

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

D uninterruptible sleep (usually IO)  
I Idle kernel thread  
R running or runnable (on run queue)  
S interruptible sleep (waiting for an event to complete)  
T stopped by job control signal  
t stopped by debugger during the tracing  
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

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

Pstree -hvx

```
root@db1:/etc# pstree -h
systemd--accounts-daemon--2*[{accounts-daemon}]
      |
      |--atd
      |--cron
      |--dbus-daemon
      |--login--bash--sudo--su--bash
      |--multipathd--6*[{multipathd}]
      |--mysqld--36*[{mysqld}]
      |--networkd-dispat
      |--polkitd--2*[{polkitd}]
      |--rsyslogd--3*[{rsyslogd}]
      |--snapd--9*[{snapd}]
      |--sshd--sshd--sshd--bash--2*[man--pager]
      |                                     |
      |                                     |--sudo--su--bash--4*[less]
      |                                     |
      |                                     |--man--pager
      |                                     |--pstree
      |                                     |--vi
      |
      |--sshd--sshd--sftp-server
      |
      |--systemd--(sd-pam)
      |--systemd-journal
      |--systemd-logind
      |--systemd-network
      |--systemd-resolve
      |--systemd-timesyn--{systemd-timesyn}
      |--systemd-udev
      |--udisksd--4*[{udisksd}]
      |--unattended-upgr--{unattended-upgr}
```

3. **What is a proc file system?**

The `proc` filesystem is a process information pseudo-filesystem which provides an interface to kernel data structures. It does not contain real files and is commonly mounted at `/proc`.

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

cat /proc/cpuinfo | head -n 15

```
root@db1:/etc# cat /proc/cpuinfo | head -n 15
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 140
model name     : 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz
stepping       : 1
cpu MHz        : 2803.198
cache size     : 12288 KB
physical id    : 0
siblings       : 1
core id        : 0
cpu cores      : 1
apicid         : 0
initial apicid : 0
fpu            : yes
```

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.

```
root@db1:/etc# ps
  PID TTY          TIME CMD
 1548 pts/0    00:00:00 sudo
 1550 pts/0    00:00:00 su
 1551 pts/0    00:00:00 bash
 1605 pts/0    00:00:00 vi
 1788 pts/0    00:00:00 less
 1800 pts/0    00:00:00 less
 2220 pts/0    00:00:00 less
 2222 pts/0    00:00:00 less
 2556 pts/0    00:00:00 man
 2566 pts/0    00:00:00 pager
14508 pts/0    00:00:00 ps
root@db1:/etc# ps -o user,args,group,pid,%cpu --pid 2556
USER      COMMAND                                GROUP      PID %CPU
root      man id                                root       2556 0.0
root@db1:/etc#
```

6. How to define kernel processes and user processes?

Kernel processes:

```
root@db1:/etc# ps --ppid 2 -p 2 -o uname,pid,ppid,args | head
USER      PID    PPID  COMMAND
root        2         0 [kthreadd]
root        3         2 [rcu_gp]
root        4         2 [rcu_par_gp]
root        6         2 [kworker/0:0H-kblockd]
root        9         2 [mm_percpu_wq]
root       10         2 [ksoftirqd/0]
root       11         2 [rcu_sched]
root       12         2 [migration/0]
root       13         2 [idle_inject/0]
root@db1:/etc#
```

User processes

```
root@db1:/etc# ps --ppid 2 -p 2 -o uname,pid,ppid,args --deselect | head
USER      PID    PPID  COMMAND
root        1         0 /sbin/init maybe-ubiquity
root       355         1 /lib/systemd/systemd-journald
root       383         1 /lib/systemd/systemd-udev
root       535         1 /sbin/multipathd -d -s
systemd+   575         1 /lib/systemd/systemd-timesyncd
systemd+   625         1 /lib/systemd/systemd-networkd
systemd+   627         1 /lib/systemd/systemd-resolved
root       640         1 /usr/lib/accountsservice/accounts-daemon
root       643         1 /usr/sbin/cron -f
```

