

**Липецкий государственный технический университет**

Факультет автоматизации и информатики Кафедра  
автоматизированных систем управления

**ЛАБОРАТОРНАЯ РАБОТА №3**

по курсу «Операционная система Linux»  
на тему «Управление процессами в Ubuntu.»

Студент

Киренский Д. К.

Группа ПИ-19-1

Руководитель

Доцент

Кургасов В.В.

Липецк 2021 г.

## Задание

1. Запустить программу виртуализации Oracle VM VirtualBox.
2. Запустить виртуальную машину Ubuntu.
3. Открыть окно интерпретатора команд
4. Вывести общую информацию о системе
  - (a) Вывести информацию о текущем интерпретаторе команд
  - (b) Вывести информацию о текущем пользователе
  - (c) Вывести информацию о текущем каталоге
  - (d) Вывести информацию об оперативной памяти и области подкачки
  - (e) Вывести информацию о дисковой памяти
5. Выполнить команды получения информации о процессах
  - (a) Получить идентификатор текущего процесса(PID)
  - (b) Получить идентификатор родительского процесса(PPID)
  - (c) Получить идентификатор процесса инициализации системы
  - (d) Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе команд
    - (e) Отобразить все процессы
6. Выполнить команды управления процессами
  - (a) Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе
    - (b) Определить текущее значение nice по умолчанию
    - (c) Запустить интерпретатор bash с понижением приоритета nice  
-n 10 bash
    - (d) Определить PID запущенного интерпретатора
    - (e) Установить приоритет запущенного интерпретатора равным 5 renice -n 5  
<PID процесса>
  - (f) Получить информацию о процессах bash ps lax | grep bash
7. Задание для варианта номер 5
  - (a) Отобразить информацию о процессах указанного пользователя в виде иерархии, вывод отсортировать по значениям PID.

(b) С помощью сигнала SIGSTOP приостановить выполнение процесса, владельцем которого является текущий пользователь. Через несколько секунд возобновить выполнение процесса.

(c) Определить идентификаторы и имена процессов, не связанных с указанным терминалом.

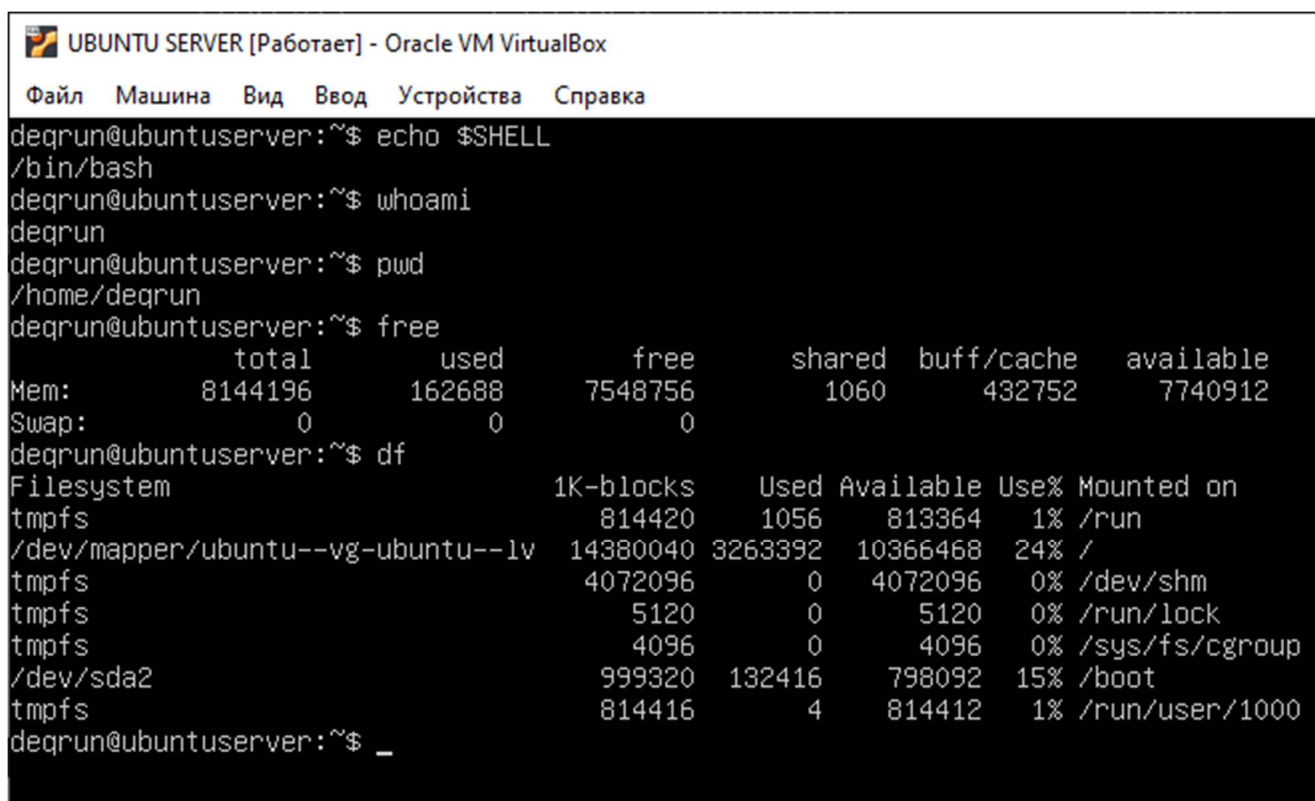
(d) В отчете предоставьте все шаги ваших действий. То есть следует привести следующее: текст задания, а следом за ним снимок экрана консоли с результатами выполнения задания. Кроме того, перед скриншотом следует привести текстовую запись использованных команд. Кратко поясните результаты выполнения всех команд.

