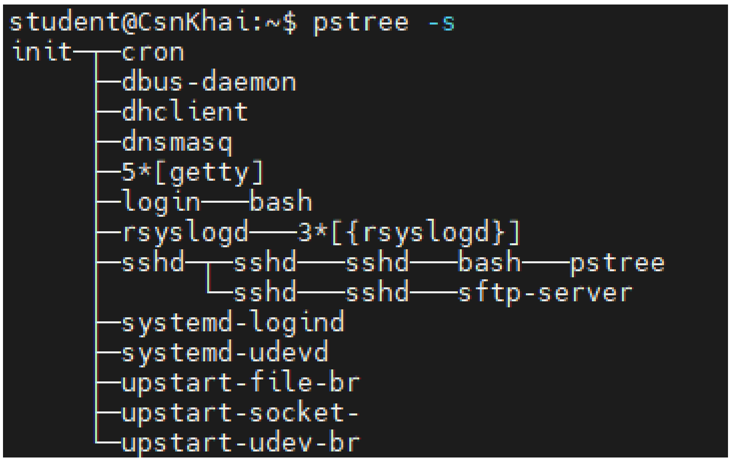
TASK 3. Part1

1. How many states could has a process in Linux?

There are five Linux process states. They are as follows: running & runnable, interruptable_sleep, uninterruptable_sleep, stopped, and zombie.

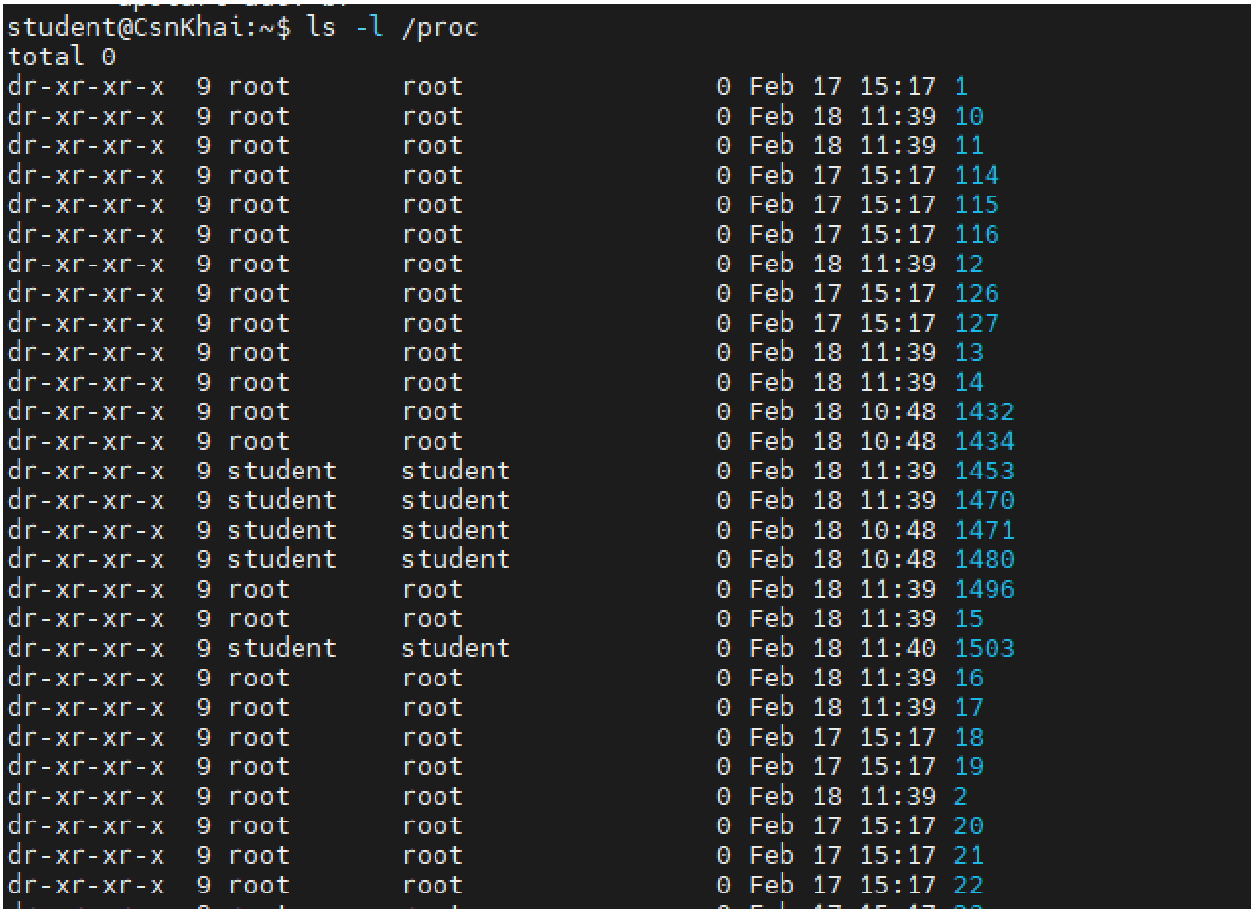
2. Examine the pstree command. Make output (highlight) the chain (ancestors) of the current process.



```
student@CsnKhai:~$ pstree -s
init─┬─cron
     ├─dbus-daemon
     ├─dhclient
     ├─dnsmasq
     ├─5*[getty]
     ├─login───bash
     ├─rsyslogd───3*[{rsyslogd}]
     ├─sshd─┬─sshd───sshd───bash───pstree
     │      └─sshd───sshd───sftp-server
     ├─systemd-logind
     ├─systemd-udevd
     ├─upstart-file-br
     ├─upstart-socket-
     └─upstart-udev-br
```

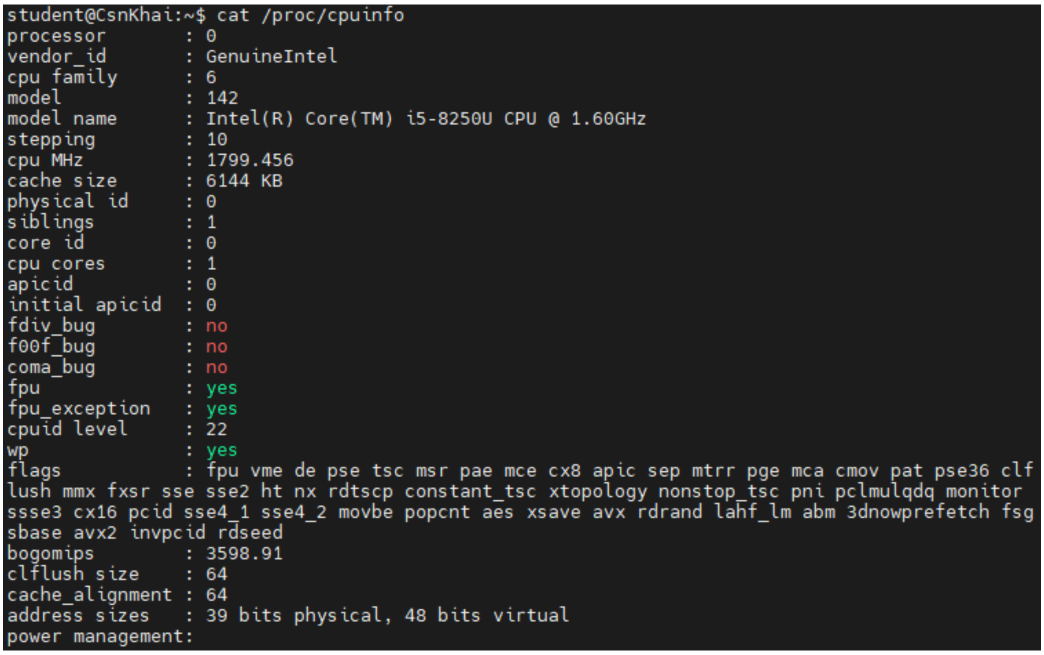
2. What is a proc file system?

Proc file system (procfs) is virtual file system created on fly when system boots and is dissolved at time of system shut down.



