TASK5.3
Part1
1. How many states could a process have in Linux?
5 states: created, ready, running, waiting and terminated.
2. Examine the pstree command. Make output (highlight) the chain (ancestors) of the current Process.

```
root@CsnKhai:~# pstree
init─┬─cron
     ├─dbus-daemon
     ├─dhclient
     ├─6*[getty]
     ├─rsyslogd───3*[{rsyslogd}]
     ├─sshd─┬─sshd───sshd───bash───sudo───su───bash───pstree
     │      └─sshd───sshd───sftp-server
     ├─systemd-logind
     ├─systemd-udevd
     ├─upstart-file-br
     ├─upstart-socket-
     └─upstart-udev-br
root@CsnKhai:~#
```

So, basically all the processes that do exist are children to the main init process. One tree to point out is an sshd process, that actually manages the current user session - it can be seen by a tree sshd-sshd-bash-sudo-su-bash-pstree. What actually happened is that a remote connection was managed by ssh demon, that opened bash for a user that logged in. That user (student) used sudo to change the user to root using the su command, and when it changed su opened bash again, but for a next user (root). Root then opened a pstree, that can be seen on the end of a tree.
3. What is a proc file system?
The /proc file system in Linux is a virtual file system that provides an interface to the kernel's internal data structures and runtime information. It allows users and processes to interact with the kernel and system resources, like hardware, as if they were accessing usual files.
4. Print information about the processor (its type, supported technologies, etc.).

```
root@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:                 142
Stepping:              10
CPU MHz:               0.000
BogoMIPS:              10422.27
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              6144K
root@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.

```
root@CsnKhai:~# ps -F
UID        PID  PPID  C    SZ   RSS PSR STIME TTY          TIME CMD
root       875   843  0  1685  2036   0 00:22 pts/0    00:00:00 sudo su -
root       876   875  0  1576  1600   0 00:22 pts/0    00:00:00 su -
root       877   876  0  1634  2892   0 00:22 pts/0    00:00:00 -su
root       899   877  0  1304  1152   0 00:45 pts/0    00:00:00 ps -F
root@CsnKhai:~#
```

To print out additional info about the processes, key -F was used.
Other variant is to use ps -aux.
6. How to define kernel processes and user processes?
From the output of a ps command, for example, ps -aux, the processes, whose commands are enclosed in square brackets, are those of the kernel processes, with others being user processes:

```
root@CsnKhai:~# ps -aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.1  0.9   4332  2320 ?        Ss   00:21   0:02 /sbin/init
root         2  0.0  0.0      0     0 ?        S    00:21   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    00:21   0:00 [ksoftirqd/0]
root         5  0.0  0.0      0     0 ?        S<   00:21   0:00 [kworker/0:0H]
root         7  0.0  0.0      0     0 ?        S    00:21   0:01 [rcu_sched]
root         8  0.0  0.0      0     0 ?        S    00:21   0:00 [rcu_bh]
root         9  0.0  0.0      0     0 ?        S    00:21   0:00 [migration/0]
root        10  0.0  0.0      0     0 ?        S    00:21   0:01 [watchdog/0]
root        11  0.0  0.0      0     0 ?        S<   00:21   0:00 [khelper]
root        12  0.0  0.0      0     0 ?        S    00:21   0:00 [kdevtmpfs]
root        13  0.0  0.0      0     0 ?        S<   00:21   0:00 [netns]
root        14  0.0  0.0      0     0 ?        S<   00:21   0:00 [writeback]
root        15  0.0  0.0      0     0 ?        S<   00:21   0:00 [kintegrityd]
root        16  0.0  0.0      0     0 ?        S<   00:21   0:00 [bioset]
root        17  0.0  0.0      0     0 ?        S<   00:21   0:00 [kworker/u3:0]
root        18  0.0  0.0      0     0 ?        S<   00:21   0:00 [kblockd]
root        19  0.0  0.0      0     0 ?        S<   00:21   0:00 [ata_sff]
root        20  0.0  0.0      0     0 ?        S    00:21   0:00 [khubd]
root        21  0.0  0.0      0     0 ?        S<   00:21   0:00 [md]
root        22  0.0  0.0      0     0 ?        S<   00:21   0:00 [devfreq_wq]
root        23  0.9  0.0      0     0 ?        R    00:21   0:15 [kworker/0:1]
root        25  0.0  0.0      0     0 ?        S    00:21   0:00 [khungtaskd]
root        26  0.0  0.0      0     0 ?        S    00:21   0:00 [kswapd0]
root        27  0.0  0.0      0     0 ?        SN   00:21   0:00 [ksmd]
root        28  0.0  0.0      0     0 ?        S    00:21   0:00 [fsnotify_mark]
root        29  0.0  0.0      0     0 ?        S    00:21   0:00 [ecryptfs-kthrea]
root        30  0.0  0.0      0     0 ?        S<   00:21   0:00 [crypto]
root        42  0.0  0.0      0     0 ?        S<   00:21   0:00 [kthrotld]
root        44  0.0  0.0      0     0 ?        S    00:21   0:01 [kworker/u2:2]
root        45  0.0  0.0      0     0 ?        S    00:21   0:00 [scsi_eh_0]
root        46  0.0  0.0      0     0 ?        S    00:21   0:00 [scsi_eh_1]
root        67  0.0  0.0      0     0 ?        S<   00:21   0:00 [deferwq]
root        68  0.0  0.0      0     0 ?        S<   00:21   0:00 [charger_manager]
root       110  0.0  0.0      0     0 ?        S<   00:21   0:00 [kworker/u3:1]
root       111  0.0  0.0      0     0 ?        S<   00:21   0:00 [kpsmoused]
root       112  0.0  0.0      0     0 ?        S    00:21   0:00 [kworker/0:2]
root       113  0.0  0.0      0     0 ?        S    00:21   0:00 [scsi_eh_2]
root       122  0.0  0.0      0     0 ?        S    00:21   0:00 [jbd2/sda1-8]
root       123  0.0  0.0      0     0 ?        S<   00:21   0:00 [ext4-rsv-conver]
root       249  0.0  0.3   3008   876 ?        S    00:22   0:00 upstart-udev-bridge --daemon
```

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?

See screenshot above with the output of a ps -aux command. There is a column named “STAT” that describes current process statuses. The possible ones are:

- R: Running or runnable (on run queue)
- D: Uninterruptible sleep (usually IO)
- 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

There’s also a list of additional symbols, that can tell more about process state:

- <: High-priority (not nice to other users)
- N: Low-priority (nice to other users)
- L: Has pages locked into memory (for real-time or custom IO)
- s: Is a session leader
- l: Is multi-threaded (using CLONE_THREAD, like NPTL pthreads do)
- +: Is in the foreground process group

So, as can be seen from a screenshot, most of the processes are in the state of interruptible sleep or they are simply waiting for other processes or events. Process kworker is in a running state, and additional symbols show that the leading process is /sbin/init, ksmd is a low-priority process, and a bunch of other processes are not nice for others, having a higher priority.

8. Display only the processes of a specific user.
Using a key -u with a username:

```
root@CsnKhai:~# ps -u student
  PID TTY          TIME CMD
  822 ?        00:00:00 sshd
  841 ?        00:00:00 sshd
  842 ?        00:00:00 sftp-server
  843 pts/0    00:00:00 bash
root@CsnKhai:~#
```

9. What utilities can be used to analyze existing running tasks (by analyzing the help for the ps command)?
Using the ps command, processes can be accessed in such ways:
-Showing command names, real group ids or names, serrion or group names, PIDs or PPIDs, real user IDs, terminals:

```
root@CsnKhai:~# ps --help list

Usage:
 ps [options]

Selection by list:
 -C <command>         command name
 -G, --Group <gid>    real group id or name
 -g, --group <group>  session or effective group name
 -p, --pid <pid>      process id
     --ppid <pid>     select by parent process id
 -s, --sid <session>  session id
 -t, t, --tty <tty>   terminal
 -u, U, --user <uid>  effective user id or name
 -U, --User <uid>     real user id or name

  selection <arguments> take either:
    comma-separated list e.g. '-u root,nobody' or
    blank-separated list e.g. '-p 123 4567'

For more details see ps(1).
```

The output format can be such:
-full format, ascii process tree, process hierarchy, BSD formats, user-oriented format, register format, virtual memory format, with or without SELinux security data:

```
root@CsnKhai:~# ps --help output

Usage:
 ps [options]

Output formats:
 -F                   extra full
 -f                   full-format, including command lines
  f, --forest         ascii art process tree
 -H                   show process hierarchy
 -j                   jobs format
  j                   BSD job control format
 -l                   long format
  l                   BSD long format
 -M, Z                add security data (for SELinux)
 -O <format>          preloaded with default columns
  O <format>          as -O, with BSD personality
 -o, o, --format <format>
                      user defined format
  s                   signal format
  u                   user-oriented format
  v                   virtual memory format
  X                   register format
 -y                   do not show flags, show rrs vs. addr (used with -l)
     --context        display security context (for SELinux)
     --headers        repeat header lines, one per page
     --no-headers     do not print header at all
     --cols, --columns, --width <num>
                      set screen width
     --rows, --lines <num>
                      set screen height
```

Threads can be shown as if they were processes, with LWP and NLWP columns, SPID column, and after processes:

```
root@CsnKhai:~# ps --help threads

Usage:
 ps [options]

Show threads:
  H                   as if they where processes
 -L                   possibly with LWP and NLWP columns
 -m, m                after processes
 -T                   possibly with SPID column

For more details see ps(1).
```

Miscellaneous options include:

Showing the true command name, using a particular sorting order, using numeric UID and wchan, setting up unlimited width for output, showing the environment after command, listing format specifiers et cetera:

```
root@CsnKhai:~# ps --help misc

Usage:
 ps [options]

Miscellaneous options:
 -c                  show scheduling class with -l option
  c                  show true command name
  e                  show the environment after command
  k,    --sort       specify sort order as: [+|-]key[,[+|-]key[,...]]
  L                  list format specifiers
  n                  display numeric uid and wchan
  S,    --cumulative include some dead child process data
 -y                  do not show flags, show rss (only with -l)
 -V, V, --version   display version information and exit
 -w, w               unlimited output width

        --help <simple|list|output|threads|misc|all>
                     display help and exit

For more details see ps(1).
```

10. What information does the top command display?

It is an interactive table of processes, set to update itself each N seconds (usually 3 or 5). It depicts a detailed description of processes, including process uptime, memory and processor time usage, command, process id, process user, priorities, niceness, et cetera.

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

It can be done using the -u key, like “top -u student”:

```
top - 01:18:43 up 56 min,  2 users,  load average: 0.00, 0.01, 0.05
Tasks:  77 total,   1 running,  76 sleeping,   0 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
KiB Mem:    247792 total,   127032 used,   120760 free,    11636 buffers
KiB Swap:        0 total,        0 used,        0 free.    70288 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
  822 student   20   0   11192   2264   1476 S  0.0  0.9   0:00.13 sshd
  841 student   20   0   11192   1856   1104 S  0.0  0.7   0:00.01 sshd
  842 student   20   0    2460    820    692 S  0.0  0.3   0:00.00 sftp-server
  843 student   20   0    6668   3104   1740 S  0.0  1.3   0:00.05 bash
  941 student   20   0   11192   2260   1468 S  0.0  0.9   0:00.06 sshd
  942 student   20   0   11192   1856   1104 S  0.0  0.7   0:00.00 sshd
  943 student   20   0    2460    628    528 S  0.0  0.3   0:00.01 sftp-server
  944 student   20   0    6668   3024   1660 S  0.0  1.2   0:00.04 bash
```

Here, no root processes are shown, only student processes.

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

```
Help for Interactive Commands - procps-ng version 3.3.9
Window 1:Def: Cumulative mode Off.  System: Delay 3.0 secs; Secure mode Off.

