

Лабораторная работа №2

Сбор сведений о системе и управление процессами

Цель работы: знакомство с командами по работе с процессами и командами, предоставляющими сведения о системе.

Ход выполнения работы:

Часть 1. Сбор сведений о системе

1) С помощью команды `uname -a` вывел в терминал имя текущей UNIX-системы.

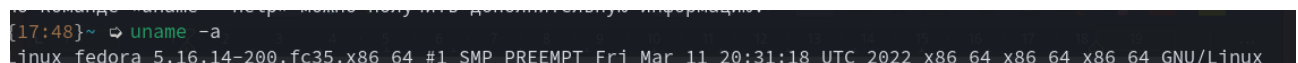


Рисунок 1. Результат выполнения команды.

2) Вывел содержимое каталога `proc`.

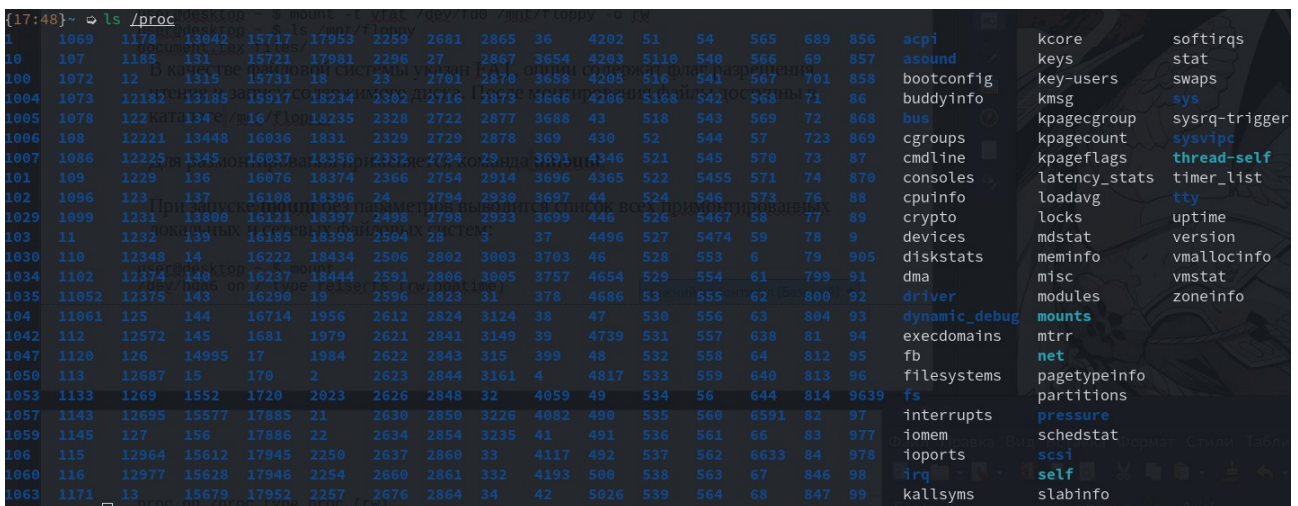


Рисунок 2. Результат выполнения команды.

3) Вывел в терминал текущие пользовательские сеансы, используя команду `who`.



Рисунок 3. Результат выполнения команды.

4) Вывел список всех примонтированных устройств с помощью команды `mount`, введенной без параметров.

```

[18:02]~ ➔ mount
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=4096k,nr_inodes=131072,mode=755,inode64)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=5729696k,nr_inodes=819200,mode=755,inode64)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,seclabel,nsdelegate,memory_recursiveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime,seclabel)
efivarfs on /sys/firmware/efi/efivars type efivarfs (rw,nosuid,nodev,noexec,relatime)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
/dev/nvme0n1p7 on / type btrfs (rw,relatime,seclabel,compress=zstd:1,ssd,space_cache,subvol=257,subvol=/root)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=31,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=18901)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)

```

Рисунок 4. Результат выполнения команды.

