

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Вятский государственный университет»
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

Методические указания по лабораторной работе №1

ПОЛУЧЕНИЕ СВЕДЕНИЙ О СИСТЕМЕ

Киров 2022

1 Краткие сведения о лабораторной работе

Целью выполнения лабораторной работы является знакомство студента с версией серверной операционной системы Astra Linux.

В ходе лабораторной работы студент должен получить основные сведения о системе.

Для простоты выполнения лабораторной работы можно воспользоваться виртуальными машинами фирм: Oracle VM VirtualBox, VMware Workstation.

При выполнении лабораторной работы студенту понадобится командная строка для ввода команд на выполнение.

Для вызова командной строки необходимо воспользоваться сочетанием клавиш «Alt + t». Пример вызова приведен на рисунке 1.

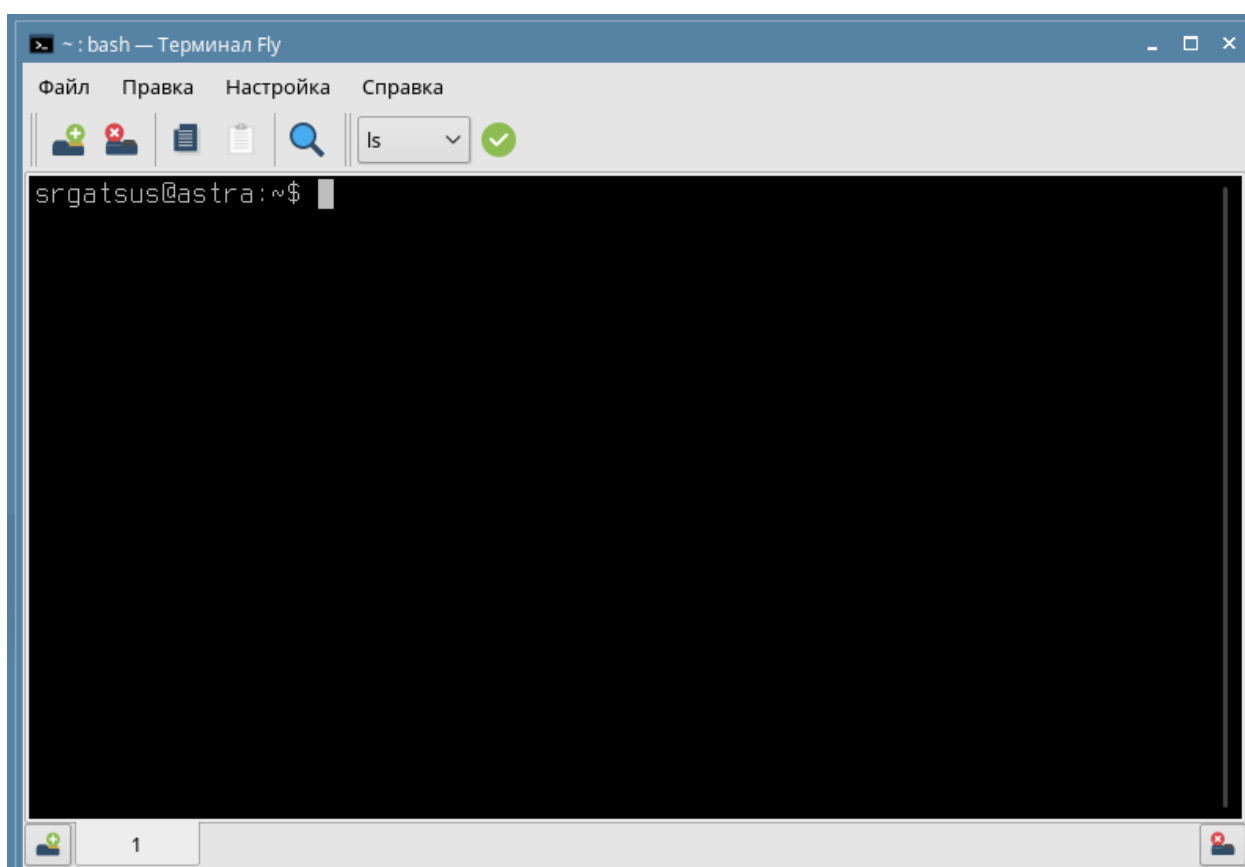


Рисунок 1 – Вызов командной строки

2 Получение имени компьютера

Получить имя компьютера можно из командной строки, воспользовавшись командой `hostname` или `hostnamectl` (выводит полную информацию). На рисунке 2.1-2.3 изображено получение имени компьютера.

