

## Раздел 1.

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

```
SmarTTY - 217.71.138.1:44556

File Edit View SCP Tools Help

root@eltex-practice2-pg1-v9:~# su - user1
user1@eltex-practice2-pg1-v9:~$
```

### 2. Подсчитайте количество процессов, имеющих несколько потоков выполнения

```
SmarTTY - 217.71.138.1:44556

File Edit View SCP Tools Help

user1@eltex-practice2-pg1-v9:~$ ps -eLf | awk '{print $2}' | uniq -c | awk '$1 > 1 {print}'
 7 345
 2 684
 4 701
 2 702
 6 712
 4 723
 2 729
 4 766
 6 829
 4 865

user1@eltex-practice2-pg1-v9:~$ ps -eLf | awk '{print $2}' | uniq -c | awk '$1 > 1 {print}' | wc -l
10

user1@eltex-practice2-pg1-v9:~$
```

### 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;

```
SmarTTY - 217.71.138.1:44556

File Edit View SCP Tools Help

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

```
SmarTTY - 217.71.138.1:44556
File Edit View SCP Tools Help
top - 08:00:08 up 11 days, 20:27, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 111 total, 1 running, 110 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 99.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3916.0 total, 2297.6 free, 464.2 used, 1402.1 buff/cache
MiB Swap: 3185.0 total, 3185.0 free, 0.0 used. 3451.8 avail Mem

  PID USER   RUSER   PR  NI  S   %CPU %MEM    TIME+  COMMAND
  345 root    root    rt   0   S    0.0  0.7   1:20.73 multipathd
  702 root    root    20   0   S    0.0  0.1  10:41.08 qemu-ga
    1 root    root    20   0   S    0.0  0.3   0:11.81 systemd
    2 root    root    20   0   S    0.0  0.0   0:00.08 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-slub_
    7 root    root    0 -20  I    0.0  0.0   0:00.00 kworker/R-netns
   10 root    root    0 -20  I    0.0  0.0   0:00.00 kworker/0:0H-events_highpri
   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   0:00.10 ksoftirqd/0
   17 root    root    20   0   I    0.0  0.0   0:05.35 rcu_preempt
   18 root    root    rt   0   S    0.0  0.0   0:04.15 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   0:03.70 migration/1
   24 root    root    20   0   S    0.0  0.0   0:00.12 ksoftirqd/1
   26 root    root    0 -20  I    0.0  0.0   0:00.00 kworker/1:0H-events_highpri
   27 root    root    20   0   S    0.0  0.0   0:00.00 kdevtmpfs
   28 root    root    0 -20  I    0.0  0.0   0:00.00 kworker/R-inet_
   30 root    root    20   0   S    0.0  0.0   0:00.00 kauditd
```

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

```
SmarTTY - 217.71.138.1:44556
File Edit View SCP Tools Help
root@eltex-practice2-pg1-v9:~# su - user1
user1@eltex-practice2-pg1-v9:~$ 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)

SmarTTY - 217.71.138.1:44556

File Edit View SCP Tools Help

```
top - 08:05:03 up 11 days, 20:32, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 115 total, 1 running, 114 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.2 sy, 0.0 ni, 99.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3916.0 total, 2289.8 free, 471.9 used, 1402.2 buff/cache
MiB Swap: 3185.0 total, 3185.0 free, 0.0 used. 3444.0 avail Mem
```

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
33811	user1	user1	20	0	S	0.0	0.1	0:00.01	bash
33920	user1	user1	20	0	S	0.0	0.1	0:00.01	bash
33930	root	user1	20	0	S	0.0	0.1	0:00.00	passwd

SmarTTY - 217.71.138.1:44556

File Edit View SCP Tools Help

```
top - 08:06:09 up 11 days, 20:33, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 112 total, 1 running, 111 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 : 3916.0 total, 2289.8 free, 471.9 used, 1402.2 buff/cache
MiB Swap: 3185.0 total, 3185.0 free, 0.0 used. 3444.0 avail Mem
PID to signal/kill [default pid = 33838] 33930
```

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
33839	user1	user1	20	0	R	0.0	0.2	0:00.43	top
33920	user1	user1	20	0	S	0.0	0.1	0:00.01	bash
33930	root	user1	20	0	S	0.0	0.1	0:00.00	passwd

SmarTTY - 217.71.138.1:44556

File Edit View SCP Tools Help

```
top - 08:06:09 up 11 days, 20:33, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 112 total, 1 running, 111 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 : 3916.0 total, 2289.8 free, 471.9 used, 1402.2 buff/cache
MiB Swap: 3185.0 total, 3185.0 free, 0.0 used. 3444.0 avail Mem
Send pid 33930 signal [15/sigterm] 15
```

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
33839	user1	user1	20	0	R	0.0	0.2	0:00.43	top
33920	user1	user1	20	0	S	0.0	0.1	0:00.01	bash
33930	root	user1	20	0	S	0.0	0.1	0:00.00	passwd

