Демоны - Так же, говоря о процессах в линуксе, можно выделить особый вид процессов - **демоны**. Данный вид процессов работает в фоне (подобно **службам** в Windows), без терминала и выполняет задачи для других процессов. Данный вид процессов на серверных системах является основным.

**Фоновый режим**

Команды можно запускать в фоновом режиме, что бы получить сразу доступ к терминалу (например мы пингуем сайт и записываем данные в файл, и после хотим сразу посмотреть изменения в файле с помощью tail -f. Если не запустить в фоновом режиме, то терминал будет заблокирован командой ping, до тех пор пока мы ее не отменим, а тогда не будет смысла смотреть динамические изменения

Задачу (например, команду или скрипт) можно перевести в фоновый режим, добавив символ «&» в конец командной строки. Этот оператор переводит команду в фоновый режим и освобождает место в терминале. Команда, выполняемая в фоновом режиме, называется заданием. Во время работы фоновой команды можно выполнять любые другие команды

**Примеры:**

Чтобы перевести команду ls в фоновый режим, введите:

ls \*.py > output.txt &

jobs – находит задания запущенные в фоновом режиме

jobs –l – отображает PID процессов

%идентификаторПроцесса – переводит задачу из фонового режима в приоритетный

Что бы обратно перевести задачу в фоновый режим, нужно нажать CTRL + Z, при этом выполнение будет приостановлено

%идентификаторПроцесса & - опять запустить задачу в фоновом режиме, после остановки

**kill %**номер\_job – убить фоновый процесс

**fg** – вывести job из фона

**bg** – отправить job в фон

**Управление процессами:**

Процесс зомби – завершился, но код завершения не отправил, или родительский процесс не смог его убить. Что бы убить процесс зомби нужно убить процесс родителя

Процесс сирота – завершился родительский процесс, и родительский процесс не завершил дочерний процесс. Такой процес переходит под контроль init.d (самый первый процесс в системе) и init.d его завершает

**systemctl (start, stop disable, restart,..)** (название приложения) – управление основными процессами. Запускает приложение на заднем фоне.

**ps aux** – посмотреть все процессы

**ps -e** – посмотреть все процессы

**kill** номерпроцесса – завершить процесс

**kill** -сигнал PIDпроцесса – завершить процесс

* 2 (INT) - Interrupt. В случае выполнения простых команд, вызывает прекращение выполнения, в интерактивных программах — прекращение активного процесса
* 9 (KILL) - Всегда прекращает выполнение процесса
* 15 (TERM) - Software Termination. Требование закончить процесс (программное завершение)
* 18 (CONT) - Продолжение выполнения приостановленного процесса
* 19 (STOP) - Приостановка выполнения процесса

**cat /proc/meminfo** – информация о потреблении RAM

**free** – информация о потреблении RAM

**cat /proc/cpuinfo** – информация о потреблении CPU

**cat /proc/loadavg** – хранит среднюю загрузку процессора за последние 1,5 и 15 минут

**top** – посмотреть динамическое изменение загруженности системы

**PID** – айдишник процесса

**USER** – пользователь от которого запущен процесс

**PR** – приоритет процесса

**NI** – nice value процесса

**VIRT** – виртуальная память

**RES** – физическа память, которая не в swap

**SHR** – объединенная память

**S** – статус процесса

• 'D' = uninterruptible sleep

• 'R' = running

• 'S' = sleeping

• 'T' = traced or stopped

• 'Z' = zombie (Такое может произойти, если **процесс-потомок** (дочерний процесс) завершился раньше, чем этого ожидал **процесс-родитель**. **Процессы-зомби** не занимают процессорного времени (т. е. их выполнение прекращается), но соответствующие им структуры ядра не освобождаются. В некотором смысле это «мертвые» процессы.

**%CPU** – загруженность процессора

**%MEM** – загруженномть памяти

**TIME+** - время которое процесс выполняется

**COMMAND** – название процесса