

Практика 3.

Выполнила: Парамонова Кристина – 2 подгруппа

Раздел 1.

1. Войдите под пользователем user1 из практики 2 (su - user1)

```
root@eltex-practice2-pg2-v4:~# su user1
$
```

2. Подсчитайте количество процессов, имеющих несколько потоков выполнения

```
root@eltex-practice2-pg2-v4:~# ps -eo nlwp,comm | awk '$1 > 1 {count ++} END {print "Всего: ", count}'
Всего: 12
```

3. Запустите top и настройте вывод полей с информацией о процессе следующим образом:

- удалите поля VIRT, RES, SHR;
- добавьте поле RUSER и сделайте так, чтобы это поле было показано после поля USER;

```
top - 13:19:47 up 2 days, 5:22, 1 user, load average: 0.00, 0.01, 0.00
Tasks: 111 total, 2 running, 107 sleeping, 2 stopped, 0 zombie
%Cpu(s): 0.2 us, 0.0 sy, 0.0 ni, 99.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3771.7 total, 2154.3 free, 522.9 used, 1386.2 buff/cache
MiB Swap: 3185.0 total, 3184.5 free, 0.5 used. 3248.8 avail Mem
```

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
42688	root	root	20	0	R	0.3	0.0	0:00.08	kworker/0:0-events
1	root	root	20	0	S	0.0	0.3	0:07.99	systemd
2	root	root	20	0	S	0.0	0.0	0:00.03	kthreadd
3	root	root	20	0	S	0.0	0.0	0:00.00	pool_workqueue_release
4	root	root	0	-20	I	0.0	0.0	0:00.00	kworker/R-rcu_g
5	root	root	0	-20	I	0.0	0.0	0:00.00	kworker/R-rcu_p
6	root	root	0	-20	I	0.0	0.0	0:00.00	kworker/R-slab_
7	root	root	0	-20	I	0.0	0.0	0:00.00	kworker/R-netns
12	root	root	0	-20	I	0.0	0.0	0:00.00	kworker/R-mm_pe
13	root	root	20	0	I	0.0	0.0	0:00.00	rcu_tasks_kthread
14	root	root	20	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_kthread
15	root	root	20	0	I	0.0	0.0	0:00.00	rcu_tasks_trace_kthread
16	root	root	20	0	S	0.0	0.0	0:00.15	ksoftirqd/0
17	root	root	20	0	I	0.0	0.0	0:01.64	rcu_preempt
18	root	root	rt	0	S	0.0	0.0	0:00.71	migration/0
19	root	root	-51	0	S	0.0	0.0	0:00.00	idle_inject/0
20	root	root	20	0	S	0.0	0.0	0:00.00	cpuhp/0
21	root	root	20	0	S	0.0	0.0	0:00.00	cpuhp/1
22	root	root	-51	0	S	0.0	0.0	0:00.00	idle_inject/1
23	root	root	rt	0	S	0.0	0.0	0:00.83	migration/1
24	root	root	20	0	S	0.0	0.0	0:00.13	ksoftirqd/1
26	root	root	0	-20	I	0.0	0.0	0:00.00	kworker/1:0H-events_highpri

4. В другом терминальном окне выполните команду `passwd` и оставьте ее в состоянии запроса текущего пароля

```
$ passwd
Changing password for user1.
Current password: █
```

5. Перейдите в терминальное окно с `top` и выполните следующие действия:

- выведите все процессы, для которых реальным пользователем является пользователь, которым вы вошли в сеанс;
- найдите процесс, запущенный командой `passwd`;

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
42716	user1	user1	20	0	S	0.0	0.1	0:00.03	screen
42717	user1	user1	20	0	S	0.0	0.2	0:00.97	top
42719	root	root	20	0	I	0.0	0.0	0:00.13	kworker/u4:0-events_power_efficient
42725	root	root	20	0	S	0.0	0.1	0:00.00	su
42726	user1	user1	20	0	S	0.0	0.0	0:00.00	sh
42733	root	root	20	0	I	0.0	0.0	0:00.12	kworker/u4:3-flush-252:0
42735	user1	user1	20	0	S	0.0	0.1	0:00.00	screen
42736	user1	user1	20	0	S	0.0	0.1	0:00.21	screen
42737	user1	user1	20	0	S	0.0	0.0	0:00.00	sh
42739	user1	user1	20	0	S	0.0	0.0	0:00.00	sh
42747	root	root	20	0	I	0.0	0.0	0:00.01	kworker/u4:1-flush-252:0
42754	root	root	20	0	I	0.0	0.0	0:00.00	kworker/1:3
42763	user1	user1	20	0	R	0.0	0.2	0:00.44	top
42771	root	root	20	0	I	0.0	0.0	0:00.00	kworker/0:2-events
42773	root	user1	20	0	S	0.0	0.1	0:00.00	passwd
42778	root	root	20	0	I	0.0	0.0	0:00.00	kworker/0:1

