

Запущенные процессы

Цели урока



Разобраться, что такое PID



Научиться находить нужные процессы



Узнать о том, какие бывают состояния процессов


PID и PPID

PID — идентификатор процесса.

PPID — идентификатор родительского процесса.

Например:

```
root      1254  0.0  0.5 490772 44512 ?        Ss   08:34   0:01 /usr/sbin/apache2 -k start
www-data  2449  0.0  0.1 490804 10352 ?        S    08:39   0:00 /usr/sbin/apache2 -k start
www-data  2450  0.0  0.1 490804 10352 ?        S    08:39   0:00 /usr/sbin/apache2 -k start
www-data  2451  0.0  0.1 490804 10352 ?        S    08:39   0:00 /usr/sbin/apache2 -k start
www-data  2454  0.0  0.1 490804 10352 ?        S    08:39   0:00 /usr/sbin/apache2 -k start
www-data  2455  0.0  0.1 490804 10352 ?        S    08:39   0:00 /usr/sbin/apache2 -k start
```



В системе **не существует** двух процессов с одинаковым PID. Новому процессу может быть назначен ранее использованный свободный PID.

Исключение

Init — прародитель всех процессов в системе. Это первая программа, которая выполняется при загрузке Linux и управляет остальными процессами в системе.

Он запускается самим ядром и всегда имеет PID = 1, поэтому у него в принципе **нет родительского процесса.**

Идентификатор процесса

Pidof — команда для отображения идентификатора процесса.



Поиск процессов

Способы поиска процесса:

- утилита ps
- команда pgrep

Поиск процесса: утилита ps

Утилита **ps** позволяет найти процесс по имени.



PS Command

Поиск процесса: команда `pgrep`

Команда **`pgrep`** просматривает текущие запущенные процессы и перечисляет идентификаторы процессов, которые соответствуют критериям выбора.

Какой способ выбрать?

Если нужен только идентификатор процесса → **pgrep**.

Если нужна более подробная информация о процессе → **ps**.

Чаще всего только идентификатора мало для анализа процесса, поэтому все утилиты можно использовать в комбинации.

Состояние процесса

Состояние процесса — статус, в котором находится процесс в текущий момент.

Статусы состояний:

- выполнение
- ожидание
- завершение
- зомби

Выполнение

- 1 Процесс запущен или готов к запуску
- 2 Что делать? Ожидать пользовательский ввод
- 3 Активные процессы можем узнать утилитой ps -A

```
PID TTY          TIME CMD
  1 ?             00:00:11 systemd
  2 ?             00:00:00 kthreadd
  3 ?             00:00:00 rcu_gp
  4 ?             00:00:00 rcu_par_gp
  6 ?             00:00:00 kworker/0:0H-kblockd
  8 ?             00:00:00 mm_percpu_wq
  9 ?             00:00:00 ksoftirqd/0
 10 ?            00:00:00 rcu_sched
 11 ?            00:00:00 migration/0
 12 ?            00:00:00 idle_inject/0
 14 ?            00:00:00 cpuhp/0
 15 ?            00:00:00 kdevtmpfs
 16 ?            00:00:00 netns
 17 ?            00:00:00 rcu_tasks_kthre
 18 ?            00:00:00 kthreadd
```

Ожидание

- 1 Процесс ожидает наступления некоторого события или выделения системных ресурсов
- 2 Что делать?
 - «Дать» процессу ресурсы
 - Осуществить пользовательский ввод
- 3 Чтобы увидеть процессы в состоянии ожидания, надо ввести команду `ps -aux`

root	1	0.0	0.3	164136	10384	?	Ss	16:31	0:01	/sbin/init
root	2	0.0	0.0	0	0	?	S	16:31	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	I<	16:31	0:00	[rcu_gp]
root	4	0.0	0.0	0	0	?	I<	16:31	0:00	[rcu_par_gp]
root	6	0.0	0.0	0	0	?	I<	16:31	0:00	[kworker/0:0H-events_highpri]
root	9	0.0	0.0	0	0	?	I<	16:31	0:00	[mm_percpu_wq]
root	10	0.0	0.0	0	0	?	S	16:31	0:00	[rcu_tasks_rude_]
root	11	0.0	0.0	0	0	?	S	16:31	0:00	[rcu_tasks_trace]
root	12	0.0	0.0	0	0	?	S	16:31	0:00	[ksoftirqd/0]
root	13	0.0	0.0	0	0	?	I	16:31	0:00	[rcu_sched]
root	14	0.0	0.0	0	0	?	S	16:31	0:00	[migration/0]
root	15	0.0	0.0	0	0	?	S	16:31	0:00	[cpuhp/0]
root	16	0.0	0.0	0	0	?	S	16:31	0:00	[cpuhp/1]
root	17	0.0	0.0	0	0	?	S	16:31	0:00	[migration/1]