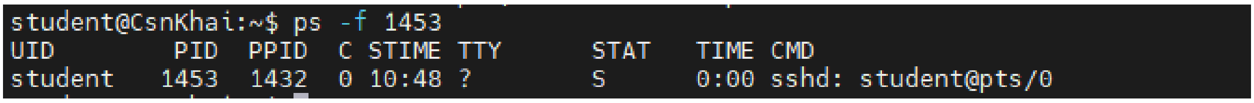
```
student@CsnKhai:~$ ls -l /proc
total 0
dr-xr-xr-x  9 root       root              0 Feb 17 15:17 1
dr-xr-xr-x  9 root       root              0 Feb 18 11:39 10
dr-xr-xr-x  9 root       root              0 Feb 18 11:39 11
dr-xr-xr-x  9 root       root              0 Feb 17 15:17 114
dr-xr-xr-x  9 root       root              0 Feb 17 15:17 115
dr-xr-xr-x  9 root       root              0 Feb 17 15:17 116
dr-xr-xr-x  9 root       root              0 Feb 18 11:39 12
dr-xr-xr-x  9 root       root              0 Feb 17 15:17 126
dr-xr-xr-x  9 root       root              0 Feb 17 15:17 127
dr-xr-xr-x  9 root       root              0 Feb 18 11:39 13
dr-xr-xr-x  9 root       root              0 Feb 18 11:39 14
dr-xr-xr-x  9 root       root              0 Feb 18 10:48 1432
dr-xr-xr-x  9 root       root              0 Feb 18 10:48 1434
dr-xr-xr-x  9 student    student           0 Feb 18 11:39 1453
dr-xr-xr-x  9 student    student           0 Feb 18 11:39 1470
dr-xr-xr-x  9 student    student           0 Feb 18 10:48 1471
dr-xr-xr-x  9 student    student           0 Feb 18 10:48 1480
dr-xr-xr-x  9 root       root              0 Feb 18 11:39 1496
dr-xr-xr-x  9 root       root              0 Feb 18 11:39 15
dr-xr-xr-x  9 student    student           0 Feb 18 11:40 1503
dr-xr-xr-x  9 root       root              0 Feb 18 11:39 16
dr-xr-xr-x  9 root       root              0 Feb 18 11:39 17
dr-xr-xr-x  9 root       root              0 Feb 17 15:17 18
dr-xr-xr-x  9 root       root              0 Feb 17 15:17 19
dr-xr-xr-x  9 root       root              0 Feb 18 11:39 2
dr-xr-xr-x  9 root       root              0 Feb 17 15:17 20
dr-xr-xr-x  9 root       root              0 Feb 17 15:17 21
dr-xr-xr-x  9 root       root              0 Feb 17 15:17 22
```

4. **Print information about the processor (its type, supported technologies, etc.).**



```
student@CsnKhai:~$ cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 142
model name      : Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz
stepping        : 10
cpu MHz         : 1799.456
cache size      : 6144 KB
physical id     : 0
siblings        : 1
core id         : 0
cpu cores       : 1
apicid          : 0
initial apicid  : 0
fdiv_bug        : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 22
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clf
lush mmx fxsr sse sse2 ht nx rdtscp constant_tsc xtopology nonstop_tsc pni pclmulqdq monitor
ssse3 cx16 pcid sse4_1 sse4_2 movbe popcnt aes xsave avx rdrand lahf_lm abm 3dnowprefetch fsg
sbase avx2 invpcid rdseed
bogomips        : 3598.91
clflush size    : 64
cache_alignment : 64
address sizes   : 39 bits physical, 48 bits virtual
power management:
```

5. Use the ps command to get information about the process. The information should be as follows: the owner of the process, the arguments with which the process was launched for execution, the group owner of this process, etc.



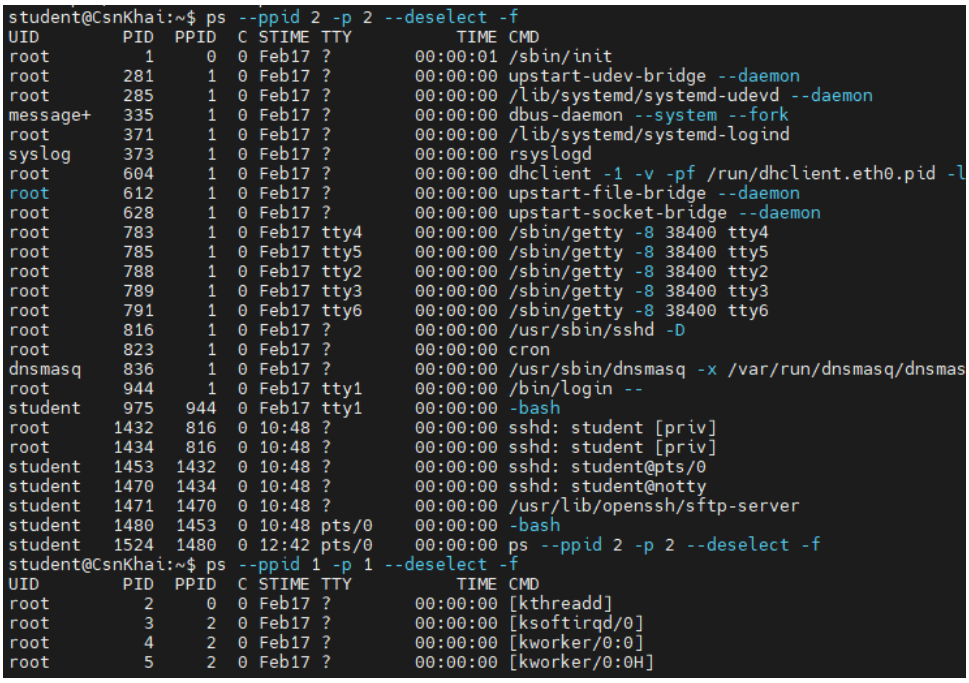
```
student@CsnKhai:~$ ps -f 1453
UID        PID  PPID  C STIME TTY      STAT   TIME CMD
student   1453  1432  0 10:48 ?        S      0:00 sshd: student@pts/0
```

6. How to define kernel processes and user processes?

User-space processes have its own virtual address space.

Kernel processes or threads do not have their own address space, they operate within kernel address space only. And they may be started before the kernel has started any user process (e.g. init).

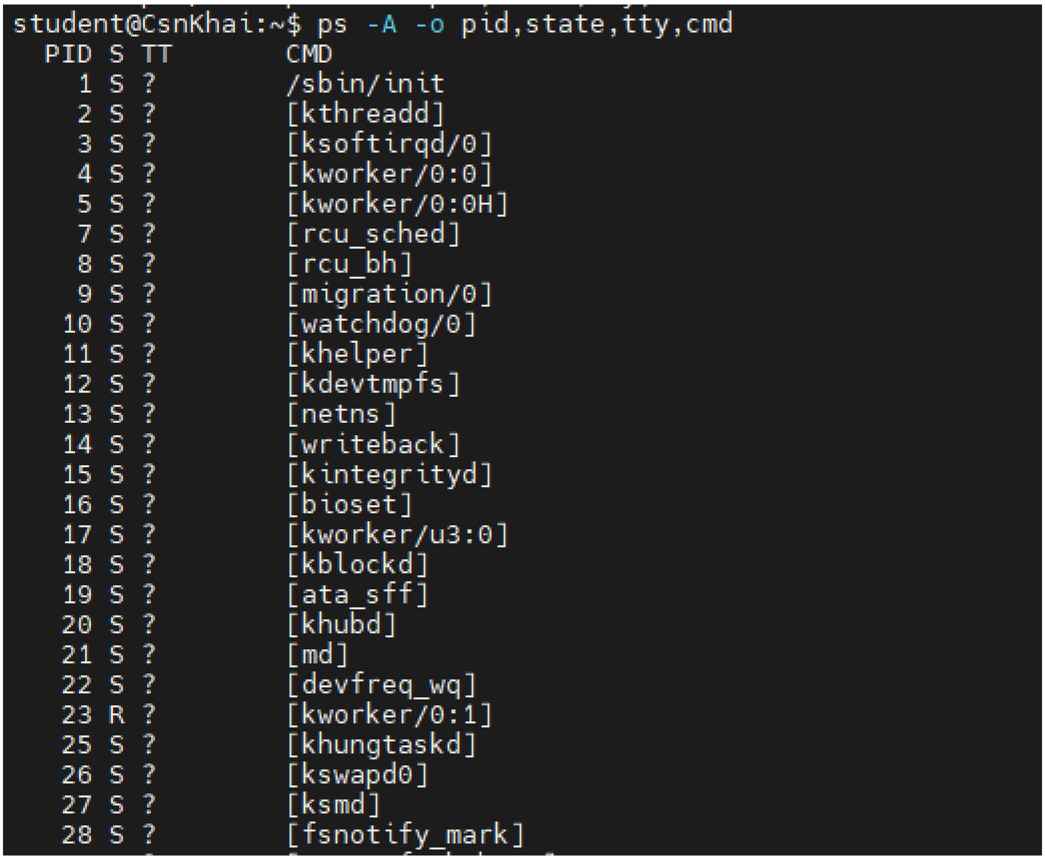
Kernel threads run only in Kernel Mode, while regular processes run alterna- tively in Kernel Mode and in User Mode.



