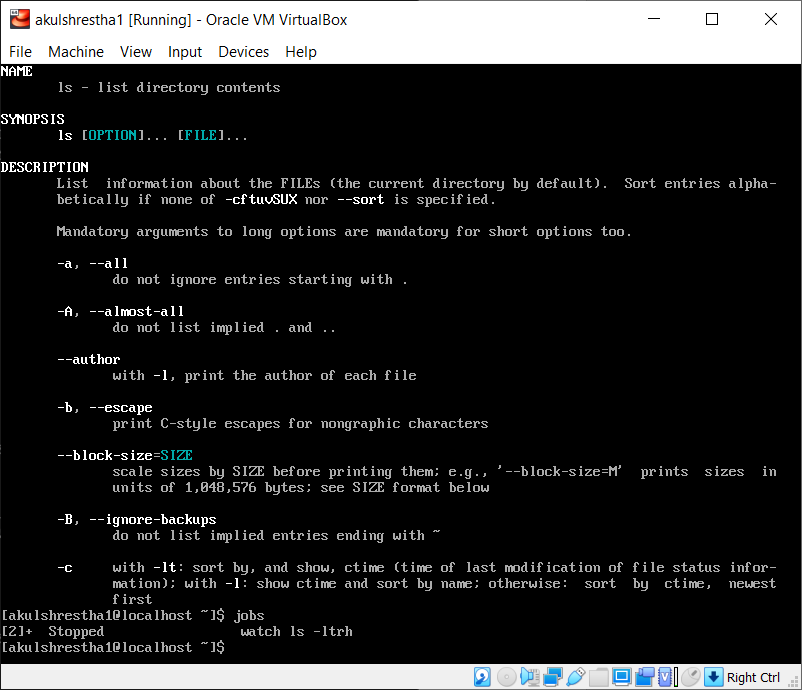




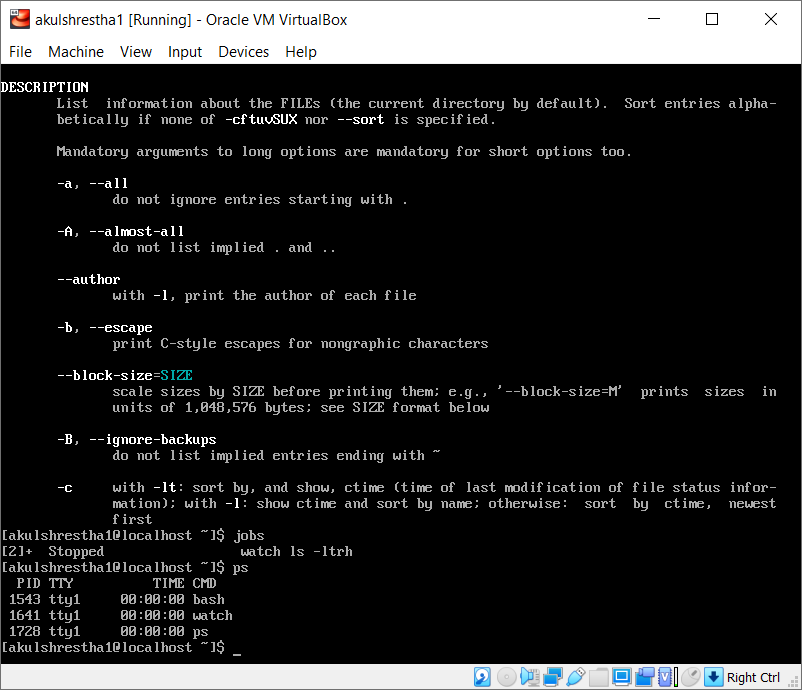






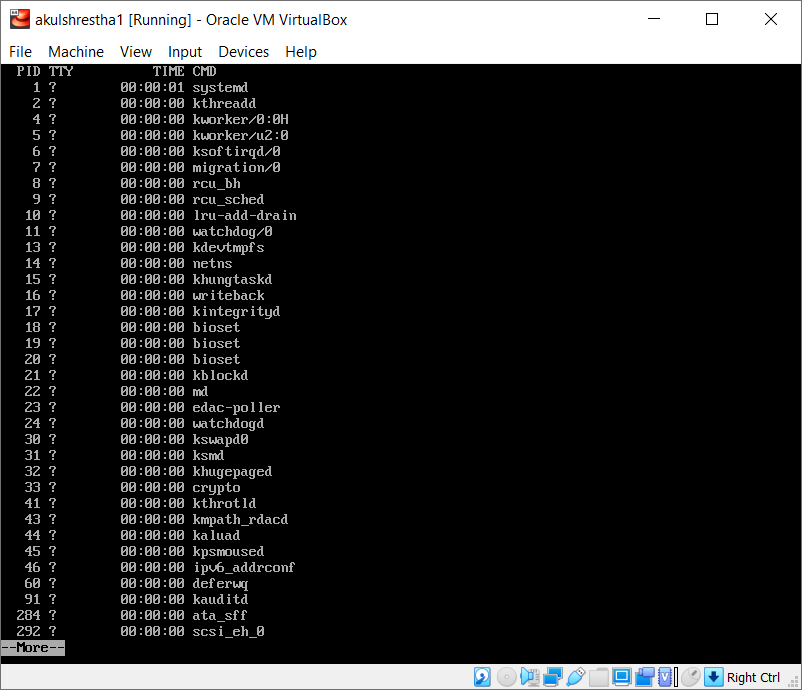


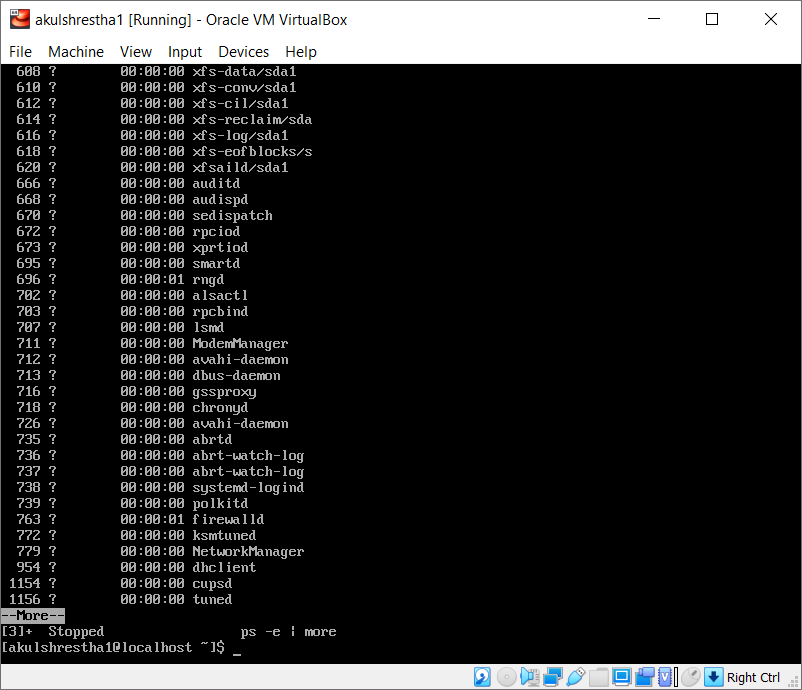
How many Jobs do you see? 1 job



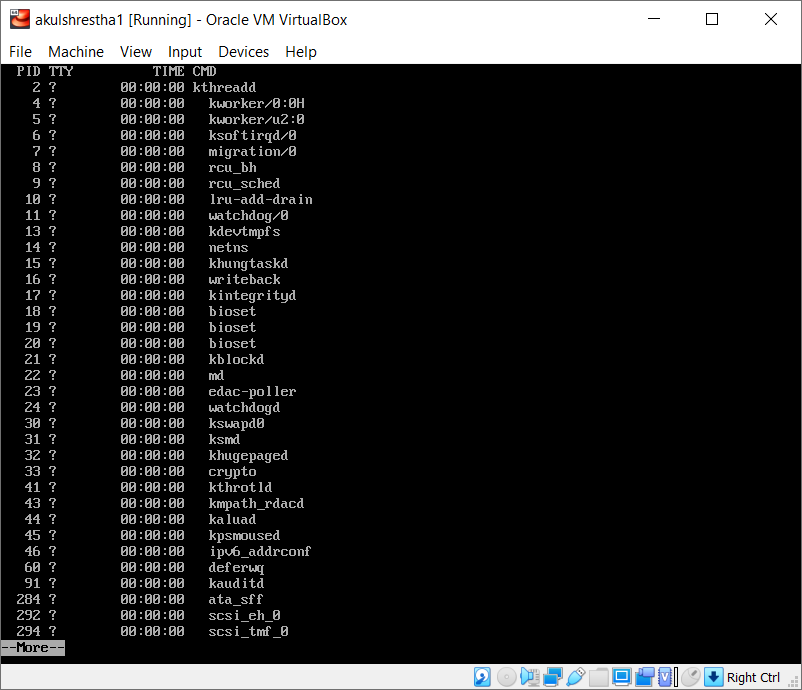
What is the watch command process id? 1641

