

Пономарев Руслан, группы 2. Модуль 1, практика 3.

Раздел 1.

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

```
root@eltex-practice2-pg2-v8:~# su - user1
user1@eltex-practice2-pg2-v8:~$
```

2. Подсчитайте количество процессов, имеющих несколько потоков выполнения

```
user1@eltex-practice2-pg2-v8:~$ ps -eL | awk '{print $1}' | sort | uniq -c | wc -l
121
user1@eltex-practice2-pg2-v8:~$ ps -eL | awk '{print $1}' | sort | uniq -c | awk '{if($1 > 1) print $1}' | wc -l
10
user1@eltex-practice2-pg2-v8:~$
```

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;

F – для входа в режим изменения полей

Отключать поля можно клавишей d, перемещать через k и j

```
Fields Management for window 1:Def, whose current sort field is %CPU
  Navigate with Up/Dn, Right selects for move then <Enter> or Left commits,
  'd' or <Space> toggles display, 's' sets sort. Use 'q' or <Esc> to end!
* PID      = Process Id          USED     = Res+Swap Size (KiB)
* USER     = Effective User Name nsIPC   = IPC namespace Inode
* PR       = Priority             nsMNT   = MNT namespace Inode
* NI       = Nice Value           nsNET   = NET namespace Inode
* VIRT     = Virtual Image (KiB) nsPID   = PID namespace Inode
* RES      = Resident Size (KiB) nsUSER  = USER namespace Inode
* SHR      = Shared Memory (KiB) nsUTS   = UTS namespace Inode
* S        = Process Status       LXC     = LXC container name
* %CPU    = CPU Usage             RSan    = RES Anonymous (KiB)
* %MEM    = Memory Usage (RES)   RSfd   = RES File-based (KiB)
* TIME+   = CPU Time, hundredths RSlk   = RES Locked (KiB)
* COMMAND = Command Name/Line   RSsh   = RES Shared (KiB)
PPID      = Parent Process pid  CGNAME = Control Group name
UID       = Effective User Id   NU     = Last Used NUMA node
RUID     = Real User Id         LOGID  = Login User Id
RUSER    = Real User Name       EXE    = Executable Path
SUID     = Saved User Id        RSS    = Res Mem (smaps), KiB
SUSER    = Saved User Name      PSS    = Proportion RSS, KiB
GID      = Group Id             PSan   = Proportion Anon, KiB
GROUP    = Group Name           PSFd   = Proportion File, KiB
PGRP     = Process Group Id    PSsh   = Proportion Shrd, KiB
TTY       = Controlling Tty     USS    = Unique RSS, KiB
TPGID    = Tty Process Grp Id  ioR    = I/O Bytes Read
SID      = Session Id           ioRop  = I/O Read Operations
nTH      = Number of Threads   ioW    = I/O Bytes Written
P        = Last Used Cpu (SMP)  iowop  = I/O Write Operations
TIME     = CPU Time              AGID   = Autogroup Identifier
SWAP     = Swapped Size (KiB)   AGNI   = Autogroup Nice Value
CODE     = Code Size (KiB)       STARTED = Start Time from boot
DATA     = Data+Stack (KiB)      ELAPSED = ETapsed Running Time
nMaj     = Major Page Faults   %CUU   = CPU Utilization
nMin     = Minor Page Faults   %CUC   = Utilization + child
nDRT     = Dirty Pages Count   nsCGROUP = CGRP namespace Inode
WCHAN   = Sleeping in Function nTIME  = TIME namespace Inode
Flags    = Task Flags <sched.h>
CGROUPS = Control Groups
SUPGIDS = Supp Groups IDs
SUPGRPS = Supp Groups Names
TGID    = Thread Group Id
OOMa    = OOMEM Adjustment
OOMs    = OOMEM Score current
ENVIRON = Environment vars
vMj    = Major Faults delta
vMn    = Minor Faults delta
```

Fields Management for window 4:User, whose current sort field is USER
 Navigate with Up/Dn, Right selects for move then <Enter> or Left commits,
 'd' or <Space> toggles display, 's' sets sort. Use 'q' or <Esc> to end!