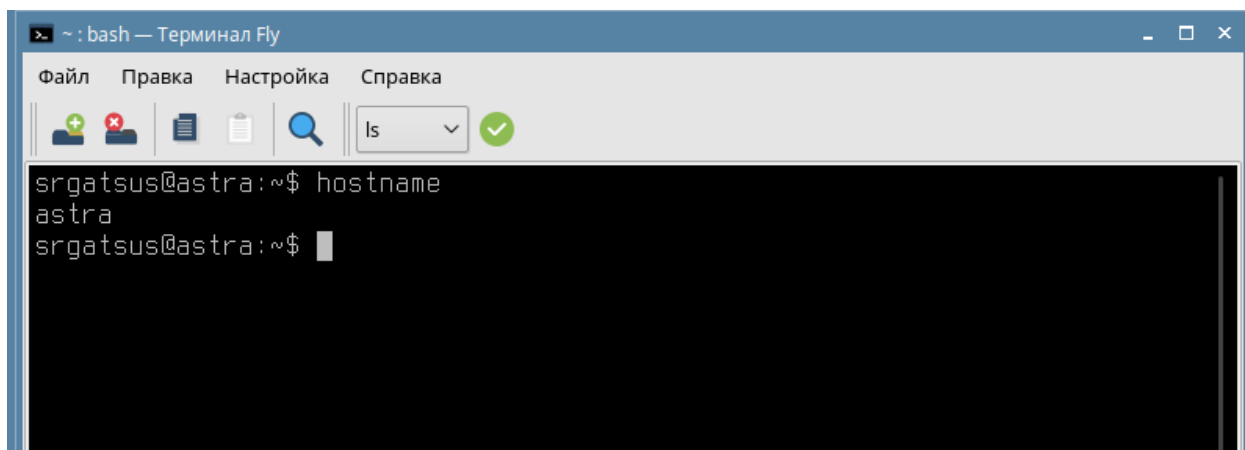


Рисунок 2.1 – Получение имени через `hostname`

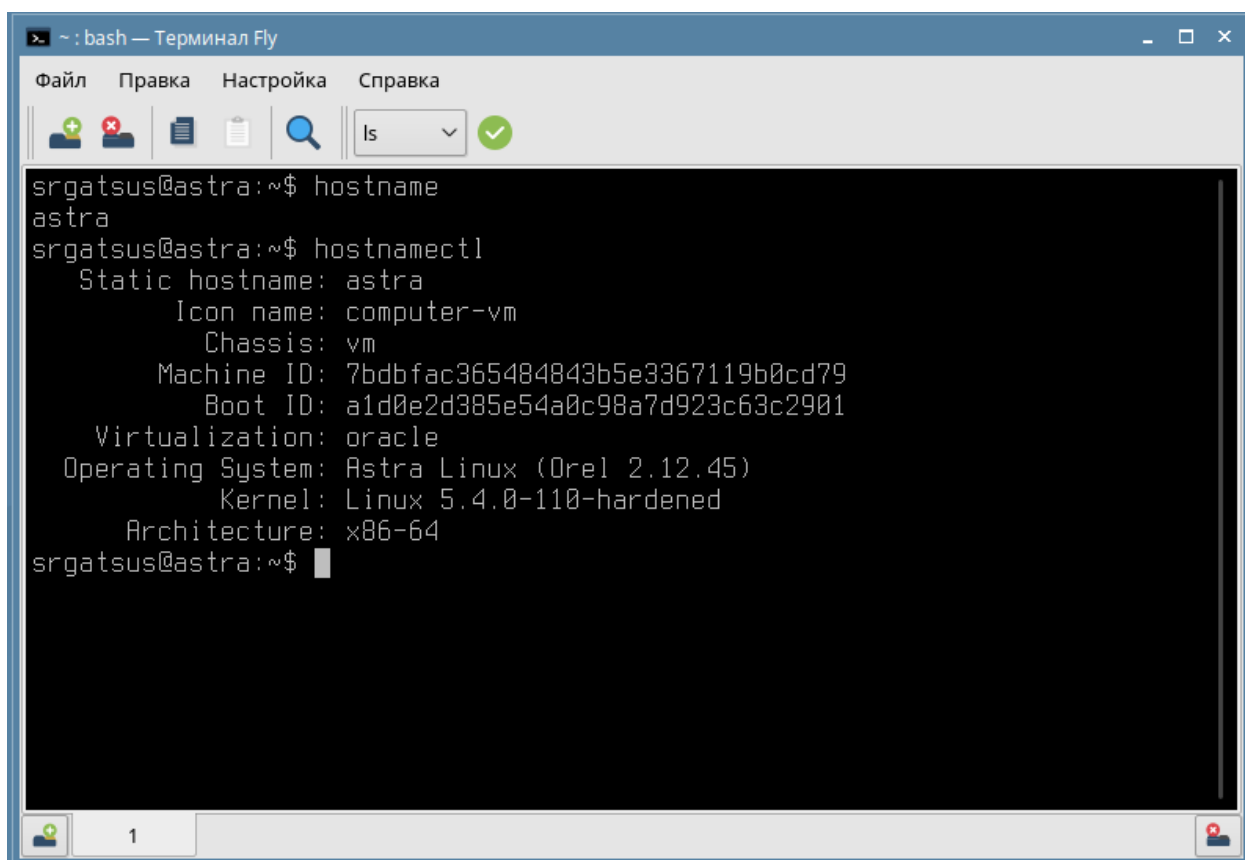


Рисунок 2.2 – Получение имени через `hostnamectl`

```
sergatsus@astra:~$ whoami
sergatsus
sergatsus@astra:~$
```