## Оглавление

Задание .....	1
1. Общая информация.....	4
2. Команды получения информации о процессах .....	4
3. Команды управления процессами.....	6
4. Задание для варианта номер 5.....	7
Заключение.....	10

## 1. Общая информация

- 1.1 Вывести информацию о текущем интерпретаторе команд (команда `echo $SHELL`)
- 1.2 Вывести информацию о текущем пользователе (команда `whoami`)
- 1.3 Вывести информацию о текущем каталоге (команда `pwd`)
- 1.4 Вывести информацию об оперативной памяти и области подкачки (команда `free`)
- 1.5 Вывести информацию о дисковой памяти (команда `df`)



```
UBUNTU SERVER [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
degrun@ubuntuserver:~$ echo $SHELL
/bin/bash
degrun@ubuntuserver:~$ whoami
degrun
degrun@ubuntuserver:~$ pwd
/home/degrun
degrun@ubuntuserver:~$ free
              total        used         free       shared  buff/cache   available
Mem:      8144196      162688      7548756          1060        432752      7740912
Swap:           0             0             0
degrun@ubuntuserver:~$ df
Filesystem                1K-blocks    Used Available Use% Mounted on
tmpfs                      814420      1056    813364   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 14380040 3263392 10366468 24% /
tmpfs                     4072096         0    4072096   0% /dev/shm
tmpfs                      5120         0       5120   0% /run/lock
tmpfs                      4096         0       4096   0% /sys/fs/cgroup
/dev/sda2                 999320    132416    798092  15% /boot
tmpfs                     814416         4    814412   1% /run/user/1000
degrun@ubuntuserver:~$ _
```