5) Вывел загрузенность всех файлов с помощью команды `df` и аргументом `-h` для большего понимания.

```

[18:09]~ ➔ df -h
Файловая система  Размер  Использовано  Дост  Использовано%  Смонтировано в
devtmpfs          4,0М      0      4,0М      0% /dev
tmpfs             14Г      2,3М      14Г      1% /dev/shm
tmpfs             5,5Г      2,2М      5,5Г      1% /run
/dev/nvme0n1p7    270Г      69Г      201Г      26% /
tmpfs             14Г      1,6М      14Г      1% /tmp
/dev/nvme0n1p7    270Г      69Г      201Г      26% /home
/dev/loop2        128К      128К      0      100% /var/lib/snapd/snap/bare/5
/dev/loop3        256К      256К      0      100% /var/lib/snapd/snap/gtk-theme-nordic/2
/dev/loop7        111М      111М      0      100% /var/lib/snapd/snap/core/12821
/dev/loop0        210М      210М      0      100% /var/lib/snapd/snap/code/91
/dev/loop1        157М      157М      0      100% /var/lib/snapd/snap/dotnet-sdk/158
/dev/loop4        165М      165М      0      100% /var/lib/snapd/snap/gnome-3-28-1804/161
/dev/loop5        56М      56М      0      100% /var/lib/snapd/snap/core18/2284
/dev/loop8        82М      82М      0      100% /var/lib/snapd/snap/onenote-desktop/15
/dev/loop6        66М      66М      0      100% /var/lib/snapd/snap/gtk-common-themes/1519
/dev/loop9        92М      92М      0      100% /var/lib/snapd/snap/p3x-onenote/138
/dev/loop11       44М      44М      0      100% /var/lib/snapd/snap/snapd/15177
/dev/loop10       1,1Г      1,1Г      0      100% /var/lib/snapd/snap/rider/263
/dev/nvme0n1p6    974М      289М      618М      32% /boot
/dev/nvme0n1p3    96М      41М      56М      43% /boot/efi
tmpfs             2,8Г      100К      2,8Г      1% /run/user/1000

```

Рисунок 5. Результат выполнения команды.

6) Командой `ps` аух вывел информацию о всех выполняющихся процессах.

```

USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root            1  0.0  0.0 172412 17748 ?        Ss   16:10   0:01 /usr/lib/systemd/systemd rhgb --switch
root            2  0.0  0.0      0      0 ?        S    16:10   0:00 [kthreadd]
root            3  0.0  0.0      0      0 ?        I<   16:10   0:00 [rcu_gp]
root            4  0.0  0.0      0      0 ?        I<   16:10   0:00 [rcu_par_gp]
root            6  0.0  0.0      0      0 ?        I<   16:10   0:00 [kworker/0:0H-events_highpri]
root            9  0.0  0.0      0      0 ?        I<   16:10   0:00 [mm_percpu_wq]
root           10  0.0  0.0      0      0 ?        S    16:10   0:00 [rcu_tasks_kthre]
root           11  0.0  0.0      0      0 ?        S    16:10   0:00 [rcu_tasks_rude_]
root           12  0.0  0.0      0      0 ?        S    16:10   0:00 [rcu_tasks_trace]
root           13  0.0  0.0      0      0 ?        S    16:10   0:00 [ksoftirqd/0]
root           14  0.3  0.0      0      0 ?        I    16:10   0:28 [rcu_preempt]
root           15  0.0  0.0      0      0 ?        S    16:10   0:00 [migration/0]
root           16  0.0  0.0      0      0 ?        S    16:10   0:00 [cpuhp/0]
root           17  0.0  0.0      0      0 ?        S    16:10   0:00 [cpuhp/1]
root           18  0.0  0.0      0      0 ?        S    16:10   0:00 [migration/1]
root           19  0.0  0.0      0      0 ?        S    16:10   0:00 [ksoftirqd/1]
root           21  0.0  0.0      0      0 ?        I<   16:10   0:00 [kworker/1:0H-kblockd]
root           22  0.0  0.0      0      0 ?        S    16:10   0:00 [cpuhp/2]
root           23  0.0  0.0      0      0 ?        S    16:10   0:00 [migration/2]
root           24  0.0  0.0      0      0 ?        S    16:10   0:00 [ksoftirqd/2]
root           27  0.0  0.0      0      0 ?        S    16:10   0:00 [cpuhp/3]
root           28  0.0  0.0      0      0 ?        S    16:10   0:00 [migration/3]
root           29  0.0  0.0      0      0 ?        S    16:10   0:00 [ksoftirqd/3]
root           31  0.0  0.0      0      0 ?        I<   16:10   0:00 [kworker/3:0H-events_highpri]
root           32  0.0  0.0      0      0 ?        S    16:10   0:00 [cpuhp/4]
root           33  0.0  0.0      0      0 ?        S    16:10   0:00 [migration/4]
root           34  0.0  0.0      0      0 ?        S    16:10   0:00 [ksoftirqd/4]