Рисунок 2.3 –Получение имени пользователя whoami

3 Просмотр версии Astra Linux

Linux является сервером так что воспользуемся командой **cat /etc/astra_version/**. Результат приведен на рисунке 3.

```
sergatsus@astra:~$ cat /etc/astra_version
CE 2.12.45 (orel)
sergatsus@astra:~$
```

Рисунок 3 – Просмотр информации о версии Astra Linux

4 Получение сведений о системе

На рисунке 4.1-4. Изображено получение сведений о системе. С помощью данного инструмента можно получить много аппаратной и программной информации о системе.

```
sergatsus@astra:~$ cat /etc/*release
DISTRIB_ID="AstraLinuxCE"
DISTRIB_DESCRIPTION="Astra Linux CE 2.12.45 (Orel)"
DISTRIB_RELEASE=2.12.45
DISTRIB_CODENAME=orel
PRETTY_NAME="Astra Linux (Orel 2.12.45)"
NAME="Astra Linux (Orel)"
ID=astra
ID_LIKE=debian
ANSI_COLOR="1;31"
HOME_URL="http://astralinux.ru"
SUPPORT_URL="http://astralinux.ru/support"
VARIANT_ID=orel
VARIANT=Orel
LOGO=astra
VERSION_ID=2.12.45
VERSION_CODENAME=orel
sergatsus@astra:~$
```

Рисунок 4.1 – Получение через командную строку

Для просмотра сведений о системе необходимо перейти в меню. Затем необходимо нажать «Пуск», выбрать «Системные», далее переходим «Информация о системе». Результат приведен на рисунке 4.2.