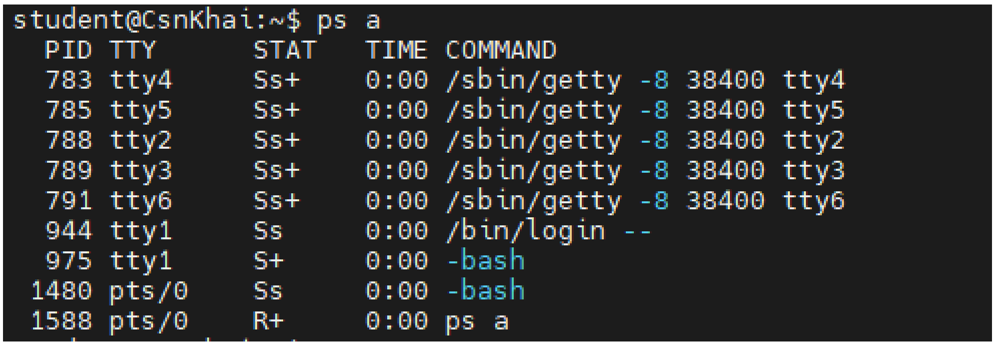
```
student@CsnKhai:~$ ps --ppid 2 -p 2 --deselect -f
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 Feb17 ?        00:00:01 /sbin/init
root       281     1  0 Feb17 ?        00:00:00 upstart-udev-bridge --daemon
root       285     1  0 Feb17 ?        00:00:00 /lib/systemd/systemd-udevd --daemon
message+   335     1  0 Feb17 ?        00:00:00 dbus-daemon --system --fork
root       371     1  0 Feb17 ?        00:00:00 /lib/systemd/systemd-logind
syslog     373     1  0 Feb17 ?        00:00:00 rsyslogd
root       604     1  0 Feb17 ?        00:00:00 dhclient -1 -v -pf /run/dhclient.eth0.pid -l
root       612     1  0 Feb17 ?        00:00:00 upstart-file-bridge --daemon
root       628     1  0 Feb17 ?        00:00:00 upstart-socket-bridge --daemon
root       783     1  0 Feb17 tty4     00:00:00 /sbin/getty -8 38400 tty4
root       785     1  0 Feb17 tty5     00:00:00 /sbin/getty -8 38400 tty5
root       788     1  0 Feb17 tty2     00:00:00 /sbin/getty -8 38400 tty2
root       789     1  0 Feb17 tty3     00:00:00 /sbin/getty -8 38400 tty3
root       791     1  0 Feb17 tty6     00:00:00 /sbin/getty -8 38400 tty6
root       816     1  0 Feb17 ?        00:00:00 /usr/sbin/sshd -D
root       823     1  0 Feb17 ?        00:00:00 cron
dnsmasq    836     1  0 Feb17 ?        00:00:00 /usr/sbin/dnsmasq -x /var/run/dnsmasq/dnsmas
root       944     1  0 Feb17 tty1     00:00:00 /bin/login --
student    975   944  0 Feb17 tty1     00:00:00 -bash
root      1432   816  0 10:48 ?        00:00:00 sshd: student [priv]
root      1434   816  0 10:48 ?        00:00:00 sshd: student [priv]
student   1453  1432  0 10:48 ?        00:00:00 sshd: student@pts/0
student   1470  1434  0 10:48 ?        00:00:00 sshd: student@notty
student   1471  1470  0 10:48 ?        00:00:00 /usr/lib/openssh/sftp-server
student   1480  1453  0 10:48 pts/0    00:00:00 -bash
student   1524  1480  0 12:42 pts/0    00:00:00 ps --ppid 2 -p 2 --deselect -f
student@CsnKhai:~$ ps --ppid 1 -p 1 --deselect -f
UID        PID  PPID  C STIME TTY          TIME CMD
root         2     0  0 Feb17 ?        00:00:00 [kthreadd]
root         3     2  0 Feb17 ?        00:00:00 [ksoftirqd/0]
root         4     2  0 Feb17 ?        00:00:00 [kworker/0:0]
root         5     2  0 Feb17 ?        00:00:00 [kworker/0:0H]
```

Kernel processes are with brackets ("[ ]").

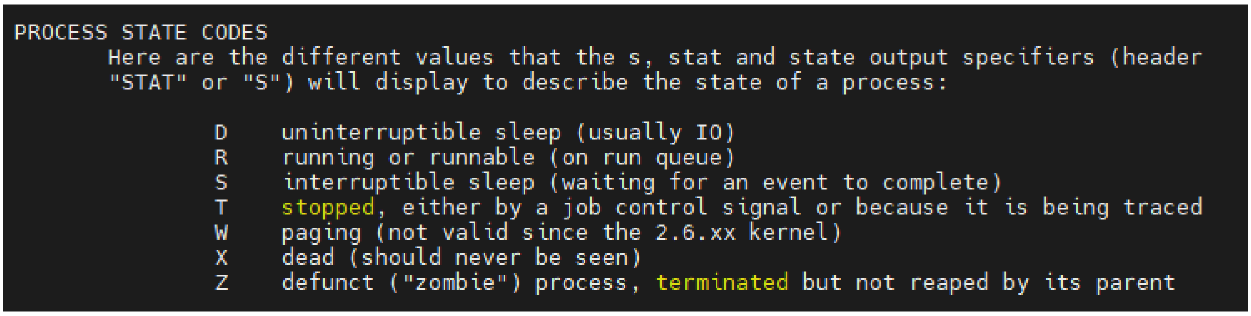
7. Print the list of processes to the terminal. Briefly describe the statuses of the processes. What condition are they in, or can they be arriving in?



```
student@CsnKhai:~$ ps -A -o pid,state,tty,cmd
  PID S TT       CMD
    1 S ?        /sbin/init
    2 S ?        [kthreadd]
    3 S ?        [ksoftirqd/0]
    4 S ?        [kworker/0:0]
    5 S ?        [kworker/0:0H]
    7 S ?        [rcu_sched]
    8 S ?        [rcu_bh]
    9 S ?        [migration/0]
   10 S ?        [watchdog/0]
   11 S ?        [khelper]
   12 S ?        [kdevtmpfs]
   13 S ?        [netns]
   14 S ?        [writeback]
   15 S ?        [kintegrityd]
   16 S ?        [bioset]
   17 S ?        [kworker/u3:0]
   18 S ?        [kblockd]
   19 S ?        [ata_sff]
   20 S ?        [khubd]
   21 S ?        [md]
   22 S ?        [devfreq_wq]
   23 R ?        [kworker/0:1]
   25 S ?        [khungtaskd]
   26 S ?        [kswapd0]
   27 S ?        [ksmd]
   28 S ?        [fsnotify_mark]
```

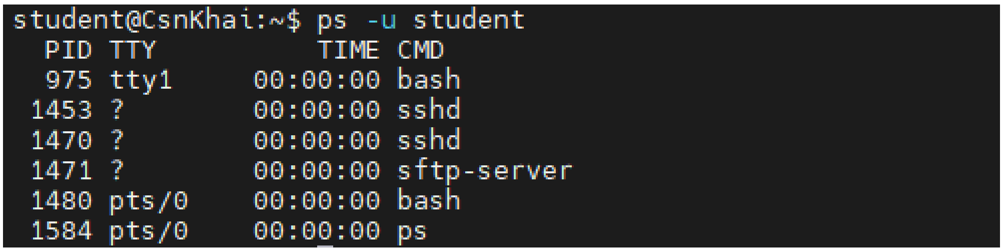


```
student@CsnKhai:~$ ps a
  PID TTY      STAT   TIME COMMAND
  783 tty4     Ss+    0:00 /sbin/getty -8 38400 tty4
  785 tty5     Ss+    0:00 /sbin/getty -8 38400 tty5
  788 tty2     Ss+    0:00 /sbin/getty -8 38400 tty2
  789 tty3     Ss+    0:00 /sbin/getty -8 38400 tty3
  791 tty6     Ss+    0:00 /sbin/getty -8 38400 tty6
  944 tty1     Ss     0:00 /bin/login --
  975 tty1     S+     0:00 -bash
 1480 pts/0    Ss     0:00 -bash
 1588 pts/0    R+     0:00 ps a
```