  Z,B,E,e   Global: 'Z' colors; 'B' bold; 'E'/'e' summary/task memory scale
  l,t,m     Toggle Summary: 'l' load avg; 't' task/cpu stats; 'm' memory info
  0,1,2,3,I Toggle: '0' zeros; '1/2/3' cpus or numa node views; 'I' Irix mode
  f,F,X     Fields: 'f'/'F' add/remove/order/sort; 'X' increase fixed-width

  L,&,<,> . Locate: 'L'/'&' find/again; Move sort column: '<'/'>' left/right
  R,H,V,J . Toggle: 'R' Sort; 'H' Threads; 'V' Forest view; 'J' Num justify
  c,i,S,j . Toggle: 'c' Cmd name/line; 'i' Idle; 'S' Time; 'j' Str justify
  x,y     . Toggle highlights: 'x' sort field; 'y' running tasks
  z,b     . Toggle: 'z' color/mono; 'b' bold/reverse (only if 'x' or 'y')
  u,U,o,O . Filter by: 'u'/'U' effective/any user; 'o'/'O' other criteria
  n,#,^O  . Set: 'n'/'#' max tasks displayed; Show: Ctrl+'O' other filter(s)
  C,...   . Toggle scroll coordinates msg for: up,down,left,right,home,end

  k,r       Manipulate tasks: 'k' kill; 'r' renice
  d or s    Set update interval
  W,Y       Write configuration file 'W'; Inspect other output 'Y'
  q         Quit
          ( commands shown with '.' require a visible task display window )
```

For example, k command allows you to kill a process with a PID that will be set interactively. Also, you can apply sorting by a particular column or filtering by user or other criterias.

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

Sorted by memory (shift+M):

```
top - 02:09:12 up  1:47,  2 users,  load average: 0.03, 0.04, 0.05
Tasks:  77 total,   1 running,  76 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,   127412 used,   120380 free,    11636 buffers
KiB Swap:        0 total,        0 used,        0 free.    70588 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
  903 root      20   0   11192   3792   3028 S  0.0  1.5   0:00.52 sshd
  801 root      20   0   11192   3788   3028 S  0.0  1.5   0:00.05 sshd
  905 root      20   0   11192   3784   3032 S  0.0  1.5   0:00.08 sshd
  804 root      20   0   11192   3780   3028 S  0.0  1.5   0:00.08 sshd
  843 student   20   0    6668   3104   1740 S  0.0  1.3   0:00.05 bash
  944 student   20   0    6668   3024   1660 S  0.0  1.2   0:00.04 bash
  962 root      20   0    6536   2896   1660 S  0.0  1.2   0:00.02 bash
  877 root      20   0    6536   2892   1660 S  0.0  1.2   0:00.03 bash
  713 root      20   0    7796   2488   1996 S  0.0  1.0   0:00.26 sshd
  941 student   20   0   11192   2488   1692 S  0.0  1.0   0:01.23 sshd
    1 root      20   0    4332   2320   1420 S  0.0  0.9   0:02.21 init
  822 student   20   0   11192   2264   1476 S  0.0  0.9   0:00.13 sshd
  875 root      20   0    6740   2036   1616 S  0.0  0.8   0:00.03 sudo
  960 root      20   0    6740   2028   1608 S  0.0  0.8   0:00.01 sudo
  410 root      20   0    5512   1856    140 S  0.0  0.7   0:00.00 dhclient
  841 student   20   0   11192   1856   1104 S  0.0  0.7   0:00.01 sshd
  942 student   20   0   11192   1856   1104 S  0.0  0.7   0:00.00 sshd
  360 root      20   0    4212   1688   1396 S  0.0  0.7   0:00.01 systemd-logind
  961 root      20   0    6304   1604   1216 S  0.0  0.6   0:00.00 su
  876 root      20   0    6304   1600   1216 S  0.0  0.6   0:00.00 su
  253 root      20   0   12024   1412    980 S  0.0  0.6   0:00.18 systemd-udevd
  998 root      20   0    5420   1360   1004 R  0.0  0.5   0:00.00 top
  364 syslog    20   0   30476   1064    732 S  0.0  0.4   0:00.64 rsyslogd
  337 message+  20   0    4236    988    704 S  0.0  0.4   0:00.14 dbus-daemon
  648 root      20   0    3132    924    456 S  0.0  0.4   0:00.02 upstart-socket-
  249 root      20   0    3008    876    664 S  0.0  0.4   0:00.28 upstart-udev-br
  691 root      20   0    4644    836    720 S  0.0  0.3   0:00.00 getty
  696 root      20   0    4644    836    720 S  0.0  0.3   0:00.00 getty
  699 root      20   0    4644    836    720 S  0.0  0.3   0:00.00 getty
  697 root      20   0    4644    832    720 S  0.0  0.3   0:00.00 getty
  792 root      20   0    4644    832    720 S  0.0  0.3   0:00.00 getty
  693 root      20   0    4644    828    720 S  0.0  0.3   0:00.00 getty
  842 student   20   0    2460    820    692 S  0.0  0.3   0:00.00 sftp-server
  724 root      20   0    3052    792    624 S  0.0  0.3   0:00.00 cron
  943 student   20   0    2460    628    528 S  0.0  0.3   0:00.01 sftp-server
```

Sorted by CPU (shift+P):