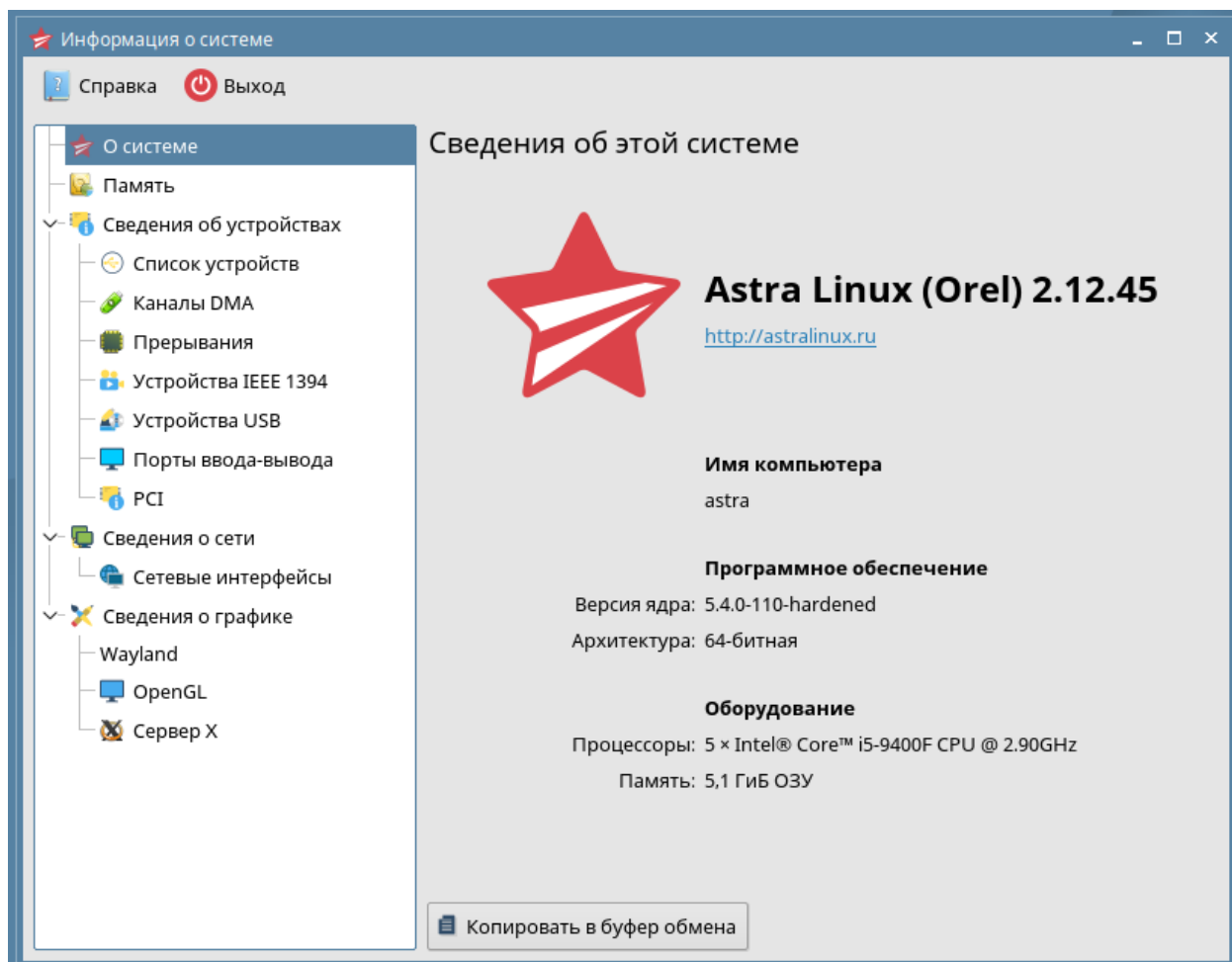


Рисунок 4.2 – Просмотр сведений о системе

```
srgatsus@astra:~$ cat /proc/version
Linux version 5.4.0-110-hardened (builder@build) (gcc version 6.3.0 20170516 (Debian 6.3.0-18+deb9u1)) #astra35+ci74-Ubuntu SMP Tue Jul 19 20:27:06 UTC 2022
srgatsus@astra:~$
```

Рисунок 4.3 – Получение версии ядра cat /proc/version

lscpu- информация о процессоре

```
~: bash — Терминал Fly
Файл  Правка  Настройка  Справка
ls
srgatsus@astra:~$ lscpu
Аrchитектура:x86_64
CPU op-mode(s): 32-bit, 64-bit
Порядок байт:Little Endian
CPU(s): 5
On-line CPU(s) list: 0-4
Thread(s) per core: 1
Ядер на сокет:5
Сокетов: 1
NUMA node(s): 1
ID прроизводителя:GenuineIntel
Семейство ЦПУ:6
Модель: 158
Имя модели: Intel(R) Core(TM) i5-9400F CPU @ 2.90GHz
Степпинг: 10
CPU MHz: 2903.996
БогоMIPS: 5807.99
Разработчик гипервизора:KVM
Тип Виртуализации:полный
L1d cache: 32K
L1i cache: 32K
L2 cache: 256K
L3 cache: 9216K
NUMA node0 CPU(s): 0-4
Флаги: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx rdtscp lm constant
_tsc rep_good nopl xtopology nonstop_tsc cpuid tsc_known_freq pni pclmulqd
q ssse3 cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt aes xsave avx rdrand h
ypervisor lahf_lm abm 3dnowprefetch invpcid_single pti fsgsbase bmi1 avx2
```

