Part1

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

two types of processes in Linux: foreground and background

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

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

3. What is a proc file system?

The proc file system acts as an interface to internal data structures in the kernel. It can be used to obtain information about the system and to change certain kernel parameters at runtime (sysctl).

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:             AuthenticAMD
CPU family:            23
Model:                 113
Stepping:              0
CPU MHz:               3599.481
BogoMIPS:              7198.96
L1d cache:             32K
L1i cache:             32K
L2 cache:              512K
L3 cache:              32768K
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 -Ffu student
UID        PID  PPID  C    SZ   RSS PSR STIME TTY          TIME CMD
student    889   867  0  2235  2124   0 06:07 ?        00:00:00 sshd: student@pts/0
student    903   889  0  1667  3024   0 06:07 pts/0    00:00:00 -bash
student    914   870  0  2158  1704   0 06:07 ?        00:00:00 sshd: student@notty
student    915   914  0   615   824   0 06:07 ?        00:00:00 /usr/lib/openssh/sftp-server
root@CsnKhai:~#
```

6. How to define kernel processes and user processes?

the kernel runs in kernel space, and normal programs run in user space. it restricts user programs so they can't mess with memory (and other resources) owned by other programs or by the OS kernel. The kernel is the core of the operating system. It normally has full access to all memory and machine hardware (and everything else on the machine). When managing processes, it is easy to recognize the kernel processes because they have a name that is between square 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?