* PID	= Process Id	USED	= Res+Swap Size (KiB)
* PPID	= Parent Process pid	nsIPC	= IPC namespace Inode
* UID	= Effective User Id	nsMNT	= MNT namespace Inode
* USER	= Effective User Name	nsNET	= NET namespace Inode
* RUSER	= Real User Name	nsPID	= PID namespace Inode
* TTY	= Controlling Tty	nsUSER	= USER namespace Inode
* TIME+	= CPU Time, hundredths	nsUTS	= UTS namespace Inode
* %CPU	= CPU Usage	LXC	= LXC container name
* %MEM	= Memory Usage (RES)	RSan	= RES Anonymous (KiB)
* S	= Process Status	RSfd	= RES File-based (KiB)
* COMMAND	= Command Name/Line	RSTk	= RES Locked (KiB)
RUID	= Real User Id	RSch	= RES Shared (KiB)

PID	PPID	UID	USER	RUSER	TTY	
91090	91089	1001	user1	user1	pts/3	0
91144	91090	1001	user1	user1	pts/3	0

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

```
root@eltex-practice2-pg2-v8:~# su - user1
user1@eltex-practice2-pg2-v8:~$ passwd
Changing password for user1.
Current password: [REDACTED]
```

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;

Клавишей **u** сортируем по имени пользователя.

PID	PPID	UID	USER	RUSER	TTY	TIME+	%CPU	%MEM	S	COMMAND
91090	91089	1001	user1	user1	pts/3	0:00.02	0.0	0.2	S	bash
91229	91228	1001	user1	user1	pts/4	0:00.01	0.0	0.2	S	bash
91246	91090	1001	user1	user1	pts/3	0:00.02	0.0	0.2	R	top

- найдите процесс, запущенный командой passwd;

PID	PPID	UID	USER	RUSER	TTY	TIME+	%CPU	%MEM	S	COMMAND
91229	91228	1001	user1	user1	pts/4	0:00.01	0.0	0.2	S	bash
91246	91090	1001	user1	user1	pts/3	0:00.02	0.0	0.2	R	top

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

PID	PPID	UID	USER	RUSER	TTY	TIME+	%CPU	%MEM	S	COMMAND
91229	91228	1001	user1	user1	pts/4	0:00.01	0.0	0.2	S	bash
91246	91090	1001	user1	user1	pts/3	0:00.02	0.0	0.2	R	top

```

top - 08:20:26 up 17 days, 27 min, 3 users, load average: 0.00, 0.00, 0.00
Tasks: 122 total, 1 running, 121 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 2.5 sy, 0.0 ni, 97.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 2510.0 total, 518.4 free, 497.2 used, 1786.4 buff/cache
MiB Swap: 3185.0 total, 3184.5 free, 0.5 used. 2012.8 avail Mem
Send pid 91229 signal [15/sigterm] 2

```

PID	PPID	UID	USER	RUSER	TTY	TIME+	%CPU	%MEM	S	COMMAND
91229	91228	1001	user1	user1	pts/4	0:00.01	0.0	0.2	S	bash
91090	91089	1001	user1	user1	pts/3	0:00.02	0.0	0.2	S	bash
91246	91090	1001	user1	user1	pts/3	0:01.96	0.0	0.2	R	top

```

top - 08:20:26 up 17 days, 27 min, 3 users, load average: 0.00, 0.00, 0.00
Tasks: 122 total, 1 running, 121 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 2.5 sy, 0.0 ni, 97.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 2510.0 total, 518.4 free, 497.2 used, 1786.4 buff/cache
MiB Swap: 3185.0 total, 3184.5 free, 0.5 used. 2012.8 avail Mem
Send pid 91229 signal [15/sigterm] 3