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?

```

root@db1:/etc# ps -eo uname,pid,ppid,args,s | tail
root      2763      2 [loop8] S
root      2785      1 /usr/lib/upower/upowerd S
root      2824      1 /usr/lib/snapd/snapd S
root     14399      2 [kworker/0:2-events] I
root     14498      2 [kworker/0:0-mm_percpu_wq] I
root     14499      2 [kworker/u2:1-events_unboun I
root     14510      2 [kworker/u2:2-events_power_ I
root     14513      2 [kworker/u2:0-events_power_ I
root     14522    1551 ps -eo uname,pid,ppid,args, R
root     14523    1551 tail S

```

8. Display only the processes of a specific user.

```

root@db1:/etc# ps -u mysql
  PID TTY          TIME CMD
   728 ?           00:01:59 mysqld
root@db1:/etc# █

```

9. What utilities can be used to command)?

`pgrep`, `pstree`, `top`, `proc`.

```

root@db1:/etc# pgrep mysql
728
root@db1:/etc# ps -u mysql
      PID TTY          TIME CMD
      728 ?           00:01:59 mysqld
root@db1:/etc# pgrep mysql
728
root@db1:/etc# pstree | head
systemd+-accounts-daemon---2*[{accounts-daemon}]
        |-atd
        |-cron
        |-dbus-daemon
        |-login---bash---sudo---su---bash
        |-multipathd---6*[{multipathd}]
        |-mysqld---38*[{mysqld}]
        |-networkd-dispat
        |-polkitd---2*[{polkitd}]
        |-rsyslogd---3*[{rsyslogd}]
root@db1:/etc# top
top - 12:29:07 up 23:28, 2 users, load average: 0.00, 0.04, 0.02
Tasks: 121 total, 1 running, 109 sleeping, 11 stopped, 0 zombie
%Cpu(s):  0.0/0.0  0[
MiB Mem :   981.2 total,    79.8 free,   377.0 used,   524.5 buff/cache
MiB Swap: 1231.0 total, 1066.5 free,   164.5 used.  421.1 avail Mem
Unknown command - try 'h' for help

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
14565	root	20	0	0	0	0	I	1.9	0.0	0:00.07	kworker/u2:0-events_unbound
1	root	20	0	168788	10448	6492	S	0.0	1.0	0:03.96	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-kblockd
9	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
10	root	20	0	0	0	0	S	0.0	0.0	0:00.64	ksoftirqd/0
11	root	20	0	0	0	0	I	0.0	0.0	0:17.72	rcu_sched
12	root	rt	0	0	0	0	S	0.0	0.0	0:00.53	migration/0
13	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
15	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
16	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
17	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_kthre
18	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kauditd
19	root	20	0	0	0	0	S	0.0	0.0	0:00.03	khungtaskd
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper
21	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	writeback
22	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kcompactd0
23	root	25	5	0	0	0	S	0.0	0.0	0:00.00	ksmd
24	root	39	19	0	0	0	S	0.0	0.0	0:00.06	khugepaged
70	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kintegrityd
71	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kblockd
72	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	blkcg_punt_bio
73	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	tpm_dev_wq
74	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	ata_sff
75	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	md
76	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	edac-poller
77	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	devfreq_wq
78	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdogd
81	root	20	0	0	0	0	S	0.0	0.0	0:01.11	kswapd0
82	root	20	0	0	0	0	S	0.0	0.0	0:00.00	ecryptfs-kthrea
84	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kthrotld
85	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	acpi_thermal_pm

#### 10. What information does top command display?

**top** can display system summary information as well as a list of processes or threads current managed by the Linux kernel.

