

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

```
root@eltex-practice2-pg1-v17:~# su user1
Warning: your password will expire in 0 days.
user1@eltex-practice2-pg1-v17:/root$ _
```

2. Подсчитайте количество процессов, имеющих несколько потоков выполнения

```
user1@eltex-practice2-pg1-v17:/root$ ps -eLo pid,nlwp | awk '$2 > 1 {print $1}' | wc -l
53
```

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;

```
Tasks: 122 total, 1 running, 117 sleeping, 4 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3916.0 total, 2611.8 free, 479.2 used, 1078.3 buff/cache
MiB Swap: 3185.0 total, 3185.0 free, 0.0 used. 3436.8 avail Mem
```

PID	PPID	UID	USER	RUSER	TTY	TIME+	%CPU	%MEM	S	COMMAND
14973	14972	1001	user1	user1	pts/3	0:00.00	0.0	0.1	S	bash
693	1	997	systemd+	systemd+	?	0:00.73	0.0	0.2	S	systemd-times+
692	1	992	systemd+	systemd+	?	0:00.35	0.0	0.3	S	systemd-resol+
551	1	998	systemd+	systemd+	?	0:02.73	0.0	0.2	S	systemd-netwo+
729	1	103	syslog	syslog	?	0:00.19	0.0	0.2	S	rsyslogd
1	0	0	root	root	?	0:04.24	0.0	0.3	S	systemd
2	0	0	root	root	?	0:00.02	0.0	0.0	S	kthreadd

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

```
user1@eltex-practice2-pg1-v17:/root$ 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;
- отправьте этому процессу сигналы 15 (SIGTERM), 2 (SIGINT), 3 (SIGQUIT), 9(SIGKILL)

PID	PPID	UID	USER	RUSER	TTY	TIME+	%CPU	%MEM	S	COMMAND
14973	14972	1001	user1	user1	pts/3	0:00.00	0.0	0.1	S	bash
15496	15495	1001	user1	user1	pts/6	0:00.00	0.0	0.1	S	bash
15503	15496	0	root	user1	pts/6	0:00.00	0.0	0.1	S	passwd

PID	PPID	UID	USER	RUSER	TTY	TIME+	%CPU	%MEM	S	COMMAND
14973	14972	1001	user1	user1	pts/3	0:00.00	0.0	0.1	S	bash
15496	15495	1001	user1	user1	pts/6	0:00.00	0.0	0.1	S	bash

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

```
user1@eltex-practice2-pg1-v17:/root$ vim ~/file_task3.txt
[1]+  Stopped                  vim ~/file_task3.txt
```

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

```
user1@eltex-practice2-pg1-v17:/root$ sleep 600
^Z
[2]+  Stopped                  sleep 600
user1@eltex-practice2-pg1-v17:/root$ jobs
[1]-  Stopped                  vim ~/file_task3.txt
[2]+  Stopped                  sleep 600
```

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

```
user1@eltex-practice2-pg1-v17:/root$ bg %2
[2]+  sleep 600 &
```

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

```
user1@eltex-practice2-pg1-v17:/root$ ps aux | grep sleep
user1      15537  0.0  0.0   5684  2048 pts/6    S    06:55   0:00 sleep 600
user1      15539  0.0  0.0   6544  2304 pts/6    S+   06:57   0:00 grep --col
ep
user1@eltex-practice2-pg1-v17:/root$ renice 10 -p 15537
15537 (process ID) old priority 0, new priority 10
user1@eltex-practice2-pg1-v17:/root$
```

10. Проверьте, что число NICE у этого задания изменилось

```
user1@eltex-practice2-pg1-v17:/root$ ps -eo pid,comm,nice | grep sleep
15537 sleep 10
```

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

```
user1@eltex-practice2-pg1-v17:/root$ fg %1
vim ~/file task3.txt
```

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

```
user1@eltex-practice2-pg1-v17:/root$ kill -15 15537
user1@eltex-practice2-pg1-v17:/root$ jobs
[2]+  Terminated              sleep 600
```

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

```
user1@eltex-practice2-pg1-v17:/root$ trap 'echo "Меня голыми руками не возьмёшь!"' INT QUIT
Меня голыми руками не возьмёшь!root$ ^C
Меня голыми руками не возьмёшь!root$ ^\
user1@eltex-practice2-pg1-v17:/root$ trap - INT QUIT
```

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

```
root@eltex-practice2-pg1-v17:~# touch template_task.sh
root@eltex-practice2-pg1-v17:~# nano template_task.sh
```

```
root@eltex-practice2-pg1-v17:~# bash template_task.sh
я бригадир, сам не работаю
```

```
#!/bin/bash

if [ "$(basename "$0")" = "template_task.sh" ]; then echo "я бригадир, сам не работаю"
    exit 1
fi

script_name=$(basename "$0" .sh) log_file="report_${script_name}.log" pid=$$
echo "[${pid}] $(date '+%Y-%m-%d %H:%M:%S') Скрипт запущен" >> "${log_file}"

sleep_time=$((RANDOM % 1771 + 30)) # От 30 до 1800 секунд sleep "${sleep_time}"

work_time=$((sleep_time / 60))

echo "[${pid}] $(date '+%Y-%m-%d %H:%M:%S') Скрипт завершился, работал ${work_time} мин"
```

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

```
#!/bin/bash
log_file="observer.log"

while read -r script; do if [ -n "$script" ] && [ -f "$script" ]; then
    # Проверка, работает ли скрипт
    if ! pgrep -f "$(basename "$script" .sh)" > /dev/null; then echo "$(date '+%Y-%m-%d %H:%M:%S') Запускаю $script" >> $log_file
        nohup "$script" > /dev/null 2>&1 &
    fi fi
done < observer.conf
```

3. Настройте запуск observer.sh посредством cron по расписанию – 1 раз в минуту
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-pg1-v17:~# ln -s template_task.sh task1.sh
root@eltex-practice2-pg1-v17:~# ln -s template_task.sh task2.sh
root@eltex-practice2-pg1-v17:~# ln -s template_task.sh task3.sh
```

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-pg1-v17:~# tar -czvf task_report.tar.gz report_*.log observer.log
template_task.sh observer.sh observer.conf
report_task1.log
report_task2.log
report_task3.log
report_template_task.log
observer.log
template_task.sh
observer.sh
observer.conf
root@eltex-practice2-pg1-v17:~# ls
eltex_module1_2025      observer.conf          task1.sh
eltex_module1_2025_clone observer.log            task2.sh
etc_backup              observer.sh            task3_part1.log
etc_backup.7z           practice2_eltex-pg1-v17_part1.log task3.sh
etc_backup.7zip         report_task1.log       task_report.tar.gz
etc_backup.gzip         report_task2.log       template_task.sh
etc_backup.tar.bz2     report_task3.log       tmp
etc_backup.tar.gz      report_template_task.log
etc_backup_with_motd.tar.gz sudo
root@eltex-practice2-pg1-v17:~# crontab -e
crontab: installing new crontab
```