```

PID	PPID	UID	USER	RUSER	TTY	TIME+	%CPU	%MEM	S	COMMAND
91229	91228	1001	user1	user1	pts/4	0:00.01	0.0	0.2	S	bash
91090	91089	1001	user1	user1	pts/3	0:00.02	0.0	0.2	S	bash
91246	91090	1001	user1	user1	pts/3	0:01.96	0.0	0.2	R	top

```

top - 08:21:13 up 17 days, 28 min, 3 users, load average: 0.00, 0.00, 0.00
Tasks: 123 total, 1 running, 122 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 99.2 id, 0.8 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 2510.0 total, 517.7 free, 497.9 used, 1786.4 buff/cache
MiB Swap: 3185.0 total, 3184.5 free, 0.5 used. 2012.1 avail Mem

```

PID	PPID	UID	USER	RUSER	TTY	TIME+	%CPU	%MEM	S	COMMAND
91229	91228	1001	user1	user1	pts/4	0:00.01	0.0	0.2	S	bash
91090	91089	1001	user1	user1	pts/3	0:00.02	0.0	0.2	S	bash
91246	91090	1001	user1	user1	pts/3	0:01.97	0.0	0.2	R	top

```

top - 08:21:25 up 17 days, 28 min, 3 users, load average: 0.00, 0.00, 0.00
Tasks: 123 total, 1 running, 122 sleeping, 0 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 : 2528.0 total, 517.7 free, 515.9 used, 1786.4 buff/cache
MiB Swap: 3185.0 total, 3184.5 free, 0.5 used. 2012.1 avail Mem
Send pid 91229 signal [15/sigterm] 9

```

PID	PPID	UID	USER	RUSER	TTY	TIME+	%CPU	%MEM	S	COMMAND
91229	91228	1001	user1	user1	pts/4	0:00.01	0.0	0.2	S	bash
91090	91089	1001	user1	user1	pts/3	0:00.02	0.0	0.2	S	bash
91246	91090	1001	user1	user1	pts/3	0:01.98	0.3	0.2	R	top

```

top - 08:21:48 up 17 days, 28 min, 3 users, load average: 0.00, 0.00, 0.00
Tasks: 121 total, 1 running, 120 sleeping, 0 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 : 2528.0 total, 517.7 free, 515.9 used, 1786.4 buff/cache
MiB Swap: 3185.0 total, 3184.5 free, 0.5 used. 2012.1 avail Mem

```

PID	PPID	UID	USER	RUSER	TTY	TIME+	%CPU	%MEM	S	COMMAND
91090	91089	1001	user1	user1	pts/3	0:00.02	0.0	0.2	S	bash
91246	91090	1001	user1	user1	pts/3	0:01.98	0.0	0.2	R	top

```

root@eltex-practice2-pg2-v8:~# su - user1
user1@eltex-practice2-pg2-v8:~$ passwd
Changing password for user1.
Current password: Killed
root@eltex-practice2-pg2-v8:~#

```

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

```
user1@eltex-practice2-pg2-v8:~$ vim ~/file_task3.txt
[1]+ Stopped vim ~/file_task3.txt
user1@eltex-practice2-pg2-v8:~$ █
```

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

```
[1]+ Stopped vim ~/file_task3.txt
user1@eltex-practice2-pg2-v8:~$ sleep 600
^Z
[2]+ Stopped sleep 600
user1@eltex-practice2-pg2-v8:~$ jobs
[1]- Stopped vim ~/file_task3.txt
[2]+ Stopped sleep 600
user1@eltex-practice2-pg2-v8:~$ █
```

8. Последнее задание (sleep 600) сделайте фоновым

```
user1@eltex-practice2-pg2-v8:~$ bg %2
[2]+ sleep 600 &
user1@eltex-practice2-pg2-v8:~$ █
```

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

```
user1@eltex-practice2-pg2-v8:~$ ps aux | grep sleep
user1    91354  0.0  0.0  5684  2048 pts/3    S     08:26   0:00 sleep 600
user1    91357  0.0  0.0  6544  2304 pts/3    S+    08:28   0:00 grep --color=au
to sleep
user1@eltex-practice2-pg2-v8:~$ renice 10 -p 91354
91354 (process ID) old priority 0, new priority 10
```