```
top - 02:10:10 up  1:48,  2 users,  load average: 0.01, 0.03, 0.05
Tasks:  77 total,   1 running,  76 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.7 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:    247792 total,   127412 used,   120380 free,    11636 buffers
KiB Swap:        0 total,        0 used,        0 free.    70588 cached Mem
  scroll coordinates: y = 1/77 (tasks), x = 1/12 (fields)
  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
  941 student   20   0   11192   2488   1692 S  0.3  1.0   0:01.29 sshd
    1 root      20   0    4332   2320   1420 S  0.0  0.9   0:02.21 init
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.01 kthreadd
    3 root      20   0       0      0      0 S  0.0  0.0   0:01.38 ksoftirqd/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   0:01.34 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   0:01.48 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   0:00.17 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   0:17.27 kworker/0:1
   25 root      20   0       0      0      0 S  0.0  0.0   0:00.00 khungtaskd
   26 root      20   0       0      0      0 S  0.0  0.0   0:00.00 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
   30 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 crypto
   42 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kthrotld
   44 root      20   0       0      0      0 S  0.0  0.0   0:01.74 kworker/u2:2
   45 root      20   0       0      0      0 S  0.0  0.0   0:00.24 scsi_eh_0
   46 root      20   0       0      0      0 S  0.0  0.0   0:00.02 scsi_eh_1
   67 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 deferwq
   68 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 charger_manager
  110 root       0 -20       0      0      0 S  0.0  0.0   0:00.02 kworker/u3:1
```

Sorted by time (shift+T):

```
top - 02:11:03 up  1:49,  2 users,  load average: 0.00, 0.03, 0.05
Tasks:  77 total,   1 running,  76 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,   127412 used,   120380 free,    11636 buffers
KiB Swap:        0 total,        0 used,        0 free.    70588 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
   23 root      20   0       0      0      0 S  0.0  0.0   0:17.29 kworker/0:1
    1 root      20   0    4332   2320   1420 S  0.0  0.9   0:02.21 init
   44 root      20   0       0      0      0 S  0.0  0.0   0:01.75 kworker/u2:2
   10 root      rt   0       0      0      0 S  0.0  0.0   0:01.48 watchdog/0
    3 root      20   0       0      0      0 S  0.0  0.0   0:01.40 ksoftirqd/0
    7 root      20   0       0      0      0 S  0.0  0.0   0:01.34 rcu_sched
  941 student   20   0   11192   2488   1692 S  0.0  1.0   0:01.33 sshd
  364 syslog    20   0   30476   1064    732 S  0.0  0.4   0:00.64 rsyslogd
  903 root      20   0   11192   3792   3028 S  0.0  1.5   0:00.52 sshd
  249 root      20   0    3008    876    664 S  0.0  0.4   0:00.28 upstart-udev-br
  713 root      20   0    7796   2488   1996 S  0.0  1.0   0:00.26 sshd
   45 root      20   0       0      0      0 S  0.0  0.0   0:00.24 scsi_eh_0
  253 root      20   0   12024   1412    980 S  0.0  0.6   0:00.18 systemd-udevd
   20 root      20   0       0      0      0 S  0.0  0.0   0:00.17 khubd
  337 message+  20   0    4236    988    704 S  0.0  0.4   0:00.14 dbus-daemon
  822 student   20   0   11192   2264   1476 S  0.0  0.9   0:00.13 sshd
  122 root      20   0       0      0      0 S  0.0  0.0   0:00.09 jbd2/sda1-8
  804 root      20   0   11192   3780   3028 S  0.0  1.5   0:00.08 sshd
  905 root      20   0   11192   3784   3032 S  0.0  1.5   0:00.08 sshd
  375 root      20   0    3008    524    276 S  0.0  0.2   0:00.07 upstart-file-br
  801 root      20   0   11192   3788   3028 S  0.0  1.5   0:00.05 sshd
  843 student   20   0    6668   3104   1740 S  0.0  1.3   0:00.05 bash
  944 student   20   0    6668   3024   1660 S  0.0  1.2   0:00.04 bash
  875 root      20   0    6740   2036   1616 S  0.0  0.8   0:00.03 sudo
  877 root      20   0    6536   2892   1660 S  0.0  1.2   0:00.03 bash
   46 root      20   0       0      0      0 S  0.0  0.0   0:00.02 scsi_eh_1
  110 root       0 -20       0      0      0 S  0.0  0.0   0:00.02 kworker/u3:1
  648 root      20   0    3132    924    456 S  0.0  0.4   0:00.02 upstart-socket-
  962 root      20   0    6536   2896   1660 S  0.0  1.2   0:00.02 bash
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.01 kthreadd
  360 root      20   0    4212   1688   1396 S  0.0  0.7   0:00.01 systemd-logind
  841 student   20   0   11192   1856   1104 S  0.0  0.7   0:00.01 sshd
  943 student   20   0    2460    628    528 S  0.0  0.3   0:00.01 sftp-server
  960 root      20   0    6740   2028   1608 S  0.0  0.8   0:00.01 sudo
    5 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kworker/0:0H
```

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

Process priority is a measure of how important the process is relative to other processes - how much memory, processor time and other system resources is supposed to be given to this process comparatively to other ones - more or less. The processes with higher priority are given more resources and vice versa. To change the process priority, or to set a basic one, command “nice” is used. Niceness is an opposite of a priority: using niceness 20 on a process with a priority 20 will set it to 0. To change the niceness value of the already “nicened” (or “de-nicened”) process, use the command “renice”.

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

Yes, it can be done interactively by using the “r” key while the top is open (r for renice). After that, it will be needed to specify the process id to change and the amount of niceness to apply. Here, niceness is set relatively to priority, e.g. “-10” will reduce priority by 10 and vice versa.

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

```
KILL(1)                                User Commands                                KILL(1)

NAME
       kill - send a signal to a process

SYNOPSIS
       kill [options] <pid> [...]

DESCRIPTION
       The  default signal for kill is TERM.  Use -l or -L to list available signals.  Particularly useful signals include HUP, INT, KILL, STOP, CONT, and
       0.  Alternate signals may be specified in three ways: -9, -SIGKILL or -KILL.  Negative PID values may be used to choose whole process  groups;  see
       the PGID column in ps command output.  A PID of -1 is special; it indicates all processes except the kill process itself and init.

OPTIONS
       <pid> [...]
              Send signal to every <pid> listed.

       -<signal>
       -s <signal>
       --signal <signal>
              Specify the signal to be sent.  The signal can be specified by using name or number.  The behavior of signals is explained in signal(7) man-
              ual page.