Ожидающие процессы

Типы ожидающих процессов:

- прерываемые — могут быть прерваны сигналами
- непрерываемые — ожидают на аппаратном уровне и не могут быть прерваны каким-либо событием/сигналом

Завершён

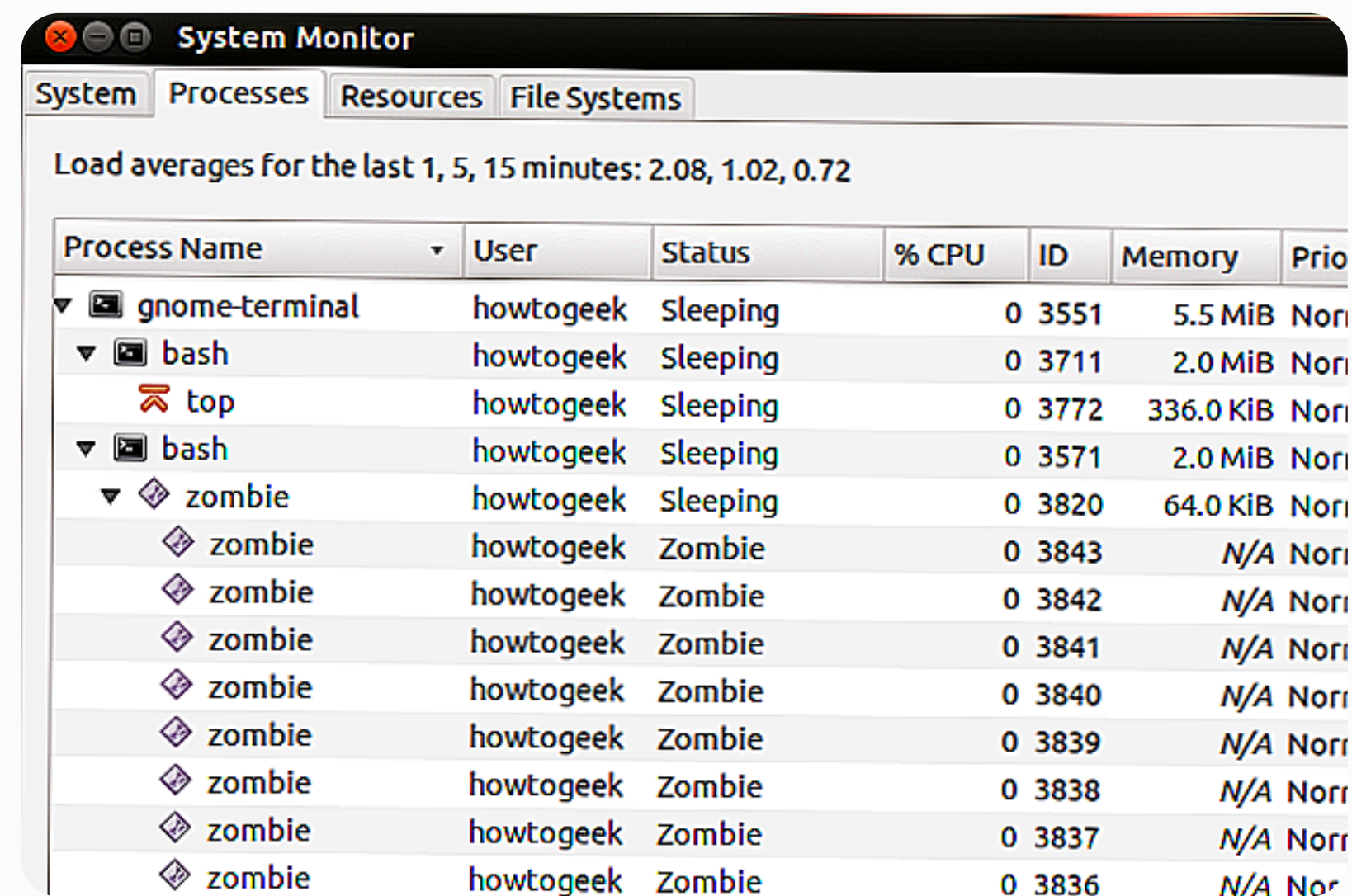
- Процесс отработал и завершился, освободив выделенные ресурсы
- После завершения процесса он нигде отображаться не будет

Сначала завершаются дочерние процессы, затем — родительские.