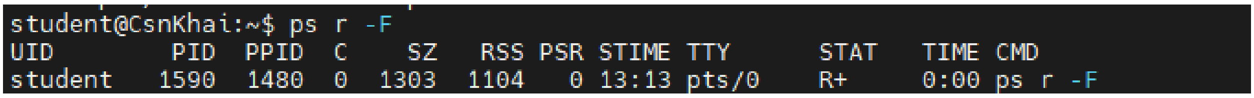
```
PROCESS STATE CODES
       Here are the different values that the s, stat and state output specifiers (header
       "STAT" or "S") will display to describe the state of a process:

               D    uninterruptible sleep (usually IO)
               R    running or runnable (on run queue)
               S    interruptible sleep (waiting for an event to complete)
               T    stopped, either by a job control signal or because it is being traced
               W    paging (not valid since the 2.6.xx kernel)
               X    dead (should never be seen)
               Z    defunct ("zombie") process, terminated but not reaped by its parent
```

8. Display only the processes of a specific user.



```
student@CsnKhai:~$ ps -u student
  PID TTY          TIME CMD
  975 tty1     00:00:00 bash
 1453 ?        00:00:00 sshd
 1470 ?        00:00:00 sshd
 1471 ?        00:00:00 sftp-server
 1480 pts/0    00:00:00 bash
 1584 pts/0    00:00:00 ps
```

9. What utilities can be used to analyze existing running tasks (by analyzing the help for the ps command)?

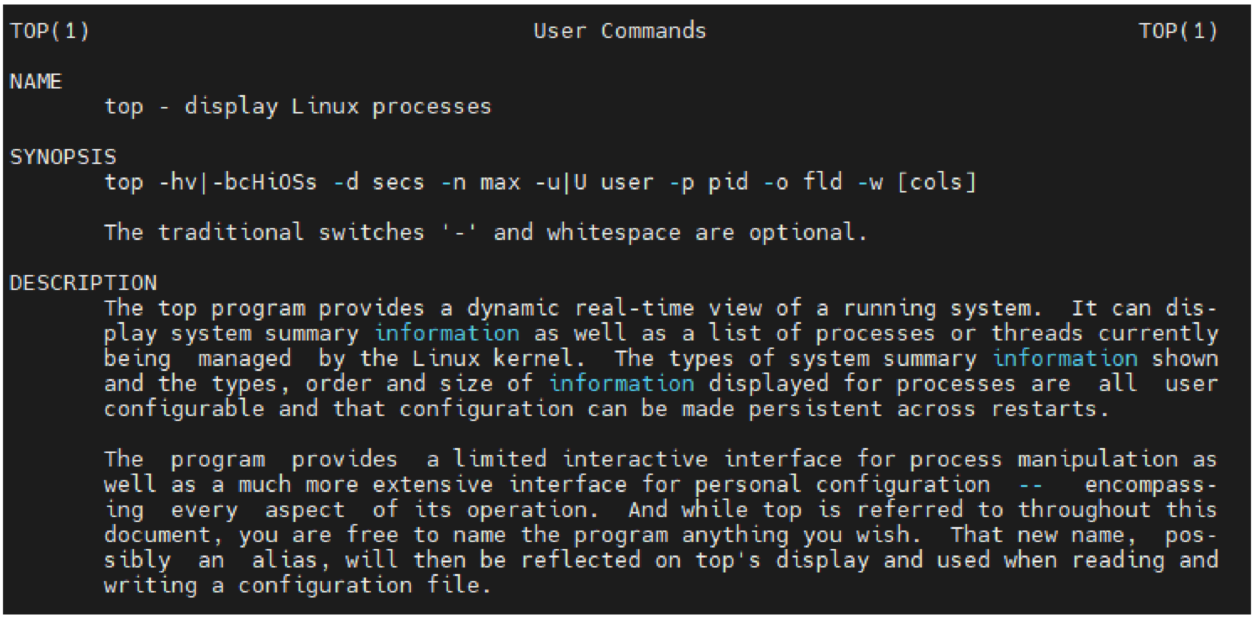


```
student@CsnKhai:~$ ps r -F
UID         PID  PPID  C    SZ   RSS PSR STIME TTY      STAT   TIME CMD
student    1590  1480  0  1303  1104   0 13:13 pts/0    R+     0:00 ps r -F
```