       -l, --list [signal]
              List signal names.  This option has optional argument, which will convert signal number to signal name, or other way round.

       -L, --table
              List signal names in a nice table.

       NOTES  Your shell (command line interpreter) may have a built-in kill command.  You may need to run the command  described  here  as  /bin/kill  to
              solve the conflict.

EXAMPLES
       kill -9 -1
              Kill all processes you can kill.

       kill -l 11
              Translate number 11 into a signal name.

       kill -L
              List the available signal choices in a nice table.
 Manual page kill(1) line 1 (press h for help or q to quit)
```

By default, the SIGTERM signal is being sent. There are other signals, seen from this list:

```
root@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
root@CsnKhai:~#
```

Most commonly used signals are SIGINT, SIGTERM, SIGKILL, SIGSTOP.

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 used to display and manipulate jobs. For example, without options the list of active jobs will be listed.

```
root@CsnKhai:~# help jobs
jobs: jobs [-lnprs] [jobspec ...] or jobs -x command [args]
    Display status of jobs.

    Lists the active jobs.  JOBSPEC restricts output to that job.
    Without options, the status of all active jobs is displayed.

    Options:
      -l        lists process IDs in addition to the normal information
      -n        lists only processes that have changed status since the last
        notification
      -p        lists process IDs only
      -r        restrict output to running jobs
      -s        restrict output to stopped jobs

    If -x is supplied, COMMAND is run after all job specifications that
    appear in ARGS have been replaced with the process ID of that job's
    process group leader.

    Exit Status:
    Returns success unless an invalid option is given or an error occurs.
    If -x is used, returns the exit status of COMMAND.
root@CsnKhai:~#
```

Fg is used to move jobs to foreground:

```
root@CsnKhai:~# help fg
fg: fg [job_spec]
    Move job to the foreground.

    Place the job identified by JOB_SPEC in the foreground, making it the
    current job.  If JOB_SPEC is not present, the shell's notion of the
    current job is used.

    Exit Status:
    Status of command placed in foreground, or failure if an error occurs.
root@CsnKhai:~#
```

Bg is the same but to background:

```
root@CsnKhai:~# help bg
bg: bg [job_spec ...]
    Move jobs to the background.

    Place the jobs identified by each JOB_SPEC in the background, as if they
    had been started with `&'.  If JOB_SPEC is not present, the shell's notion
    of the current job is used.

    Exit Status:
    Returns success unless job control is not enabled or an error occurs.
root@CsnKhai:~#
```

Nohup is used to running a command with output to a non-tty and ignoring the hangup signals:

```
NAME
       nohup - run a command immune to hangups, with output to a non-tty

SYNOPSIS
       nohup COMMAND [ARG]...
       nohup OPTION

DESCRIPTION
       Run COMMAND, ignoring hangup signals.

       --help display this help and exit

       --version
              output version information and exit

       If  standard  input  is  a  terminal,  redirect  it  from  /dev/null.   If standard output is a terminal, append output to 'nohup.out' if possible,
       '$HOME/nohup.out' otherwise.  If standard error is a terminal, redirect it to standard output.  To save output to FILE, use 'nohup COMMAND > FILE'.

       NOTE: your shell may have its own version of nohup, which usually supersedes the version described here.  Please refer to your  shell's  documenta-
       tion for details about the options it supports.

AUTHOR
       Written by Jim Meyering.

REPORTING BUGS
       Report nohup bugs to bug-coreutils@gnu.org
       GNU coreutils home page: <http://www.gnu.org/software/coreutils/>
       General help using GNU software: <http://www.gnu.org/gethelp/>
       Report nohup translation bugs to <http://translationproject.org/team/>

COPYRIGHT
       Copyright © 2013 Free Software Foundation, Inc.  License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
       This is free software: you are free to change and redistribute it.  There is NO WARRANTY, to the extent permitted by law.

SEE ALSO
       The  full  documentation for nohup is maintained as a Texinfo manual.  If the info and nohup programs are properly installed at your site, the com-
       mand

              info coreutils 'nohup invocation'
 Manual page nohup(1) line 1 (press h for help or q to quit)
```

