

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Липецкий Государственный Технический Университет
Факультет автоматизации и информатики
Кафедра автоматизированных систем управления

Лабораторная работа
по информатике №5
“ Управление процессами в операционной системе Linux”

Студент

(подпись, дата)

Станиславчук С. М.

Группа АС-21-1

Руководитель

Доцент, кандидат наук

(подпись, дата)

Харитоненко А. А.

Липецк 2022 г.

Задание 1

0. Создадим шел-скрипт script.sh

```
cat > script.sh
```

1. Сделаем его исполняемым.

```
chmod +x script.sh
```

Запустим в фоновом режиме

```
sh script.sh &
```

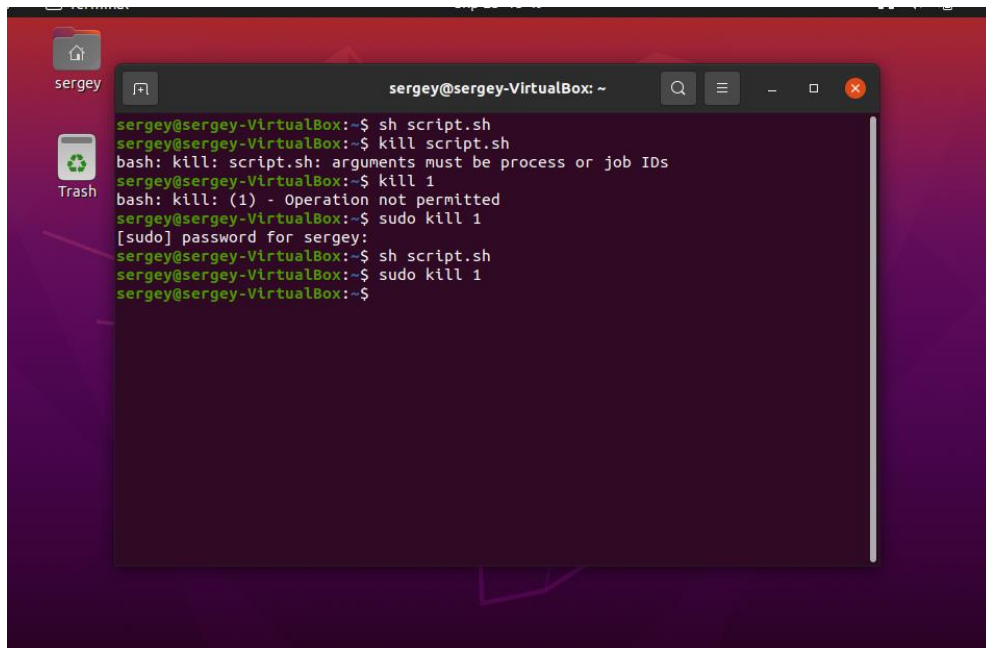
2. Запустим на переднем плане, приостановим выполнение программы, заново запустим и завершим

```
sh script.sh
```

```
kill 1
```

```
sh script.sh
```

```
kill 1
```



The screenshot shows a terminal window titled 'sergey@sergey-VirtualBox: ~'. The user 'sergey' is logged in. The terminal output is as follows:

```
sergey@sergey-VirtualBox:~$ sh script.sh
sergey@sergey-VirtualBox:~$ kill script.sh
bash: kill: script.sh: arguments must be process or job IDs
sergey@sergey-VirtualBox:~$ kill 1
bash: kill: (1) - Operation not permitted
sergey@sergey-VirtualBox:~$ sudo kill 1
[sudo] password for sergey:
sergey@sergey-VirtualBox:~$ sh script.sh
sergey@sergey-VirtualBox:~$ sudo kill 1
sergey@sergey-VirtualBox:~$
```

5. Перевод выполняемого процесса на передний план и наоборот

jobs – проверка состояний процессов

```
sh script.sh &
```

```
fg 1
```

```
sudo kill 1
```