```

Рисунок 6. Результат выполнения команды.

7) Вывел системные процессы.

```
{18:23}~ ➔ ps aux | grep -v user
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	172412	17748	?	Ss	16:10	0:01	/usr/lib/systemd/
root	2	0.0	0.0	0	0	?	S	16:10	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	I<	16:10	0:00	[rcu_gp]
root	4	0.0	0.0	0	0	?	I<	16:10	0:00	[rcu_par_gp]
root	6	0.0	0.0	0	0	?	I<	16:10	0:00	[kworker/0:0H-eve
root	9	0.0	0.0	0	0	?	I<	16:10	0:00	[mm_percpu_wq]
root	10	0.0	0.0	0	0	?	S	16:10	0:00	[rcu_tasks_kthre]
root	11	0.0	0.0	0	0	?	S	16:10	0:00	[rcu_tasks_rude_]
root	12	0.0	0.0	0	0	?	S	16:10	0:00	[rcu_tasks_trace]
root	13	0.0	0.0	0	0	?	S	16:10	0:00	[ksoftirqd/0]
root	14	0.3	0.0	0	0	?	I	16:10	0:31	[rcu_preempt]
root	15	0.0	0.0	0	0	?	S	16:10	0:00	[migration/0]
root	16	0.0	0.0	0	0	?	S	16:10	0:00	[cpuhp/0]
root	17	0.0	0.0	0	0	?	S	16:10	0:00	[cpuhp/1]
root	18	0.0	0.0	0	0	?	S	16:10	0:00	[migration/1]
root	19	0.0	0.0	0	0	?	S	16:10	0:00	[ksoftirqd/1]
root	21	0.0	0.0	0	0	?	I<	16:10	0:00	[kworker/1:0H-kbl
root	22	0.0	0.0	0	0	?	S	16:10	0:00	[cpuhp/2]
root	23	0.0	0.0	0	0	?	S	16:10	0:00	[migration/2]
root	24	0.0	0.0	0	0	?	S	16:10	0:00	[ksoftirqd/2]
root	27	0.0	0.0	0	0	?	S	16:10	0:00	[cpuhp/3]
root	28	0.0	0.0	0	0	?	S	16:10	0:00	[migration/3]
root	29	0.0	0.0	0	0	?	S	16:10	0:00	[ksoftirqd/3]
root	31	0.0	0.0	0	0	?	I<	16:10	0:00	[kworker/3:0H-eve
root	32	0.0	0.0	0	0	?	S	16:10	0:00	[cpuhp/4]
root	33	0.0	0.0	0	0	?	S	16:10	0:00	[migration/4]
root	34	0.0	0.0	0	0	?	S	16:10	0:00	[ksoftirqd/4]
root	36	0.0	0.0	0	0	?	I<	16:10	0:00	[kworker/4:0H-eve
root	37	0.0	0.0	0	0	?	S	16:10	0:00	[cpuhp/5]
root	38	0.0	0.0	0	0	?	S	16:10	0:00	[migration/5]

Рисунок 7. Результат выполнения команды.

8) Вывел иерархию процессов с помощью команды `ps tree`. В вершине стоит процесс `systemmd`, это главный процесс в Fedora Linux.