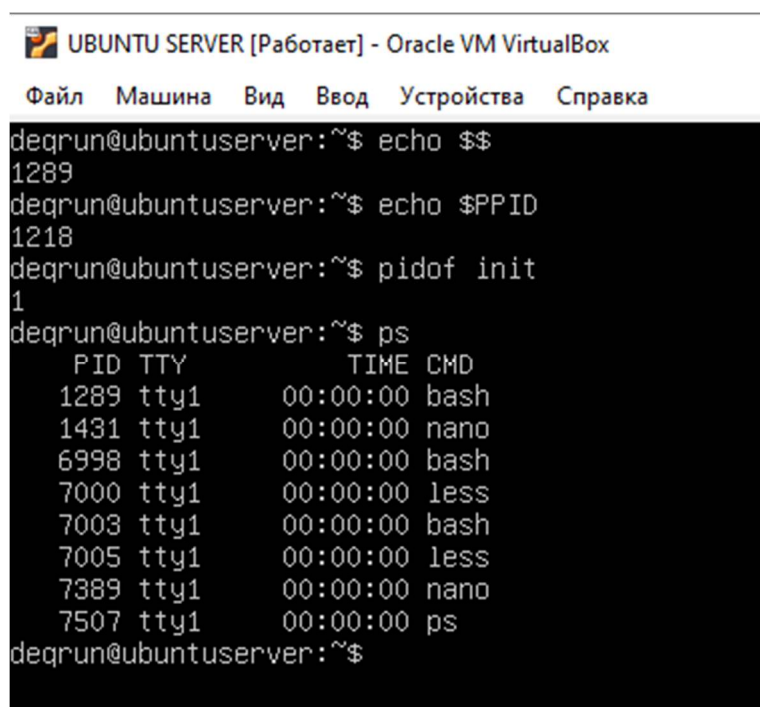
Рисунок 1 – общая информация.

## 2. Команды получения информации о процессах

- 2.1 Получить идентификатор текущего процесса(PID) (команда `echo $$`)
- 2.2 Получить идентификатор родительского процесса(PPID) (команда `echo $PPID`)
- 2.3 Получить идентификатор процесса инициализации системы (команда `pidof init`)
- 2.4 Получить информацию о выполняющихся процессах текущего (команда `ps`)

пользователя в текущем интерпретаторе команд

## 2.5 Отобразить все процессы (команда `ps -e`)



The screenshot shows a terminal window titled "UBUNTU SERVER [Работает] - Oracle VM VirtualBox". The menu bar includes "Файл", "Машина", "Вид", "Ввод", "Устройства", and "Справка". The terminal output shows the user 'deqrun' at 'ubuntuserver' with a tilde prompt. The user enters 'echo \$\$' and gets '1289'. Then 'echo \$PPID' and gets '1218'. Then 'pidof init' and gets '1'. Finally, 'ps' is entered, resulting in a table of processes.

```
deqrun@ubuntuserver:~$ echo $$
1289
deqrun@ubuntuserver:~$ echo $PPID
1218
deqrun@ubuntuserver:~$ pidof init
1
deqrun@ubuntuserver:~$ ps
  PID TTY          TIME CMD
 1289 tty1        00:00:00 bash
 1431 tty1        00:00:00 nano
 6998 tty1        00:00:00 bash
 7000 tty1        00:00:00 less
 7003 tty1        00:00:00 bash
 7005 tty1        00:00:00 less
 7389 tty1        00:00:00 nano
 7507 tty1        00:00:00 ps
deqrun@ubuntuserver:~$
```

Рисунок 2 - идентификаторы процессов.

```
589 ?      00:00:00 jbd2/sda2-8
590 ?      00:00:00 ext4-rsv-conver
593 ?      00:00:00 loop8
597 ?      00:00:00 loop9
622 ?      00:00:00 systemd-timesyn
666 ?      00:00:00 systemd-network
668 ?      00:00:00 systemd-resolve
678 ?      00:00:00 accounts-daemon
680 ?      00:00:00 cron
682 ?      00:00:00 dbus-daemon
690 ?      00:00:00 networkd-dispat
691 ?      00:00:00 rsyslogd
694 ?      00:00:01 snapd
696 ?      00:00:00 systemd-logind
698 ?      00:00:00 udisksd
734 ?      00:00:00 sshd
740 ?      00:00:00 polkitd
746 ?      00:00:00 unattended-upgr
993 ?      00:00:00 systemd
994 ?      00:00:00 (sd-pam)
1046 tty6   00:00:00 agetty
1218 tty1    00:00:00 login
1289 tty1    00:00:00 bash
1431 tty1    00:00:00 nano
1584 ?      00:00:04 kworker/0:2-events
6998 tty1    00:00:00 bash
7000 tty1    00:00:00 less
7003 tty1    00:00:00 bash
7005 tty1    00:00:00 less
7228 ?      00:00:00 packagekitd
7389 tty1    00:00:00 nano
7450 ?      00:00:00 kworker/0:0-events
7490 ?      00:00:00 kworker/u2:2-events_power_efficient
7497 ?      00:00:00 kworker/u2:1-events_unbound
7502 ?      00:00:00 kworker/u2:0-events_power_efficient
7508 tty1    00:00:00 ps
degrun@ubuntu-server:~$
```