```

root@db1:/etc# pgrep mysql
728
root@db1:/etc# pstree | head
systemd+-accounts-daemon---2*[{accounts-daemon}]
|-atd
|-cron
|-dbus-daemon
|-login---bash---sudo---su---bash
|-multipathd---6*[{multipathd}]
|-mysqld---38*[{mysqld}]
|-networkd-dispat
|-polkitd---2*[{polkitd}]
|-rsyslogd---3*[{rsyslogd}]
root@db1:/etc# top
top - 12:26:28 up 23:26, 2 users, load average: 0.02, 0.07, 0.03
Tasks: 121 total, 1 running, 109 sleeping, 11 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.3 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 981.2 total, 79.8 free, 377.0 used, 524.5 buff/cache
MiB Swap: 1231.0 total, 1066.5 free, 164.5 used. 421.1 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
  728 mysql     20   0 1309244 231072 11804 S   0.3   23.0   2:00.83 mysqld
 1399 db1       20   0  14060    3508   2708 S   0.3    0.3   0:02.12 sshd
    1 root       20   0 168788   10448   6492 S   0.0    1.0   0:03.96 systemd
    2 root       20   0      0      0      0 S   0.0    0.0   0:00.01 kthreadd
    3 root        0 -20      0      0      0 I   0.0    0.0   0:00.00 rcu_gp
    4 root        0 -20      0      0      0 I   0.0    0.0   0:00.00 rcu_par_gp
    6 root        0 -20      0      0      0 I   0.0    0.0   0:00.00 kworker/0:0H-kblockd
    9 root        0 -20      0      0      0 I   0.0    0.0   0:00.00 mm_percpu_wq
   10 root       20   0      0      0      0 S   0.0    0.0   0:00.63 ksoftirqd/0
   11 root       20   0      0      0      0 I   0.0    0.0   0:17.69 rcu_sched
   12 root        rt   0      0      0      0 S   0.0    0.0   0:00.53 migration/0
   13 root      -51   0      0      0      0 S   0.0    0.0   0:00.00 idle_inject/0
   14 root       20   0      0      0      0 S   0.0    0.0   0:00.00 cpuhp/0
   15 root       20   0      0      0      0 S   0.0    0.0   0:00.00 kdevtmpfs
   16 root        0 -20      0      0      0 I   0.0    0.0   0:00.00 netns
   17 root       20   0      0      0      0 S   0.0    0.0   0:00.00 rcu_tasks_kthre
   18 root       20   0      0      0      0 S   0.0    0.0   0:00.00 kauditd
   19 root       20   0      0      0      0 S   0.0    0.0   0:00.03 khungtaskd
   20 root       20   0      0      0      0 S   0.0    0.0   0:00.00 oom_reaper
   21 root        0 -20      0      0      0 I   0.0    0.0   0:00.00 writeback
   22 root       20   0      0      0      0 S   0.0    0.0   0:00.00 kcompactd0
   23 root       25   5      0      0      0 S   0.0    0.0   0:00.00 ksmd
   24 root       39  19      0      0      0 S   0.0    0.0   0:00.06 khugepaged
   70 root        0 -20      0      0      0 I   0.0    0.0   0:00.00 kintegrityd
   71 root        0 -20      0      0      0 I   0.0    0.0   0:00.00 kblockd
   72 root        0 -20      0      0      0 I   0.0    0.0   0:00.00 blkcg_punt_bio
   73 root        0 -20      0      0      0 I   0.0    0.0   0:00.00 tpm_dev_wq

```

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

```

top - 12:45:26 up 23:45, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 124 total, 1 running, 109 sleeping, 14 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
MiB Mem : 981.2 total, 78.0 free, 378.7 used, 524.5 buff/cache
MiB Swap: 1231.0 total, 1066.5 free, 164.5 used. 419.3 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
  728 mysql     20   0 1309244 231072 11804 S   0.0   23.0   2:02.29 mysqld

```

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

- 'h' help
- 'ESC' update
- 'k' kill process
- 'u' filter by user
- 'n' set max number of tasks displayed

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

In order to run several processes on a single CPU, a special mechanism is provided by linux kernel. It schedules processes execution time using priority concept. Priority determines how much CPU time will the process get in contrast to other processes.

**Commands:**

set priority for new process: **nice -n 10 firefox**

for existing process. **renice 10 -p \$(pgrep firefox)**

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

execute top command;

push r;

enter PID;

enter nice value.

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

