Task 3

Part 1

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:~$ ps
  PID TTY          TIME CMD
  865 pts/0    00:00:00 bash
 1175 pts/0    00:00:00 ps
student@CsnKhai:~$ pstree -h -s 865
init---sshd---sshd---sshd---bash-+-pstree
                                 `-sudo
student@CsnKhai:~$
```

3) What is a proc file system?

Proc file system (procfs) is a virtual file system created on the fly when the system boots and is dissolved at the time of system shutdown. It contains useful information about the processes that are currently running, it is regarded as a control and information center for the kernel. The proc file system also provides a communication medium between kernel space and user space.

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

```
student@CsnKhai:~$ lscpu
Architecture:          i686
CPU op-mode(s):        32-bit
Byte Order:            Little Endian
CPU(s):                1
On-line CPU(s) list:   0
Thread(s) per core:    1
Core(s) per socket:    1
Socket(s):             1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 158
Stepping:              10
CPU MHz:               0.000
BogoMIPS:              8081.40
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              8192K
student@CsnKhai:~$
```

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 -o user,pid,cmd,group
USER       PID CMD                         GROUP
student    865 -bash                       student
student   1178 ps -o user,pid,cmd,group    student
```

6) How to define kernel processes and user processes?

Kernel processes, also known as system processes or kernel threads, are processes that are essential for the functioning of the operating system. These processes are created and managed by the kernel itself. They handle critical system tasks and provide services required for the proper operation of the system.

User processes are application-level processes that are created by users or user-space applications. These processes are not part of the kernel itself and operate in user mode. They interact with the kernel through system calls to request services and resources.

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 aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0   4204  2212 ?        Ss   10:27   0:01 /sbin/init
root           2  0.0  0.0      0     0 ?        S    10:27   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    10:27   0:05 [ksoftirqd/0]
root           5  0.0  0.0      0     0 ?        S<   10:27   0:00 [kworker/0:0H]
root           7  0.0  0.0      0     0 ?        R    10:27   0:00 [rcu_sched]
root           8  0.0  0.0      0     0 ?        S    10:27   0:00 [rcu_bh]
root           9  0.0  0.0      0     0 ?        S    10:27   0:00 [migration/0]
root          10  0.0  0.0      0     0 ?        S    10:27   0:00 [watchdog/0]
root          11  0.0  0.0      0     0 ?        S<   10:27   0:00 [khelper]
root          12  0.0  0.0      0     0 ?        S    10:27   0:00 [kdevtmpfs]
root          13  0.0  0.0      0     0 ?        S<   10:27   0:00 [netns]
root          14  0.0  0.0      0     0 ?        S<   10:27   0:00 [writeback]
root          15  0.0  0.0      0     0 ?        S<   10:27   0:00 [kintegrityd]
root          16  0.0  0.0      0     0 ?        S<   10:27   0:00 [bioset]
root          17  0.0  0.0      0     0 ?        S<   10:27   0:00 [kworker/u3:0]
root          18  0.0  0.0      0     0 ?        S<   10:27   0:00 [kblockd]
root          19  0.0  0.0      0     0 ?        S<   10:27   0:00 [ata_sff]
```

From the status we can see, that most of the processes are sleeping.

8) Display only the processes of a specific user.

```
student@CsnKhai:~$ ps -u student
  PID TTY          TIME CMD
  829 tty1     00:00:00 bash
  864 ?        00:00:00 sshd
  865 pts/0    00:00:00 bash
  885 ?        00:00:00 sshd
  888 ?        00:00:00 sftp-server
 1181 pts/0    00:00:00 ps
student@CsnKhai:~$
```

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

The top or ps aux commands are frequently used to analyze existing running tasks, which can display a detailed list of all running processes, including information about the user, PID, CPU usage, memory usage, command, and more.

10) What information does top command display?

The top command provides real-time monitoring of system processes. It displays a dynamic view of processes sorted by various criteria like CPU usage, memory usage, and more. It's interactive and can be used to monitor ongoing activity.

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

```
student@CsnKhai:~$ top -u student
top - 15:18:43 up  4:51,  2 users,  load average: 0.03, 0.04, 0.05
Tasks:  68 total,   2 running,  64 sleeping,   2 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.7 sy,  0.0 ni, 99.0 id,  0.0 wa,  0.0 hi,  0.3 si,  0.0 st
KiB Mem:   2908896 total,   111744 used,  2797152 free,    12668 buffers
KiB Swap:        0 total,        0 used,        0 free.    70564 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
  864 student   20   0   11192   2340   1548 S  0.3  0.1   0:00.98 sshd
 1186 student   20   0    5424   1356   1008 R  0.3  0.0   0:00.01 top
  829 student   20   0    6668   2788   1500 S  0.0  0.1   0:00.11 bash
  865 student   20   0    6680   3156   1780 S  0.0  0.1   0:00.09 bash
  885 student   20   0   11192   1704    956 S  0.0  0.1   0:00.02 sshd
  888 student   20   0    2460    824    692 S  0.0  0.0   0:00.00 sftp-server
 1182 student   20   0    5420   1348   1000 T  0.0  0.0   0:00.24 top
```

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

We can use Q to quit top, K to kill the process, Z to change the color mode, R to renice the process etc.

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

To sort processes by CPU usage, we should use P. To sort processes by memory usage, we should use M. To sort processes by process name, we should use N etc.

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