- отправьте этому процессу сигналы 15 (SIGTERM), 2 (SIGINT), 3 (SIGQUIT), 9 (SIGKILL)

15 (SIGTERM)

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
42707	root	root	20	0	I	0.0	0.0	0:00.15	kworker/u4:2-events_power_efficient
42716	user1	user1	20	0	S	0.0	0.1	0:00.04	screen
42717	user1	user1	20	0	S	0.0	0.2	0:01.04	top
42719	root	root	20	0	I	0.0	0.0	0:00.14	kworker/u4:0-events_power_efficient
42725	root	root	20	0	S	0.0	0.1	0:00.00	su
42726	user1	user1	20	0	S	0.0	0.0	0:00.00	sh
42733	root	root	20	0	I	0.0	0.0	0:00.12	kworker/u4:3-events_unbound
42735	user1	user1	20	0	S	0.0	0.1	0:00.00	screen
42736	user1	user1	20	0	S	0.0	0.1	0:00.22	screen
42737	user1	user1	20	0	S	0.0	0.0	0:00.00	sh
42739	user1	user1	20	0	S	0.0	0.0	0:00.00	sh
42754	root	root	20	0	I	0.0	0.0	0:00.00	kworker/1:3
42771	root	root	20	0	I	0.0	0.0	0:00.00	kworker/0:2-events
42773	root	user1	20	0	S	0.0	0.1	0:00.00	passwd
42778	root	root	20	0	I	0.0	0.0	0:00.00	kworker/0:1-mm_percpu_wq

2 (SIGINT)

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
42707	root	root	20	0	I	0.0	0.0	0:00.15	kworker/u4:2-events_unbound
42716	user1	user1	20	0	S	0.0	0.1	0:00.04	screen
42719	root	root	20	0	I	0.0	0.0	0:00.14	kworker/u4:0-events_power_efficient
42725	root	root	20	0	S	0.0	0.1	0:00.00	su
42726	user1	user1	20	0	S	0.0	0.0	0:00.00	sh
42733	root	root	20	0	I	0.0	0.0	0:00.12	kworker/u4:3-events_unbound
42735	user1	user1	20	0	S	0.0	0.1	0:00.00	screen
42736	user1	user1	20	0	S	0.0	0.1	0:00.22	screen
42737	user1	user1	20	0	S	0.0	0.0	0:00.00	sh
42739	user1	user1	20	0	S	0.0	0.0	0:00.00	sh
42754	root	root	20	0	I	0.0	0.0	0:00.00	kworker/1:3
42763	user1	user1	20	0	R	0.0	0.2	0:00.57	top
42771	root	root	20	0	I	0.0	0.0	0:00.00	kworker/0:2-events
42773	root	user1	20	0	S	0.0	0.1	0:00.00	passwd
42778	root	root	20	0	I	0.0	0.0	0:00.00	kworker/0:1-events

3 (SIGQUIT)

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
42707	root	root	20	0	I	0.0	0.0	0:00.16	kworker/u4:2-events_power_efficient
42716	user1	user1	20	0	S	0.0	0.1	0:00.04	screen
42717	user1	user1	20	0	S	0.0	0.2	0:01.09	top
42719	root	root	20	0	I	0.0	0.0	0:00.15	kworker/u4:0-events_power_efficient
42725	root	root	20	0	S	0.0	0.1	0:00.00	su
42726	user1	user1	20	0	S	0.0	0.0	0:00.00	sh
42733	root	root	20	0	I	0.0	0.0	0:00.13	kworker/u4:3-events_power_efficient
42735	user1	user1	20	0	S	0.0	0.1	0:00.00	screen
42736	user1	user1	20	0	S	0.0	0.1	0:00.23	screen
42737	user1	user1	20	0	S	0.0	0.0	0:00.00	sh
42739	user1	user1	20	0	S	0.0	0.0	0:00.00	sh
42754	root	root	20	0	I	0.0	0.0	0:00.00	kworker/1:3
42771	root	root	20	0	I	0.0	0.0	0:00.00	kworker/0:2-events
42773	root	user1	20	0	S	0.0	0.1	0:00.00	passwd
42778	root	root	20	0	I	0.0	0.0	0:00.00	kworker/0:1-events

9 (SIGKILL)