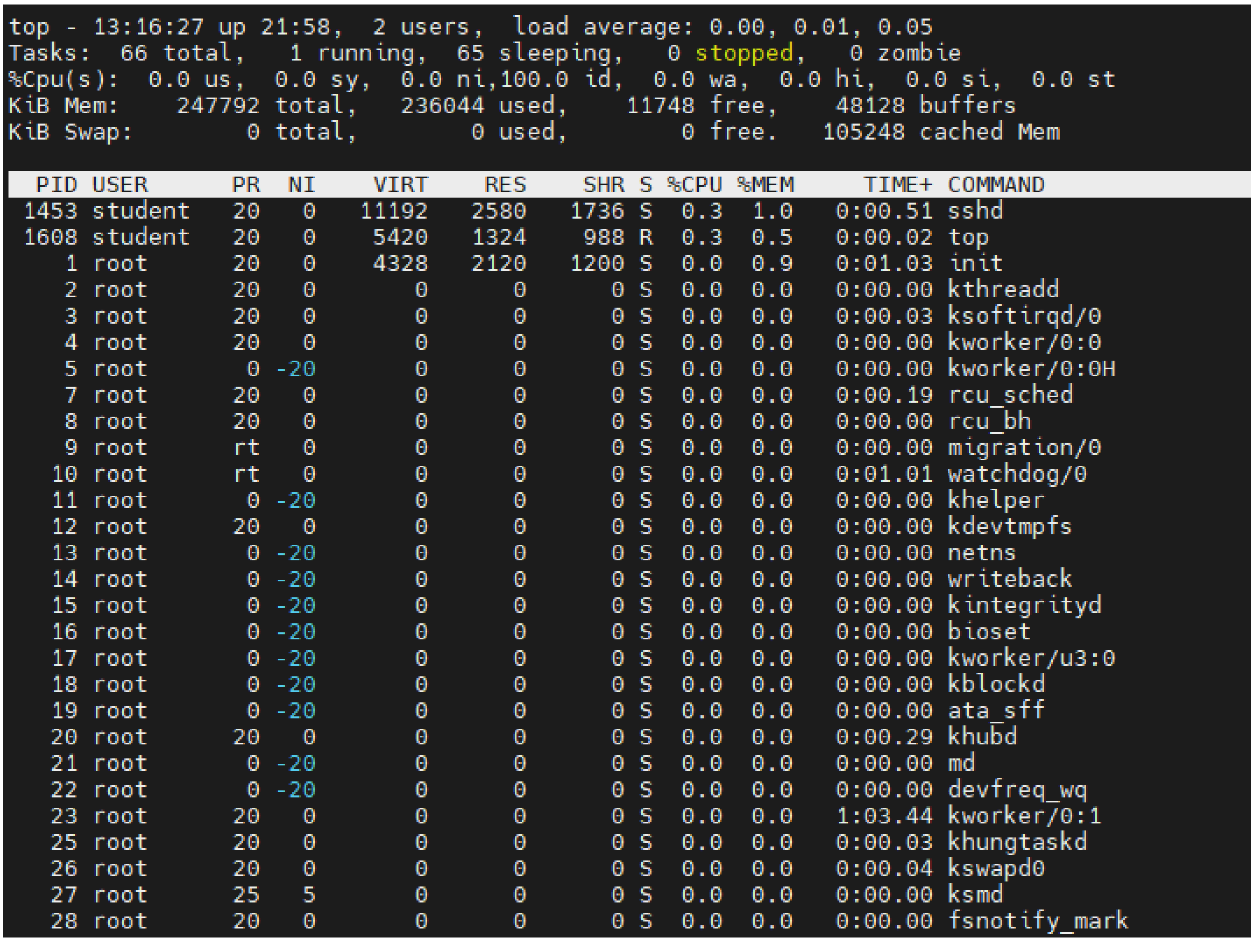
r – running processes

-F – full info

10. What information does top command display?



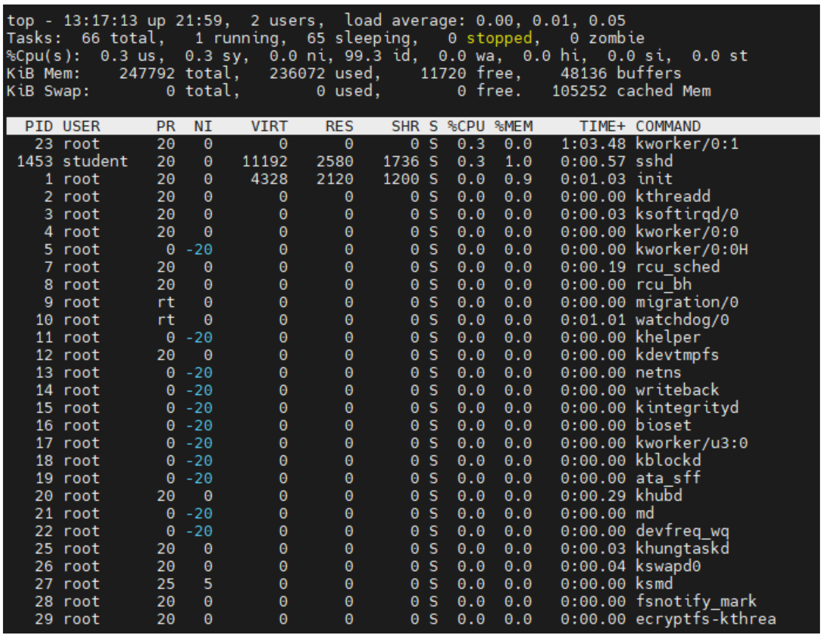
```
TOP(1)                          User Commands                         TOP(1)

NAME
       top - display Linux processes

SYNOPSIS
       top -hv|-bcHiOSs -d secs -n max -u|U user -p pid -o fld -w [cols]

       The traditional switches '-' and whitespace are optional.

DESCRIPTION
       The top program provides a dynamic real-time view of a running system.  It can dis-
       play system summary information as well as a list of processes or threads currently
       being  managed  by the Linux kernel.  The types of system summary information shown
       and the types, order and size of information displayed for processes are  all  user
       configurable and that configuration can be made persistent across restarts.

       The  program  provides  a limited interactive interface for process manipulation as
       well as a much more extensive interface for personal configuration  --   encompass-
       ing  every  aspect  of its operation.  And while top is referred to throughout this
       document, you are free to name the program anything you wish.  That new name,  pos-
       sibly  an  alias, will then be reflected on top's display and used when reading and
       writing a configuration file.
```



```
top - 13:16:27 up 21:58,  2 users,  load average: 0.00, 0.01, 0.05
Tasks:  66 total,   1 running,  65 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:    247792 total,   236044 used,    11748 free,    48128 buffers
KiB Swap:        0 total,        0 used,        0 free.   105248 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1453 student   20   0   11192   2580   1736 S  0.3  1.0   0:00.51 sshd
 1608 student   20   0    5420   1324    988 R  0.3  0.5   0:00.02 top
    1 root      20   0    4328   2120   1200 S  0.0  0.9   0:01.03 init
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0       0      0      0 S  0.0  0.0   0:00.03 ksoftirqd/0
    4 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kworker/0:0
    5 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kworker/0:0H
    7 root      20   0       0      0      0 S  0.0  0.0   0:00.19 rcu_sched
    8 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcu_bh
    9 root      rt   0       0      0      0 S  0.0  0.0   0:00.00 migration/0
   10 root      rt   0       0      0      0 S  0.0  0.0   0:01.01 watchdog/0
   11 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 khelper
   12 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kdevtmpfs
   13 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 netns
   14 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 writeback
   15 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kintegrityd
   16 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 bioset
   17 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kworker/u3:0
   18 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kblockd
   19 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 ata_sff
   20 root      20   0       0      0      0 S  0.0  0.0   0:00.29 khubd
   21 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 md
   22 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 devfreq_wq
   23 root      20   0       0      0      0 S  0.0  0.0   1:03.44 kworker/0:1
   25 root      20   0       0      0      0 S  0.0  0.0   0:00.03 khungtaskd
   26 root      20   0       0      0      0 S  0.0  0.0   0:00.04 kswapd0
   27 root      25   5       0      0      0 S  0.0  0.0   0:00.00 ksmd
   28 root      20   0       0      0      0 S  0.0  0.0   0:00.00 fsnotify_mark
```

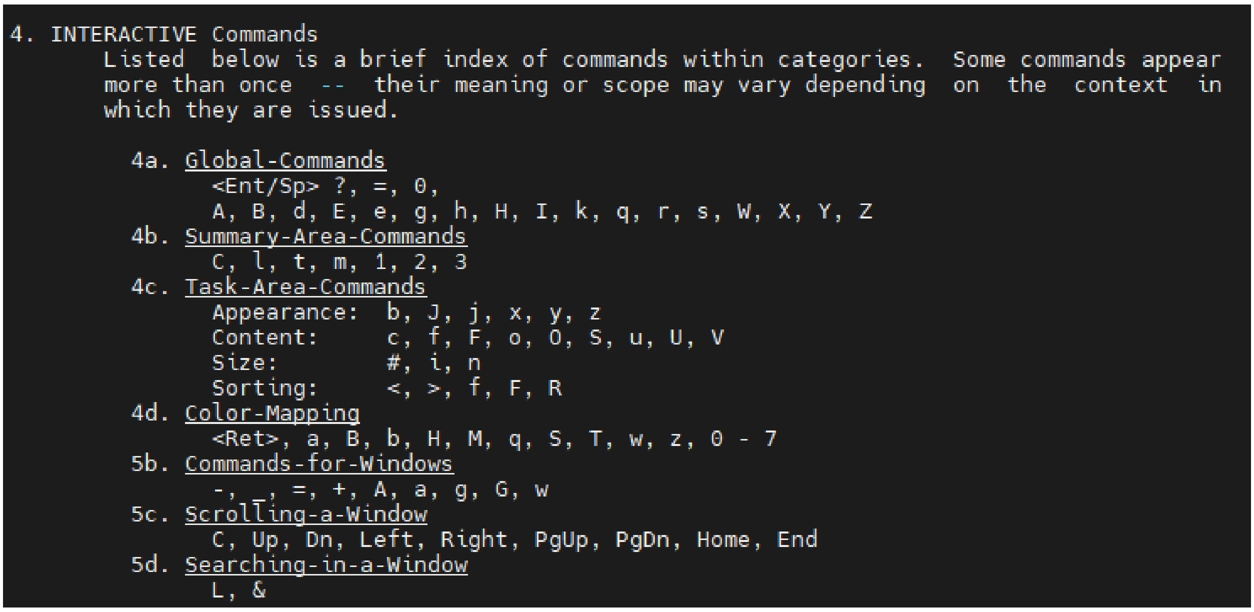
11. Display the processes of the specific user using the top command.

```
top –U root
```



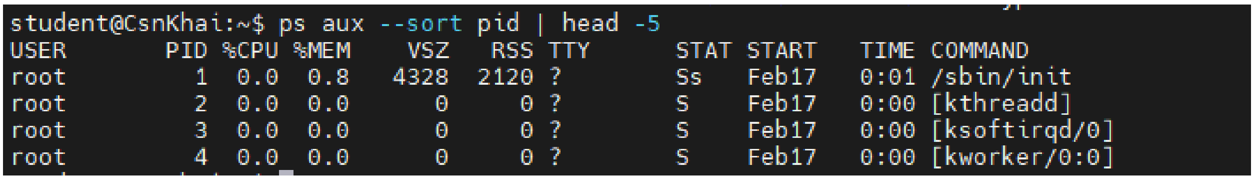
```
top - 13:17:13 up 21:59,  2 users,  load average: 0.00, 0.01, 0.05
Tasks:  66 total,   1 running,  65 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.3 us,  0.3 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:    247792 total,   236072 used,    11720 free,    48136 buffers
KiB Swap:        0 total,        0 used,        0 free.   105252 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
   23 root      20   0       0      0      0 S  0.3  0.0   1:03.48 kworker/0:1
 1453 student   20   0   11192   2580   1736 S  0.3  1.0   0:00.57 sshd
    1 root      20   0    4328   2120   1200 S  0.0  0.9   0:01.03 init
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0       0      0      0 S  0.0  0.0   0:00.03 ksoftirqd/0
    4 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kworker/0:0
    5 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kworker/0:0H
    7 root      20   0       0      0      0 S  0.0  0.0   0:00.19 rcu_sched
    8 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcu_bh
    9 root      rt   0       0      0      0 S  0.0  0.0   0:00.00 migration/0
   10 root      rt   0       0      0      0 S  0.0  0.0   0:01.01 watchdog/0
   11 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 khelper
   12 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kdevtmpfs
   13 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 netns
   14 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 writeback
   15 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kintegrityd
   16 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 bioset
   17 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kworker/u3:0
   18 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kblockd
   19 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 ata_sff
   20 root      20   0       0      0      0 S  0.0  0.0   0:00.29 khubd
   21 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 md
   22 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 devfreq_wq
   25 root      20   0       0      0      0 S  0.0  0.0   0:00.03 khungtaskd
   26 root      20   0       0      0      0 S  0.0  0.0   0:00.04 kswapd0
   27 root      25   5       0      0      0 S  0.0  0.0   0:00.00 ksmd
   28 root      20   0       0      0      0 S  0.0  0.0   0:00.00 fsnotify_mark
   29 root      20   0       0      0      0 S  0.0  0.0   0:00.00 ecryptfs-kthrea
```