10. Проверьте, что число NICE у этого задания изменилось

```
user1@eltex-practice2-pg2-v8:~$ ps -l
F S  UID      PID  PPID  C PRI  NI ADDR SZ WCHAN TTY          TIME CMD
4 S  1001    91090  91089  0  80    0 - 2161 do_wai pts/3    00:00:00 bash
0 T  1001    91352  91090  0  80    0 - 6192 do_sig pts/3    00:00:00 vim
0 S  1001    91354  91090  0  90   10 - 1421 do_sys pts/3    00:00:00 sleep
0 R  1001    91361  91090  0  80    0 - 2729 -           pts/3    00:00:00 ps
user1@eltex-practice2-pg2-v8:~$ █
```

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

```
user1@eltex-practice2-pg2-v8:~$ fg %1
vim ~/file_task3.txt
user1@eltex-practice2-pg2-v8:~$ █
```

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

```
user1@eltex-practice2-pg2-v8:~$ kill -15 %2
user1@eltex-practice2-pg2-v8:~$ jobs
[2]+ Terminated sleep 600
user1@eltex-practice2-pg2-v8:~$ █
```

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

```
bash: trap: invalid signal specification
user1@eltex-practice2-pg2-v8:~$ trap 'echo "Меня голыми руками не возьмёшь!"' SIGINT SIGQUIT
user1@eltex-practice2-pg2-v8:~$ kill -SIGINT @@
Меня голыми руками не возьмёшь!
user1@eltex-practice2-pg2-v8:~$ kill -SIGQUIT @@
Меня голыми руками не возьмёшь!
user1@eltex-practice2-pg2-v8:~$
```

Раздел 2

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

При запуске проверят, 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

```
#!/bin/bash

SCRIPT_NAME=$(basename "$0")
REPORT_FILE="report_${SCRIPT_NAME}.log"

if [ "$SCRIPT_NAME"=="template_task.sh" ]; then
    echo "Я бригадир, сам не работаю"
    exit 1
fi

PID=$$
echo "[${PID}] $(date '+%Y-%m-%d %H:%M:%S') Скрипт запущен" >> "$REPORT_FILE"
RANDOM_SEC=$((RANDOM % 1771 + 30))
sleep $RANDOM_SEC

MIN=$((($RANDOM_SEC + 59)/60))
echo "[${PID}] $(date '+%Y-%m-%d %H:%M:%S') Скрипт завершился, работал $MIN минут"
>> "$REPORT_FILE"
```

```
root@eltex-practice2-pg2-v8:~# ./template_task.sh
Я бригадир, сам не работаю
root@eltex-practice2-pg2-v8:~#
```

```
root@eltex-practice2-pg2-v8:~# cat report_template_task.sh.log
[PID] 2025-11-01 08:54:30 Скрипт запущен
[91545] 2025-11-01 08:54:30 Скрипт завершился, работал 1 минут
[PID] 2025-11-01 08:54:54 Скрипт запущен
[91550] 2025-11-01 08:54:54 Скрипт завершился, работал 20 минут
[PID] 2025-11-01 08:55:44 Скрипт запущен
[91563] 2025-11-01 08:55:44 Скрипт завершился, работал 7 минут
[PID] 2025-11-01 09:00:40 Скрипт запущен
root@eltex-practice2-pg2-v8:~#
```

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

```
root@eltex-practice2-pg2-v8: ~
#!/bin/bash
CONFIG_FILE="observer.conf"
LOG_FILE="observer.log"

log_msg() {
    echo "[$(date '+%Y-%m-%d %H:%M:%S') $1]" >> "$LOG_FILE"
}

if [ ! -f "$CONFIG_FILE" ]; then
    log_msg "Error. Файл конфигурации $CONFIG_FILE не найден"
    exit 1
fi

while IFS= read -r script || [ -n "$script" ]; do
    if [[ -z "$script" || "$script" =~ ^[:space:]*# ]]; then
        continue
    fi
    script=$(echo "$script" | xargs)
    script_basename=${basename "$script"}
    running=false
    for proc_dir in /proc/[0-9]*; do
        if [ -d "$proc_dir" ]; then
            if [ -r "$proc_dir/cmdline" ]; then
                cmdline=$(tr '\0' '\n' < "$proc_dir/cmdline")
                if [[ "$cmdline" == *"$script_basename"* ]]; then
                    running=true
                    break
                fi
            fi
        fi
    done
    if ! $running; then
        if [ ! -f "$script" ]; then
            log_msg "Скрипт $script не найден, пропуск"
            continue
        fi
        if [ ! -x "$script" ]; then
            chmod +x "$script"
        fi
        #скрипт в отключенном от терминала режиме
        nohup "./$script" > /dev/null 2>&1 &
        log_msg "Скрипт $script_basename [PID: $!] запущен"
    else
        log_msg "Скрипт $script_basename уже был запущен"
    fi
done < "$CONFIG_FILE"
log_msg "Проверка завершена"
```

```
[2025-11-02 09:17:54 Скрипт observer.sh уже был запущен]
[2025-11-02 09:17:54 Скрипт template_task.sh [PID: 94738] запущен]
[2025-11-02 09:17:54 Проверка завершена]
root@eltex-practice2-pg2-v8:~# cat observer.log
```

