***Матеріал підготував Довбешко В.А.***

- *Чим відрізняється фоновий процес від звичайного. Де вони використовуються?
  A background process is a process that runs in the background, i.e. without the active participation of the user. It can be started when the operating system boots or to perform long operations that can take a long time, and a normal process is a process that runs in active mode when the user interacts with the system or program.

- *Опишіть наступні команди та поясніть що вони виконують – команда jobs, bg, fg.

**Jobs.**

This command is used to display a list of current tasks (processes) that are running in the shell and are either in the background or paused.
Each job is assigned a unique identifier that is used to refer to it during further management.

**bg**

Switches a background process to the background execution state. This means that the process will continue to run, but will not block the terminal from entering other commands. Syntax: 'bg% job _ id'.

**fg**

Used to switch a background task to active execution mode, i.e. bring the background process to the foreground.
This can be useful if you want to interact with a task that was previously moved to the background, for example, debug it or view the output.

- **Якою командою можна переглянути інформацію про запущені в системи фонові процеси та задачі?

You can view it using the jobs command. It displays a list of active processes and their states, including the process ID, state, and the command associated with each process.

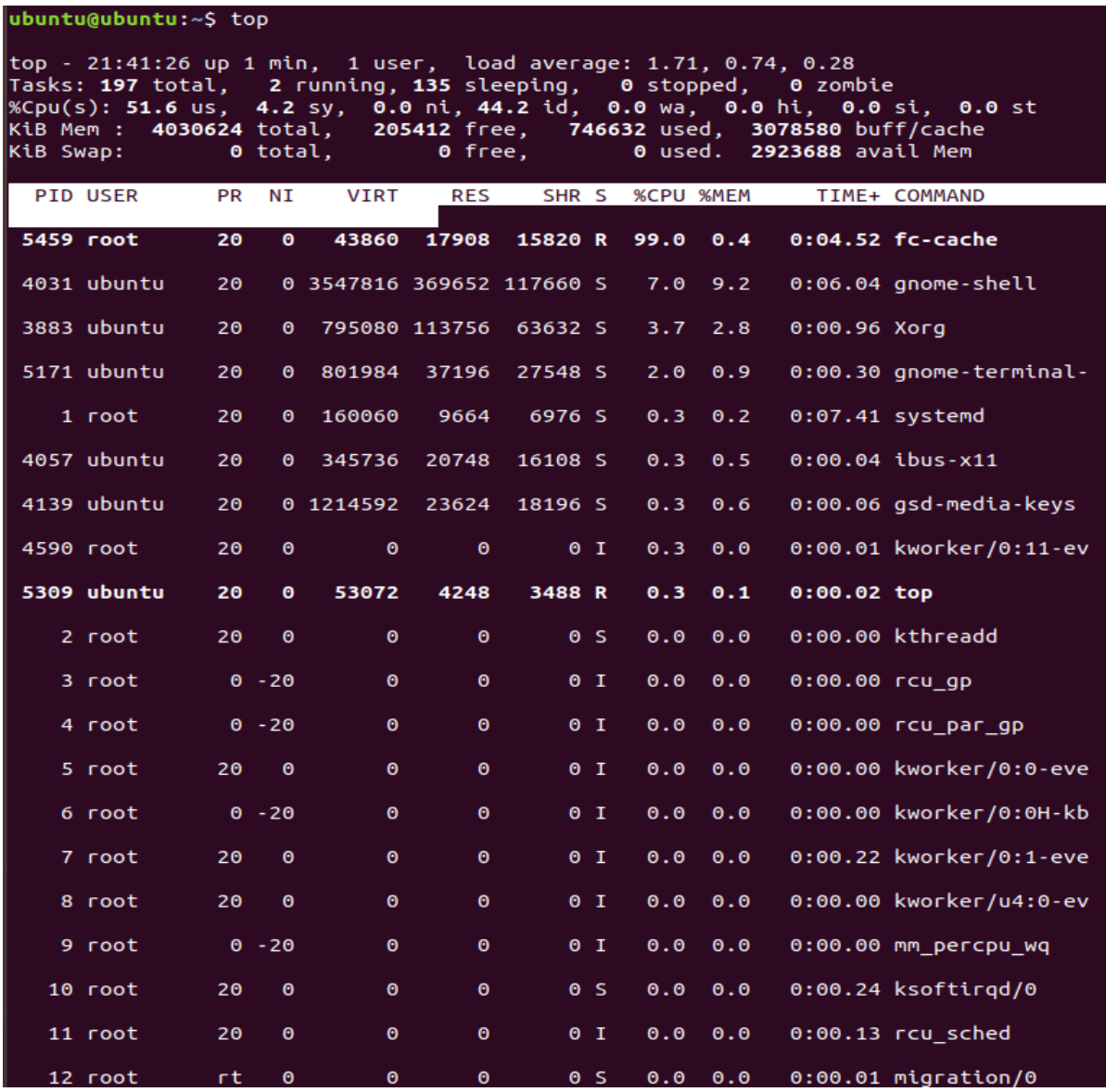
- **Як призупинити фоновий процес, як його потім відновити та при необхідності перезапусти?

