



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

---

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных  
технологий

## **Отчет по практической работе №1**

по дисциплине «Системное программное обеспечение»

**Тема практической работы:** «Командная строка Linux»

**Выполнил:**

Студент группы ИВБО-01-21

Апостолов Антон Игоревич

**Проверил:**

ассистент Воронцов Ю.А.

МОСКВА 2023 г.

## СОДЕРЖАНИЕ

1 ПРАКТИЧЕСКАЯ РАБОТА №1.....	5
2 ПРАКТИЧЕСКАЯ РАБОТА №2.....	7
3 ПРАКТИЧЕСКАЯ РАБОТА №3.....	11
4 ПРАКТИЧЕСКАЯ РАБОТА №4.....	15
ЗАКЛЮЧЕНИЕ.....	19

# 1 ПРАКТИЧЕСКАЯ РАБОТА №1

В практической работе №1 нам нужно реализовать следующую структуру, представленную на рисунке 1.1.

```
|-ivanov_dir_1
  |-file_1.txt
  |
  |-ivanov_dir_2
    |-cp_file_1.txt
    |-file_2.txt
    |-file_3.txt
    |
    |-ivanov_dir_3
      |-cp_file_3.txt
      |-mv_file_4_dir_1.txt
```

**Рисунок 1.1. – структура директорий и файлов**

В первой строке терминала с помощью команды `mkdir` создаем 3 директории. Далее с помощью команды `cd` переходим в директорию `ivanov_dir_1` и с помощью команды `nano` два текстовых файла: `file1.txt` и `mv_file4_dir_1.txt`. Выходим из директории `ivanov_dir_1` и с помощью команды `cp`, копируем `file_1.txt` в директорию `ivanov_dir_2`, а также с помощью команды `mv` перемещаем файл `mv_file4_dir_1.txt` в директорию `ivanov_dir_3`. Далее с помощью команды `cp` копируем файл `file_3.txt` в директорию `ivanov_dir_3`. Все команды представлены ниже, на рисунке 1.2.

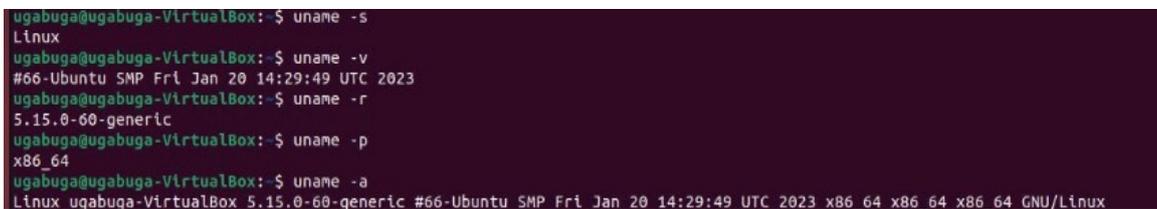
```
ugabuga@ugabuga-VirtualBox: ~
ugabuga@ugabuga-VirtualBox:~$ mkdir ivanov_dir_1 ivanov_dir_2 ivanov_dir_3
ugabuga@ugabuga-VirtualBox:~$ cd ivanov_dir_1
ugabuga@ugabuga-VirtualBox:~/ivanov_dir_1$ nano file_1.txt
ugabuga@ugabuga-VirtualBox:~/ivanov_dir_1$ nano mv_file_4_dir_1.txt
ugabuga@ugabuga-VirtualBox:~/ivanov_dir_1$ cd ~
ugabuga@ugabuga-VirtualBox:~$ cp ivanov_dir_1/file_1.txt ivanov_dir_2
ugabuga@ugabuga-VirtualBox:~$ mv ivanov_dir_1/mv_file_4_dir_1.txt ivanov_dir_3
ugabuga@ugabuga-VirtualBox:~$ cd ivanov_dir_2
ugabuga@ugabuga-VirtualBox:~/ivanov_dir_2$ nano file_2.txt
ugabuga@ugabuga-VirtualBox:~/ivanov_dir_2$ nano file_3.txt
ugabuga@ugabuga-VirtualBox:~/ivanov_dir_2$ cp file_3 ivanov_dir_3/cp_file_3.txt
cp: не удалось выполнить stat для 'file_3': Нет такого файла или каталога
ugabuga@ugabuga-VirtualBox:~/ivanov_dir_2$ cp file_3.txt ivanov_dir_3/cp_file_3.txt
cp: невозможно создать обычный файл 'ivanov_dir_3/cp_file_3.txt': Нет такого файла или каталога
ugabuga@ugabuga-VirtualBox:~/ivanov_dir_2$ cd~
Команда «cd~» не найдена. Возможно, вы имели в виду:
  command 'cdi' from deb cdo (2.0.4-1)
  command 'cdo' from deb cdo (2.0.4-1)
  command 'cdb' from deb tinycdb (0.78build3)
  command 'cdp' from deb irpas (0.10-9)
  command 'cdw' from deb cdw (0.8.1-2)
  command 'cde' from deb cde (0.1+git9-g551e54d-1.2)
  command 'cd5' from deb cd5 (0.1-4)
Try: sudo apt install <deb name>
ugabuga@ugabuga-VirtualBox:~/ivanov_dir_2$ cd ~
ugabuga@ugabuga-VirtualBox:~$ cp ivanov_dir_2/file_3.txt ivanov_dir_3/cp_file_3.txt
ugabuga@ugabuga-VirtualBox:~$
```

