

TASK5.3

Part1

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

Running, Waiting, Terminated, Ready, Created

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

```
root@CsnKhai:/home/student# 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-udev
      |--upstart-file-br
      |--upstart-socket-
      |--upstart-udev-br
```

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

```
root@CsnKhai:/home/student# cat /proc/cpuinfo
processor       : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 158
model name    : Intel(R) Core(TM) i5-9400F CPU @ 2.90GHz
stepping      : 10
cpu MHz       : 2903.973
cache size    : 9216 KB
physical id   : 0
siblings      : 1
core id       : 0
cpu cores     : 1
apicid        : 0
initial apicid : 0
fdiv_bug      : no
f00f_bug      : no
coma_bug      : no
fpu           : yes
fpu_exception : yes
cpuid level   : 22
wp            : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush
h mmx fxsr sse sse2 ht nx rdtscp constant_tsc xtopology nonstop_tsc pni pclmulqdq monitor ssse3
cx16 pcid sse4_1 sse4_2 movbe popcnt aes xsave avx rdrand lahf_lm abm 3dnowprefetch fsgsbase bmi
1 avx2 bmi2 invpcid rdseed
bogomips      : 5807.94
clflush size  : 64
cache_alignment : 64
address sizes  : 39 bits physical, 48 bits virtual
power management:
```

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

```
root@CsnKhai:/home/student# ps -axu
USER      PID  %CPU  %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1   0.0   0.8   4196   2188 ?        Ss   10:13   0:00 /sbin/init
root         2   0.0   0.0      0      0 ?        S    10:13   0:00 [kthreadd]
root         3   0.0   0.0      0      0 ?        S    10:13   0:00 [ksoftirqd/0]
root         4   0.0   0.0      0      0 ?        S    10:13   0:00 [kworker/0:0]
root         5   0.0   0.0      0      0 ?        S<   10:13   0:00 [kworker/0:0H]
root         6   0.0   0.0      0      0 ?        S    10:13   0:00 [kworker/u2:0]
root         7   0.0   0.0      0      0 ?        S    10:13   0:00 [rcu_sched]
root         8   0.0   0.0      0      0 ?        S    10:13   0:00 [rcu_bh]
root         9   0.0   0.0      0      0 ?        S    10:13   0:00 [migration/0]
root        10   0.0   0.0      0      0 ?        S    10:13   0:00 [watchdog/0]
root        11   0.0   0.0      0      0 ?        S<   10:13   0:00 [khelper]
root        12   0.0   0.0      0      0 ?        S    10:13   0:00 [kdevtmpfs]
root        13   0.0   0.0      0      0 ?        S<   10:13   0:00 [netns]
root        14   0.0   0.0      0      0 ?        S<   10:13   0:00 [writeback]
```

6. How to define kernel processes and user processes?

```
root@CsnKhai:/home/student# ps aux | grep '\[k'
root         2   0.0   0.0      0      0 ?        S    10:13   0:00 [kthreadd]
root         3   0.0   0.0      0      0 ?        S    10:13   0:00 [ksoftirqd/0]
root         4   0.0   0.0      0      0 ?        S    10:13   0:00 [kworker/0:0]
root         5   0.0   0.0      0      0 ?        S<   10:13   0:00 [kworker/0:0H]
root         6   0.0   0.0      0      0 ?        S    10:13   0:00 [kworker/u2:0]
root         7   0.0   0.0      0      0 ?        S    10:13   0:00 [rcu_sched]
root         8   0.0   0.0      0      0 ?        S    10:13   0:00 [rcu_bh]
root         9   0.0   0.0      0      0 ?        S    10:13   0:00 [migration/0]
root        10   0.0   0.0      0      0 ?        S    10:13   0:00 [watchdog/0]
root        11   0.0   0.0      0      0 ?        S<   10:13   0:00 [khelper]
root        12   0.0   0.0      0      0 ?        S    10:13   0:00 [kdevtmpfs]
root        13   0.0   0.0      0      0 ?        S<   10:13   0:00 [netns]
root        14   0.0   0.0      0      0 ?        S<   10:13   0:00 [writeback]
root        15   0.0   0.0      0      0 ?        S<   10:13   0:00 [kintegrityd]
root        16   0.0   0.0      0      0 ?        S<   10:13   0:00 [bioset]
```

```
root@CsnKhai:/home/student# ps -u
USER      PID  %CPU  %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root       708   0.0   0.3   4644   824 tty4      Ss+  10:13   0:00 /sbin/getty -8 38400 tty4
root       710   0.0   0.3   4644   828 tty5      Ss+  10:13   0:00 /sbin/getty -8 38400 tty5
root       713   0.0   0.3   4644   832 tty2      Ss+  10:13   0:00 /sbin/getty -8 38400 tty2
root       714   0.0   0.3   4644   836 tty3      Ss+  10:13   0:00 /sbin/getty -8 38400 tty3
root       716   0.0   0.3   4644   828 tty6      Ss+  10:13   0:00 /sbin/getty -8 38400 tty6
root       809   0.0   0.3   4644   828 tty1      Ss+  10:13   0:00 /sbin/getty -8 38400 tty1
root       882   0.0   0.8   6740  2024 pts/0    S    10:34   0:00 sudo su
root       883   0.0   0.6   6304  1596 pts/0    S    10:34   0:00 su
root       884   0.0   0.8   5736  1996 pts/0    S    10:34   0:00 bash
root      1011   0.0   0.4   5216  1160 pts/0    R+   11:37   0:00 ps -u
root@CsnKhai:/home/student#
```