Рисунок 3 - все процессы (команда ps -e).

### 3. Команды управления процессами

3.1 Получить информацию о выполняющихся процессах текущего пользователя в текущем интерпретаторе.

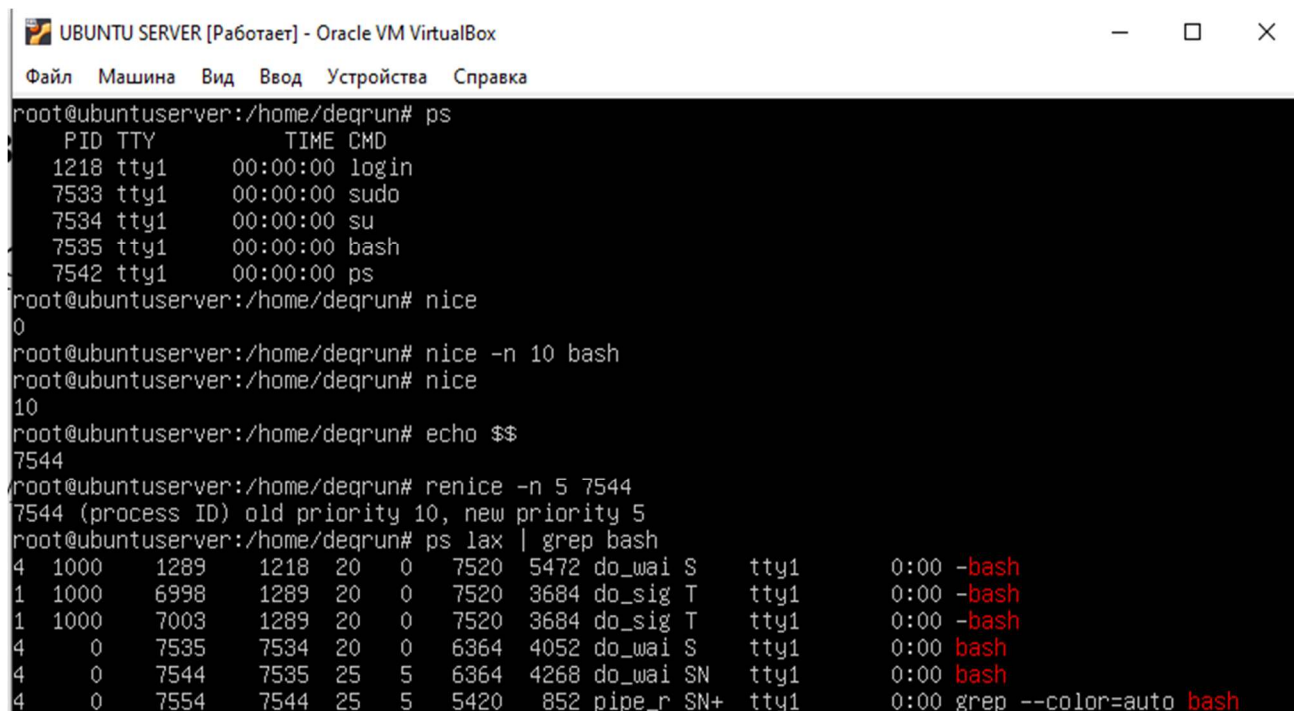
3.2 Определить текущее значение nice по умолчанию