```
{18:29}~ ➔ pstree
systemd
├── ModemManager──3*[{ModemManager}] r4 #1 Mon Oct 17 3
├── NetworkManager──2*[{NetworkManager}]
├── abrt-dbus──2*[{abrt-dbus}] amu
├── 3*[abrt-dump-journ]
├── abrttd──2*[{abrttd}]
├── accounts-daemon──3*[{accounts-daemon}]
├── alsactl
├── auditd──{auditd} используется для отправки сигнала пр
├── avahi-daemon──avahi-daemon
├── bluetoothd──bluetoothd для kth имеет один из следующих формат
├── chronyd
├── colord──3*[{colord}] вание сигнала] идентифи
├── cupsd
├── dbus-broker-lau──dbus-brokerнала идентификато
├── firewalld──{firewalld}
├── gdm
│   ├── gdm-session-wor──gdm-wayland-ses──gnome-sess
│   │   └── 2*[{gdm-session-wor}]
│   └── 2*[{gdm}] чтобы отправить сигнал, необходимо
├── geoclue──3*[{geoclue}] идентификатор можно с по
├── gnome-keyring-d──3*[{gnome-keyring-d}] зан, то посыла
├── gssproxy──5*[{gssproxy}]
├── low-memory-monit──2*[{low-memory-monit}]
├── packagekitd──2*[{packagekitd}]
├── pcscd──25*[{pcscd}]
└── polkitd──11*[{polkitd}]
```

Рисунок 8. Результат выполнения команды.

9) С помощью команды `top` просмотрел поведение процессов интерактивно.

```
top - 18:31:41 up 2:20, 1 user, load average: 0,90, 1,20, 1,14
Tasks: 384 total, 3 running, 381 sleeping, 0 stopped, 0 zombie
%Cpu(s): 6,8 us, 3,5 sy, 0,0 ni, 89,3 id, 0,0 wa, 0,0 hi, 0,4 si, 0,0 st
MiB Mem : 27977,0 total, 18538,2 free, 3953,8 used, 5485,0 buff/cache
MiB Swap: 8192,0 total, 8192,0 free, 0,0 used. 23488,6 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4496	SergoGa+	20	0	4522876	817640	359844	S	79,1	2,9	32:38.03	firefox
18734	SergoGa+	20	0	2936492	278964	115444	R	48,2	1,0	8:52.60	Isolated Web Co
2366	SergoGa+	20	0	7438152	460040	199780	R	33,9	1,6	23:06.30	gnome-shell
23225	SergoGa+	20	0	1897044	17512	8876	S	5,0	0,1	0:16.88	kite-update
527	root	-2	0	0	0	0	S	3,3	0,0	0:59.19	gfx
856	root	20	0	0	0	0	S	2,3	0,0	1:36.07	napi/phy0-8194
446	root	-51	0	0	0	0	S	2,0	0,0	0:46.80	irq/54-MSFT0001
905	root	-2	0	0	0	0	S	2,0	0,0	1:22.69	mt76-tx phy0
13042	SergoGa+	20	0	2851404	174644	109964	S	1,7	0,6	1:02.86	Isolated Web Co
858	root	20	0	0	0	0	S	1,3	0,0	0:49.25	napi/phy0-8196
2623	SergoGa+	9	-11	306240	66560	7516	S	1,3	0,2	2:05.55	pipewire-pulse
21010	root	20	0	0	0	0	I	1,3	0,0	0:01.19	kworker/u32:8-mt76
2621	SergoGa+	9	-11	334376	21900	9308	S	1,0	0,1	1:16.77	pipewire
12221	SergoGa+	20	0	3112884	325256	169164	S	0,7	1,1	4:22.55	Isolated Web Co
12977	SergoGa+	20	0	3044628	276756	171336	S	0,7	1,0	1:35.53	Isolated Web Co
14	root	20	0	0	0	0	I	0,3	0,0	0:33.14	rcu_preempt
1053	rtkit	21	1	153916	1552	1332	S	0,3	0,0	0:00.26	rtkit-daemon
12375	SergoGa+	20	0	2744012	142196	112720	S	0,3	0,5	1:12.93	Isolated Web Co
18374	root	20	0	0	0	0	I	0,3	0,0	0:00.27	kworker/12:2-events
23471	SergoGa+	20	0	225976	4388	3520	R	0,3	0,0	0:00.03	top
1	root	20	0	172412	17748	11248	S	0,0	0,1	0:01.74	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.02	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-events_highpri