Рисунок 4.4 – Получение информации о процессоре lscpu

5 Конфигурация системы

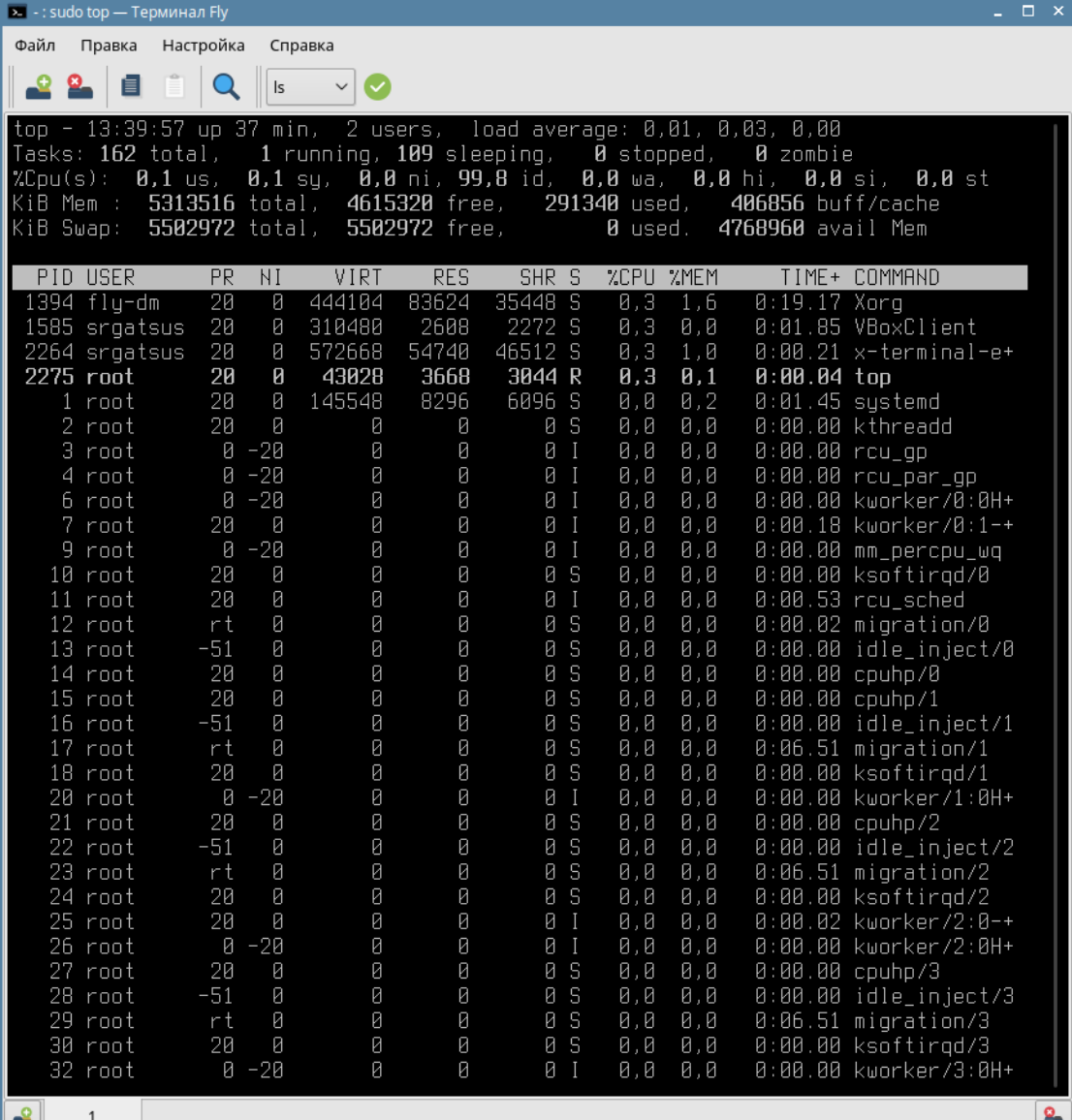
На рисунке 5 представлена команда, позволяющая получить информацию о загрузке системы.

```
~: bash — Терминал Fly
ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.1 145548 8296 ?        Ss   13:02   0:01 /sbin/init
root         2  0.0  0.0      0     0 ?        S    13:02   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        I<   13:02   0:00 [rcu_gp]
root         4  0.0  0.0      0     0 ?        I<   13:02   0:00 [rcu_par_gp]
root         6  0.0  0.0      0     0 ?        I<   13:02   0:00 [kworker/0:0H-kb]
root         7  0.0  0.0      0     0 ?        I    13:02   0:00 [kworker/0:1-cgr]
root         9  0.0  0.0      0     0 ?        I<   13:02   0:00 [mm_percpu_wq]
root        10  0.0  0.0      0     0 ?        S    13:02   0:00 [ksoftirqd/0]
root        11  0.0  0.0      0     0 ?        I    13:02   0:00 [rcu_sched]
root        12  0.0  0.0      0     0 ?        S    13:02   0:00 [migration/0]
root        13  0.0  0.0      0     0 ?        S    13:02   0:00 [idle_inject/0]
root        14  0.0  0.0      0     0 ?        S    13:02   0:00 [cpuhp/0]
root        15  0.0  0.0      0     0 ?        S    13:02   0:00 [cpuhp/1]
root        16  0.0  0.0      0     0 ?        S    13:02   0:00 [idle_inject/1]
root        17  0.3  0.0      0     0 ?        S    13:02   0:06 [migration/1]
root        18  0.0  0.0      0     0 ?        S    13:02   0:00 [ksoftirqd/1]
root        20  0.0  0.0      0     0 ?        I<   13:02   0:00 [kworker/1:0H-kb]
root        21  0.0  0.0      0     0 ?        S    13:02   0:00 [cpuhp/2]
root        22  0.0  0.0      0     0 ?        S    13:02   0:00 [idle_inject/2]
root        23  0.3  0.0      0     0 ?        S    13:02   0:06 [migration/2]
root        24  0.0  0.0      0     0 ?        S    13:02   0:00 [ksoftirqd/2]
root        25  0.0  0.0      0     0 ?        I    13:02   0:00 [kworker/2:0-eve]
root        26  0.0  0.0      0     0 ?        I<   13:02   0:00 [kworker/2:0H-kb]
root        27  0.0  0.0      0     0 ?        S    13:02   0:00 [cpuhp/3]
root        28  0.0  0.0      0     0 ?        S    13:02   0:00 [idle_inject/3]
root        29  0.3  0.0      0     0 ?        S    13:02   0:06 [migration/3]
root        30  0.0  0.0      0     0 ?        S    13:02   0:00 [ksoftirqd/3]
```