Рисунок 1.2. – команды в терминале

## 2 ПРАКТИЧЕСКАЯ РАБОТА №2

В практической работе №2 необходимо отразить следующее: скриншот с информацией о ядре: наименование ядра, версия ядра, выпуск ядра, аппаратная архитектура, полная информация о ядре (если данная информация не помещается на 1 скриншот, разделите на несколько и укажите в отчете, что отражено на каждом) скриншоты и пояснения по исследованию команд: `ls` и дополнительных параметров к ней, `less` и `file`. А также показать свое исследование 3 важных каталогов системы из списка к практике.

Для отображения информации о ядре воспользуемся командой `uname` и следующими параметрами: `-s`, чтобы показать наименование ядра, `-v`, для отображения версии ядра, `-r`, чтобы показать выпуск ядра, `-p` для визуализации архитектуры ядра и `-a`, чтобы отобразить всю информацию о ядре. Все команды приведены ниже, на рисунке 2.1.



```
ugabuga@ugabuga-VirtualBox:~$ uname -s
Linux
ugabuga@ugabuga-VirtualBox:~$ uname -v
#66-Ubuntu SMP Fri Jan 20 14:29:49 UTC 2023
ugabuga@ugabuga-VirtualBox:~$ uname -r
5.15.0-60-generic
ugabuga@ugabuga-VirtualBox:~$ uname -p
x86_64
ugabuga@ugabuga-VirtualBox:~$ uname -a
Linux ugabuga-VirtualBox 5.15.0-60-generic #66-Ubuntu SMP Fri Jan 20 14:29:49 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
```

**Рисунок 2.1. – команды для отображения информации о ядре**

Команда `ls` необходима для просмотра все файлов и директорий там, где вы сейчас находитесь, `less` для просмотра файлов и их редактирования, а команда `file` предоставляет полную информацию о файле, какое расширение, архитектура и т.д. Для большего понимания всех этих команд, проведем исследование 3 важных каталогов системы.

Зайдем под пользователя `root` и перейдем в корневую папку системы, после чего воспользуемся командой `ls`. Как видно из рисунка 2.2, команда `ls` вывела нам все папки и файлы находящиеся в корневом каталоге системы. Зайдем в некоторые из них и посмотрим их содержимое и по одному файлу

В КАЖДОМ ИЗ НИХ.

```
root@ugabuga-VirtualBox:/# ls
bin  cdrom  etc  lib  lib64  lost+found  mnt  proc  run  snap  swapfile  tmp  var
boot  dev  home  lib32  libx32  media  opt  root  sbin  srv  sys  usr
```

Рисунок 2.2. – вызов команды ls в корневом каталоге системы

Зайдем, например в папку bin (там хранятся все бинарные файлы), вызовем ls. На рисунке 2.3 представлены лишь некоторые файлы, так как их очень много.

```
root@ugabuga-VirtualBox:/bin# ls
[
411toppm  git-receive-pack  open
aa-enabled  git-shell  openssl
aa-exec  git-upload-archive  openvt
aa-features-abi  git-upload-pack  opldecode
aconect  gjs  orca
acpi_listen  gjs-console  orca-dm-wrapper
addpart  gkbd-keyboard-display  os-prober
addr2line  glib-compile-schemas  p11-kit
airscan-discover  gmake  pacat
alsabat  gnome-calculator  pacmd
  gnome-characters  partl
```

Рисунок 2.3. – вызов ls в каталоге bin

Для отображения содержимого файла применим команду less, например к файлу git. После чего мы увидим, что находится в этом файле. Как результат там, находится большое количество бинарного кода, представленного на рисунке 2.4. Для выходы нажмем клавишу отвечающую за английскую букву ‘q’.

```
ELF
...
GNU
...
q
```