3.3 Запустить интерпретатор bash с понижением приоритета  
nice -n 10 bash

3.4 Определить PID запущенного интерпретатора

3.5 Установить приоритет запущенного интерпретатора равным 5  
renice -n 5 <PID процесса>

3.6 Получить информацию о процессах bash  
ps lax | grep bash



```
root@ubuntuserver:/home/deqrun# ps
  PID TTY          TIME CMD
 1218 tty1        00:00:00 login
 7533 tty1        00:00:00 sudo
 7534 tty1        00:00:00 su
 7535 tty1        00:00:00 bash
 7542 tty1        00:00:00 ps
root@ubuntuserver:/home/deqrun# nice
0
root@ubuntuserver:/home/deqrun# nice -n 10 bash
root@ubuntuserver:/home/deqrun# nice
10
root@ubuntuserver:/home/deqrun# echo $$
7544
root@ubuntuserver:/home/deqrun# renice -n 5 7544
7544 (process ID) old priority 10, new priority 5
root@ubuntuserver:/home/deqrun# ps lax | grep bash
4  1000  1289  1218  20  0  7520  5472 do_wai S   tty1    0:00 -bash
1  1000  6998  1289  20  0  7520  3684 do_sig T   tty1    0:00 -bash
1  1000  7003  1289  20  0  7520  3684 do_sig T   tty1    0:00 -bash
4    0  7535  7534  20  0  6364  4052 do_wai S   tty1    0:00 bash
4    0  7544  7535  25  5  6364  4268 do_wai SN  tty1    0:00 bash
4    0  7554  7544  25  5  5420   852 pipe_r SN+  tty1    0:00 grep --color=auto bash
```

Рисунок 4 - команды управления процессами.

#### 4. Задание для варианта номер 5.

4.1 Отобразить информацию о процессах указанного пользователя в виде иерархии, вывод отсортировать по значениям PID.



```

366 ?      S<s    0:00  /lib/systemd/systemd-journald
398 ?      Ss     0:00  /lib/systemd/systemd-udevd
555 ?      Ssl    0:00  /sbin/multipathd -d -s
622 ?      Ssl    0:00  /lib/systemd/systemd-timesyncd
666 ?      Ss     0:00  /lib/systemd/systemd-networkd
668 ?      Ss     0:00  /lib/systemd/systemd-resolved
678 ?      Ssl    0:00  /usr/lib/accounts-service/accounts-daemon
680 ?      Ss     0:00  /usr/sbin/cron -f -P
682 ?      Ss     0:00  @dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd=
690 ?      Ss     0:00  /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
691 ?      Ssl    0:00  /usr/sbin/rsyslogd -n -iNONE
694 ?      Ssl    0:01  /usr/lib/snapd/snapd
696 ?      Ss     0:00  /lib/systemd/systemd-logind
698 ?      Ssl    0:00  /usr/libexec/udisks2/udisksd
734 ?      Ss     0:00  sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
740 ?      Ssl    0:00  /usr/libexec/polkitd --no-debug
746 ?      Ssl    0:00  /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-sh
993 ?      Ss     0:00  /lib/systemd/systemd --user
994 ?      S      0:00  (sd-pam)
1046 tty6   Ss+    0:00  /sbin/agetty -o -p -- \u --noclear tty6 linux
1218 tty1   Ss     0:00  /bin/login -p --
1289 tty1   S      0:00  -bash
1431 tty1   T      0:00  nano script17
6998 tty1   T      0:00  -bash
7000 tty1   T      0:00  less
7003 tty1   T      0:00  -bash
7005 tty1   T      0:00  less
7389 tty1   T      0:00  nano script20_new
7533 tty1   S      0:00  sudo su
7534 tty1   S      0:00  su
7535 tty1   S      0:00  bash
7544 tty1   SN     0:00  bash
7555 tty1   SN     0:00  sudo -i
7556 tty1   SN     0:00  -bash
7573 tty1   RN+    0:00  ps U deqrun -A -H --sort pid
7228 ?      Ssl    0:00  /usr/libexec/packagekitd
root@ubuntu:~#

```

Рисунок 5 – информация о процессах указанного пользователя. (команда `ps U deqrun -A -H --sort pid`)

4.2 С помощью сигнала SIGSTOP приостановить выполнение процесса, владельцем которого является текущий пользователь. Через несколько секунд возобновить выполнение процесса.