```
root@CsnKhai:~# ps ax
  PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:01 /sbin/init
    2 ?        S      0:00 [kthreadd]
    3 ?        S      0:00 [ksoftirqd/0]
    4 ?        S      0:00 [kworker/0:0]
    5 ?        S<     0:00 [kworker/0:0H]
    6 ?        S      0:00 [kworker/u2:0]
    7 ?        S      0:00 [rcu_sched]
    8 ?        S      0:00 [rcu_bh]
    9 ?        S      0:00 [migration/0]
   10 ?        S      0:01 [watchdog/0]
   11 ?        S<     0:00 [khelper]
   12 ?        S      0:00 [kdevtmpfs]
   13 ?        S<     0:00 [netns]
   14 ?        S<     0:00 [writeback]
   15 ?        S<     0:00 [kintegrityd]
   16 ?        S<     0:00 [bioset]
   17 ?        S<     0:00 [kworker/u3:0]
   18 ?        S<     0:00 [kblockd]
   19 ?        S<     0:00 [ata_sff]
   20 ?        S      0:00 [khubd]
   21 ?        S<     0:00 [md]
   22 ?        S<     0:00 [devfreq_wq]
   23 ?        R      0:29 [kworker/0:1]
   25 ?        S      0:00 [khungtaskd]
   26 ?        S      0:00 [kswapd0]
   27 ?        SN     0:00 [ksmd]
   28 ?        S      0:00 [fsnotify_mark]
   29 ?        S      0:00 [ecryptfs-kthrea]
   30 ?        S<     0:00 [crypto]
   42 ?        S<     0:00 [kthrotld]
   44 ?        S      0:00 [scsi_eh_0]
   45 ?        S      0:00 [scsi_eh_1]
   67 ?        S<     0:00 [deferwq]
   68 ?        S<     0:00 [charger_manager]
  114 ?        S<     0:00 [kpsmoused]
  115 ?        S      0:00 [scsi_eh_2]
  124 ?        S<     0:00 [kworker/u3:1]
  125 ?        S      0:00 [jbd2/sda1-8]
  126 ?        S<     0:00 [ext4-rsv-conver]
  268 ?        S      0:00 upstart-udev-bridge --daemon
  273 ?        Ss     0:00 /lib/systemd/systemd-udevd --daemon
  321 ?        Ss     0:00 dbus-daemon --system --fork
  354 ?        Ss     0:00 /lib/systemd/systemd-logind
  356 ?        Ssl    0:00 rsyslogd
  374 ?        S      0:00 upstart-file-bridge --daemon
  417 ?        S<     0:00 [ttm_swap]
  586 ?        Ss     0:00 dhclient -1 -v -pf /run/dhclient.eth0.pid -lf /var/lib/dhcp/dhclient.eth0.leases eth0
  703 ?        S      0:00 upstart-socket-bridge --daemon
  763 tty4     Ss+    0:00 /sbin/getty -8 38400 tty4
  765 tty5     Ss+    0:00 /sbin/getty -8 38400 tty5
  768 tty2     Ss+    0:00 /sbin/getty -8 38400 tty2
  769 tty3     Ss+    0:00 /sbin/getty -8 38400 tty3
  771 tty6     Ss+    0:00 /sbin/getty -8 38400 tty6
  794 ?        Ss     0:00 /usr/sbin/sshd -D
  806 ?        Ss     0:00 cron
  858 tty1     Ss     0:00 /bin/login --
  867 ?        Ss     0:00 sshd: student [priv]
  869 ?        S      0:00 [kauditd]
```

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.

ps – u username

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

ps -aux | less

10. What information does top command display?

top command is used to show the Linux processes. It provides a dynamic real-time view of the running system. Usually, this command shows the summary information of the system and the list of processes or threads which are currently managed by the Linux Kernel.

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

top –u username

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

-a : Sort by memory usage

-n : Number of iterations limit as: -n number

Specifies the maximum number of iterations, or frames, top should produce before ending

-d : Delay time interval as: -d ss.tt (seconds.tenths)

Specifies the delay between screen updates, and overrides the corresponding value in one's personal configuration file or the startup default.

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

<Shift>+<N>—sort by PID;
<Shift>+<P>—sort by CPU usage;
<Shift>+<M>—sort by Memory usage;
<Shift>+<T>—sort by Time usage;

<Shift>+<Z>—change colors;

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

The kernel stores a great deal of information about processes including process priority which is simply the scheduling priority attached to a process. Processes with a higher priority will be executed before those with a lower priority, while processes with the same priority are scheduled one after the next, repeatedly.

There are a total of 140 priorities and two distinct priority ranges implemented in Linux. The first one is a ***nice*** value (niceness) which ranges from -20 (highest priority value) to 19 (lowest priority value) and the default is 0, this is what we will uncover in this guide. The other is the real-time priority, which ranges from 1 to 99 by default, then 100 to 139 are meant for user-space.

One important characteristic of Linux is dynamic priority-based scheduling, which allows the ***nice*** value of processes to be changed (increased or decreased) depending on your needs.

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

you can use the r command from the top utility to change the priority of a currently running process

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

To send a signal to a process, the kill command is used. The most common use is the need to stop a process, which you can do by using the kill command followed by the PID of the process. This sends the SIGTERM signal to the process, which normally causes the process to cease its activity.

Sometimes the kill command does not work because the process you want to kill is busy. In that case, you can use kill 9 to send the SIGKILL signal to the process.

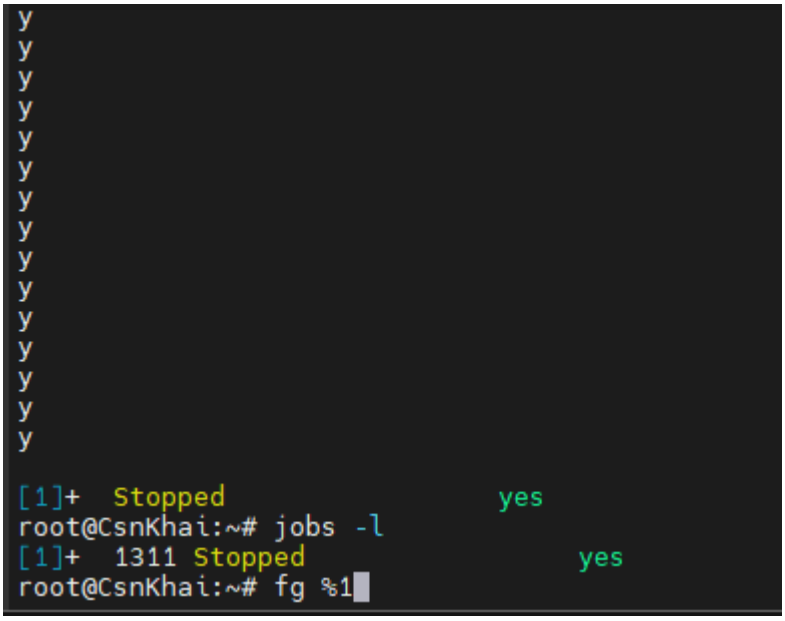
SIGINT	2	Signals when the Linux user presses 'CONTROL-C'
SIGHUP	1	Hangs up signals when controlling the terminal or at the end of the controlling processes.
SIGQUIT	3	Signals when the Linux user presses 'CONTROL-D'
SIGFPE	8	Signals when any unexpected mathematical operation is performed.
SIGKILL	9	When any of the process issues this signal, it will quit immediately.
SIGALRM	14	Signals for alarm clock
SIGTERM	15	Signals to terminate the process or the software.
SIGSTOP	17,19,23	Signals to stop the process in Linux.

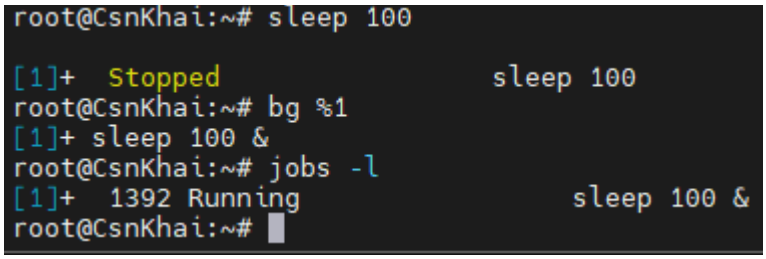
17. Commands jobs, fg, bg, nohup. What are they for? Use the sleep, yes command to demonstrate the process control mechanism with fg, bg.

Jobs command is used to list the jobs that you are running in the background and in the foreground

The fg command, short for the foreground, is a command that moves a background process on your current Linux shell to the foreground. This contrasts the bg command, short for background, that sends a process running in the foreground to the background in the current shell.

nohup (No Hang Up) is a command in Linux systems that runs the process even after logging out from the shell/terminal. Usually, every process in Linux systems is sent a SIGHUP (Signal Hang UP) which is responsible for terminating the process after closing/exiting the terminal.





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)