## Рисунок 2.4. – файл git

Далее применяем команду `file`, к файлу `git`, после чего нам выдастся результат этой команды, показанный на рисунке 2.5

```
root@ugabuga-VirtualBox:/bin# file git
git: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=9554b695d998574fcf0ed8e733dbf7490871f380, for GNU/Linux 3.2.0, stripped
```

Рисунок 2.5. – информация о файле git

Аналогично посмотрим еще два каталога. Каталог `var`, который должен содержать файлы, которые часто изменяются. Размер этих файлов постоянно увеличивается. Здесь содержатся файлы системных журналов, различные кэши, базы данных и так далее, показанные на рисунке 2.6. Пример файла из этого каталога показан на рисунке 2.7.

```
backups cache crash lib local lock log mail metrics opt run snap spool tmp
```

Рисунок 2.6. – каталог var



Рисунок 2.7. – пример вывода файла каталога var

И каталог `tmp`. В этом каталоге содержатся временные файлы, созданные системой, любыми программами или пользователями. Все пользователи имеют право записи в эту директорию. Файлы удаляются при каждой перезагрузке. Аналогом Windows является папка `Windows\Temp`, здесь тоже хранятся все временные файлы, показанные на рисунке 2.8. Пример файла из этого каталога показан на рисунке 2.9.

```
root@toxa-VirtualBox:/# cd tmp
root@toxa-VirtualBox:/tmp# ls
config-err-Azs1fQ
snap-private-tmp
ssh-7x3z5hXCQvT1
systemd-private-035fab595b124ab785a36cc56f2476ef-color.service-LLYq5g
systemd-private-035fab595b124ab785a36cc56f2476ef-ModemManager.service-gki3Df
systemd-private-035fab595b124ab785a36cc56f2476ef-switcheroo-control.service-B89Vai
systemd-private-035fab595b124ab785a36cc56f2476ef-systemd-logind.service-vMDgig
systemd-private-035fab595b124ab785a36cc56f2476ef-systemd-resolved.service-FWnhEh
systemd-private-035fab595b124ab785a36cc56f2476ef-systemd-timesyncd.service-Q0HmJh
systemd-private-035fab595b124ab785a36cc56f2476ef-upower.service-a5huwg
tracker-extract-files.1000
tracker-extract-files.125
VMwareDnD
root@toxa-VirtualBox:/tmp# nano ssh-7x3z5hXCQvT1
root@toxa-VirtualBox:/tmp#
```

Рисунок 2.8. – файлы каталога tmp

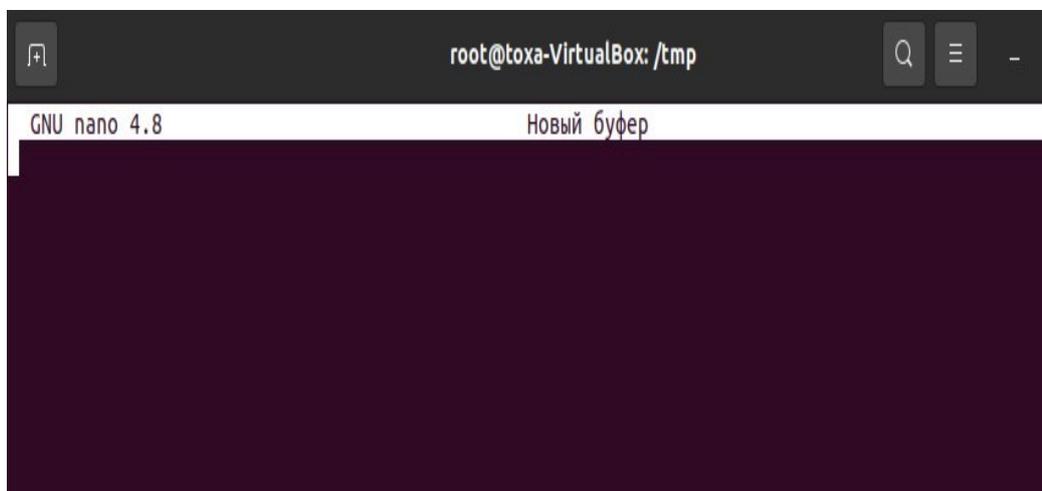


Рисунок 2.9. – вывод файла из каталога tmp

### 3 ПРАКТИЧЕСКАЯ РАБОТА №3

В практической работе №3 необходимо создать 3 пользователей с обычными правами, создать набор файлов и каталогов с разными модификаторами доступа в домашних каталогах пользователей, реализовать различные доступы к файлам и каталогам других пользователей, реализовать с использованием различных команд управления на запрет и проверку доступа после запрета, выдать sudo одному из пользователей и осуществить доступ к файлам и каталогам с ограничениями, перейти под пользователя root и выполнить любые действия без применения команды sudo.