Рисунок 5 – Получение информации о загрузке системы – ps aux

6 Диспетчер задач

Диспетчер - `sudo top`. Предназначен для вывода на экран списка запущенных процессов и потребляемых ими ресурсов, в частности процессорное время и оперативная память. Результат выполнения программы представлен на рисунке 6.



```
top - 13:39:57 up 37 min, 2 users, load average: 0,01, 0,03, 0,00
Tasks: 162 total, 1 running, 109 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,1 us, 0,1 sy, 0,0 ni, 99,8 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 5313516 total, 4615320 free, 291340 used, 406856 buff/cache
KiB Swap: 5502972 total, 5502972 free, 0 used, 4768960 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 1394 fly-dm    20   0 444104  83624 35448 S   0,3   1,6   0:19.17 Xorg
 1585 srgatsus  20   0 310400   2608  2272 S   0,3   0,0   0:01.85 VBoxClient
 2264 srgatsus  20   0 572668  54740 46512 S   0,3   1,0   0:00.21 x-terminal-e+
 2275 root      20   0  43028   3668  3044 R   0,3   0,1   0:00.04 top
    1 root      20   0 145548   8296  6096 S   0,0   0,2   0:01.45 systemd
    2 root      20   0         0         0      0 S   0,0   0,0   0:00.00 kthreadd
    3 root       0 -20         0         0      0 I   0,0   0,0   0:00.00 rcu_gp
    4 root       0 -20         0         0      0 I   0,0   0,0   0:00.00 rcu_par_gp
    6 root       0 -20         0         0      0 I   0,0   0,0   0:00.00 kworker/0:0H+
    7 root      20   0         0         0      0 I   0,0   0,0   0:00.18 kworker/0:1-+
    9 root       0 -20         0         0      0 I   0,0   0,0   0:00.00 mm_percpu_wq
   10 root      20   0         0         0      0 S   0,0   0,0   0:00.00 ksoftirqd/0
   11 root      20   0         0         0      0 I   0,0   0,0   0:00.53 rcu_sched
   12 root      rt    0         0         0      0 S   0,0   0,0   0:00.02 migration/0
   13 root     -51   0         0         0      0 S   0,0   0,0   0:00.00 idle_inject/0
   14 root      20   0         0         0      0 S   0,0   0,0   0:00.00 cpuhp/0
   15 root      20   0         0         0      0 S   0,0   0,0   0:00.00 cpuhp/1
   16 root     -51   0         0         0      0 S   0,0   0,0   0:00.00 idle_inject/1
   17 root      rt    0         0         0      0 S   0,0   0,0   0:06.51 migration/1
   18 root      20   0         0         0      0 S   0,0   0,0   0:00.00 ksoftirqd/1
   20 root       0 -20         0         0      0 I   0,0   0,0   0:00.00 kworker/1:0H+
   21 root      20   0         0         0      0 S   0,0   0,0   0:00.00 cpuhp/2
   22 root     -51   0         0         0      0 S   0,0   0,0   0:00.00 idle_inject/2
   23 root      rt    0         0         0      0 S   0,0   0,0   0:06.51 migration/2
   24 root      20   0         0         0      0 S   0,0   0,0   0:00.00 ksoftirqd/2
   25 root      20   0         0         0      0 I   0,0   0,0   0:00.02 kworker/2:0-+
   26 root       0 -20         0         0      0 I   0,0   0,0   0:00.00 kworker/2:0H+
   27 root      20   0         0         0      0 S   0,0   0,0   0:00.00 cpuhp/3
   28 root     -51   0         0         0      0 S   0,0   0,0   0:00.00 idle_inject/3
   29 root      rt    0         0         0      0 S   0,0   0,0   0:06.51 migration/3
   30 root      20   0         0         0      0 S   0,0   0,0   0:00.00 ksoftirqd/3
   32 root       0 -20         0         0      0 I   0,0   0,0   0:00.00 kworker/3:0H+
```

Рисунок 6 – Диспетчер - `sudo top`.

7 Информация о BIOS

Узнать версию BIOS можно набрав в командной строке команду `sudo dmidecode -s bios`. Данная команда выдаст список команд, которые можно получить, например, `sudo dmidecode -s bios-version`. Результат выполнения приведен на рисунке 7.



```
sergatsus@astra:~$ sudo dmidecode -s bios
Invalid string keyword: bios
Valid string keywords are:
  bios-vendor
  bios-version
  bios-release-date
  system-manufacturer
  system-product-name
  system-version
  system-serial-number
  system-uuid
  baseboard-manufacturer
  baseboard-product-name
  baseboard-version
  baseboard-serial-number
  baseboard-asset-tag
  chassis-manufacturer
  chassis-type
  chassis-version
  chassis-serial-number
  chassis-asset-tag
  processor-family
  processor-manufacturer
  processor-version
  processor-frequency
sergatsus@astra:~$
```

Рисунок 7 – Получение информации о BIOS.

8 Применение Visual Basic Script

Visual Basic Script— скриптовый язык программирования. Скрипты в Astra Linux пишутся в расширении `.sh`. Для того, чтобы создать скрипт, необходимо создать текстовый файл и дать ему имя в формате:

< название >.sh

Далее нужно открыть файл (можно открыть правой кнопкой по ярлыку -> открыть с помощью-> выбираем kate).

Скрипт №1

```
while true
do
echo -n "still running ..."
date
sleep 300      # sleep 5 minutes
done
```

Для запуска скрипта нужно открыть консоль и ввести команду ls - если нет нашего скрипта, но есть название Desktop, вводим cd ./Desktop , смотрим там(должен там, если вы создавали на рабочем столе). Результат создания скрипта приведен на рисунке 8.1.

