

Задание №7

Введение

Это и последующее задание будет выполняться в операционной системе Astra Linux. Если пункт выделен желтым, то это значит, что пункт необходимо зафиксировать скриншотом, либо записать необходимую информацию в отчет. Количество скриншотов определяйте из содержимого, если несколько команд можно разместить на один скриншот, то можете сделать это.

1. Изучение понятия процесса

1. Отобразите список процессов системы без и с использованием опций с помощью команды ps:

```
ps
```

```
ps -aux
```

```
ps -eux
```

2. Получите выборочную информацию обо всех процессах в системе. Включите в формат: состояние, ID процесса, номер терминала и команду, которая запустила процесс.

```
ps -eo s,pid,tty,command
```

3. Отобразите состояния процессов в реальном времени.

```
top
```

Сделайте скриншот результата работы и разместите в области ответа

student@prac-work-question:~# Предоставьте ответ в виде скриншота(-ов).

Если необходимо предоставить скрипт, то ответ может содержать ссылку на скрипт решения (только git):

Скриншот для шага №3

```
top - 22:37:46 up 6:58, 5 users, load average: 0.31, 0.24, 0.16
tasks: 150 total, 1 running, 157 sleeping, 0 stopped, 0 zombie
Cpus(s): 0.2 us, 0.0 sy, 0.0 ni, 90.7 id, 0.2 wa, 0.0 hi, 0.2 si, 0.0 st
MiB Mem : 3914.3 total, 147.7 free, 653.1 used, 3113.5 buff/cache
MiB Swap: 975.0 total, 972.7 free, 2.3 used, 2946.4 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR     S    %CPU  %MEM     time+ COMMAND
 744 root        0   0 378924 159632 42416 S    0.7   4.0   1:12.00 korg
29823 kalinin  20   0 447892 68648 56584 S    0.7   1.7   0:15.58 x-terminal-emul
 895 kalinin  20   0 11496  4388   3788 R    0.3   0.1   0:00.01 top
   1 root        0   0 103260 11736  8848 S    0.0   0.3   0:05.07 systemd
   2 root        0   0 0 0 0 S    0.0   0.0   0:00.00 kthreadd
   3 root        0 -20 0 0 0 S    0.1   0.0   0:00.00 rcu_gp
   4 root        0 -20 0 0 0 S    0.1   0.0   0:00.00 rcu_par_gp
   5 root        0 -20 0 0 0 S    0.1   0.0   0:00.00 slab_flushwq
   6 root        0 -20 0 0 0 S    0.1   0.0   0:00.00 netns
   8 root        0 -20 0 0 0 S    0.1   0.0   0:00.00 kworker/0:0H-kblockd
  10 root        0 -20 0 0 0 S    0.1   0.0   0:00.00 mm_percpu_wq
  11 root        0  0 0 0 0 S    0.5   0.0   0:00.00 rcu_tasks_rude
  12 root        0  0 0 0 0 S    0.5   0.0   0:00.00 rcu_tasks_trace
  13 root        0  0 0 0 0 S    0.5   0.0   0:01.01 ksftirqd/0
  14 root        0  0 0 0 0 S    0.1   0.0   0:02.58 rcu_sched
  15 root        0  0 0 0 0 S    0.5   0.0   0:00.12 migration/0
  16 root       -51  0 0 0 0 S    0.5   0.0   0:00.00 idle_inject/0
  18 root        0  0 0 0 0 S    0.5   0.0   0:00.00 cphp/0
  19 root        0  0 0 0 0 S    0.5   0.0   0:00.00 cphp/1
  20 root       -51  0 0 0 0 S    0.5   0.0   0:00.00 idle_inject/1
  21 root        0  0 0 0 0 S    0.5   0.0   0:00.57 migration/1
  22 root        0  0 0 0 0 S    0.5   0.0   0:03.02 ksftirqd/1
  24 root        0 -20 0 0 0 S    0.1   0.0   0:00.00 kworker/1:0H-events_highpri
  25 root        0 -20 0 0 0 S    0.1   0.0   0:00.00 kdevpts
  26 root        0 -20 0 0 0 S    0.1   0.0   0:00.00 inet_frag_wq
  27 root        0  0 0 0 0 S    0.5   0.0   0:00.11 kauditd
  28 root        0  0 0 0 0 S    0.5   0.0   0:00.01 khungtaskd
  29 root        0  0 0 0 0 S    0.5   0.0   0:00.00 oom_reaper
  30 root        0 -20 0 0 0 S    0.1   0.0   0:00.00 writeback
  31 root        0  0 0 0 0 S    0.5   0.0   0:01.51 kcompactd8
  32 root        0  0 0 0 0 S    0.5   0.0   0:00.00 ksd
  33 root        0  0 0 0 0 S    0.5   0.0   0:00.00 khugepaged
  80 root        0 -20 0 0 0 S    0.1   0.0   0:00.00 kintegrityd
  81 root        0 -20 0 0 0 S    0.1   0.0   0:00.00 nblockd
  82 root        0 -20 0 0 0 S    0.1   0.0   0:00.00 blkcg_punt_bio
  83 root        0 -20 0 0 0 S    0.1   0.0   0:00.00 tpa_dev_wq
  84 root        0 -20 0 0 0 S    0.1   0.0   0:00.00 ata_sff
```

2. Управление процессами

1. Выведите список процессов пользователя root, используя команду ps. С помощью утилиты для подсчета количества строк посчитайте, сколько процессов принадлежит этому пользователю. Перенаправьте список процессов root в файл proc1.

```
ps -u root
```

```
ps -u root | wc -l
```

```
ps -u root > proc1
```

```
less proc1
```

2. Запустите браузер firefox в фоновом режиме в первом терминале. Во втором терминале запустите утилиту top в фоновом режиме.

```
firefox &
```

```
top &
```

3. В первом терминале найдите первый запущенный вами процесс с помощью команды `ps` и перенаправьте вывод этой команды на ввод команды `grep`. Определите ID процесса и номер терминала.

```
ps | grep firefox
```

4. Во втором терминале по идентификатору процесса найдите запущенный вами процесс, связанный с утилитой `top`.

```
ps | grep {ID_процесса}
```

5. Вернитесь в первый терминал и найдите поддерево для процесса `firefox`, изучите список составляющих его процессов.

```
pstree
```

6. Измените приоритет для процесса, порожденного программой `firefox`. Сделайте его равным 10. Подтвердите, что приоритет изменился.

```
ps | grep firefox
```

```
renice -n 10 {номер процесса firefox}
```

```
ps -o s,pid,tty,command,nice | grep firefox | grep 10
```

Сделайте скриншот результата работы и разместите в области ответа

7. Завершите все процессы вашего пользователя с помощью команды `killall`.

```
killall -u {имя_текущего_пользователя}
```