```
root@db1:/etc# kill -L
1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
6) SIGABRT     7) SIGBUS     8) SIGFPE      9) SIGKILL     10) SIGUSR1
11) SIGSEGV    12) SIGUSR2    13) SIGPIPE    14) SIGALRM     15) SIGTERM
16) SIGSTKFLT  17) SIGCHLD   18) SIGCONT    19) SIGSTOP     20) SIGTSTP
21) SIGTTIN    22) SIGTTOU   23) SIGURG     24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF   28) SIGWINCH   29) SIGIO       30) SIGPWR
31) SIGSYS     34) SIGRTMIN  35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
```

Commonly used signals:

**SIGTERM (15)** - requests the termination of the process, could be ignored

**SIGKILL (9)** - causes the process to terminate immediately

**SIGSTOP (19)** - pause the process in its current state

**SIGCONT (18)** - resume process execution.

**Kill all you can kill: kill -9 -1**

**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, fg, bg, nohup** are the bash's job control commands. **nohup** stands for "no hangup". Command allows a process to continue running even after logout or disconnection from current shell.

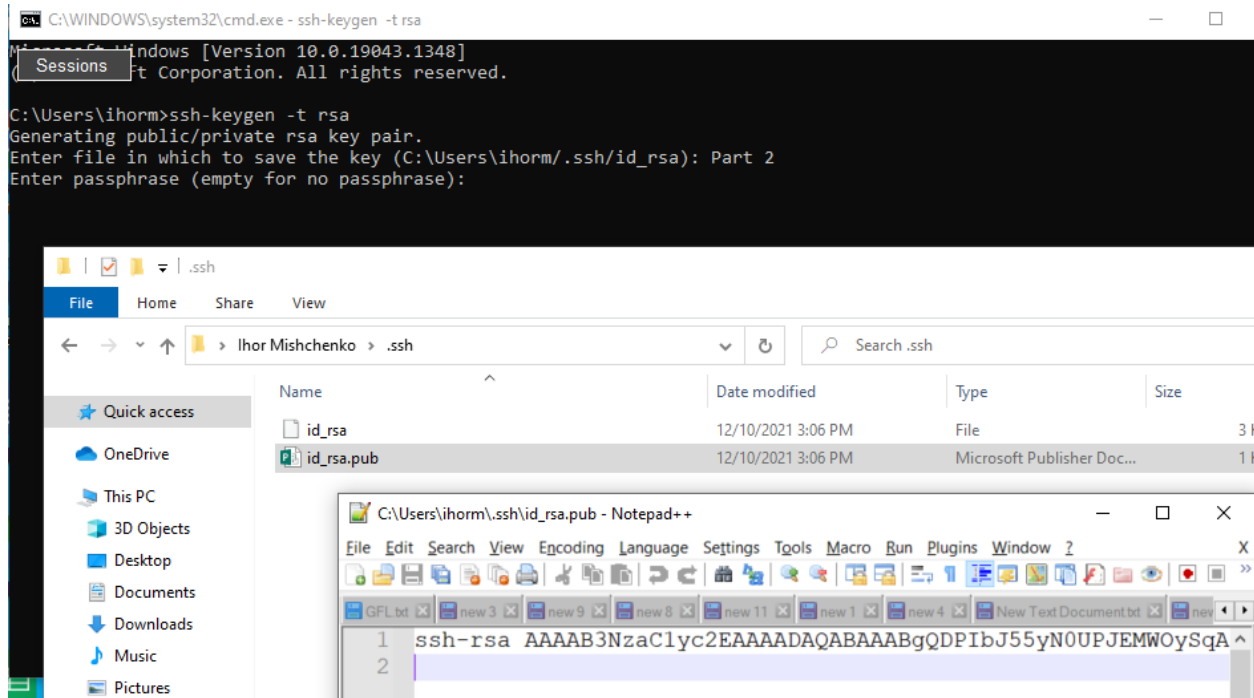
**Part 2**

**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)**

**ssh user@host** command is used to connect to a remote host

**ssh-keygen -t keytype** command is used to generate authentication key pair





2. Implement basic SSH settings to increase the security of the client-server connection (at least)
3. List the options for choosing keys for encryption in SSH. Implement 3 of them.