12. **What interactive commands can be used to control the top command? Give a couple of examples.**

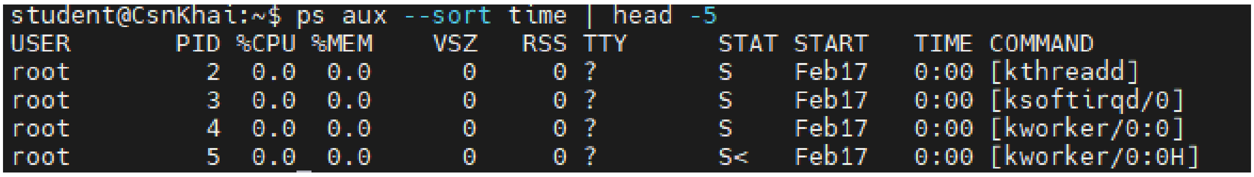


```
4. INTERACTIVE Commands
       Listed  below is a brief index of commands within categories.  Some commands appear
       more than once  --  their meaning or scope may vary depending  on  the  context  in
       which they are issued.

         4a. Global-Commands
               <Ent/Sp> ?, =, 0,
               A, B, d, E, e, g, h, H, I, k, q, r, s, W, X, Y, Z
         4b. Summary-Area-Commands
               C, l, t, m, 1, 2, 3
         4c. Task-Area-Commands
               Appearance:  b, J, j, x, y, z
               Content:     c, f, F, o, O, S, u, U, V
               Size:        #, i, n
               Sorting:     <, >, f, F, R
         4d. Color-Mapping
               <Ret>, a, B, b, H, M, q, S, T, w, z, 0 - 7
         5b. Commands-for-Windows
               -, _, =, +, A, a, g, G, w
         5c. Scrolling-a-Window
               C, Up, Dn, Left, Right, PgUp, PgDn, Home, End
         5d. Searching-in-a-Window
               L, &
```

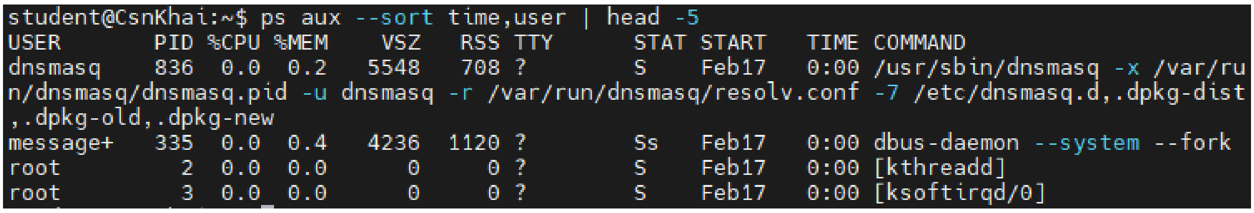
13. **Sort the contents of the processes window using various parameters (for example, the amount of processor time taken up, etc.)**



```
student@CsnKhai:~$ ps aux --sort pid | head -5
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.8   4328  2120 ?        Ss   Feb17   0:01 /sbin/init
root         2  0.0  0.0      0     0 ?        S    Feb17   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    Feb17   0:00 [ksoftirqd/0]
root         4  0.0  0.0      0     0 ?        S    Feb17   0:00 [kworker/0:0]
```



```
student@CsnKhai:~$ ps aux --sort time | head -5
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         2  0.0  0.0      0     0 ?        S    Feb17   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    Feb17   0:00 [ksoftirqd/0]
root         4  0.0  0.0      0     0 ?        S    Feb17   0:00 [kworker/0:0]
root         5  0.0  0.0      0     0 ?        S<   Feb17   0:00 [kworker/0:0H]
```



```
student@CsnKhai:~$ ps aux --sort time,user | head -5
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
dnsmasq    836  0.0  0.2   5548   708 ?        S    Feb17   0:00 /usr/sbin/dnsmasq -x /var/ru
n/dnsmasq/dnsmasq.pid -u dnsmasq -r /var/run/dnsmasq/resolv.conf -7 /etc/dnsmasq.d,.dpkg-dist
,.dpkg-old,.dpkg-new
message+   335  0.0  0.4   4236  1120 ?        Ss   Feb17   0:00 dbus-daemon --system --fork
root         2  0.0  0.0      0     0 ?        S    Feb17   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    Feb17   0:00 [ksoftirqd/0]
```

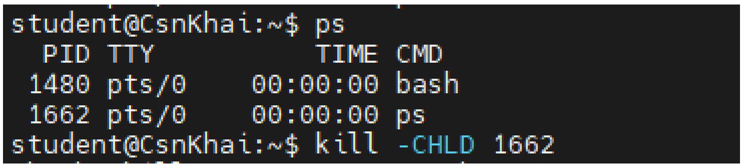
14. **Concept of priority, what commands are used to set priority?**

nice / renice

15. Can I change the priority of a process using the top command? If so, how?

Once given top command, press r. Give PID value of the process you want to change the process value. Give renice value (from -20 to +19)

16. Examine the kill command. How to send with the kill command process control signal? Give an example of commonly used signals.



```
student@CsnKhai:~$ ps
  PID TTY          TIME CMD
 1480 pts/0    00:00:00 bash
 1662 pts/0    00:00:00 ps
student@CsnKhai:~$ kill -CHLD 1662
```

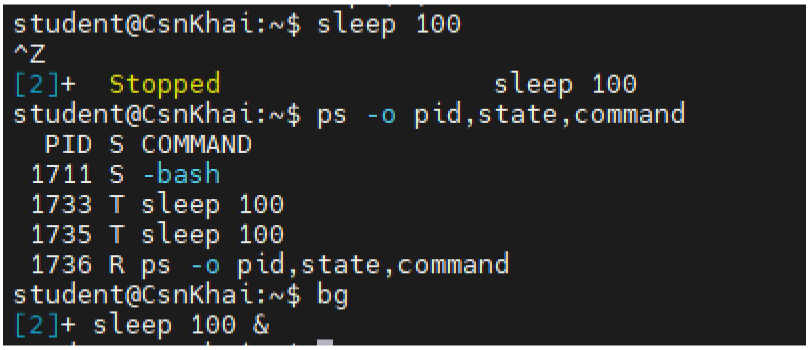
17. Commands jobs, fg, bg, nohup. What are they for? Use the sleep, yes command to demonstrate the process control mechanism with fg, bg.

Jobs is for listing background processes.

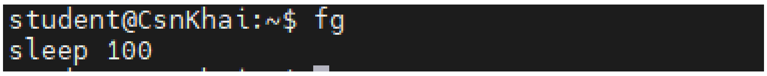
Fg is for moving background processes.

Bg is for moving foreground processes to the background

If you know when starting the process that you will want to close the terminal before the process completes, you can start it using the nohup command.



```
student@CsnKhai:~$ sleep 100
^Z
[2]+  Stopped                 sleep 100
student@CsnKhai:~$ ps -o pid,state,command
  PID S COMMAND
 1711 S -bash
 1733 T sleep 100
 1735 T sleep 100
 1736 R ps -o pid,state,command
student@CsnKhai:~$ bg
[2]+ sleep 100 &
```



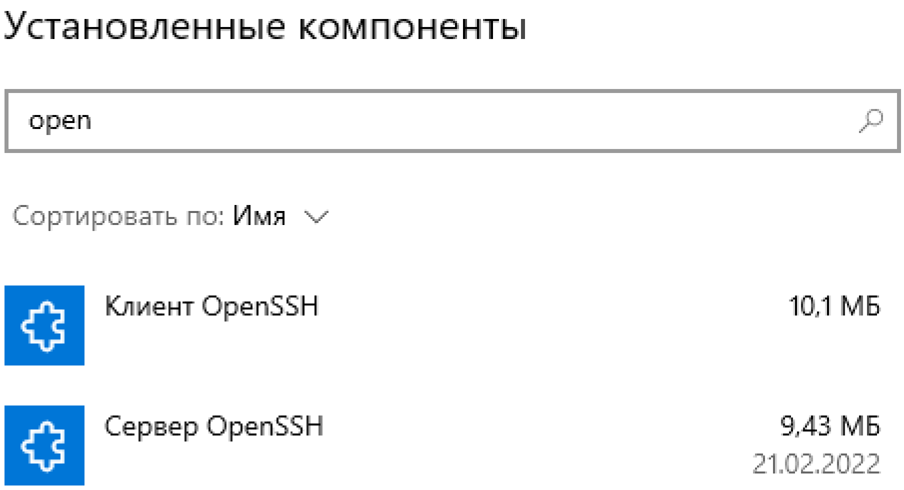
```
student@CsnKhai:~$ fg
sleep 100
```

*can’t make a screenshot of yes command with bg/fg since the output is too large.