Для отображения создания пользователей с обычными правами воспользуемся командой `useradd` с параметром `-m` (автоматическое создание домашней директории) и командой `passwd` для установки пароля для каждого из новых пользователей, как показано на рисунке 3.1.

```
ugabuga@ugabuga-VirtualBox:~$ sudo useradd -m user1
[sudo] пароль для ugabuga:
ugabuga@ugabuga-VirtualBox:~$ sudo useradd -m user2
ugabuga@ugabuga-VirtualBox:~$ sudo useradd -m user2
useradd: пользователь «user2» уже существует
ugabuga@ugabuga-VirtualBox:~$ sudo useradd -m user3
ugabuga@ugabuga-VirtualBox:~$ sudo passwd user1
Новый пароль:
Повторите ввод нового пароля:
passwd: пароль успешно обновлён
ugabuga@ugabuga-VirtualBox:~$ sudo passwd user2
Новый пароль:
Повторите ввод нового пароля:
passwd: пароль успешно обновлён
ugabuga@ugabuga-VirtualBox:~$ sudo passwd user3
Новый пароль:
Повторите ввод нового пароля:
passwd: пароль успешно обновлён
ugabuga@ugabuga-VirtualBox:~$
```

Рисунок 3.1. – создание пользователей

Далее зайдём за каждого из пользователей и с помощью команды `touch` и `mkdir` создадим файлы и директорию и с помощью команды `chmod`

изменим параметры доступа для каждого из них, на чтение (604), выполнение (601), и запись (602), как показано на рисунке 3.2.

```
ugabuga@ugabuga-VirtualBox:~$ su - user1
Пароль:
user1@ugabuga-VirtualBox:~$ touch pr3_user1.txt
user1@ugabuga-VirtualBox:~$ chmod 601 pr3_user1.txt
user1@ugabuga-VirtualBox:~$ ls -l pr3_user1.txt
-rw-----x 1 user1 user1 0 фев 19 14:18 pr3_user1.txt
user1@ugabuga-VirtualBox:~$ exit
выход
ugabuga@ugabuga-VirtualBox:~$ su - user2
Пароль:
user2@ugabuga-VirtualBox:~$ touch pr3_user2.txt
user2@ugabuga-VirtualBox:~$ chmod 602 pr3_user2.txt
user2@ugabuga-VirtualBox:~$ ls -l pr3_user2.txt
-rw----w- 1 user2 user2 0 фев 19 14:20 pr3_user2.txt
user2@ugabuga-VirtualBox:~$ exit
выход
ugabuga@ugabuga-VirtualBox:~$ su - user3
Пароль:
user3@ugabuga-VirtualBox:~$ mkdir pr3_user3
user3@ugabuga-VirtualBox:~$ chmod 604 pr_user3
chmod: невозможно получить доступ к 'pr_user3': Нет такого файла или ката
лога
user3@ugabuga-VirtualBox:~$ chmod 604 pr3_user3
user3@ugabuga-VirtualBox:~$ ls l pr3_user
ls: невозможно получить доступ к 'l': Нет такого файла или каталога
ls: невозможно получить доступ к 'pr3_user': Нет такого файла или каталог
а
user3@ugabuga-VirtualBox:~$ ls l pr3_user3
ls: невозможно получить доступ к 'l': Нет такого файла или каталога
pr3_user3:
user3@ugabuga-VirtualBox:~$ ls -l pr3_user3
итого 0
user3@ugabuga-VirtualBox:~$ ls -l
итого 4
drw---r-- 2 user3 user3 4096 фев 19 14:21 pr3_user3
user3@ugabuga-VirtualBox:~$
```

Рисунок 3.2. – создание файлов и директорий с разным уровнем доступа

Зайдем, за пользователя user1 и попробуем осуществить доступ к файлам пользователя user2 с помощью команды less. Результат показан ниже на рисунке 3.3.

```
user1@ugabuga-VirtualBox:~/home/user2$ less pr3_user2.txt
pr3_user2.txt: Отказано в доступе
```

Рисунок 3.3. – попытка открыть файл пользователя user2 с помощью команды less за пользователя user1

Зайдем, за пользователя user2 и попробуем осуществить доступ к файлам пользователя user1 с помощью команды nano. Результат показан ниже на рисунке 3.4.