You can use the keyboard shortcut **Ctrl+Z** to pause a process, or the **fg** command, for example: **fg% 1**
This will put the background process with ID 1 in the active execution state. To restart a background process that has been paused or that has completed execution, you need to start it again, usually with the same command or another appropriate command.

1. Запустіть термінал, та в командному рядку виконайте наступні дії для ознайомлення з роботою з процесами:

***Матеріал підготував Довбешко В.А.***

- запустіть команду top, проаналізуйте отриманий в цій команді результат та охарактеризуйте найбільш активні процеси у системі;




After analyzing the "top" command, we saw how much memory is available, used, free, and cached. Here are the most active processes:

**fc-cache:** This process is used to cache fonts. It uses 99% of CPU and 0.4% of memory.

**gnome-shell:** This is the graphical shell of GNOME. It uses 7% of CPU and 9.2% of memory.

**gnome-terminal :** This is the GNOME terminal. It uses 2% CPU and 0.9% memory.

- призупинити виконання команди top (треба використати комбінацію клавіш);
  За допомогою комбінації клавіш Ctrl+Z призупиняємо виконання команди:

```
  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 oom_reaper

  27 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 writeback

  28 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kcompactd0

  29 root      25   5       0      0      0 S   0.0   0.0   0:00.00 ksmd

  30 root      39  19       0      0      0 S   0.0   0.0   0:00.00 khugepaged

  35 root      20   0       0      0      0 I   0.0   0.0   0:00.12 kworker/1:1-eve

  77 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kintegrityd

  78 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kblockd

[1]+  Stopped                 top
ubuntu@ubuntu:~$
```

- вивести інформацію про процеси за допомогою команди ps;

```
  78 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kblockd

[1]+  Stopped                 top
ubuntu@ubuntu:~$ ps
  PID TTY          TIME CMD
 5237 pts/0    00:00:00 bash
 5309 pts/0    00:00:00 top
 5844 pts/0    00:00:00 ps
ubuntu@ubuntu:~$
```

- *наведіть 5 прикладів з використанням різних параметрів команди ps (наприклад, вивести тільки системні процеси, вивести процеси конкретного користувача, вивести дерево процесів тощо). Опишіть, що саме роблять обрані Вами параметри

```
ubuntu@ubuntu:~$ ps -e
  PID TTY          TIME CMD
    1 ?        00:00:07 systemd
    2 ?        00:00:00 kthreadd
    3 ?        00:00:00 rcu_gp
    4 ?        00:00:00 rcu_par_gp
    6 ?        00:00:00 kworker/0:0H-kb
    9 ?        00:00:00 mm_percpu_wq
   10 ?        00:00:00 ksoftirqd/0
   11 ?        00:00:00 rcu_sched
   12 ?        00:00:00 migration/0
   13 ?        00:00:00 idle_inject/0
   14 ?        00:00:00 cpuhp/0
   15 ?        00:00:00 cpuhp/1
   16 ?        00:00:00 idle_inject/1
   17 ?        00:00:00 migration/1
   18 ?        00:00:00 ksoftirqd/1
   20 ?        00:00:00 kworker/1:0H-kb
   21 ?        00:00:00 kdevtmpfs
   22 ?        00:00:00 netns
   23 ?        00:00:00 rcu_tasks_kthre
   24 ?        00:00:00 kauditd
   25 ?        00:00:00 khungtaskd
   26 ?        00:00:00 oom_reaper
   27 ?        00:00:00 writeback
   28 ?        00:00:00 kcompactd0
   29 ?        00:00:00 ksmd
   30 ?        00:00:00 khugepaged
   77 ?        00:00:00 kintegrityd
   78 ?        00:00:00 kblockd
   79 ?        00:00:00 blkcg_punt_bio
   80 ?        00:00:00 tpm_dev_wq
   81 ?        00:00:00 ata_sff
   82 ?        00:00:00 md
   83 ?        00:00:00 edac-poller
   84 ?        00:00:00 devfreq_wq
   85 ?        00:00:00 watchdogd
   89 ?        00:00:00 kswapd0
   90 ?        00:00:00 ecryptfs-kthrea
   92 ?        00:00:00 kthrotld
   93 ?        00:00:00 acpi_thermal_pm
   94 ?        00:00:00 scsi_eh_0
   95 ?        00:00:00 scsi_tmf_0
   96 ?        00:00:00 scsi_eh_1
   97 ?        00:00:00 scsi_tmf_1
```