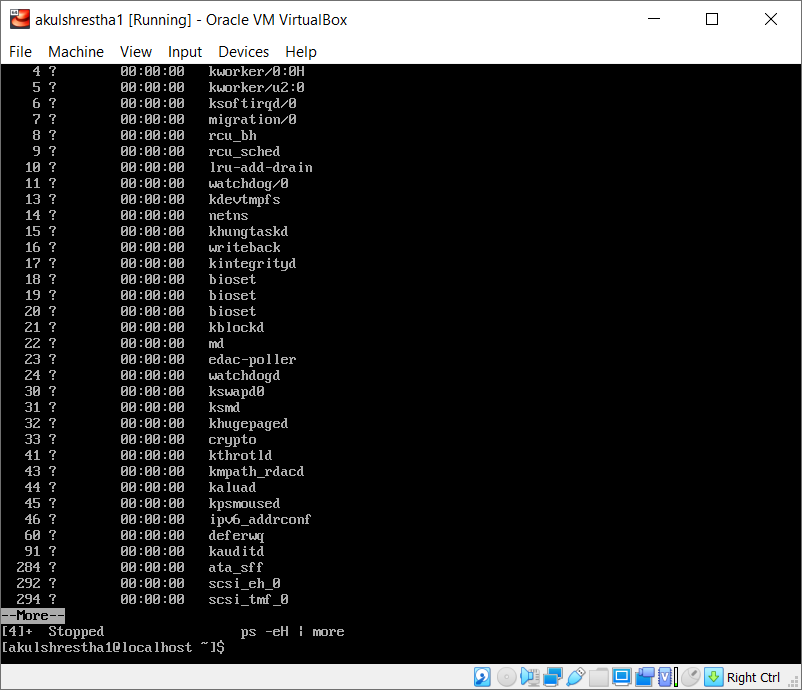
What are the other processes you see? bash, and ps

Run **ps -e | more** to see a complete list of processes, not only yours.



Run **ps -eH | more** to see the process hierarchy (forest)





Run **ps -elH | more** to also see the long format.