Рисунок 9. Результат выполнения команды.

Часть 2. Управление процессами с помощью сигналов

1) Запустил команду `yues` и прервал её нажатием `Ctrl-C`.

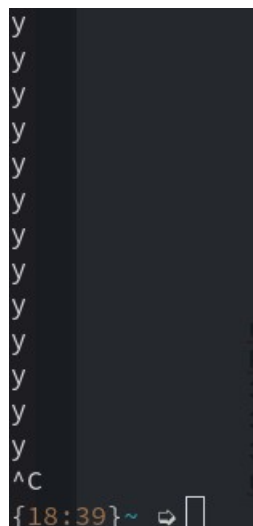


Рисунок 10. Результат выполнения команды.

2) Запустил сбор информации обо всех файлах системы.

```
{18:49}~ ➔ find / > testLab.txt
find: '/boot/loader/entries': Отказано в доступе
find: '/boot/lost+found': Отказано в доступе
find: '/boot/grub2': Отказано в доступе
find: '/boot/efi': Отказано в доступе
find: '/proc/tty/driver': Отказано в доступе
find: '/proc/1/task/1/fd': Отказано в доступе
find: '/proc/1/task/1/ns': Отказано в доступе
```

Рисунок 11. Результат выполнения команды.

3) Нашел необходимый PID.

```
{18:48}~ ➔ ps aux | grep find
SergoGa+ 26202 0.0 0.0 221824 2364 pts/1 S+ 1
e-dir=CVS --exclude-dir=.git --exclude-dir=.hg --exclud
```

Рисунок 12. Результат выполнения команды.

4) С помощью команды kill 2623 остановил этот процесс.

5) Попытался остановить системный процесс.

```
{18:50}~ ➔ kill 2
kill: kill 2 failed: Операция не позволена
{19:47}~ ➔
```

Рисунок 12. Результат выполнения команды.

6) killall bash.

7) killalll — SIGNAL find.

Часть 3. Выполнение задач в фоновом режиме

1) Запустил длительную команду и приостановил её с помощью нажатия Ctrl-Z.

```
find: '/proc/21166/task/21166/fd': Отказано в доступе
find: '/proc/21166/task/21166/ns': Отказано в доступе
find: '/proc/21166/fd': Отказано в доступе
find: '/proc/21166/map_files': Отказано в доступе
find: '/proc/21166/ns': Отказано в доступе
^Z
[2] + 29281 suspended find / > testLab.txt
{19:58}~ ➔
```

Рисунок 13. Результат выполнения команды.

2) Посмотреть текущий список запущенных задач командной оболочки.

```

[19:58]~ ↵ jobs jobs -x команда [аргументы]
[1] - suspended sudo find / > testLab.txt
[2] + suspended find / > testLab.txt

```

Рисунок 14. Результат выполнения команды.

3) С помощью команды fg запустил процесс 2.