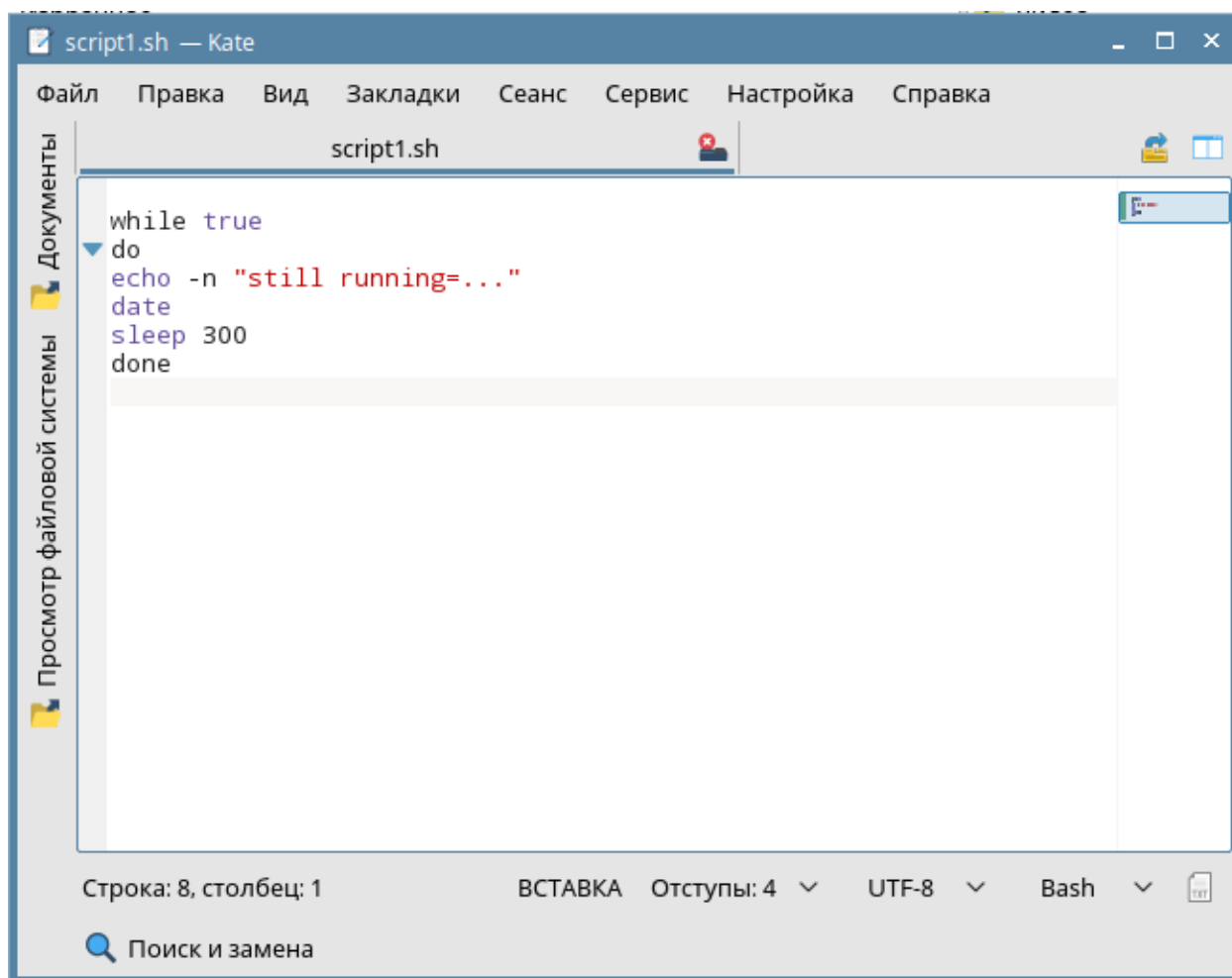


Рисунок 8.1 – Создание скрипта

Запуск скрипта происходит с помощью команды `bash ./<название скрипта>.sh`. Пример приведен на рисунке 8.2.

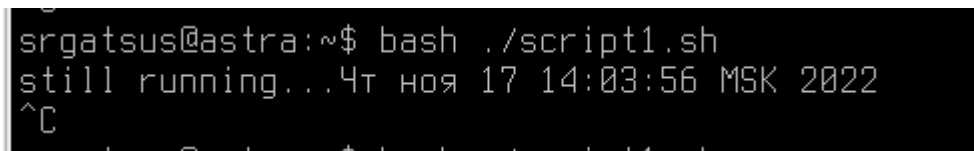
A terminal window with a black background and white text. The prompt is 'srgatsus@astra:~\$'. The user enters 'bash ./script1.sh'. The output is 'still running...Чт ноя 17 14:03:56 MSK 2022'. The user then presses Ctrl-C, indicated by '^C'.

Рисунок 8.2 – Запуск скрипта

9 Получение информации о процессах в системе

У каждого приложения есть свой `id` процесса. У ранее созданного скрипта `script1.sh` тоже есть `id`. Чтобы вывести `id` в скрипте воспользуемся `echo $$`, пример на рисунке 9.1.