```
PS C:\Users\kauk> ssh student@192.168.1.6
The authenticity of host '192.168.1.6 (192.168.1.6)' can't be established.
ECDSA key fingerprint is SHA256:yp8INOs6pk/gVv7G84N/cRT3KsgxLPiH81jZ/cRpz0o.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.1.6' (ECDSA) to the list of known hosts.
student@192.168.1.6's password:
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.13.0-63-generic i686)

 * Documentation:  https://help.ubuntu.com/
Last login: Fri Feb 18 06:22:32 2022 from 192.168.1.3
student@CsnKhai:~$ pwd
/home/student
student@CsnKhai:~$ ls -l
total 0
student@CsnKhai:~$ ls -la
total 40
drwxr-xr-x 4 student student 4096 Feb 18 06:22 .
drwxr-xr-x 4 root    root    4096 Feb 17 08:38 ..
-rw------- 1 student student  166 Feb 17 19:12 .bash_history
-rw-r--r-- 1 student student  220 Sep 15  2015 .bash_logout
-rw-r--r-- 1 student student 3637 Sep 15  2015 .bashrc
drwx------ 2 student student 4096 Sep 15  2015 .cache
-rw-rw-r-- 1 student student   34 Feb 16 09:11 .plan
-rw-r--r-- 1 student student  675 Sep 15  2015 .profile
drwx------ 2 student student 4096 Feb 15 15:04 .ssh
-rw------- 1 student student  159 Feb 18 06:22 .Xauthority
student@CsnKhai:~$ pstree -h
init─┬─cron
     ├─dbus-daemon
     ├─dhclient
     ├─6*[getty]
     ├─rsyslogd───3*[{rsyslogd}]
     ├─sshd─┬─sshd───sshd───bash───sudo───bash
     │      ├─sshd───sshd───sftp-server
     │      └─sshd───sshd───bash───pstree
     ├─systemd-logind
     ├─systemd-udevd
     ├─upstart-file-br
     ├─upstart-socket-
     └─upstart-udev-br
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
```

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

```
PS C:\Users\kauk> ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\kauk/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\kauk/.ssh/id_rsa.
Your public key has been saved in C:\Users\kauk/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:LTRW1Fuj2ExuTHZIki8+E914CIknsrF0I+HkGBa44Vw kauk\kauk@kauk
The key's randomart image is:
+---[RSA 3072]----+
| .+.o. ..=+o     |
|o. E= = =o* +    |
|ooo.oB ++& O .   |
|.o  o  o+o& o    |
|       .S+..     |
|        +.       |
|         o       |
|                 |
|                 |
+----[SHA256]-----+
```

PermitRootLogin no

ChallengeResponseAuthentication no

PasswordAuthentication no

UsePAM no

Port 300

iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m recent --set --name ssh –rsource

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

```
student@CsnKhai:~$ ssh-keygen --help
unknown option -- -
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          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.
student@CsnKhai:~$
```

ssh-keygen -t rsa -b 4096 -o -a 250

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

ssh student@192.168.1.6 -R 8080:localhost:80