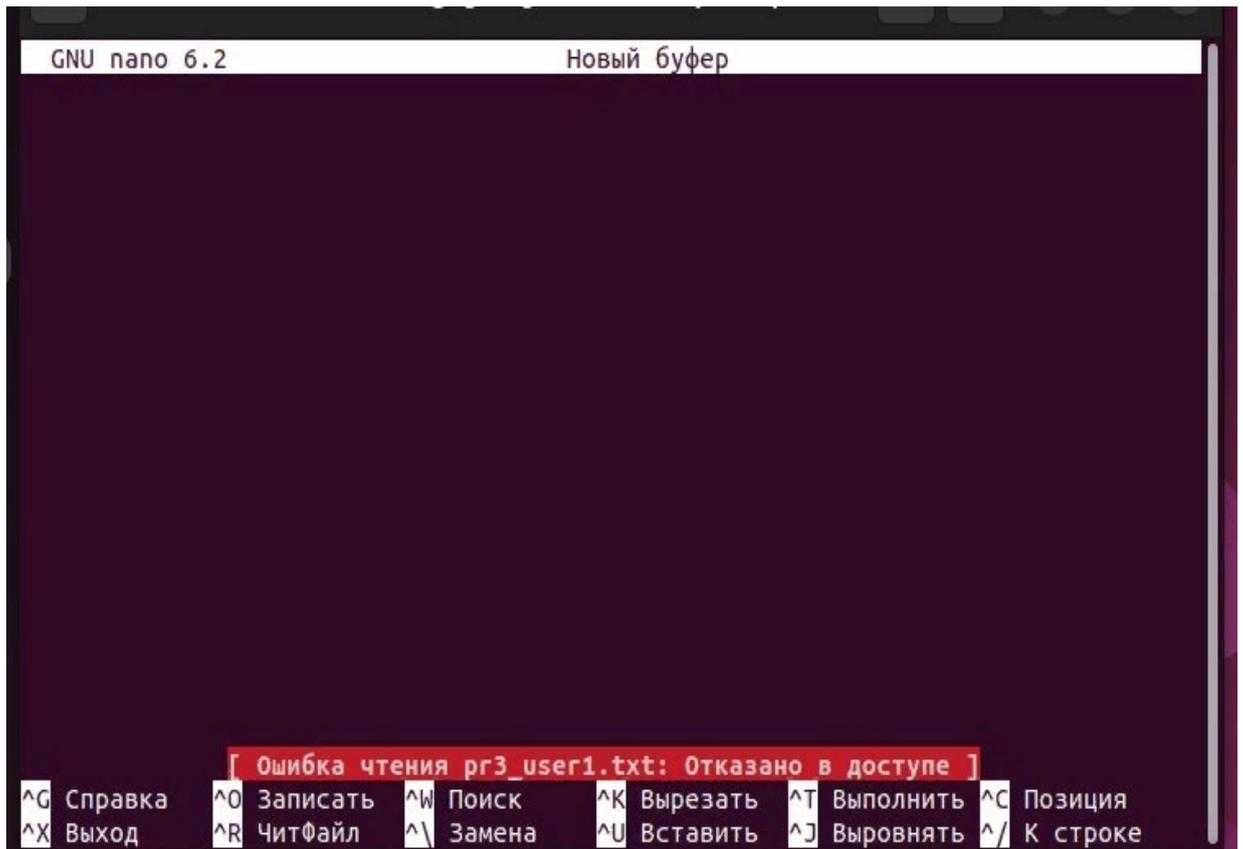


Рисунок 3.4. – попытка открыть файл пользователя user1 с помощью команды less за пользователя user2

Зайдем, за пользователя user1 и попробуем зайти в директорию, созданную пользователем user3 с помощью команды cd. Результат показан ниже на рисунке 3.5.

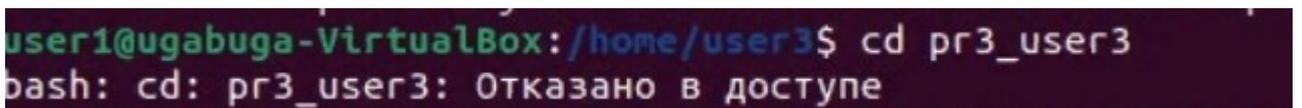


Рисунок 3.5. – попытка открыть файл пользователя user2 с помощью команды less за пользователя user1

Далее выдадим пользователю user1 права sudo, с помощью команды и параметров, представленных ниже, на рисунке 3.6

