

Часть 6. Процессы и потоки

Влад 'mend0za' Шахов
Linux & Embedded Team Leader

Linux & Embedded Department



Процессы. Общая информация

Процесс

- Программа запускает 1 или более процессов.
- Процесс состоит из инструкций, выполняемых процессором, данных и информации о выполняемой задаче (данные в памяти и стеке, открытые файлы и статус процесса).



Процессы. Общая информация

Процесс

- Программа запускает 1 или более процессов.
- Процесс состоит из инструкций, выполняемых процессором, данных и информации о выполняемой задаче (данные в памяти и стеке, открытые файлы и статус процесса).

Изоляция процессов

- Процессы изолированы друг от друга^a.
- Процессы могут обмениваться данными через систему межпроцессного взаимодействия (IPC)
- Следствие: пользователи могут запускать несколько экземпляров одной и той же программы.

^aнет доступа к памяти и стеку, открытым файлам и порядку выполнения



Файлы и процессы. Права доступа

Связь с файлами

Процессы запускаются из файлов (двоичных программ или скриптов).

Права доступа

Процессы работают с правами пользователя, их запустившего^a

^aЕсли не установлены SUID или SGID биты на файле программы



Виды процессов

Демоны

Неинтерактивные процессы, выполняются в фоновом режиме. Не связаны ни с одним пользовательским сеансом и не могут непосредственно управляться пользователем.

Примеры: `sshd`, `apache`, `cron`, `samba`



Виды процессов

Демоны

Неинтерактивные процессы, выполняются в фоновом режиме. Не связаны ни с одним пользовательским сеансом и не могут непосредственно управляться пользователем.

Примеры: `sshd`, `apache`, `cron`, `samba`

Системные процессы

Часть ядра и всегда расположены в оперативной памяти.

Примеры: диспечер подкачки памяти (`[kswapd0]`)



Виды процессов

Демоны

Неинтерактивные процессы, выполняются в фоновом режиме. Не связаны ни с одним пользовательским сеансом и не могут непосредственно управляться пользователем.

Примеры: `sshd`, `apache`, `cron`, `samba`

Системные процессы

Часть ядра и всегда расположены в оперативной памяти.

Примеры: диспечер подкачки памяти (`[kswapd0]`)

Прикладные процессы

все остальные процессы. Запускаются в рамках пользовательского сеанса.

Примеры: `ls`, `bash`, `vim`, `find`, `mysql`



Практика: утилиты просмотр процессов

- **ps** - список запущенных процессов.

Популярные ключи: **-u username**, **ax**¹, **aux**²

PID	TTY	STAT	TIME	COMMAND
1	?	Ss	0:01	init [2]
31	?	S	0:00	[kswapd0]
3451	?	Ss	0:00	/usr/sbin/apache2 -k start
3766	?	S	0:00	/usr/sbin/vsftpd
3767	?	Ss	0:00	/usr/bin/dbus-daemon —system
3810	?	Ss	0:00	/usr/sbin/sshd
5318	pts/3	Ss	0:01	—bash
6259	pts/3	Sl	0:03	evince 06_Processes.pdf
6877	pts/3	R+	0:00	ps ax

- **top** - интерактивный список процессов

```
~$ top
top - 18:55:42 up 29 min, 0 users, load average: 0.17, 0.24, 0.24
Tasks: 152 total, 1 running, 151 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.9 us, 0.7 sy, 0.0 ni, 94.0 id, 2.4 wa, 0.0 hi, 0.0 si, 0.0 st

  PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
    1 root        20   0 10648   836  696 S   0.0   0.0   0:01.20 init
  .....
7494 mendoza    20   0 25452 1444 1076 R   0.0   0.0   0:00.01 top
```

¹**ax** для BSD, **-e** в SystemV

²**aux** для BSD, **-ef** в SystemV



Состояния процесса



Виды межпроцессного взаимодействия (IPC)

Пользовательские IPC

- 1 файлы
- 2 каналы (трубы) - именованные (FIFO) и неименованные (в Shell)



Виды межпроцессного взаимодействия (IPC)

Пользовательские IPC

- 1 **файлы**
- 2 **каналы (трубы)** - именованные (FIFO) и неименованные (в Shell)
- 3 **сигналы** - уведомление о возникновении события



Виды межпроцессного взаимодействия (IPC)

Пользовательские IPC

- 1 **файлы**
- 2 **каналы (трубы)** - именованные (FIFO) и неименованные (в Shell)
- 3 **сигналы** - уведомление о возникновении события

IPC Для программистов

- 1 **разделяемая память**
- 2 **семафоры**
- 3 **очереди сообщений**



Сигнал

- 1 Способ передачи уведомления о возникновении какого-либо события.
- 2 Сигнал может идти от одного процесса другому или от ядра ОС какому-либо процессу.
- 3 Номер сигнала - единственная информация, которую он передаёт.



Сигналы

Сигнал

- 1 Способ передачи уведомления о возникновении какого-либо события.
- 2 Сигнал может идти от одного процесса другому или от ядра ОС какому-либо процессу.
- 3 Номер сигнала - единственная информация, которую он передаёт.

Права доступа

- Пользователь может посылать сигналы только тем процессам, владельцем которых он является.
- **root** может посылать сигналы любому процессу.



Работа с сигналами

Информация о сигналах: **man 7 signal**, **kill -l**

Популярные сигналы

Сигнал	Значение	Действие .умолч.	Комментарий
SIGHUP	1	Term	Обрыв соед. терминала или смерть упр. процесса
SIGINT	2	Term	Прерывание с клавиатуры (Ctrl+C)
SIGKILL	9	Term	Убить процесс
SIGSEGV	11	Core	Segmentation fault
SIGPIPE	13	Term	Broken pipe: запись в канал без читателей
SIGTERM	15	Term	Прекратить процесс
SIGCONT	19,18,25	Cont	Продолжить выполнение
SIGSTOP	17,19,23	Stop	Остановить



Работа с сигналами

Информация о сигналах: **man 7 signal**, **kill -l**

Популярные сигналы

Сигнал	Значение	Действие .умолч.	Комментарий
SIGHUP	1	Term	Обрыв соедин. терминала или смерть упр. процесса
SIGINT	2	Term	Прерывание с клавиатуры (Ctrl+C)
SIGKILL	9	Term	Убить процесс
SIGSEGV	11	Core	Segmentation fault
SIGPIPE	13	Term	Broken pipe: запись в канал без читателей
SIGTERM	15	Term	Прекратить процесс
SIGCONT	19,18,25	Cont	Продолжить выполнение
SIGSTOP	17,19,23	Stop	Остановить

Утилиты управления

- **kill** - послать сигнал процессу (по PID)
- **killall** - послать сигнал процессам по имени