```
student@CsnKhai:~$ nice -10 cat example_file.txt
Bohdan Lesyk
student@CsnKhai:~$
```

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

When you start top command, you can change the priority of procced by using r key, this will prompt you to enter a new priority value.

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

```
student@CsnKhai:~$ kill 1182
student@CsnKhai:~$ 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
student@CsnKhai:~$
```

Example of basic kill command and list of all kill signals.

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

The commands jobs, fg, bg, and nohup are used for process management and control.

```
y
y
y
y

^Z[3]+  Stopped                 yes
student@CsnKhai:~$ jobs
[1]-  Stopped                 top
[2]   Running                 sleep 300 &
[3]+  Stopped                 yes
student@CsnKhai:~$
```

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-keygen: Generate SSH key pairs for secure authentication.

```
C:\Users\acer>ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\acer/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\acer/.ssh/id_rsa.
Your public key has been saved in C:\Users\acer/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:seli5AUpZFI3vXafXvzGFWKMYZMSPP74lbYBDsQiKKc acer@Bohdan
The key's randomart image is:
+---[RSA 3072]----+
|  ..= o.o.. .    |
| . * o +.* =     |
|  + . + =.+ =    |
| E   . .o*.o + . |
|      ..S.=.ooo .|
|     o o . oo=o .|
|      + . ..o.oo.|
|     . .   ...  +|
|              . |
+----[SHA256]-----+

C:\Users\acer>
```

ssh command allows to connect to our virtual machine.

```
C:\Users\acer>ssh student@192.168.1.107
The authenticity of host '192.168.1.107 (192.168.1.107)' can't be established.
ECDSA key fingerprint is SHA256:yp8INOs6pk/gVv7G84N/cRT3KsgxLPiH81jZ/cRpz0o.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.107' (ECDSA) to the list of known hosts.
student@192.168.1.107'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: Thu Aug 17 18:31:03 2023 from 192.168.1.106
student@CsnKhai:~$
```

scp securely copy files between a local and remote system.

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

The default SSH connection port is 22. Of course, all attackers know this and therefore, it is necessary to change the default port number to ensure SSH security.

```
  GNU nano 2.2.6

# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 5922
```

There may be users without passwords on your system. To prevent such users from accessing the servers you can set the PermitEmptyPasswords line value to no.

```
PermitEmptyPasswords no
```

Attackers can try to gain access to your other systems by port forwarding through SSH connections. I should turn off X11Forwarding.

```
X11Forwarding no
X11DisplayOffset 10
PrintMotd no
PrintLastLog yes
TCPKeepAlive yes
#UseLogin no
```

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

RSA keys are widely used for encryption and digital signatures. They are known for their security and compatibility.

```
student@CsnKhai:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/student/.ssh/id_rsa):
Created directory '/home/student/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/student/.ssh/id_rsa.
Your public key has been saved in /home/student/.ssh/id_rsa.pub.
The key fingerprint is:
24:55:70:44:15:d6:5e:50:73:c7:1c:7a:a4:ff:49:a9 student@CsnKhai
The key's randomart image is:
+--[ RSA 2048]----+
|        o==.+o.O=|
|       . . .  = B|
|      . .    + o |
|       o      + .|
|        S      + |
|               o o|
|              E ..|
|                  |
|                  |
+-----------------+
```

DSA keys are an older key type, but their usage has decreased due to security concerns and the recommendation to use RSA or ECDSA keys instead.

```
student@CsnKhai:~$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/student/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/student/.ssh/id_dsa.
Your public key has been saved in /home/student/.ssh/id_dsa.pub.
The key fingerprint is:
8e:de:c9:3b:ea:a9:df:42:d3:e1:b8:11:75:d9:e4:1f student@CsnKhai
The key's randomart image is:
+--[ DSA 1024]----+
|             +.  |
|          . o..  |
|         . .  .E |
|        . .    ..|
|        S= .    .|
|       .= o      |
|       .o+       |
|       .o*..     |
|      .+*o*o     |
+-----------------+
```

ECDSA keys offer strong security with smaller key sizes compared to RSA, making them efficient for resource-constrained environments.

```
student@CsnKhai:~$ ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/home/student/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/student/.ssh/id_ecdsa.
Your public key has been saved in /home/student/.ssh/id_ecdsa.pub.
The key fingerprint is:
29:cc:42:e3:08:d9:e9:c4:ae:4a:5a:0c:47:4d:bf:d4 student@CsnKhai
The key's randomart image is:
+--[ECDSA  256]---+
|     .           |
| + + . .         |
|o * + o E        |
| * + = . .       |
|. = o = S        |
| =   . .         |
|..o              |
|+.               |
|o                |
+-----------------+
student@CsnKhai:~$
```

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