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:/home/student# ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.8  4196  2188 ?        Ss   10:13   0:00 /sbin/init
root         2  0.0  0.0      0     0 ?        S    10:13   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    10:13   0:00 [ksoftirqd/0]
root         4  0.0  0.0      0     0 ?        S    10:13   0:00 [kworker/0:0]
root         5  0.0  0.0      0     0 ?        S<   10:13   0:00 [kworker/0:0H]
root         6  0.0  0.0      0     0 ?        S    10:13   0:00 [kworker/u2:0]
root         7  0.0  0.0      0     0 ?        S    10:13   0:00 [rcu_sched]
```

8. Display only the processes of a specific user.

```
root@CsnKhai:/home/student# ps -u student
PID TTY      TIME CMD
846 ?        00:00:00 sshd
847 pts/0    00:00:00 bash
879 ?        00:00:00 sshd
880 ?        00:00:00 sftp-server
root@CsnKhai:/home/student#
root@CsnKhai:/home/student#
root@CsnKhai:/home/student#
```

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

```
root@CsnKhai:/home/student# pgrep
pgrep: no matching criteria specified
Try 'pgrep --help' for more information.
root@CsnKhai:/home/student#
root@CsnKhai:/home/student# pstree
init--cron
    |--dbus-daemon
    |--dhclient
    |--6*[getty]
    |--rsyslogd--3*[{rsyslogd}]
    |--sshd--sshd--sshd--bash--sudo--su--b
    |   |--sshd--sshd--sftp-server
    |--systemd-logind
    |--systemd-udevd
    |--upstart-file-br
    |--upstart-socket-
    |--upstart-udev-br
root@CsnKhai:/home/student#
root@CsnKhai:/home/student#
root@CsnKhai:/home/student# top
```

10. What information does top command display?

```
top - 11:39:10 up 1:25, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 68 total, 1 running, 67 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 99.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 247792 total, 126512 used, 121280 free, 15268 buffers
KiB Swap: 0 total, 0 used, 0 free, 73520 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    1 root        20   0   4196    2188    1392 S   0.0   0.9   0:00.77 init
    2 root        20   0       0       0       0 S   0.0   0.0   0:00.00 kthreadd
    3 root        20   0       0       0       0 S   0.0   0.0   0:00.00 ksoftirqd/0
    4 root        20   0       0       0       0 S   0.0   0.0   0:00.00 kworker/0:0
    5 root         0 -20    0       0       0 S   0.0   0.0   0:00.00 kworker/0:0H
    6 root        20   0       0       0       0 S   0.0   0.0   0:00.12 kworker/u2:0
    7 root        20   0       0       0       0 S   0.0   0.0   0:00.13 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   0:00.07 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
```

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

```
root@CsnKhai:/home/student# top --help
top: inappropriate '-help'
Usage:
  top -hv | -bcHiOSs -d secs -n max -u|U user -p pid(s) -o field -w [cols]
root@CsnKhai:/home/student#
root@CsnKhai:/home/student#
```

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

To set priority we use command `-nice-`

`nice _command_ -n priority -p process-id`

16. Examine the kill command. How to send with the kill command

process control signal? Give an example of commonly used signals.

To get list of available signals we use `kill -l`

```
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:~$
student@CsnKhai:~$
```

If we need to kill process we should know PIDs of process, for example firefox

input "pidof firefox"

output "6252 2525 2525 2525"

kill -9 6252 2525 2525 2525

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 – show running and suspended jobs

fg – take back job to front.

bg – continues suspended job

nohup – run immune job (process will continue even if the terminal is close)