Part2

1. Check the implementability of the most frequently used OPENSSH commands in the MS Windows operating system. (Description of the expected result of the commands + screenshots: command – result should be presented)

Download OpenSSH:



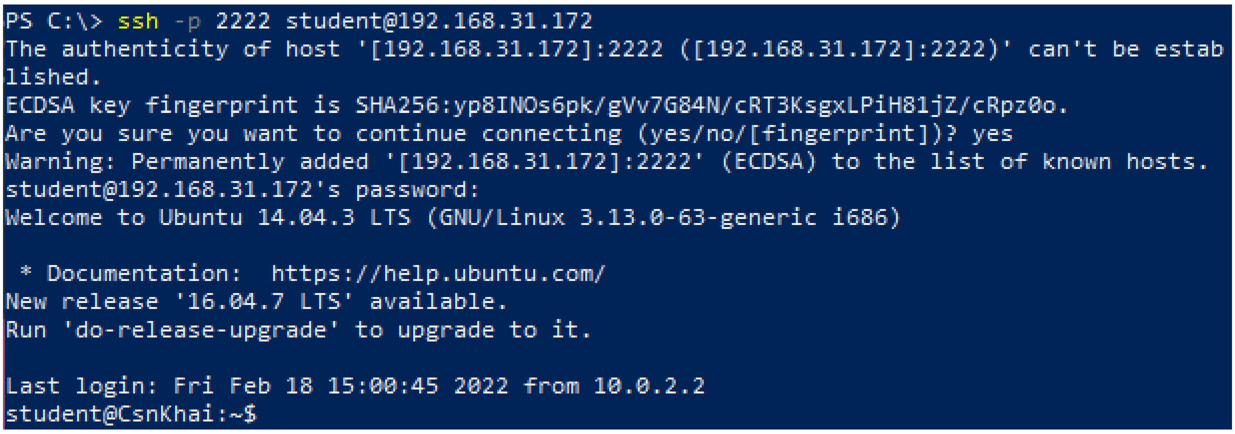
Установленные компоненты

open

Сортировать по: Имя

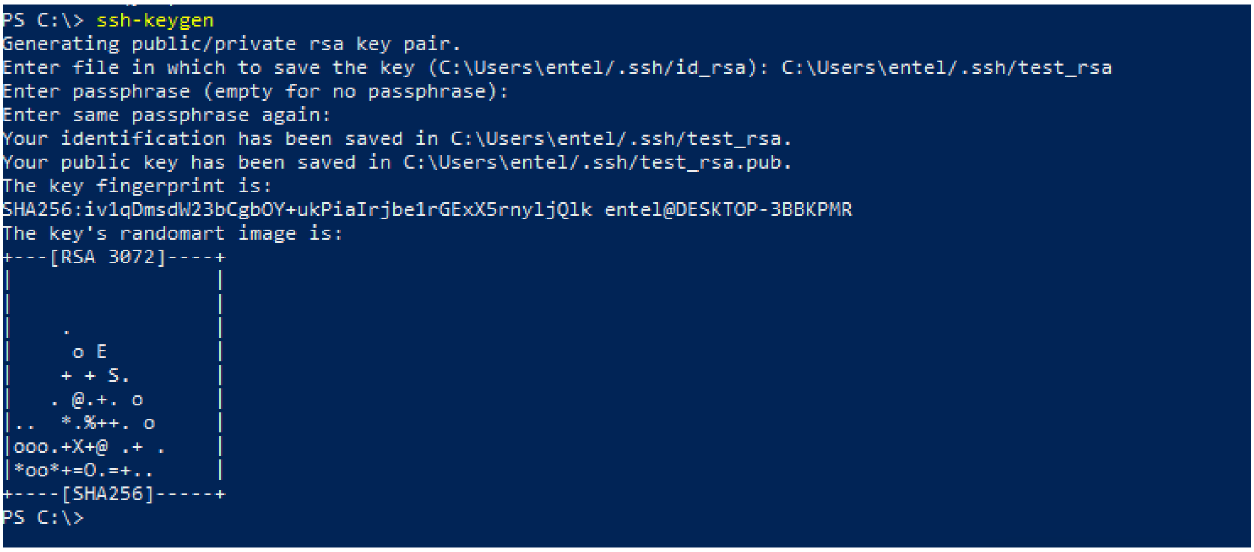
Клиент OpenSSH	10,1 МБ
Сервер OpenSSH	9,43 МБ 21.02.2022

Connect to machine:



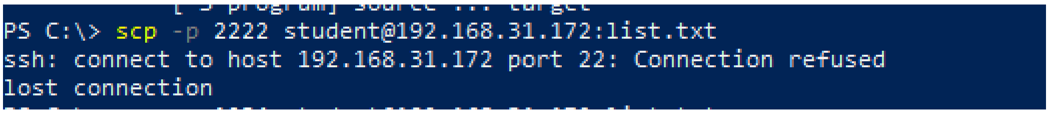
```
PS C:\> ssh -p 2222 student@192.168.31.172
The authenticity of host '[192.168.31.172]:2222 ([192.168.31.172]:2222)' can't be estab
lished.
ECDSA key fingerprint is SHA256:yp8INOs6pk/gVv7G84N/cRT3KsgxLPiH81jZ/cRpz0o.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.31.172]:2222' (ECDSA) to the list of known hosts.
student@192.168.31.172's password:
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.13.0-63-generic i686)

 * Documentation:  https://help.ubuntu.com/
New release '16.04.7 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Feb 18 15:00:45 2022 from 10.0.2.2
student@CsnKhai:~$
```

ssh-keygen:



```
PS C:\> ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\entel/.ssh/id_rsa): C:\Users\entel/.ssh/test_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\entel/.ssh/test_rsa.
Your public key has been saved in C:\Users\entel/.ssh/test_rsa.pub.
The key fingerprint is:
SHA256:iv1qDmsdW23bCgbOY+ukPiaIrjbe1rGExX5rny1jQ1k entel@DESKTOP-3BBKPMR
The key's randomart image is:
+---[RSA 3072]----+
|                 |
|                 |
|     .           |
|      o E        |
|     + + S.      |
|    . @.+. o     |
|..   *.%++. o    |
|ooo.+X+@ .+ .    |
|*oo*+=O.=+..     |
+----[SHA256]-----+
PS C:\>
```

-p option doesn’t help with scp command:



```
             [-J program] source ... target
PS C:\> scp -p 2222 student@192.168.31.172:list.txt
ssh: connect to host 192.168.31.172 port 22: Connection refused
lost connection
```

2. Implement basic SSH settings to increase the security of the client-server connection

1st – we made port forwarding which creates a secure connection between a local computer and a remote machine through which services can be relayed.

By default, all systems user can login via SSH using their password or public key.