4) Снова приостановил процесс и запустил его в фоновом режиме с помощью команды `bg`.

```
{20:03}~ ➦ jobs
[1]      suspended sudo find / > testLab.txt
[2]      - suspended find / > testLab.txt
[3]      + suspended find / > testLab.txt
{20:03}~ ➦ bg 3
```

Рисунок 15. Результат выполнения команды.

5) Запустил программу в фоновом режиме.

```
{20:04}~ ❯ neofetch $ cat /etc/issue.d/issue используете для запуска программ в фоновом режиме. Пример:
      .',;,:;:;,'.
      .':;cccccccccccc;:,. % mplayer /usr/share/sounds/alsa/Boot.mp3&
      .;cccccccccccccccccccc;
      .:cccccccccccccccccccccc:
      .;cccccccccccc;:ddd!.;:cccccc;
      .:cccccccccccc;0WMK00XmWd;:cccccc:
      .:cccccccccccc;KMMc;cc;xMMc:cccccc:
      .:cccccccccccc;MMM.;cc;:WW:;cccccc:
      .:cccccccccccc;MMM.;cccccccccccccc:
      .:cccccc;ox000o;MMM000k.;cccccccccc:
      .:cccccc;0MMKxdd;MMMkddc.;cccccccccc:
      .:cccc;XM0';ccc;MMM.;cccccccccccccc'
      .:cccc;MMo;cccc;MMW.;cccccccccccccc;
      .:cccc;0MNC.ccc.xMMD;cccccccccccccc;
      .:cccc;dnMWXXxWM0:;cccccccccccccc;
      .:cccccc;.odl:.;cccccccccccccc:
      .:cccccc;.odl:.;cccccccccccccc:

SergoGansta777@fedora
OS: Fedora Linux 35 (Workstation Edition) x86_64
Host: 82JQ Legion 5 Pro 16ACH6H
Kernel: 5.16.14-200.fc35.x86_64
Uptime: 3 hours, 54 mins
Packages: 2454 (rpm), 44 (flatpak)
Shell: zsh 5.8.1
Resolution: 2560x1600
DE: GNOME 41.4
WM: Mutter
WM Theme: WhiteSur-dark-solid
Theme: WhiteSur-dark-nord [GTK2/3]
Icons: WhiteSur-dark [GTK2/3]
Terminal: gnome-terminal
CPU: AMD Ryzen 7 5800H with Radeon Graphics (16) @ 3.200GHz
```

Рисунок 16. Результат выполнения команды.

Часть 4. Запуск демонов

1) Запустил команду в фоновом режиме.

```
[20:22] ~ ➤ find / -name "*.html" -exec grep -Hn "linux loader" \{\}\; &
```

Рисунок 17. Результат выполнения команды.

2) Для того, чтобы программа не получила сигнал SIGHUP, ввел команду `nohup find / -name "*.txt" -exec grep -Hn "linux loader" \{\}\; &`.

```
[20:25]~ ➔ nohup find / -name "*.txt" -exec grep -Hn "linux loader" \{\}\; &.  
[1] 33736
```

Рисунок 18. Результат выполнения команды.

Часть 5. Изменение приоритетов выполняющихся программ

1) С помощью команды `ps -l` посмотрел уровни приоритетов в процессах.

```
[20:25]~ ➔ ps -l  
F S  UID      PID      PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD  
0 S   1000    26278    6591  0  80   0 - 56486 sigsus pts/2    00:00:00 zsh  
4 R   1000    33849    26278  0  80   0 - 56323 - режим pts/2    00:00:00 ps
```

Рисунок 19. Результат выполнения команды.

2) Запустил создание архива с пониженным приоритетом.

```
[20:41]~ ➔ nice -n 2 tar -cjf testing.tar.bz2 testLab.txt  
[20:42]~ ➔
```

Рисунок 20. Результат выполнения команды.

3) С помощью команды `renice` можно понизить приоритет уже запущенного процесса.