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
42707	root	root	20	0	I	0.0	0.0	0:00.16	kworker/u4:2-events_unbound
42716	user1	user1	20	0	S	0.0	0.1	0:00.04	screen
42719	root	root	20	0	I	0.0	0.0	0:00.15	kworker/u4:0-events_power_efficient
42725	root	root	20	0	S	0.0	0.1	0:00.00	su
42726	user1	user1	20	0	S	0.0	0.0	0:00.00	sh
42733	root	root	20	0	I	0.0	0.0	0:00.13	kworker/u4:3-events_power_efficient
42735	user1	user1	20	0	S	0.0	0.1	0:00.00	screen
42736	user1	user1	20	0	S	0.0	0.1	0:00.23	screen
42737	user1	user1	20	0	S	0.0	0.0	0:00.00	sh
42739	user1	user1	20	0	S	0.0	0.0	0:00.00	sh
42754	root	root	20	0	I	0.0	0.0	0:00.00	kworker/1:3
42763	user1	user1	20	0	R	0.0	0.2	0:00.61	top
42778	root	root	20	0	I	0.0	0.0	0:00.00	kworker/0:1-mm_percpu_wq

```
$ passwd
Changing password for user1.
Current password: Killed
```

6. Выполните команду `vim ~/file_task3.txt` и нажмите `Ctrl-Z`

```
~
~/file_task3.txt [New]
```

```
root@eltex-practice2-pg2-v4:~# vim ~/file_task3.txt
[1]+  Stopped                  vim ~/file_task3.txt
```

7. Выполните команду `sleep 600`, нажмите `Ctrl-Z` и выполните команду `jobs`

```
root@eltex-practice2-pg2-v4:~# sleep 600
^Z
[2]+  Stopped                  sleep 600
root@eltex-practice2-pg2-v4:~# jobs
[1]-  Stopped                  vim ~/file_task3.txt
[2]+  Stopped                  sleep 600
```

8. Последнее задание (`sleep 600`) сделайте фоновым

```
root@eltex-practice2-pg2-v4:~# sleep 600&
[3] 44812
```

9. Измените число NICE у задания (`sleep 600`), сделав его равным 10

```
root@eltex-practice2-pg2-v4:~# sudo renice 10 -p $(pidof sleep)
44812 (process ID) old priority 0, new priority 10
44811 (process ID) old priority 0, new priority 10
44807 (process ID) old priority 0, new priority 10
44803 (process ID) old priority 0, new priority 10
```

10. Проверьте, что число NICE у этого задания изменилось

```
root@eltex-practice2-pg2-v4:~# ps -l
F S    UID      PID      PPID    C  PRI   NI   ADDR  SZ  WCHAN  TTY          TIME CMD
4 S     0      44785    44686   0   80    0    -    2162 do_wai pts/0        00:00:00 bash
0 T     0      44809    44785   0   80    0    -    6186 do_sig pts/0        00:00:00 vim
0 T     0      44811    44785   0   90   10    -    1421 do_sig pts/0        00:00:00 sleep
0 S     0      44812    44785   0   90   10    -    1421 hrtime pts/0        00:00:00 sleep
4 R     0      44819    44785   0   80    0    -    2729 -      pts/0        00:00:00 ps
```

11. Сделайте задание `vim ~/file_task3.txt` активным и выйдите из редактора

E325: ATTENTION

```
Found a swap file by the name "~/.file_task3.txt.swp"
  owned by: root   dated: Sat Oct 18 07:36:18 2025
  file name: ~root/file_task3.txt
  modified: no
  user name: root   host name: eltex-practice2-pg2-v4
  process ID: 44809 (STILL RUNNING)
While opening file "/root/file_task3.txt"
  CANNOT BE FOUND
(1) Another program may be editing the same file.  If this is the case,
    be careful not to end up with two different instances of the same
    file when making changes.  Quit, or continue with caution.
(2) An edit session for this file crashed.
    If this is the case, use ":recover" or "vim -r /root/file_task3.txt"
    to recover the changes (see ":help recovery").
    If you did this already, delete the swap file "/root/.file_task3.txt.swp"
    to avoid this message.

Swap file "~/.file_task3.txt.swp" already exists!
~/file_task3.txt" [New]
:wq
```

12. Отправьте сигнал 15 (SIGTERM) заданию `sleep 600` и выполните команду `jobs`

```
root@eltex-practice2-pg2-v4:~# jobs
[1]+  Stopped                  vim ~/file_task3.txt
[2]-  Stopped                  sleep 600
[3]   Running                  sleep 600 &
root@eltex-practice2-pg2-v4:~# kill -15 %2

[2]-  Stopped                  sleep 600
[3]   Done                     sleep 600
root@eltex-practice2-pg2-v4:~# jobs
[1]+  Stopped                  vim ~/file_task3.txt
[2]-  Terminated              sleep 600
```

13. Создайте перехватчик сигналов SIGINT и SIGQUIT внутри командного интерпретатора, который выводит сообщение «Меня голыми руками не возмѣшь!» (используйте встроенную команду `trap`) и отправьте сигналы самому себе

```
root@eltex-practice2-pg2-v4:~# trap 'echo "Меня голыми руками не возмѣшь!"' SIGINT SIGQUIT
Меня голыми руками не возмѣшь!^C
Меня голыми руками не возмѣшь!^\\
```

Раздел 2.