The **ps -e** option displays all processes that exist on the system, including system processes.

```
ubuntu@ubuntu:~$ ps -ejH
  PID  PGID   SID TTY          TIME CMD
    2     0     0 ?        00:00:00 kthreadd
    3     0     0 ?        00:00:00   rcu_gp
    4     0     0 ?        00:00:00   rcu_par_gp
    6     0     0 ?        00:00:00   kworker/0:0H-kb
    9     0     0 ?        00:00:00   mm_percpu_wq
   10     0     0 ?        00:00:00   ksoftirqd/0
   11     0     0 ?        00:00:00   rcu_sched
   12     0     0 ?        00:00:00   migration/0
   13     0     0 ?        00:00:00   idle_inject/0
   14     0     0 ?        00:00:00   cpuhp/0
   15     0     0 ?        00:00:00   cpuhp/1
   16     0     0 ?        00:00:00   idle_inject/1
   17     0     0 ?        00:00:00   migration/1
   18     0     0 ?        00:00:00   ksoftirqd/1
   20     0     0 ?        00:00:00   kworker/1:0H-kb
   21     0     0 ?        00:00:00   kdevtmpfs
   22     0     0 ?        00:00:00   netns
   23     0     0 ?        00:00:00   rcu_tasks_kthre
   24     0     0 ?        00:00:00   kauditd
   25     0     0 ?        00:00:00   khungtaskd
   26     0     0 ?        00:00:00   oom_reaper
   27     0     0 ?        00:00:00   writeback
   28     0     0 ?        00:00:00   kcompactd0
   29     0     0 ?        00:00:00   ksmd
   30     0     0 ?        00:00:00   khugepaged
   77     0     0 ?        00:00:00   kintegrityd
   78     0     0 ?        00:00:00   kblockd
   79     0     0 ?        00:00:00   blkcg_punt_bio
   80     0     0 ?        00:00:00   tpm_dev_wq
   81     0     0 ?        00:00:00   ata_sff
   82     0     0 ?        00:00:00   md
   83     0     0 ?        00:00:00   edac-poller
   84     0     0 ?        00:00:00   devfreq_wq
   85     0     0 ?        00:00:00   watchdogd
   89     0     0 ?        00:00:00   kswapd0
   90     0     0 ?        00:00:00   ecryptfs-kthrea
   92     0     0 ?        00:00:00   kthrotld
   93     0     0 ?        00:00:00   acpi_thermal_pm
   94     0     0 ?        00:00:00   scsi_eh_0
   95     0     0 ?        00:00:00   scsi_tmf_0
   96     0     0 ?        00:00:00   scsi_eh_1
   97     0     0 ?        00:00:00   scsi_tmf_1
```