We add the following to sshd_config:

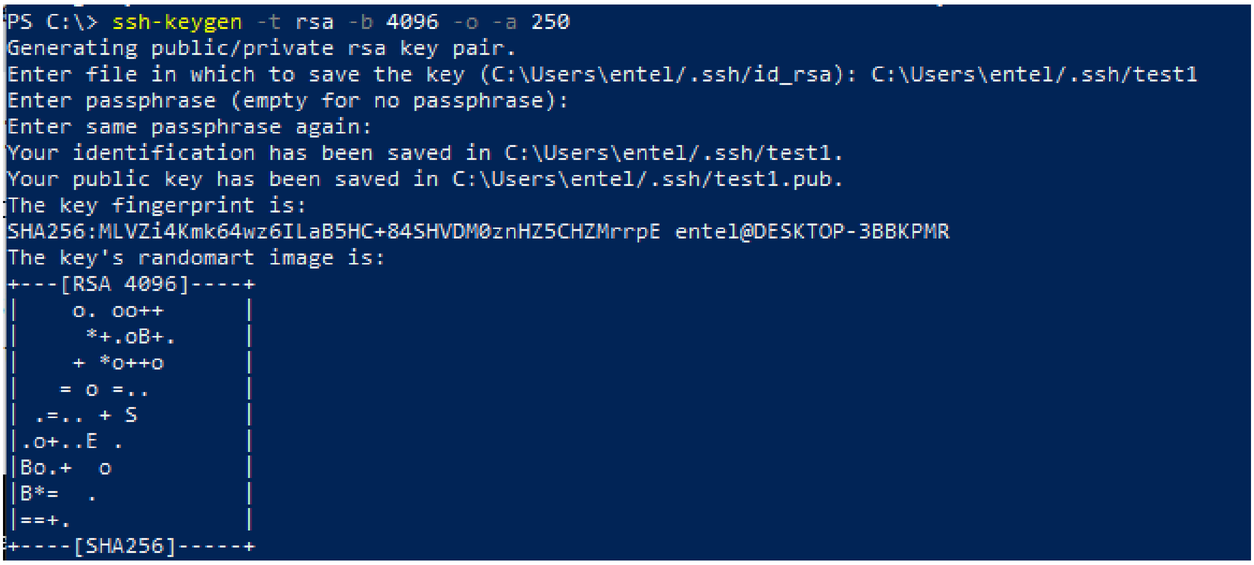
AllowUsers vivek jerry

DenyUsers root saroj anjali foo

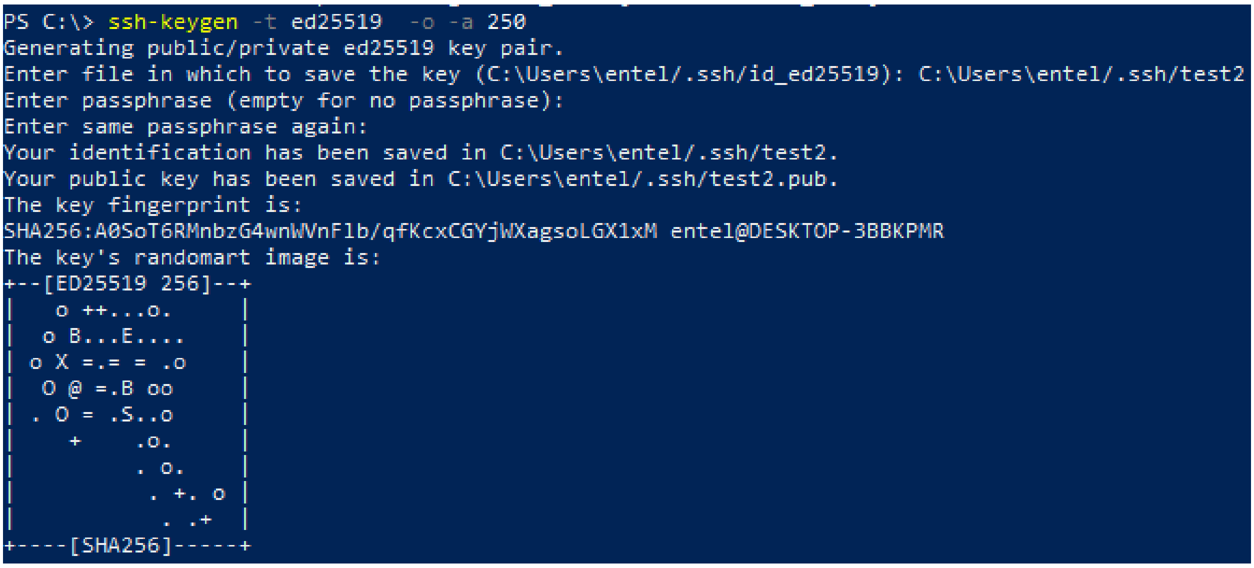
We also need to explicitly disallow remote login from accounts with empty passwords, update sshd_config with the following line:

PermitEmptyPasswords no

3. List the options for choosing keys for encryption in SSH. Implement 3 of them.



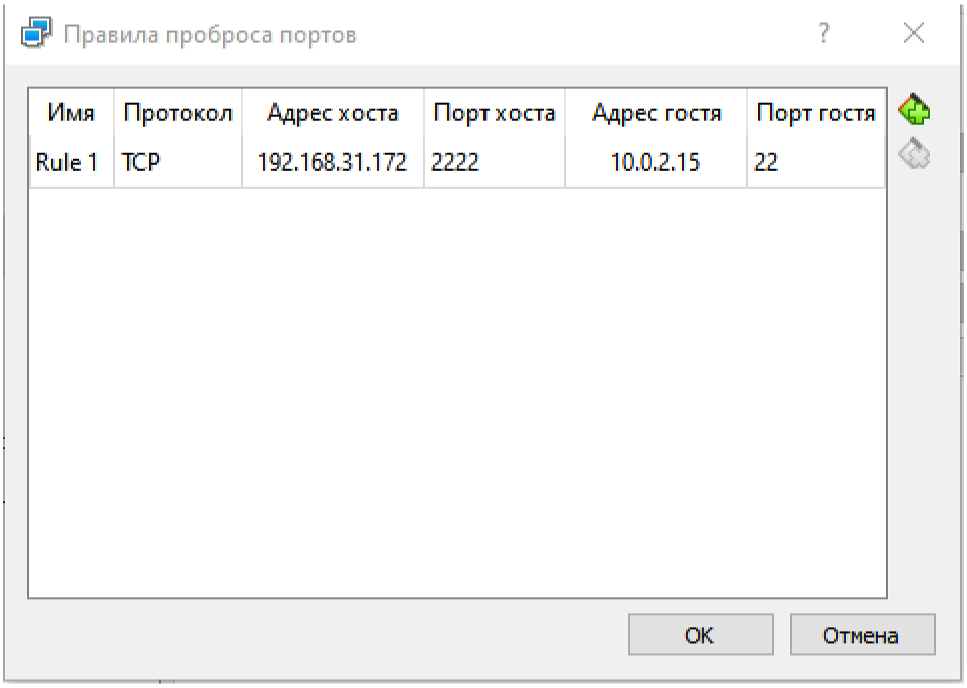
```
PS C:\> ssh-keygen -t rsa -b 4096 -o -a 250
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\entel/.ssh/id_rsa): C:\Users\entel/.ssh/test1
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\entel/.ssh/test1.
Your public key has been saved in C:\Users\entel/.ssh/test1.pub.
The key fingerprint is:
SHA256:MLVZi4Kmk64wz6ILaB5HC+84SHVDM0znHZ5CHZMrrpE entel@DESKTOP-3BBKPMR
The key's randomart image is:
+---[RSA 4096]----+
|    o. oo++      |
|     *+.oB+.     |
|    + *o++o      |
|   = o =..       |
| .=.. + S        |
|.o+..E .         |
|Bo.+  o          |
|B*=  .           |
|==+.             |
+----[SHA256]-----+
```



```
PS C:\> ssh-keygen -t ed25519  -o -a 250
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\entel/.ssh/id_ed25519): C:\Users\entel/.ssh/test2
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\entel/.ssh/test2.
Your public key has been saved in C:\Users\entel/.ssh/test2.pub.
The key fingerprint is:
SHA256:A0SoT6RMnbzG4wnWVnFlb/qfKcxCGYjWXagsoLGX1xM entel@DESKTOP-3BBKPMR
The key's randomart image is:
+--[ED25519 256]--+
|   o ++...o.     |
|  o B...E....    |
| o X =.= = .o    |
|  O @ =.B oo     |
| . O = .S..o     |
|    +    .o.     |
|         . o.    |
|          . +. o |
|           . .+  |
+----[SHA256]-----+
```

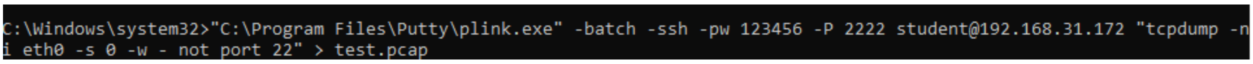
4. Implement port forwarding for the SSH client from the host machine to the guest Linux virtual machine behind NAT.

*Implemented in Linux tasks



Имя	Протокол	Адрес хоста	Порт хоста	Адрес гостя	Порт гостя
Rule 1	TCP	192.168.31.172	2222	10.0.2.15	22

5*. Intercept (capture) traffic (tcpdump, wireshark) while authorizing the remote client on the server using ssh, telnet, rlogin. Analyze the result.



```
C:\Windows\system32>"C:\Program Files\Putty\plink.exe" -batch -ssh -pw 123456 -P 2222 student@192.168.31.172 "tcpdump -n
i eth0 -s 0 -w - not port 22" > test.pcap
```