1. Создайте скрипт на языке bash с именем `template_task.sh`, делающий следующее:

- При запуске проверяет, что имя скрипта не совпадает с `template_task.sh`, если совпадает - выходит с уведомлением «я бригадир, сам не работаю»
- При запуске дописывает в файл `report_имя_скрипта_без_полного_пути.log` в рабочем каталоге информацию: [PID] ДАТА ВРЕМЯ Скрипт запущен
- Генерирует случайное число от 30 до 1800 и ждет такое количество секунд
- Дописывает в файл `report_имя_скрипта_без_полного_пути.log` сообщение: [PID] ДАТА ВРЕМЯ Скрипт завершился, работал N минут

```
filename=$(basename "$0")
if [ filename=="template_task.sh" ];
then
echo "Я бригадир, сам не работаю"
exit
fi

report_file="report_${filename}.log"

pid=$$
start=$(date '+%m-%d %H:%M')
echo "[${pid}] $start Скрипт запущен" >> "$report_file"

wait_time=$(( RANDOM % 1771 + 30 ))
sleep wait_time

minutes=$(( wait_time / 60 ))
end=$(date '+%m-%d %H:%M')
echo "[${pid}] $end Скрипт завершился, работал $minutes минут" >> "$report_file"
```

2. Создайте скрипт на языке bash с именем `observer.sh`, читающий файл конфигурации со списком скриптов `observer.conf`, проверяющим их наличие в списке работающих процессов поиском в `/proc` и запускающих их в отключенном от терминала режиме (`nohup`) в случае отсутствия в нем. Информация о перезапуске дописывайте в файл `observer.log`


```

conf_file="observer.conf"
log_file="observer.log"

if [[ ! -f $conf_file ]];
then
echo "Файл конфигурации $conf_file не найден."
exit
fi

while IFS= read -r script || [[ -n "$script" ]];
do
[[ -z "$script" || "$script" =~ ^# ]] && continue

script_name=$(basename "$script")

if ! pgrep -f "$script_name" > /dev/null;
then
nohup bash "$script" > /dev/null 2>&1 &
pid=$!

datetime=$(date '+%m-%d %H:%M')
echo "[${datetime}] Перезапущен скрипт: $script_name с PID $pid" >> "$log_file"
fi
done < "$conf_file"

```

3. Настройте запуск observer.sh посредством cron по расписанию – 1 раз в минуту

```

root@eltex-practice2-pg2-v4:~# crontab -e
no crontab for root - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano          <---- easiest
 2. /usr/bin/vim.basic
 3. /usr/bin/vim.tiny
 4. /bin/ed

* * * * * ~/observer.sh

```

4. Создайте несколько символических ссылок на файл template_task.sh с различными именами (рабочие задачи), добавьте в файл конфигурации observer.conf соответствующие записи об этих задачах, включая исходный файл template_task.sh

```

root@eltex-practice2-pg2-v4:~/task_for_template# ln -s ~/template_task.sh link_1
root@eltex-practice2-pg2-v4:~/task_for_template# ln -s ~/template_task.sh link_2
root@eltex-practice2-pg2-v4:~/task_for_template# ln -s ~/template_task.sh link_3
root@eltex-practice2-pg2-v4:~/task_for_template# ls
link_1 link_2 link_3

```

```

root@eltex-practice2-pg2-v4:~# nano observer.conf
root@eltex-practice2-pg2-v4:~# cat observer.conf
[task_template]
path=~/template_task.sh

[link_1]
path=~/task_for_template/link_1

[link_2]
path=~/task_for_template/link_2

[link_3]
path=~task_for_template/link_3

```

5. Соберите статистику работы в виде набора файлов report_*.log, observer.log, приложите их вместе с исходными текстами скриптов в качестве отчета в виде сжатого архива tar. Не забудьте остановить процесс, удалив задачу в cron!

```

root@eltex-practice2-pg2-v4:~# tar -czvf report_archive.tar.gz ~/observer.log ~/template_task.sh ~/observer.sh
tar: Removing leading '/' from member names
./root/observer.log
tar: Removing leading '/' from hard link targets
./root/template_task.sh
./root/observer.sh
root@eltex-practice2-pg2-v4:~# tar -tf report_archive.tar.gz
root/observer.log
root/template_task.sh
root/observer.sh

```

```

root@eltex-practice2-pg2-v4:~# crontab -e
crontab: installing new crontab
root@eltex-practice2-pg2-v4:~# crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

```