```
root@CsnKhai:~# sleep 100
^Z
[1]+  Stopped                 sleep 100
root@CsnKhai:~# jobs
[1]+  Stopped                 sleep 100
root@CsnKhai:~# bg %1
[1]+ sleep 100 &
root@CsnKhai:~# jobs
[1]+  Running                 sleep 100 &
root@CsnKhai:~# fg sleep
sleep 100
^Z
[1]+  Stopped                 sleep 100
root@CsnKhai:~# jobs
[1]+  Stopped                 sleep 100
root@CsnKhai:~#
```

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)

Install OpenSSH for Windows

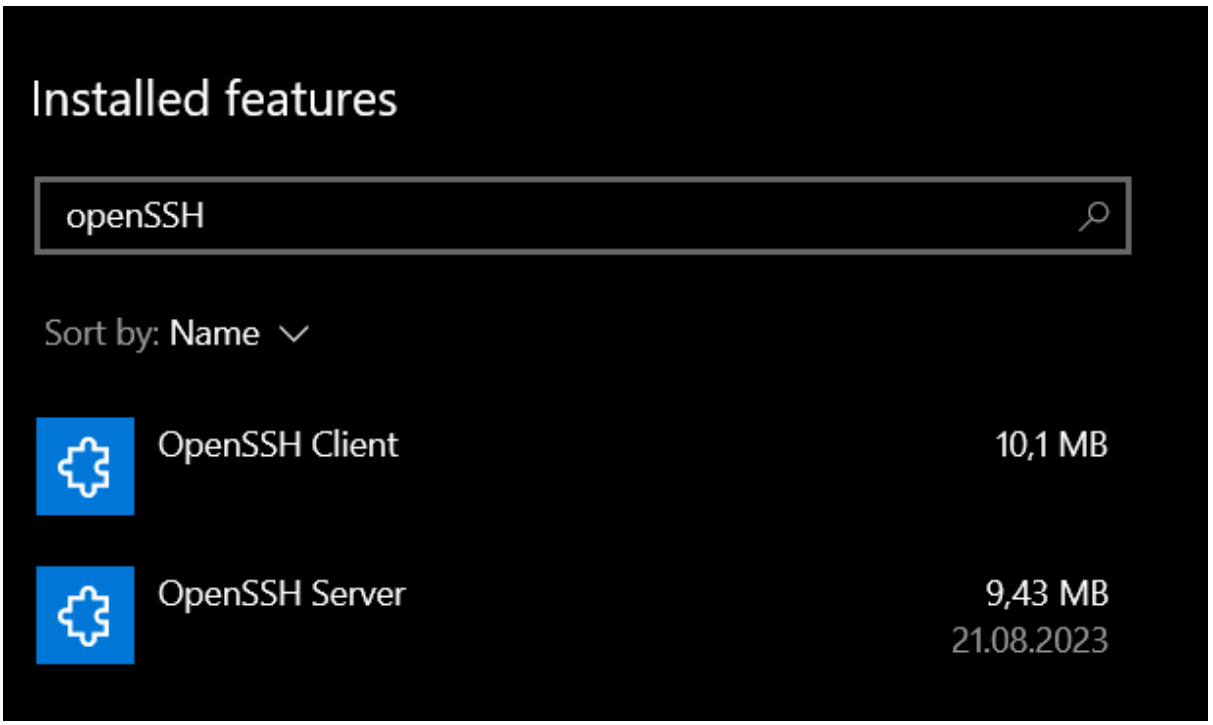
GUI | PowerShell

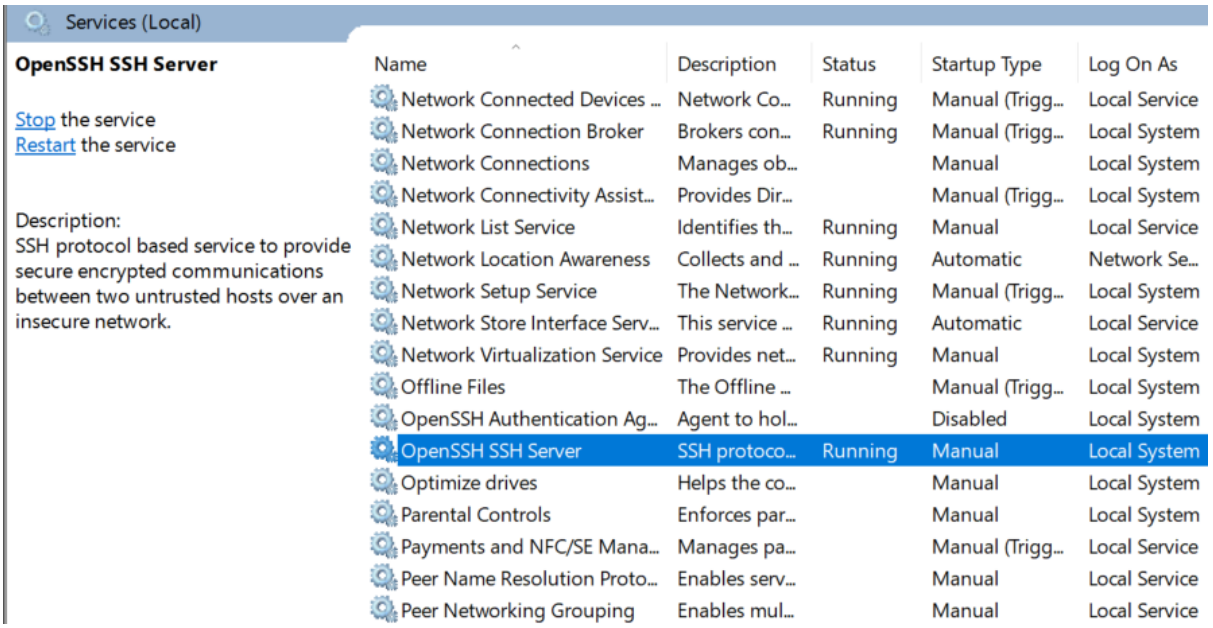
Both OpenSSH components can be installed using Windows Settings on Windows Server 2019 and Windows 10 devices.

To install the OpenSSH components:

1. Open **Settings**, select **Apps**, then select **Optional Features**.
2. Scan the list to see if the OpenSSH is already installed. If not, at the top of the page, select **Add a feature**, then:
   - Find **OpenSSH Client**, then select **Install**
   - Find **OpenSSH Server**, then select **Install**
3. Once setup completes, return to **Apps** and **Optional Features** and confirm OpenSSH is listed.
4. Open the **Services** desktop app. (Select **Start**, type *services.msc* in the search box, and then select the **Service** app or press `ENTER`.)
5. In the details pane, double-click **OpenSSH SSH Server**.
6. On the **General** tab, from the **Startup type** drop-down menu, select **Automatic**.
7. To start the service, select **Start**.

> **Note**
>
> Installing OpenSSH Server will create and enable a firewall rule named `OpenSSH-Server-In-TCP`. This allows inbound SSH traffic on port 22. If this rule is not enabled and this port is not open, connections will be refused or reset.





Useful commands:
Basic ssh connectivity test (Should connect to a remote server):

```
PS C:\Windows\system32> ssh student@172.21.254.239
The authenticity of host '172.21.254.239 (172.21.254.239)' 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 '172.21.254.239' (ECDSA) to the list of known hosts.
student@172.21.254.239'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: Sun Aug 20 22:03:24 2023 from desktop-mrfe7pj.mshome.net
student@CsnKhai:~$ exit
logout
Connection to 172.21.254.239 closed.
```

Ssh key generation:

```
PS C:\Windows\system32> ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\Shatterhand/.ssh/id_rsa): key1
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in key1.
Your public key has been saved in key1.pub.
The key fingerprint is:
SHA256:qw3JLQKzKkzheE5xT+s38lv9yDDr5vTMYCQkE0WJbOE shatterhand@DESKTOP-MRFE7PJ
The key's randomart image is:
+---[RSA 3072]----+
|      .o=o.       |
|      .+..        |
|      .E .        |
| .. . .+          |
|o +o o .S .       |
|.oo+ .oo + .      |
|o+. ..= o B .     |
|.o.  .o=o+.X o    |
|o     .++*+ * .   |
+----[SHA256]-----+
```

Scanning for existing keys on the machine:
Here, a few old keys and a newly generated can be seen:

```
PS C:\Windows\system32> ssh-keyscan localhost
# localhost:22 SSH-2.0-OpenSSH_for_Windows_8.1
localhost ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNT
# localhost:22 SSH-2.0-OpenSSH_for_Windows_8.1
localhost ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIElQ6j2fwoz8kw
# localhost:22 SSH-2.0-OpenSSH_for_Windows_8.1
localhost ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCYpu4LTVQZ2j
8JJQNIj6OncFCooiiBCdbkW489VmC2phgc38viJt8oWyJPxQ0KngWlVB7DF3
H6RAOcKbqw08PuZo8hgNkwGoZNRXLWITntBFxqIkImhPeqcNAkrxWDrxcECw
PS C:\Windows\system32>
```

Secure copy print testing:
Transferring sample pdf document to a remote host:

```
PS C:\Users\Shatterhand\Desktop> scp .\Kantarou_3.pdf student@172.21.254.239:/tmp
student@172.21.254.239's password:
Kantarou_3.pdf
PS C:\Users\Shatterhand\Desktop>
```

And the transferred file on the linux machine:

```
root@CsnKhai:/# cd tmp/
root@CsnKhai:/tmp# ls
Kantarou_3.pdf
root@CsnKhai:/tmp#
```

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

The first security option for SSH would be disabling root login. To do that, enter the file /etc/ssh/sshd_config and change the needed option to no:

```
  GNU nano 2.2.6                                    File: sshd_config

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

# What ports, IPs and protocols we listen for
Port 22
# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::
#ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 1024

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin no
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile     %h/.ssh/authorized_keys

# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh_known_hosts

^G Get Help          ^O WriteOut          ^R Read File         ^Y Pr
^X Exit              ^J Justify           ^W Where Is          ^V Ne
```

Also, optional it is possible to turn off the password-based authentication via ssh (leaving public key authentication the only viable option):

```
3. 172.21.254.239 (student)
  GNU nano 2.2.6                                    File: sshd_config

ServerKeyBits 1024

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin no
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile     %h/.ssh/authorized_keys

# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2
HostbasedAuthentication no
# Uncomment if you don't trust ~/.ssh/known_hosts for RhostsRSAAuthentication
#IgnoreUserKnownHosts yes

# To enable empty passwords, change to yes (NOT RECOMMENDED)
PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Change to no to disable tunnelled clear text passwords
#PasswordAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosGetAFSToken no

^G Get Help      ^O WriteOut      ^R Read File      ^Y Pr
^X Exit          ^J Justify       ^W Where Is       ^V Ne
```

Another option to make it even more secure would be using iptables to restrict all the ip addresses except the trusted ones from using port 22. That's a little too complex by now, but still is a viable security option.

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

Viable options are:

```
usage: ssh-keygen [options]
Options:
  -A          Generate non-existent host keys for all key types.
  -a number   Number of KDF rounds for new key format or moduli primality tests.
  -B          Show bubblebabble digest of key file.
  -b bits     Number of bits in the key to create.
  -C comment  Provide new comment.
  -c          Change comment in private and public key files.
  -D pkcs11   Download public key from pkcs11 token.
  -e          Export OpenSSH to foreign format key file.
  -F hostname Find hostname in known hosts file.
  -f filename Filename of the key file.
  -G file     Generate candidates for DH-GEX moduli.
  -g          Use generic DNS resource record format.
  -H          Hash names in known_hosts file.
  -h          Generate host certificate instead of a user certificate.
  -I key_id   Key identifier to include in certificate.
  -i          Import foreign format to OpenSSH key file.
  -J number   Screen this number of moduli lines.
  -j number   Start screening moduli at specified line.
  -K checkpt  Write checkpoints to this file.
  -k          Generate a KRL file.
  -L          Print the contents of a certificate.
  -l          Show fingerprint of key file.
  -M memory   Amount of memory (MB) to use for generating DH-GEX moduli.
  -m key_fmt  Conversion format for -e/-i (PEM|PKCS8|RFC4716).
  -N phrase   Provide new passphrase.
  -n name,... User/host principal names to include in certificate
  -O option   Specify a certificate option.
  -o          Enforce new private key format.
  -P phrase   Provide old passphrase.
  -p          Change passphrase of private key file.
  -Q          Test whether key(s) are revoked in KRL.
  -q          Quiet.
  -R hostname Remove host from known_hosts file.
  -r hostname Print DNS resource record.
  -S start    Start point (hex) for generating DH-GEX moduli.
  -s ca_key   Certify keys with CA key.
  -T file     Screen candidates for DH-GEX moduli.
  -t type     Specify type of key to create.
  -u          Update KRL rather than creating a new one.
  -V from:to  Specify certificate validity interval.
```

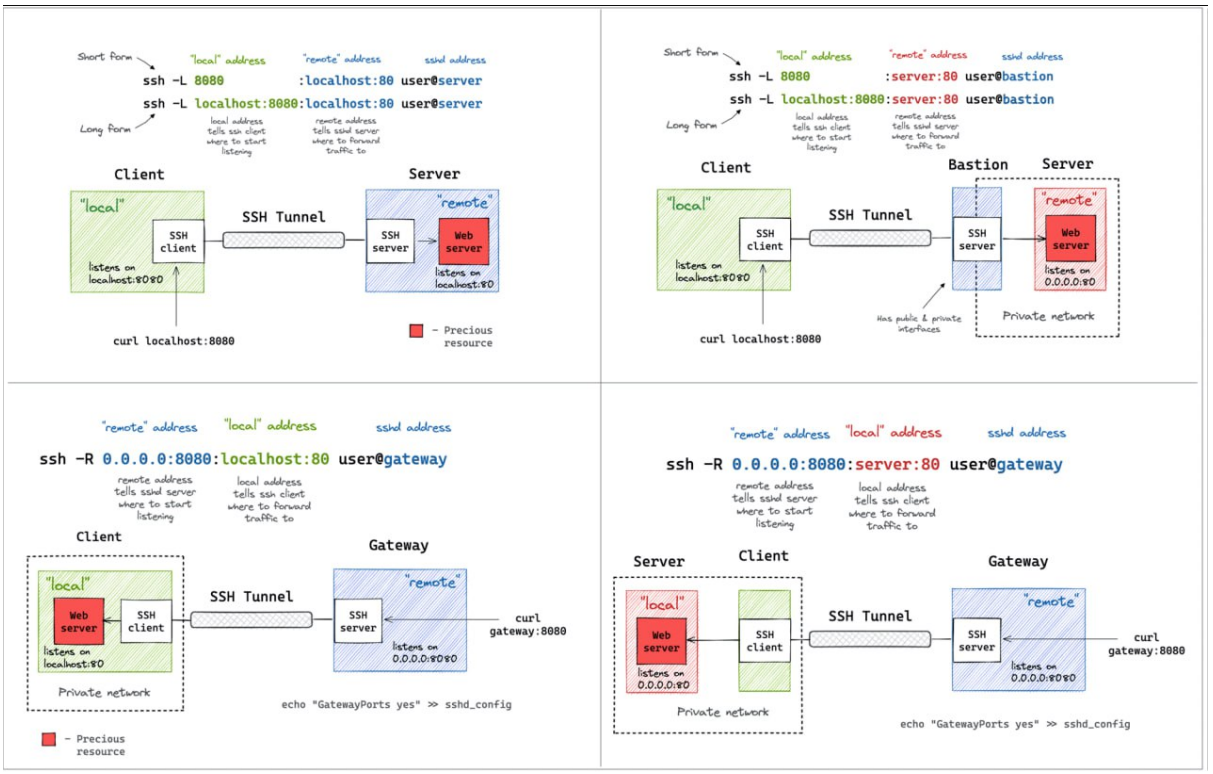
```
  -V from:to  Specify certificate validity interval.
  -v          Verbose.
  -W gen      Generator to use for generating DH-GEX moduli.
  -y          Read private key file and print public key.
  -Z cipher   Specify a cipher for new private key format.
  -z serial   Specify a serial number.
```

I have decided to use the RSA encryption algorithm, set the length of a key to 4096 bits, and to add an expiry date for this key (for a year in the future). Also, I have set the comment. To implement it, keys -t, -b, -V and -C were used:

```
root@CsnKhai:~# ssh-keygen -t rsa -b 4096 -V +365d -C "hello softserve"
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): test1
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in test1.
Your public key has been saved in test1.pub.
The key fingerprint is:
7d:1d:02:0f:92:42:33:70:a3:b5:69:04:f6:22:49:ca hello softserve
The key's randomart image is:
+--[ RSA 4096]----+
|   . ++O ..o      |
|.o o *.*.. +      |
|.Eo o =.    o .   |
|   . o   .   o .  |
|        S . . .   |
|           .      |
|                  |
|                  |
|                  |
+-----------------+
root@CsnKhai:~#
```

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

To do so, I have used this nice picture, that makes the tunnelling via SSH much more an understandable process:



So, basically, I need to use an L option, specifying the local port first, and the remote address and port after:

```
PS C:\Users\Shatterhand\Desktop> ssh -L localhost:8080:172.21.254.239:80 student@172.21.254.239
student@172.21.254.239'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: Mon Aug 21 00:18:42 2023 from desktop-mrfe7pj.mshome.net
student@CsnKhai:~$
```

So, basically, now all that goes into port 8080 on a host machine will be forwarded to the virtual Linux to port 80 via the SSH tunnel. To prove it, showing the open ports on a host machine:

```
Select Administrator: Command Prompt - netstat -ab
 EventLog
[svchost.exe]
 TCP    0.0.0.0:49667          www:0                  LISTENING
 Schedule
[svchost.exe]
 TCP    0.0.0.0:49668          www:0                  LISTENING
[spoolsv.exe]
 TCP    0.0.0.0:60947          www:0                  LISTENING
Can not obtain ownership information
 TCP    127.0.0.1:6000         DESKTOP-MRFE7PJ:52181  ESTABLISHED
[XWin_MobaX.exe]
 TCP    127.0.0.1:6000         DESKTOP-MRFE7PJ:52182  ESTABLISHED
[XWin_MobaX.exe]
 TCP    127.0.0.1:6000         DESKTOP-MRFE7PJ:52183  ESTABLISHED
[XWin_MobaX.exe]
 TCP    127.0.0.1:6463         www:0                  LISTENING
[Discord.exe]
 TCP    127.0.0.1:8080         www:0                  LISTENING
[ssh.exe]
 TCP    127.0.0.1:50615        DESKTOP-MRFE7PJ:50617  ESTABLISHED
[Zoom.exe]
 TCP    127.0.0.1:50617        DESKTOP-MRFE7PJ:50615  ESTABLISHED
[Zoom.exe]
 TCP    127.0.0.1:52181        DESKTOP-MRFE7PJ:6000   ESTABLISHED
[XWin_MobaX.exe]
 TCP    127.0.0.1:52182        DESKTOP-MRFE7PJ:6000   ESTABLISHED
[XWin_MobaX.exe]
 TCP    127.0.0.1:52183        DESKTOP-MRFE7PJ:6000   ESTABLISHED
[XWin_MobaX.exe]
 TCP    127.0.0.1:56005        DESKTOP-MRFE7PJ:56006  ESTABLISHED
```

Here indeed the open port 8080 is present with ssh.exe as owner process.

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