6. ./script.sh – запустили процесс

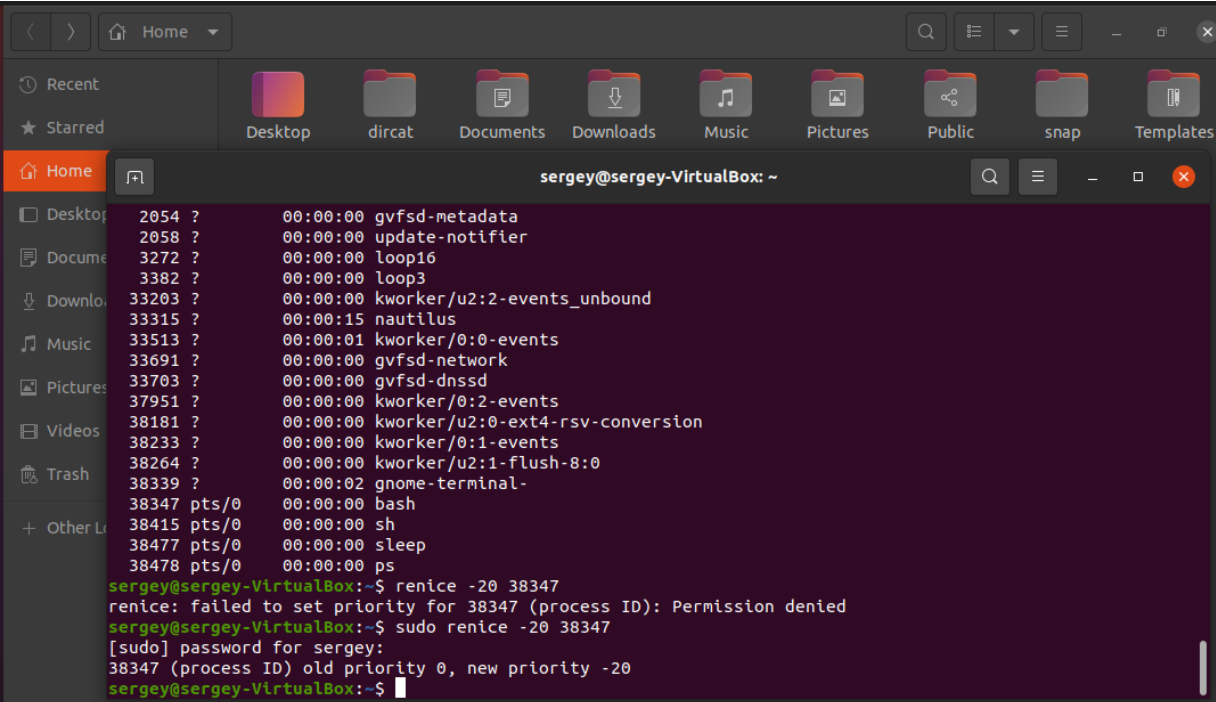
bg 1 – перевел процесс в фоновый режим

7. jobs – проверили состояние процессов, Running

8. nohup script.sh & – для работы процесса, даже после отключения терминала

Задание 2.

1. Для просмотра текущих процессов используем команду `ps -r`
2. Для просмотра всех процессов `ps -A`
3. Для просмотра процессов, запущенного конкретным пользователем `ps -U user`
4. Создал простой скрипт с бесконечным циклом выполнения.
5. При вызове листинга характеристик процессов появились новые процессы: `bash`, `sh`, `sleep`
- 6.