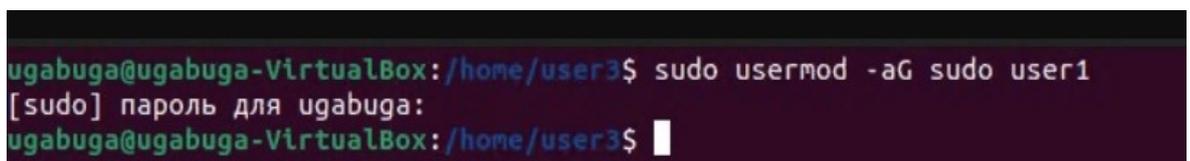


Рисунок 3.6. – предоставление прав sudo пользователю user1

Далее попробуем зайти в директорию пользователя user3, поменяв уровень доступа его файла с помощью ключевого слова sudo. Как видно из рисунка 3.7 мы успешно выдали пользователю user1 права администратора.

```
user1@ugabuga-VirtualBox:/home/user3$ sudo chmod 777 pr3_user3
user1@ugabuga-VirtualBox:/home/user3$ cd pr3_user3
user1@ugabuga-VirtualBox:/home/user3/pr3_user3$ █
```

Рисунок 3.7. – проверка выдачи user1 прав sudo

После перейдем за пользователя root и как видно из рисунка 3.8, находясь за пользователя root нам доступны абсолютно все файлы и нас больше нет необходимости применять команду sudo.

```
root@ugabuga-VirtualBox:~/snap/snapd-desktop-integration/common# exit
ВЫХОД
user1@ugabuga-VirtualBox:/home$ su
Пароль:
root@ugabuga-VirtualBox:/home# ls
ugabuga  user1  user2  user3
root@ugabuga-VirtualBox:/home# cd user1
root@ugabuga-VirtualBox:/home/user1# ls
pr3_djdd.txt  snap  Документы  Изображения  Общедоступные  Шаблоны
pr3_user1.txt  Видео  Загрузки  Музыка  'Рабочий стол'
root@ugabuga-VirtualBox:/home/user1# less pr3_user1.txt
root@ugabuga-VirtualBox:/home/user1# cd /home/user2
root@ugabuga-VirtualBox:/home/user2# ls
pr3_user2.txt
root@ugabuga-VirtualBox:/home/user2# less pr3_user2.txt
root@ugabuga-VirtualBox:/home/user2# █
```

Рисунок 3.8. – команды пользователя root

## 4 ПРАКТИЧЕСКАЯ РАБОТА №4

В практической работе №4 необходимо используя команды `ps`, `sort` и др. выбрать процессы по следующим параметрам: процессы, принадлежащий пользователю `systemd`, процессы, принадлежащий пользователю `avahi`, топ 4 процесса, занимающих наибольшее количество ресурсов процессора в процентах, топ 5 процесса, занимающих наибольшее количество ресурсов памяти в процентах, топ 3 процесса, занимающих наибольший объем виртуальной памяти, топ 4 процесса, занимающих наибольший объем физической памяти (ОЗУ). Также используя команды по управлению процессами выполнить следующее: запустить любое из графических приложений (например, `xlogo` или `gkedit`) при помощи командной строки в фоновом режиме, запустить ещё один экземпляр приложения, а затем переместить его в фоновый режим при помощи комбинаций клавиш, вывести список находящихся в фоне процессов, один из процессов завершить при помощи команды `kill`, второй процесс завершить при помощи команды `killall`. Выполнить тот же набор действий при помощи утилиты `top`. Все необходимые команды для этого можно посмотреть при помощи команды `h` в интерфейсе команды `top`.

Для отображения процессов принадлежащих пользователю `system` воспользуемся командой `ps -aux | grep systemd`, а для нахождения процессов принадлежащих пользователю `avahi` воспользуемся `ps -fu avahi`

```
root@ugabuga-VirtualBox:/home/ugabuga# ps -eo pid,ppid,cmd,%cpu --sort=-%cpu | head -5
PID    PPID  CMD                                %CPU
1100   906   /usr/bin/gnome-shell              2.8
30507  906   /usr/libexec/gnome-terminal       0.6
435    1     /lib/systemd/systemd-oomd         0.1
1225   1     /usr/libexec/packagekitd          0.1
root@ugabuga-VirtualBox:/home/ugabuga# ps -eo pid,ppid,cmd,%mem --sort=-%mem | head -6
PID    PPID  CMD                                %MEM
1100   906   /usr/bin/gnome-shell              15.2
1410   906   /snap/snap-store/582/usr/bin/gn  12.8
49434  1100  gjs /usr/share/gnome-shell/       2.9
30507  906   /usr/libexec/gnome-terminal       2.4
36069  1     /usr/bin/dockerd -H fd://        2.2
root@ugabuga-VirtualBox:/home/ugabuga# ps -eo pid,ppid,cmd,vsz --sort=-vsz | head -4
PID    PPID  CMD                                VSZ
1100   906   /usr/bin/gnome-shell              4175568
49434  1100  gjs /usr/share/gnome-shell/       2732000
1502   906   /usr/bin/gjs /usr/share/gnome-  2608584
root@ugabuga-VirtualBox:/home/ugabuga# ps -eo pid,ppid,cmd,rss --sort=-rss | head -5
PID    PPID  CMD                                RSS
1100   906   /usr/bin/gnome-shell              308888
1410   906   /snap/snap-store/582/usr/bin/gn  259660
49434  1100  gjs /usr/share/gnome-shell/       59856
30507  906   /usr/libexec/gnome-terminal       50076
```