SmarTTY - 217.71.138.1:44556

File Edit View SCP Tools Help

```
top - 08:06:55 up 11 days, 20:34, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 112 total, 1 running, 111 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 : 3916.0 total, 2289.8 free, 471.9 used, 1402.2 buff/cache
MiB Swap: 3185.0 total, 3185.0 free, 0.0 used. 3444.0 avail Mem
Send pid 33930 signal [15/sigterm] 2
```

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
33839	user1	user1	20	0	R	0.0	0.2	0:00.44	top
33920	user1	user1	20	0	S	0.0	0.1	0:00.01	bash
33930	root	user1	20	0	S	0.0	0.1	0:00.00	passwd

SmarTTY - 217.71.138.1:44556

File Edit View SCP Tools Help

```
top - 08:07:45 up 11 days, 20:35, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 112 total, 1 running, 111 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.2 us, 0.0 sy, 0.0 ni, 99.7 id, 0.2 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3916.0 total, 2289.8 free, 471.9 used, 1402.2 buff/cache
MiB Swap: 3185.0 total, 3185.0 free, 0.0 used. 3444.0 avail Mem
Send pid 33930 signal [15/sigterm] 3
```

PID	USER	RUSER	PR	NI	S	%CPU	%MEM	TIME+	COMMAND
33839	user1	user1	20	0	R	0.0	0.2	0:00.45	top
33920	user1	user1	20	0	S	0.0	0.1	0:00.01	bash
33930	root	user1	20	0	S	0.0	0.1	0:00.00	passwd

```
SmarTTY - 217.71.138.1:44556
File Edit View SCP Tools Help
top - 08:08:01 up 11 days, 20:35, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 112 total, 1 running, 111 sleeping, 0 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 : 3916.0 total, 2289.8 free, 471.9 used, 1402.2 buff/cache
MiB Swap: 3185.0 total, 3185.0 free, 0.0 used. 3444.0 avail Mem
Send pid 33930 signal [15/sigterm] 9
  PID USER      RUSER      PR  NI  S   %CPU  %MEM    TIME+  COMMAND
  33920 user1      user1      20   0  S    0.0   0.1   0:00.01  bash
  33930 root       user1      20   0  S    0.0   0.1   0:00.00  passwd

SmarTTY - 217.71.138.1:44556
File Edit View SCP Tools Help
top - 08:08:18 up 11 days, 20:35, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 111 total, 1 running, 110 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 : 3916.0 total, 2289.8 free, 471.9 used, 1402.2 buff/cache
MiB Swap: 3185.0 total, 3185.0 free, 0.0 used. 3444.0 avail Mem
  PID USER      RUSER      PR  NI  S   %CPU  %MEM    TIME+  COMMAND
  33920 user1      user1      20   0  S    0.0   0.1   0:00.01  bash
```

Только 9(SIGKILL) остановил процесс.

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

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

[1]+  Stopped                  vim ~/file_task3.txt
user1@eltex-practice2-pg1-v9:~$
```

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

```
user1@eltex-practice2-pg1-v9:~$ sleep 600
^Z
[2]+  Stopped                  sleep 600

user1@eltex-practice2-pg1-v9:~$ jobs
[1]-  Stopped                  vim ~/file_task3.txt
[2]+  Stopped                  sleep 600
```

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

```
user1@eltex-practice2-pg1-v9:~$ bg %2
[2]+ sleep 600 &
user1@eltex-practice2-pg1-v9:~$ jobs
[1]+  Stopped                  vim ~/file_task3.txt
[2]-  Running                  sleep 600 &
```

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

```

user1@eltex-practice2-pg1-v9:~$ ps
  PID TTY          TIME CMD
 33811 pts/0        00:00:00 bash
 33937 pts/0        00:00:00 vim
 33943 pts/0        00:00:00 sleep
 33946 pts/0        00:00:00 ps
user1@eltex-practice2-pg1-v9:~$ renice -n 10 33943
33943 (process ID) old priority 0, new priority 10

```

10. Проверьте, что число NICE у этого задания изменилось

```

user1@eltex-practice2-pg1-v9:~$ ps -l
F S  UID      PID     PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001    33811    33810  0  80   0  -  2166 do_wai pts/0        00:00:00 bash
0 T  1001    33937    33811  0  80   0  -  6185 do_sig pts/0        00:00:00 vim
0 S  1001    33943    33811  0  90  10  -  1421 do_sys pts/0        00:00:00 sleep
0 R  1001    33953    33811  0  80   0  -  2729 -      pts/0        00:00:00 ps

```

11. Сделайте задание vim ~/file task3.txt активным и выйдите из редактора

```

user1@eltex-practice2-pg1-v9:~$ jobs
[1]+  Stopped                  vim ~/file_task3.txt
[2]-  Running                  sleep 600 &
user1@eltex-practice2-pg1-v9:~$ fg %1

```