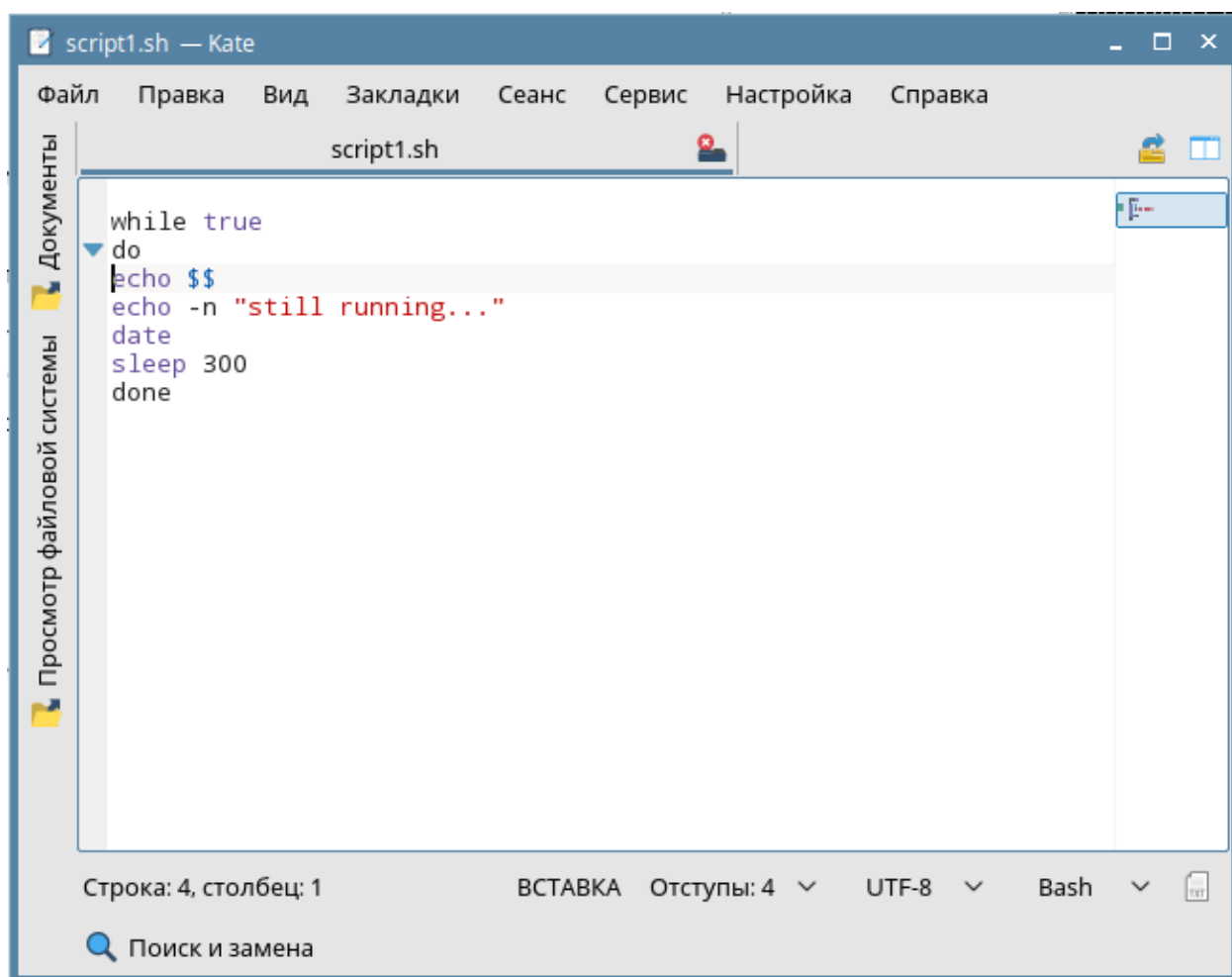


Рисунок 9.1 – Использование команды `echo $$`

В консоли необходимо запустить скрипт(рисунок 9.2).

```
srgatsus@astra:~$ bash ./script1.sh
2738
still running...Чт ноя 17 14:14:52 MSK 2022
```

Рисунок 9.2 – Запуск скрипта для получения id

В результате получаем id нашего процесса. Для того, чтобы «убить» процесс воспользуемся командой *kill <id процесса>*.

В нашем случае удаляем процесс с id «2738». На рисунке 9.3 приведен результат завершенного процесса.

```
Завершено
srgatsus@astra:~$
```

Рисунок 9.3 – Успешное удаление процесса

10 Определение сетевой информации

Для определения сетевой информации из командной строки нужно использовать команду «ip a» (рисунок 10).

```
srgatsus@astra:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP grou
p default qlen 1000
    link/ether 08:00:27:bd:59:8e brd ff:ff:ff:ff:ff:ff
    inet6 fe80::533f:402a:ce84:8167/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
srgatsus@astra:~$
```

Рисунок 10 – Получение сетевой информации командой «ip a»

Вывод:

В ходе лабораторной работы были получены базовые навыки работы с операционной системой Astra linux:

- получение имени компьютера с помощью командной строки;
- получение версии установленной OS;
- получение сведений о системе;
- получение списка параметров конфигурации системы;
- вход и просмотр диспетчера задач;
- получение информации о BIOS системы;
- применение Visual Basic Script;
- получение информации о процессах в системе;
- определение сетевой информации.