Рисунок 4.1. – нахождение топ процессов

```

ugabuga@ugabuga-VirtualBox:~$ ps -aux | grep systemd
root      226  0.0  1.1 50768 23144 ?        S<cs 12:12  0:00 /lib/systemd/systemd-journald
root      284  0.0  0.2 26584  5776 ?        Ss   12:13  0:00 /lib/systemd/systemd-udevd
systemd+  435  0.1  0.3 14828  6184 ?        Ss   12:13  0:19 /lib/systemd/systemd-oomd
systemd+  436  0.0  0.6 25392 13620 ?        Ss   12:13  0:00 /lib/systemd/systemd-resolved
systemd+  437  0.0  0.3 89380  6576 ?        Ssl  12:13  0:00 /lib/systemd/systemd-timesyncd
message+  698  0.0  0.3 10104  6120 ?        Ss   12:13  0:01 @dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
root      727  0.0  0.4 23636  8092 ?        Ss   12:13  0:00 /lib/systemd/systemd-logind
ugabuga   906  0.0  0.5 17944 10712 ?        Ss   12:13  0:01 /lib/systemd/systemd --user
ugabuga   955  0.0  0.2 10200  5996 ?        Ss   12:13  0:02 /usr/bin/dbus-daemon --session --address=systemd: --nofork --nopidfile --systemd-activation --syslog-on
ly
ugabuga  1007  0.0  0.8 602572 16244 ?        Ssl  12:13  0:00 /usr/libexec/gnome-session-binary --systemd-service --session=ubuntu
ugabuga  4294  0.0  0.1 17888  2440 pts/0    S+   16:35  0:00 grep --color=auto systemd
ugabuga@ugabuga-VirtualBox:~$ ps -fu avahi
UID      PID  PPID  C  STIME TTY          TIME CMD
avahi    695   1  0 12:13 ?        00:00:00 avahi-daemon: running [ugabuga-VirtualBox.local]
avahi    758   0 12:13 ?        00:00:00 avahi-daemon: chroot helper
ugabuga@ugabuga-VirtualBox:~$

```

**Рисунок 4.2. – нахождение процессов пользователей system и avahi**

Для нахождения топ 4 процесса, занимающих наибольшее количество ресурсов процессора в процентах, топ 5 процесса, занимающих наибольшее количество ресурсов памяти в процентах, топ 3 процесса, занимающих наибольший объем виртуальной памяти, топ 4 процесса, занимающих наибольший объем физической памяти (ОЗУ), воспользуемся командами представленными на рисунке 4.2

Запустим графическое приложение xlogo при помощи командной строки в фоновом режиме используя команду `xlogo &`. Запустим еще один экземпляр `xlogo` и затем с помощью комбинаций клавиш переместим его в фоновый режим. Далее с помощью команды `ps` выведем список находящихся в фоне процессов. Один из процессов завершим командой `kill`, второй командой `killall`. Все команды приведены ниже на рисунке 4.3

```

ugabuga@ugabuga-VirtualBox:~$ xlogo &
[1] 4694
ugabuga@ugabuga-VirtualBox:~$ xlogo
^Z
[2]+  Остановлен      xlogo
ugabuga@ugabuga-VirtualBox:~$ ps
      PID TTY          TIME CMD
      4552 pts/0        00:00:00 bash
      4694 pts/0        00:00:00 xlogo
      4695 pts/0        00:00:00 xlogo
      4696 pts/0        00:00:00 ps
ugabuga@ugabuga-VirtualBox:~$ kill %2

[2]+  Остановлен      xlogo
ugabuga@ugabuga-VirtualBox:~$ killall xlogo
[1]-  Завершено        xlogo
[2]+  Завершено        xlogo
ugabuga@ugabuga-VirtualBox:~$