Зомби

Зомби — дочерние процессы, чьи родительские завершились раньше.

Зомби-процессы не наносят вреда системе.

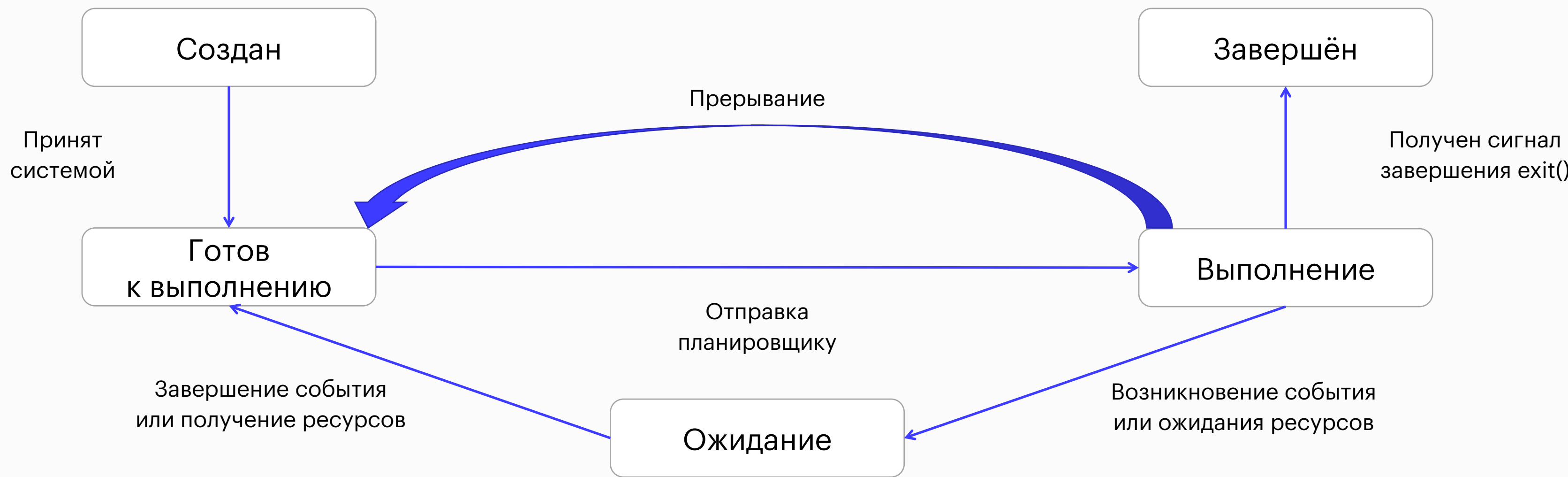


The screenshot shows the 'System Monitor' application window with the 'Processes' tab selected. It displays a table of running processes. At the top, it shows load averages for the last 1, 5, and 15 minutes: 2.08, 1.02, 0.72. The table has columns for Process Name, User, Status, % CPU, ID, Memory, and Priority. The processes listed include 'gnome-terminal', 'bash', 'top', and several 'zombie' processes. The 'zombie' processes are shown as children of the 'bash' process and have a status of 'Zombie'.

Process Name	User	Status	% CPU	ID	Memory	Prio
gnome-terminal	howtogeek	Sleeping	0	3551	5.5 MiB	Normal
bash	howtogeek	Sleeping	0	3711	2.0 MiB	Normal
top	howtogeek	Sleeping	0	3772	336.0 KiB	Normal
bash	howtogeek	Sleeping	0	3571	2.0 MiB	Normal
zombie	howtogeek	Sleeping	0	3820	64.0 KiB	Normal
zombie	howtogeek	Zombie	0	3843	N/A	Normal
zombie	howtogeek	Zombie	0	3842	N/A	Normal
zombie	howtogeek	Zombie	0	3841	N/A	Normal
zombie	howtogeek	Zombie	0	3840	N/A	Normal
zombie	howtogeek	Zombie	0	3839	N/A	Normal
zombie	howtogeek	Zombie	0	3838	N/A	Normal
zombie	howtogeek	Zombie	0	3837	N/A	Normal
zombie	howtogeek	Zombie	0	3836	N/A	Normal

Изменение состояний

Состояние процесса



Выводы урока

- ✓ Процесс можно найти по имени
- ✓ Процесс можно найти по идентификатору
- ✓ Существуют разные состояния процесса, и мы можем их менять