```
SSH-KEYGEN(1)                                BSD General Commands Manual                                SSH-KEYGEN(1)

NAME
    ssh-keygen - OpenSSH authentication key utility

SYNOPSIS
    ssh-keygen [-q] [-b bits] [-C comment] [-f output_keyfile] [-m format] [-t dsa | ecdsa | ecdsa-sk | ed25519 | ed25519-sk | rsa] [-N new_passphrase] [-O option] [-W provider]
    ssh-keygen -p [-f keyfile] [-m format] [-N new_passphrase] [-P old_passphrase]
    ssh-keygen -s [-f input_keyfile] [-m key_format]
    ssh-keygen -e [-f input_keyfile] [-m key_format]
    ssh-keygen -y [-f input_keyfile]
    ssh-keygen -c [-C comment] [-f keyfile] [-P passphrase]
    ssh-keygen -l [-v] [-f fingerprint_hash] [-f input_keyfile]
    ssh-keygen -B [-f input_keyfile]
    ssh-keygen -D pkcs11
    ssh-keygen -F hostname [-lv] [-f known_hosts_file]
    ssh-keygen -H [-f known_hosts_file]
    ssh-keygen -K [-w provider]
    ssh-keygen -R hostname [-f known_hosts_file]
    ssh-keygen -r hostname [-g] [-f input_keyfile]
    ssh-keygen -M generate [-O option] output_file
    ssh-keygen -H screen [-f input_file] [-O option] output_file
    ssh-keygen -I certificate_identity -s ca_key [-HU] [-O pkcs11_provider] [-n principals] [-O option] [-V validity_interval] [-z serial_number] file ...
    ssh-keygen -L [-f input_keyfile]
    ssh-keygen -A [-f prefix_path]
    ssh-keygen -k [-f krl_file] [-u] [-s ca_public] [-z version_number] file ...
    ssh-keygen -Q [-f krl_file] file ...
    ssh-keygen -Y find-principals -s signature_file -f allowed_signers_file
    ssh-keygen -Y check-novalidate -n namespace -s signature_file
    ssh-keygen -Y sign -f key_file -n namespace file ...
    ssh-keygen -Y verify -f allowed_signers_file -I signer_identity -n namespace -s signature_file [-r revocation_file]

DESCRIPTION
    ssh-keygen generates, manages and converts authentication keys for ssh(1).  ssh-keygen can create keys for use by SSH protocol version 2.

    The type of key to be generated is specified with the -t option.  If invoked without any arguments, ssh-keygen will generate an RSA key.

    ssh-keygen is also used to generate groups for use in Diffie-Hellman group exchange (DH-GEX).  See the MODULI GENERATION section for details.

    Finally, ssh-keygen can be used to generate and update Key Revocation Lists, and to test whether given keys have been revoked by one.  See the KEY REVOCATION LISTS section for details.

    Normally each user wishing to use SSH with public key authentication runs this once to create the authentication key in ~/.ssh/id_dsa, ~/.ssh/id_ecdsa, ~/.ssh/id_ecdsa_sk, ~/.ssh/id_ed25519, ~/.ssh/id_ed25519_sk or ~/.ssh/id_rsa.  Additionally, the system administrator may use this to generate host keys.
```

```
root@db1:/etc# ls /etc/ssh/*key*
/etc/ssh/ssh_host_dsa_key          /etc/ssh/ssh_host_ecdsa_key      /etc/ssh/ssh_host_ed25519_key    /etc/ssh/ssh_host_rsa_key
/etc/ssh/ssh_host_dsa_key.pub     /etc/ssh/ssh_host_ecdsa_key.pub  /etc/ssh/ssh_host_ed25519_key.pub /etc/ssh/ssh_host_rsa_key.pub
root@db1:/etc#
```