```

Рисунок 4.4. – команды для xlogo

Далее также создадим два xlogo и запустим top. В команде top для отключения работающих программ надо нажать клавишу К и написать программ, которую необходимо отключить. Результат представлен на рисунке 4.4 .

```

ink
      73 root          0 -20      0      0      0 I  0,0  0,0  0:00.00 kworker/u
3:0
      119 root          0 -20      0      0      0 I  0,0  0,0  0:00.00 charger_m
anager
      161 root          20   0      0      0      0 S  0,0  0,0  0:00.00 scsi_eh_2

[1]-  Завершено        xlogo
[2]+  Завершено        xlogo
ugabuga@ugabuga-VirtualBox:~$ █

```

Рисунок 4.5. – закрытие приложений xlogo через top

Далее с помощью команды top и необходимых параметров выведем, представленных на рисунке 4.5. выведем: процессы, принадлежащие

пользователю avahi, топ 4 процесса, занимающих наибольшее количество ресурсов процессора в процентах, топ 5 процесса, занимающих наибольшее количество ресурсов памяти в процентах, топ 3 процесса, занимающих наибольший объем виртуальной памяти, топ 4 процесса, занимающих наибольший объем физической памяти (ОЗУ). Результаты представлено на рисунках 4.5 и 4.6

```

top - 09:33:17 up 15:53, 1 user, load average: 0,14, 0,11, 0,04
Tasks: 183 total, 1 running, 182 sleeping, 0 stopped, 0 zombie
Cpu(s): 0,7 us, 0,4 sy, 1,4 ni, 97,5 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
МБ Mem : 1975,5 total, 369,1 free, 868,4 used, 737,9 buff/cache
МБ Swap: 2680,0 total, 2197,9 free, 482,1 used. 922,8 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
  758 avahi    20   0   7440    148     0  S   0,0   0,0   0:00.00 avahi-daemon

```

Рисунок 4.6. – использование команды top для нахождения процесса пользователя avahi

```

root@ugabuga-VirtualBox:/home/ugabuga# top -o %CPU -b -n 1|head -n 12|tail -5
 1100 ugabuga  39  19 4175456 308664 55116 S  31,2  15,3  27:38.36 gnome-shell
 52249 root      20   0   21788   4128  3432 R  12,5   0,2   0:00.02 top
    1 root      20   0  167860  11112  6824 S   0,0   0,5   0:12.04 systemd
    2 root      20   0     0     0     0 S   0,0   0,0   0:00.01 kthreadd
    3 root       0 -20     0     0     0 I   0,0   0,0   0:00.00 rcu_gp

root@ugabuga-VirtualBox:/home/ugabuga# top -o %MEM -b -n 1|head -n 12|tail -6
  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
  1100 ugabuga  39  19 4175468 308752 55200 S  25,0  15,3  27:42.11 gnome-shell
  1410 ugabuga  20   0 1184628 259660 10244 S   0,0  12,8   0:35.61 snap-store
 49434 ugabuga  39  19 2732000  59700 45308 S   0,0   3,0   0:01.02 gjs
 30507 ugabuga  20   0  621236  50012 35424 S   0,0   2,5   0:52.92 gnome-terminal-
 36069 root      20   0 1391008  45596 24228 S   0,0   2,3   0:01.76 dockerd

root@ugabuga-VirtualBox:/home/ugabuga# top -o VIRT -b -n 1|head -n 12|tail -4
 49434 ugabuga  39  19 2732000  59692 45308 S   0,0   3,0   0:01.09 gjs
  1502 ugabuga  20   0 2608584  14896 12364 S   0,0   0,7   0:00.29 gjs
  1243 ugabuga  20   0 2608544  12788 12236 S   0,0   0,6   0:00.13 gjs
   935 ugabuga   9 -11 1693928  14764 12196 S   0,0   0,7   0:03.99 pulseaudio

root@ugabuga-VirtualBox:/home/ugabuga# top -o RES -b -n 1|head -n 12|tail -5
 1100 ugabuga  39  19 4175480 308844 55224 S   6,2  15,3  27:49.38 gnome-shell
  1410 ugabuga  20   0 1184628 259660 10244 S   0,0  12,8   0:35.62 snap-store
 49434 ugabuga  39  19 2732000  59732 45308 S   0,0   3,0   0:01.15 gjs
 30507 ugabuga  20   0  621236  50012 35424 S   0,0   2,5   0:54.13 gnome-terminal-
 36069 root      20   0 1391008  45596 24228 S   0,0   2,3   0:01.80 dockerd

root@ugabuga-VirtualBox:/home/ugabuga#

```

Рисунок 4.7. – использование команды top для нахождения топ процессов

## **ЗАКЛЮЧЕНИЕ**

В результате выполнения этой работы, мы успешно выполнили все задания, которые были даны.

