



Процессы

- Отношения потомок-родитель
- Иерархия процессов `pstree`
- Вывод списка процессов `ps`
- Сигналы
- `kill`
- `killall`
- Специфика работы в `shell`
- Отключение от терминала – `nohup`
- Режимы работы программы



Потомок — Родитель

У всех процессов в системе, кроме самого первого, есть «родители» — процессы, которые запускают данный процесс.

Любой потомок может запустить другой процесс, для которого он будет родителем.

Самый первый процесс в системе — `init` с `PID=1`.

После завершения работы родительского процесса у потомка родителем становится процесс `init`.



pstree

Показывает дерево процессов.

Примеры:

`ps`

`ps xa | less`

`ps -e`

`ps xaf`

`pstree`

`ps xa | grep cupsd`

`pgrep cupsd`



ps [опции]

Показывает список процессов в системе.

Опции:

- help — выводит экран помощи.

- a — показывает список всех процессов «привязанных» к терминалам.

- x — показывает список процессов не «привязанных» к терминалу.

- e — показывает все процессы системы.

- f — показывает дерево процессов.

- u user — список процессов

пользователя.



Сигналы

Процессы могут «общаться» друг с другом при помощи сигналов.

Сигнал — это число, которое одна программа

может послать другой программе.

Реакция программы на получаемый сигнал

зависит от программиста, написавшего ее.

Для того, чтобы послать сигнал процессу можно воспользоваться программой `kill`.



kill [-сигнал] PID ...

Программа посылает сигнал процессу.

Процесс определяется его PID.

Опции:

- l — показать список всех сигналов в системе.

Примеры:

ps

kill PID_программы_bash

kill -9 PID_программы_bash



killall [-сигнал] имя ...

Программа посылает сигнал процессу.

Процесс определяется по его имени.

Пример:

```
killall firefox
```



Специфика работы в shell

Если shell заканчивает свою работу, все процессы, запущенные в этой оболочке будут завершены.

Для того, что бы программа продолжала работу после закрытия оболочки, ее необходимо запускать при помощи программы `nohup`.



nohup программа

nohup отключает программу от терминала, что позволяет продолжить выполнение программы после его выключения.

Пример запуска программы:

```
nohup programt -p1 -p2
```



Демоны

Демоны написаны таким образом, что сразу после запуска отключаются от терминала и могут продолжать работать после выхода пользователя из оболочки.



Режимы работы программы

Программа может работать в режимах:

- foreground** — занимает консоль пользователя.

- background** — запускается как параллельный процесс. После запуска программы пользователю доступна командная строка.



Режимы работы программы

Для запуска программы в `background` режиме в конце командной строки необходимо написать символ `&`

Если программа, запущенная в `background`-режиме попытается что либо прочитатъ со стандартного ввода, ее выполнение будет приостановлено.



Управление задачами

Ctrl+Z — приостановка выполнения программы

jobs — показывает список приостановленных и запущенных в background режиме программ.

fg [число] — продолжает выполнение программы в foreground режиме.

bg [число] — продолжает выполнение программы в background режиме.