Воспользуемся командами `kill -STOP <PID>` и `kill -CONT <PID>`

```

622 ?      Ssl    0:00  /lib/systemd/systemd-timesyncd
666 ?      Ss     0:00  /lib/systemd/systemd-networkd
668 ?      Ss     0:00  /lib/systemd/systemd-resolved

```

Рисунок 6 – статус процесса 666 до использования команд.

```

622 ?      Ssl    0:00  /lib/systemd/systemd-timesyncd
666 ?      Ts     0:00  /lib/systemd/systemd-networkd
668 ?      Ss     0:00  /lib/systemd/systemd-resolved

```

Рисунок 7 – статус процесса 666 после использования команды `kill -STOP 666`.

```

622 ?      Ssl    0:00  /lib/systemd/systemd-timesyncd
666 ?      Ss     0:00  /lib/systemd/systemd-networkd
668 ?      Ss     0:00  /lib/systemd/systemd-resolved

```

Рисунок 8 – статус процесса 666 после использования команды `kill -CONT 666`.

4.3 Определить идентификаторы и имена процессов, не связанных с указанным терминалом.

```

degrun@ubuntuuserver:~$ echo $$
1289
degrun@ubuntuuserver:~$ ps -p 1289 -N --ppid 1289 -N

```

Рисунок 9 – команда для определения идентификаторов и имен процессов, не связанных с bash

