

# Приоритет процессов

# Цели урока

- ✓ Разобраться, как узнать приоритет процесса
- ✓ Узнать, можем ли мы его менять

# Приоритет процесса

В системе Linux все активные процессы имеют определённый приоритет выполнения, задаваемый **nice-значением**.

# Приоритет процесса

Узнать значение приоритета команды можно по выводу команды **top**.

Только пользователь **root** может использовать отрицательные значения приоритета.

```
root@jenkins:/# ps -eo user,pid,pcpu,nice,comm
USER      PID %CPU  NI COMMAND
root        1  0.0   0 systemd
root        2  0.0   0 kthreadd
root        3  0.0  -20 rcu_gp
root        4  0.0  -20 rcu_par_gp
root        6  0.0  -20 kworker/0:0H-kblockd
root        9  0.0  -20 mm_percpu_wq
root       10  0.0   0 ksoftirqd/0
root       11  0.0   0 rcu_sched
root       12  0.0   - migration/0
root       13  0.0   - idle_inject/0
root       14  0.0   0 cpuhp/0
root       15  0.0   0 cpuhp/1
root       16  0.0   - idle_inject/1
root       17  0.0   - migration/1
root       18  0.0   0 ksoftirqd/1
root       20  0.0  -20 kworker/1:0H-kblockd
root       21  0.0   0 cpuhp/2
root       22  0.0   - idle_inject/2
root       23  0.0   - migration/2
root       24  0.0   0 ksoftirqd/2
root       26  0.0  -20 kworker/2:0H-kblockd
root       27  0.0   0 cpuhp/3
root       28  0.0   - idle_inject/3
root       29  0.0   - migration/3
root       30  0.0   0 ksoftirqd/3
```

# Приоритет процесса

Чем больше *ni*-значение, тем меньшим приоритетом будет обладать процесс.

Обычные пользователи могут задавать данный параметр в диапазоне от 0 до 20.

```
oot@jenkins:/# ps alx
F  UID      PID  PPID  PRI  NI   VSZ   RSS  WCHAN  STAT TTY      TIME COMMAND
4   0         1     0   20   0 169128 13332 ep_pol  Ss   ?        0:27 /sbin/init maybe-ubiquity
1   0         2     0   20   0      0      0 kthrea  S    ?        0:00 [kthreadd]
1   0         3     2    0  -20      0      0 rescue I<  ?        0:00 [rcu_gp]
1   0         4     2    0  -20      0      0 rescue I<  ?        0:00 [rcu_par_gp]
1   0         6     2    0  -20      0      0 worker I<  ?        0:00 [kworker/0:0H-kblockd]
1   0         9     2    0  -20      0      0 rescue I<  ?        0:00 [mm_percpu_wq]
1   0        10     2   20   0      0      0 smpboo  S    ?        0:00 [ksoftirqd/0]
1   0        11     2   20   0      0      0 rcu_gp  I    ?        4:31 [rcu_sched]
1   0        12     2  -100  -      0      0 smpboo  S    ?        0:04 [migration/0]
5   0        13     2  -51  -      0      0 smpboo  S    ?        0:00 [idle_inject/0]
1   0        14     2   20   0      0      0 smpboo  S    ?        0:00 [cpuhp/0]
1   0        15     2   20   0      0      0 smpboo  S    ?        0:00 [cpuhp/1]
5   0        16     2  -51  -      0      0 smpboo  S    ?        0:00 [idle_inject/1]
1   0        17     2  -100  -      0      0 smpboo  S    ?        0:04 [migration/1]
1   0        18     2   20   0      0      0 smpboo  S    ?        0:03 [ksoftirqd/1]
1   0        20     2    0  -20      0      0 worker I<  ?        0:00 [kworker/1:0H-kblockd]
```

# Выводы урока

- ✓ Узнали, что такое Nice-значение и его функции
- ✓ Узнали, что только root может давать отрицательные значения параметра Nice