The **ps -ejH** option allows you to display a process tree showing parent and child processes. This helps to understand the hierarchy of process execution in the system.

```
ubuntu@ubuntu:~$ ps -a
  PID TTY          TIME CMD
 3883 tty1     00:00:08 Xorg
 3897 tty1     00:00:00 gnome-session-b
 4031 tty1     00:00:23 gnome-shell
 4049 tty1     00:00:00 ibus-daemon
 4053 tty1     00:00:00 ibus-dconf
 4057 tty1     00:00:00 ibus-x11
 4087 tty1     00:00:00 gsd-power
 4088 tty1     00:00:00 gsd-print-notif
 4091 tty1     00:00:00 gsd-rfkill
 4093 tty1     00:00:00 gsd-screensaver
 4096 tty1     00:00:00 gsd-sharing
 4102 tty1     00:00:00 gsd-smartcard
 4106 tty1     00:00:00 gsd-sound
 4109 tty1     00:00:00 gsd-xsettings
 4115 tty1     00:00:00 gsd-wacom
 4120 tty1     00:00:00 gsd-a11y-settin
 4121 tty1     00:00:00 gsd-clipboard
 4124 tty1     00:00:00 gsd-color
 4125 tty1     00:00:00 gsd-datetime
 4130 tty1     00:00:00 gsd-housekeepin
 4132 tty1     00:00:00 gsd-keyboard
 4139 tty1     00:00:00 gsd-media-keys
 4144 tty1     00:00:00 gsd-mouse
 4167 tty1     00:00:00 gsd-printer
 4195 tty1     00:00:00 gsd-disk-utilit
 4206 tty1     00:00:03 nautilus-deskto
 4285 tty1     00:00:00 ibus-engine-sim
 5251 tty1     00:00:00 update-notifier
 5309 pts/0    00:00:00 top
 5692 tty1     00:00:00 deja-dup-monito
 5856 pts/0    00:00:00 ps
ubuntu@ubuntu:~$
```

The **ps - a** parameter shows information about all processes from all shells. It shows both active and suspended processes.

```
ubuntu@ubuntu:~$ ps -s
  UID   PID         PENDING         BLOCKED         IGNORED         CAUGHT STAT TTY        TIME COMMAND
  999  3881 0000000000000000 0000000000000000 0000000000001000 0000000180014000 Ssl+ tty1       0:00 /usr/lib/gdm3/gdm-x-session --
  999  3883 0000000000000000 0000000000000000 0000000000001000 00000001c18066ef Sl+  tty1       0:08 /usr/lib/xorg/Xorg vt1 -displa
  999  3897 0000000000000000 0000000000000000 0000000000001000 0000000180014a02 Sl+  tty1       0:00 /usr/lib/gnome-session/gnome-s
  999  4031 0000000000000000 0000000000000000 0000000001001000 00000001820144b0 Sl+  tty1       0:23 /usr/bin/gnome-shell
  999  4049 0000000000000000 0000000000000000 0000000001001000 0000000180010000 Sl   tty1       0:00 ibus-daemon --xim --panel disa
  999  4053 0000000000000000 0000000000000000 0000000001001000 0000000180000000 Sl   tty1       0:00 /usr/lib/ibus/ibus-dconf
  999  4057 0000000000000000 0000000000000000 0000000001001000 0000000180004002 Sl   tty1       0:00 /usr/lib/ibus/ibus-x11 --kill-
  999  4087 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-settings-daemon
  999  4088 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-settings-daemon
  999  4091 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-settings-daemon
  999  4093 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-settings-daemon
  999  4096 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-settings-daemon
  999  4102 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-settings-daemon
  999  4106 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-settings-daemon
  999  4109 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-settings-daemon
  999  4115 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-settings-daemon
  999  4120 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-settings-daemon
  999  4121 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-settings-daemon
  999  4124 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-settings-daemon
  999  4125 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-settings-daemon
  999  4130 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-settings-daemon
  999  4132 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-settings-daemon
  999  4139 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-settings-daemon
  999  4144 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-settings-daemon
  999  4167 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-settings-daemon
  999  4195 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/gnome-disk-utility/gs
  999  4206 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:03 nautilus-desktop
  999  4285 0000000000000000 0000000000000000 0000000001001000 0000000180000000 Sl   tty1       0:00 /usr/lib/ibus/ibus-engine-simp
  999  5237 0000000000000000 0000000000010000 0000000000380004 000000004b817efb Ss   pts/0      0:00 bash
  999  5251 0000000000000000 0000000000000000 0000000000001000 0000000180000002 Sl+  tty1       0:00 update-notifier
  999  5309 0000000000000000 fffffffe7ffbfeff 0000000000000000 ffffffffffffafeff T    pts/0      0:00 top
  999  5692 0000000000000000 0000000000000000 0000000000001000 0000000180000000 Sl+  tty1       0:00 /usr/lib/deja-dup/deja-dup-mon
  999  5857 0000000000000000 0000000000000000 0000000000000000 00000001f3d1fef9 R+   pts/0      0:00 ps -s
ubuntu@ubuntu:~$
```

Typically, the **ps -s** option is used to display information about the status of a process.

```
ubuntu@ubuntu:~$ ps -1
  PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:07 /sbin/init splash --- maybe-ubiquity
ubuntu@ubuntu:~$
```

The **ps -1** parameter displays extended information about processes in a long format.