```

~
~
~
~
~
~
user1@eltex-practice2-pg1-v9:~$ _

```

```

user1@eltex-practice2-pg1-v9:~$ ps
  PID TTY          TIME CMD
 33811 pts/0        00:00:00 bash
 33943 pts/0        00:00:00 sleep
 33957 pts/0        00:00:00 ps
user1@eltex-practice2-pg1-v9:~$ jobs
[2]+  Running                  sleep 600 &

```

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

```

user1@eltex-practice2-pg1-v9:~$ kill -15 33943
user1@eltex-practice2-pg1-v9:~$ 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-v9:~$ trap 'echo "you wont catch me alive!"' SIGINT SIGQUIT
you wont catch me alive!1-v9:~$ ^C
you wont catch me alive!1-v9:~$ ^\
```

## Раздел 2.

1. Создайте скрипт на языке bash с именем template\_task.sh, делающий следующее:

- При запуске проверяют, 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

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

log="${HOME}/report_${name%.*}.log"
echo "$$ $(date '+%F %T') Скрипт запущен" >> $log

sleep_time=$((30 + RANDOM % (1800 - 30)))
mins=$((sleep_time / 60))
sleep $sleep_time

echo "$$ $(date '+%F %T') Скрипт завершился, работал $mins минут" >> $log
```

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



```

File Edit View SCP Tools Help
GNU nano 7.2 observer.sh *
#!/bin/bash

list="observer.conf"
log="observer.log"

if [ ! -f $list ]; then
    echo "List not found"
    exit 1
fi

function get_name() {
    while read -r data; do
        ps -p $data | sed -n 2p | awk '{print $4}'
    done
}

function get_proc_names() {
    for pid in /proc/[0-9]*; do
        if [ -f "$pid/cmdline" ]; then
            echo "$pid" | awk -F '/' '{print $3}' | get_name 2> /dev/null
        fi
    done
}

names=$(get_proc_names)

while IFS= read -r line; do
    out=$(echo "$names" | grep -x $line)
    if [[ ! $out ]]; then
        echo "$(date +%F %T) rebooted $line" >> $log
        nohup bash "$line" &
    fi
done < $list

```

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

```

File Edit View SCP Tools Help
GNU nano 7.2 /tmp/crontab.YVgMA3/crontab
# 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
* * * * * bash /home/user1/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`

```
user1@eltex-practice2-pg1-v9:~$ ln -s template_task.sh job1.sh
user1@eltex-practice2-pg1-v9:~$ ln -s template_task.sh job2.sh
user1@eltex-practice2-pg1-v9:~$ ln -s template_task.sh job3.sh
user1@eltex-practice2-pg1-v9:~$ _
```

```
user1@eltex-practice2-pg1-v9:~$ cat observer.conf
job1.sh
job2.sh
job3.sh
user1@eltex-practice2-pg1-v9:~$
```

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`!

```
user1@eltex-practice2-pg1-v9:~$ 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
user1@eltex-practice2-pg1-v9:~$ _
```



```
user1@eltex-practice2-pg1-v9:~$ mkdir observer
user1@eltex-practice2-pg1-v9:~$ cp report_job1.log ./observer/report_job1.log
user1@eltex-practice2-pg1-v9:~$ cp report_job2.log ./observer/report_job2.log
user1@eltex-practice2-pg1-v9:~$ cp report_job3.log ./observer/report_job3.log
user1@eltex-practice2-pg1-v9:~$ cp observer.log ./observer/observer.log
user1@eltex-practice2-pg1-v9:~$ cp observer.conf ./observer/observer.conf
user1@eltex-practice2-pg1-v9:~$ cp observer.sh ./observer/observer.sh
user1@eltex-practice2-pg1-v9:~$ cp template_task.sh ./observer/template_task.h
```

```
user1@eltex-practice2-pg1-v9:~$ cd ./observer/
user1@eltex-practice2-pg1-v9:~/observer$ ls
observer.conf  observer.log  observer.sh  report_job1.log  report_job2.log  report_job3.log  template_task.h
user1@eltex-practice2-pg1-v9:~/observer$ tar -czf observer.tar.gz *
user1@eltex-practice2-pg1-v9:~/observer$ ls
observer.conf  observer.sh  report_job1.log  report_job3.log
observer.log  observer.tar.gz  report_job2.log  template_task.h
user1@eltex-practice2-pg1-v9:~/observer$ mv observer.tar.gz ../observer.tar.gz
user1@eltex-practice2-pg1-v9:~/observer$ cd ..
user1@eltex-practice2-pg1-v9:~$ ls
job1.sh  nohup.out  observer.log  report_job1.log  report_template_task.log
job2.sh  observer  observer.sh  report_job2.log  template_task.sh
job3.sh  observer.conf  observer.tar.gz  report_job3.log  test.sh
```