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

```
GNU nano 7.2 /tmp/crontab.OmKmNv/crontab *
# 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
* * * * * /root/observer.sh
```

4. Создайте несколько символьных 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-v8:~# ln -sf template_task.sh worker1.sh
root@eltex-practice2-pg2-v8:~# ln -sf template_task.sh worker2.sh
root@eltex-practice2-pg2-v8:~# ln -sf template_task.sh worker3.sh
root@eltex-practice2-pg2-v8:~# ln -sf template_task.sh worker4.sh
root@eltex-practice2-pg2-v8:~# ls -la *.sh
-rwxr-xr-x 1 root root 1269 Nov  2 09:17 observer.sh
-rwxr-xr-x 1 root root  506 Nov  1 09:11 template_task.sh
lrwxrwxrwx 1 root root   16 Nov  2 09:29 worker1.sh -> template_task.sh
lrwxrwxrwx 1 root root   16 Nov  2 09:29 worker2.sh -> template_task.sh
lrwxrwxrwx 1 root root   16 Nov  2 09:29 worker3.sh -> template_task.sh
lrwxrwxrwx 1 root root   16 Nov  2 09:29 worker4.sh -> template_task.sh
root@eltex-practice2-pg2-v8:~#
```

```
root@eltex-practice2-pg2-v8:~# cat observer.conf
template_task.sh
worker1.sh
worker2.sh
worker3.sh
worker4.sh
root@eltex-practice2-pg2-v8:~#
```

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-v8:~# pkill -f "worker" 2>/dev/null || true
root@eltex-practice2-pg2-v8:~# pkill -f "observer" 2>/dev/null || true
root@eltex-practice2-pg2-v8:~# crontab -r
root@eltex-practice2-pg2-v8:~#
```

```
root@eltex-practice2-pg2-v8:~# tar -czf monitoring_report.tar.gz monitoring_repo
rt/
root@eltex-practice2-pg2-v8:~# tar -tzf monitoring_report.tar.gz monitoring_repo
rt/
monitoring_report/
monitoring_report/observer.sh
monitoring_report/template_task.sh
monitoring_report/2
monitoring_report/observer.log
monitoring_report/observer.conf
monitoring_report/worker1.sh
monitoring_report/worker4.sh
monitoring_report/worker3.sh
monitoring_report/worker2.sh
root@eltex-practice2-pg2-v8:~#
```

```
root@eltex-practice2-pg2-v8:~/eltex_practice/module1/task3# git add .
root@eltex-practice2-pg2-v8:~/eltex_practice/module1/task3# git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   monitoring_report.tar.gz

root@eltex-practice2-pg2-v8:~/eltex_practice/module1/task3# git commit -m "added
  third practice"
[main 033e913] added third practice
  1 file changed, 0 insertions(+), 0 deletions(-)
   create mode 100644 module1/task3/monitoring_report.tar.gz
root@eltex-practice2-pg2-v8:~/eltex_practice/module1/task3# git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 2.99 KiB | 2.99 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/russichjust/eltex_practice
  b1699d4..033e913  main -> main
root@eltex-practice2-pg2-v8:~/eltex_practice/module1/task3#
```