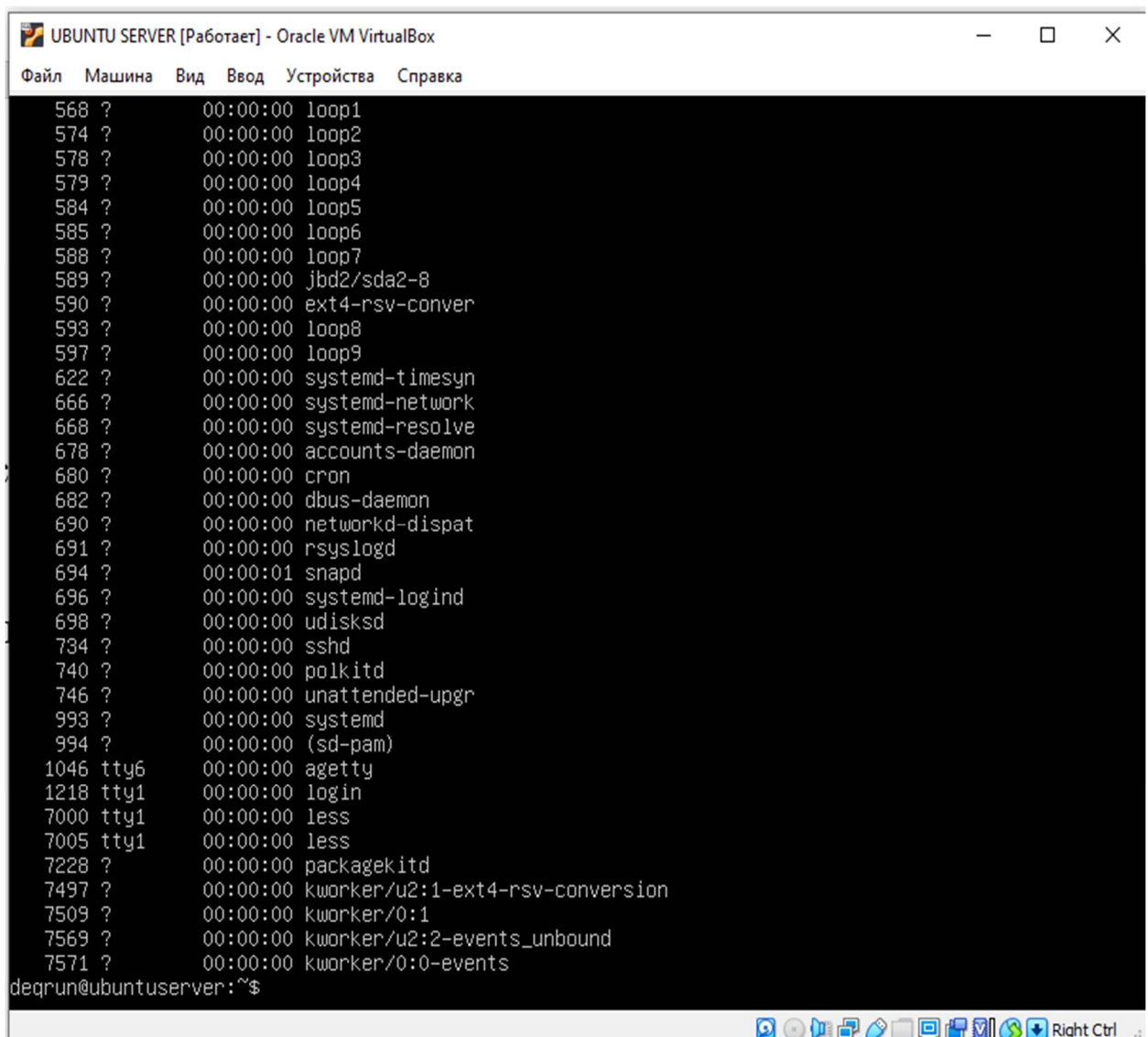


Рисунок 10 – результат работы команды.

## **Заключение**

В ходе данной лабораторной работы были изучены или повторно рассмотрены некоторые команды ОС Linux, было проведено ознакомление и анализ рекомендованной литературы, а также информации о средствах управления процессами ОС Ubuntu.

Контрольные вопросы.

1. Перечислите состояния задачи в ОС Ubuntu

Running – выполнение

Sleeping – спячка, сон

Stopped – остановка

Zombie – выполнение задачи прекратилось, но задача не удалена из системы

Dead – смерть – может быть удален из системы

Active – активное состояние

Expired – неактивное состояние

2. Как создаются задачи в ОС Ubuntu?

Задачи создаются путем вызова системной функции clone.

Любые обращения к fork или vfork преобразуются в системные вызовы clone во время компиляции

Функция fork создает дочернюю задачу, виртуальная память для которой выделяется по принципу копирования при записи.

3. Назовите классы потоков в ОС Ubuntu

Поток реального времени, обслуживаемые по алгоритму FIFO.

Поток реального времени, обслуживаемые в порядке циклической очереди.

Потоки разделения времени.

4. Как используется приоритет планирования при запуске задачи.

После создания задачи с помощью clone, она помещается в очередь выполнения (run queue) процессора, содержащую ссылки на все задачи, состязющиеся за процессорное время. Очереди выполнения, напоминают многоуровневые очереди с обратной связью, позволяют присваивать задачам различные приоритеты.

Массив приоритетов (priority array) содержит указатели на отдельные уровни очереди выполнения.

Каждая запись массива приоритетов ссылается на список задач: задача с. приоритетом { помещается в -ю ячейку массива приоритетов очереди выполнения.

Планировщик помещает задачу в начало списка на самом высоком уровне массива приоритетов. Если на этом уровне массива приоритетов существует несколько задач, они циклически упорядочены.

Во время создания каждой задачи присваивается статистический приоритет, также называемый правильным значением.

Всего 40 уровней приоритета – от -20 до 19. Чем меньше число – тем больше приоритет.

5. Как можно изменить приоритет для выполняющейся задачи?

Измененный уровень приоритета называют эффективным приоритетом.

Понизить приоритет можно с помощью следующей команды

`nice -n [коэффициент понижения] команда [аргумент]`

Для выполняемого процесса

`Renice -n [значение nice] -p [PID процесса]`