The screenshot shows a terminal window titled 'sergey@sergey-VirtualBox: ~'. The terminal displays the output of the `ps` command, listing various system and user processes. The processes listed include `gvfsd-metadata`, `update-notifier`, `loop16`, `loop3`, `kworker/u2:2-events_unbound`, `nautilus`, `kworker/0:0-events`, `gvfsd-network`, `gvfsd-dnssd`, `kworker/0:2-events`, `kworker/u2:0-ext4-rsv-conversion`, `kworker/0:1-events`, `kworker/u2:1-flush-8:0`, `gnome-terminal`, `bash`, `sh`, `sleep`, and `ps`. The terminal also shows the execution of the `renice -20 38347` command, which fails with the message 'Permission denied'. Subsequently, the `sudo renice -20 38347` command is executed, and the user is prompted for a password. After entering the password, the command succeeds, changing the priority of process 38347 to -20.

```
sergey@sergey-VirtualBox: ~  
ps -A  
2054 ?        00:00:00 gvfsd-metadata  
2058 ?        00:00:00 update-notifier  
3272 ?        00:00:00 loop16  
3382 ?        00:00:00 loop3  
33203 ?       00:00:00 kworker/u2:2-events_unbound  
33315 ?       00:00:15 nautilus  
33513 ?       00:00:01 kworker/0:0-events  
33691 ?       00:00:00 gvfsd-network  
33703 ?       00:00:00 gvfsd-dnssd  
37951 ?       00:00:00 kworker/0:2-events  
38181 ?       00:00:00 kworker/u2:0-ext4-rsv-conversion  
38233 ?       00:00:00 kworker/0:1-events  
38264 ?       00:00:00 kworker/u2:1-flush-8:0  
38339 ?       00:00:02 gnome-terminal-  
38347 pts/0      00:00:00 bash  
38415 pts/0      00:00:00 sh  
38477 pts/0      00:00:00 sleep  
38478 pts/0      00:00:00 ps  
sergey@sergey-VirtualBox:~$ renice -20 38347  
renice: failed to set priority for 38347 (process ID): Permission denied  
sergey@sergey-VirtualBox:~$ sudo renice -20 38347  
[sudo] password for sergey:  
38347 (process ID) old priority 0, new priority -20  
sergey@sergey-VirtualBox:~$
```

8. После выхода из терминала процесс выполнения скрипта пропадает
9. При помощи `screen` делаем защиту нашего процесса.
10. `screen script`
`sudo renice -20 script`
11. Избавимся от процесса, командой `kill`
12. `sh script.sh > file.txt`

13. `sh script.sh > file.txt &`
`fg`

ps -A

```
sergey@sergey-VirtualBox: ~  
1160 ?      00:00:00 gsd-usb-protect  
1163 ?      00:00:00 gsd-wacom  
1164 ?      00:00:00 gsd-wwan  
1174 ?      00:00:00 gsd-xsettings  
1188 ?      00:00:00 evolution-alarm  
1217 ?      00:00:00 gsd-printer  
1235 ?      00:00:00 colord  
1259 ?      00:00:05 snap-store  
1284 ?      00:00:00 ibus-engine-sim  
1285 ?      00:00:00 xdg-document-po  
1400 ?      00:00:00 xdg-desktop-por  
1408 ?      00:00:00 xdg-desktop-por  
1417 ?      00:00:00 fwupd  
1828 ?      00:00:00 gvfsd-metadata  
1831 ?      00:00:00 update-notifier  
2864 ?      00:00:00 kworker/0:0-events  
3021 ?      00:00:00 kworker/u2:0-events_power_efficient  
3072 ?      00:00:00 kworker/0:1-cgroup_destroy  
3101 ?      00:00:00 gnome-terminal-  
3112 pts/0   00:00:00 bash  
3138 pts/0   00:00:00 sh  
3173 pts/0   00:00:00 sleep  
3177 pts/0   00:00:00 ps  
sergey@sergey-VirtualBox:~$
```

15. Terminated

```
sergey@sergey-VirtualBox: ~  
1417 ?      00:00:00 fwupd  
1828 ?      00:00:00 gvfsd-metadata  
1831 ?      00:00:00 update-notifier  
3021 ?      00:00:00 kworker/u2:0-ext4-rsv-conversion  
3072 ?      00:00:00 kworker/0:1-cgroup_destroy  
3101 ?      00:00:03 gnome-terminal-  
3112 pts/0   00:00:00 bash  
3138 pts/0   00:00:00 sh  
3178 ?      00:00:00 kworker/u2:1-events_unbound  
3179 ?      00:00:00 kworker/u2:3-events_unbound  
3265 pts/0   00:00:00 sh  
3327 pts/0   00:00:00 sh  
3333 ?      00:00:00 tracker-store  
3343 ?      00:00:00 tracker-extract  
3370 pts/0   00:00:00 sleep  
3371 pts/0   00:00:00 sleep  
3372 pts/0   00:00:00 sleep  
3373 pts/0   00:00:00 ps  
sergey@sergey-VirtualBox:~$ bg  
bash: bg: job 3 already in background  
sergey@sergey-VirtualBox:~$ kill 3138  
sergey@sergey-VirtualBox:~$ kill 3265  
[1] Terminated sh script.sh > file.txt  
sergey@sergey-VirtualBox:~$ kill 3327
```