

# Процессы в Linux

Программы, процессы, демоны, зомби,  
сироты





# Программа

- термин, в переводе означающий «предписание»
- комбинация компьютерных инструкций и данных
- обычно хранится в энергонезависимой памяти



# Процесс

- термин, в переводе означающий «течение, ход, продвижение»
- выполнение пассивных инструкций компьютерной программы
- выполняющаяся программа и все её элементы (адресное пространство, глобальные переменные, регистры... )



# Отличия

- программа — пассивная последовательность инструкций
- процесс — непосредственное выполнение этих инструкций

# Программа



# Процесс





# Системные вызовы

- обращение прикладной программы к ядру операционной системы
- управляют процессами
- работают с файлами
- управляют устройствами
- работают с информацией
- обеспечивают связь и коммуникацию



# Системные вызовы для управления процессами

`fork()` - создание процесса

`exec()` - загрузка (запуск) программы в процесс

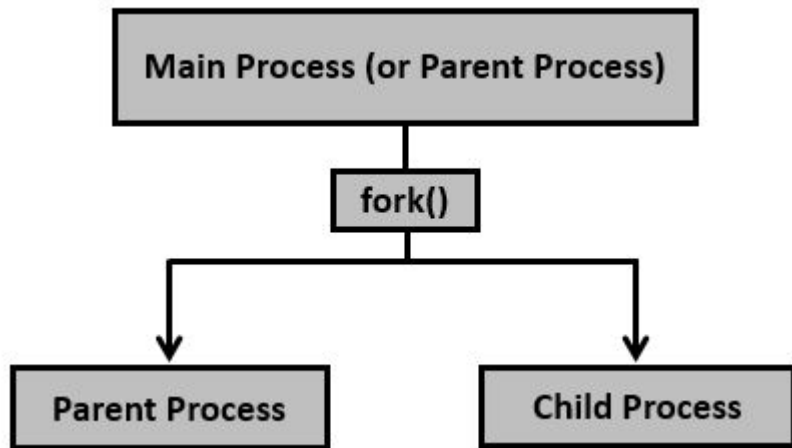
`exit()` - завершение процесса

`wait()` - считывание кода возврата процесса



# **fork() - единственный способ создать процесс в Linux**

- стартует в одном процессе, завершается в двух
- процессу родителю возвращает код возврата равный PID дочернего процесса
- дочернему процессу возвращает код 0





# **exec() - способ запустить программу в Linux**

- работает в существующем процессе
- заменяет программу процесса на ту, что была передана `exec()` в кач-ве аргумента
- `exec()` - семейство системных вызовов (`execl`, `execle`, `execlp`, `execv`, `execve`, `execvp`)



# exit() - завершение работы программы в Linux

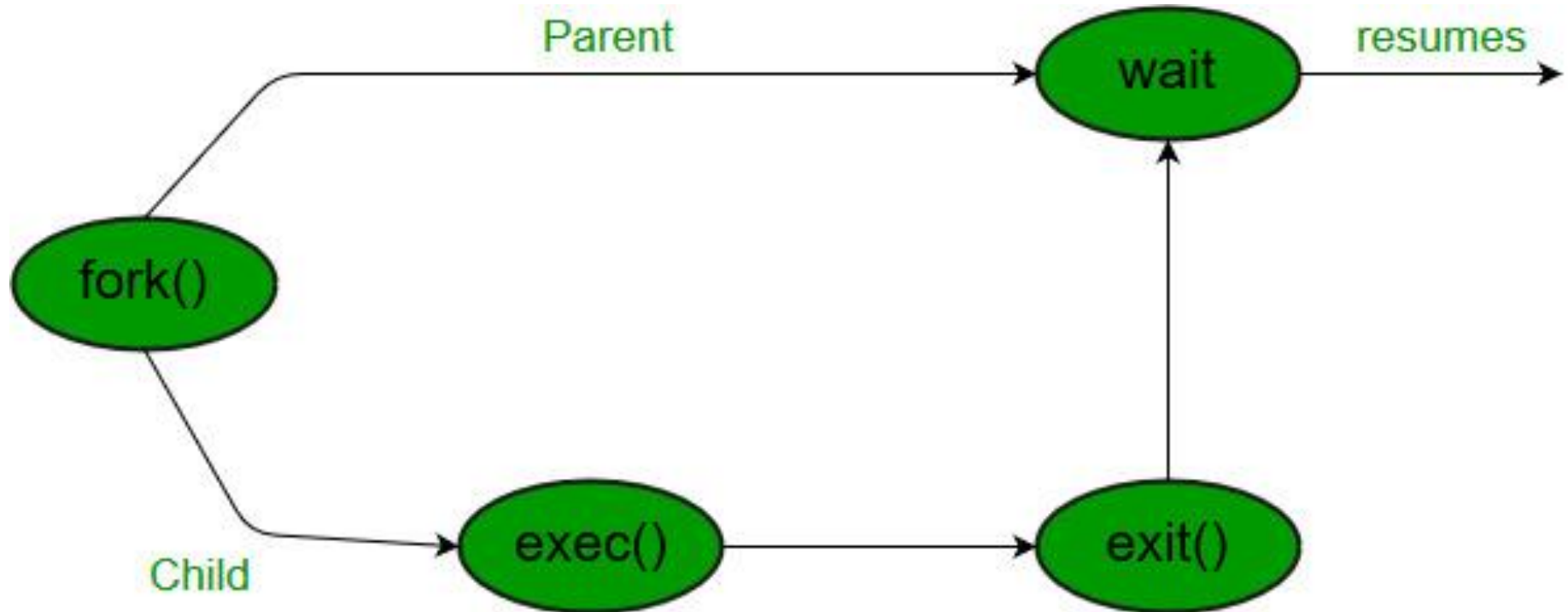
- приводит к обычному завершению программы
- после завершения программы ее переменная *status* будет доступна родительскому процессу (*status* - код возврата)
- `exit()` - не завершает процесс



# **wait() -считывает код возврата дочернего процесса в Linux**

- приостанавливает выполнение текущего процесса до тех пор, пока дочерний процесс не прекратит выполнение
- получает код возврата дочернего процесса
- после вызова wait() дочерний процесс полностью исчезает из системы

# Жизненный цикл процесса в Linux





# Некоторые состояния процессов

R - выполняется в данный момент

S - процесс ожидает

D - ожидает ввода-вывода

T - процесс остановлен

Z - процесс зомби

# Демон

- процесс работающий в фоновом режиме без прямого взаимодействия с пользователем
- sshd, httpd, ftpd и т.д. - программы работающие в режиме демона (последняя буква в названии “d” означает daemon)
- название произошло в честь демона Максвелла



# Зомби

- обязательное состояние любого процесса
- процессы зомби не потребляют ресурсы
- процесс зомби нельзя убить (завершить)

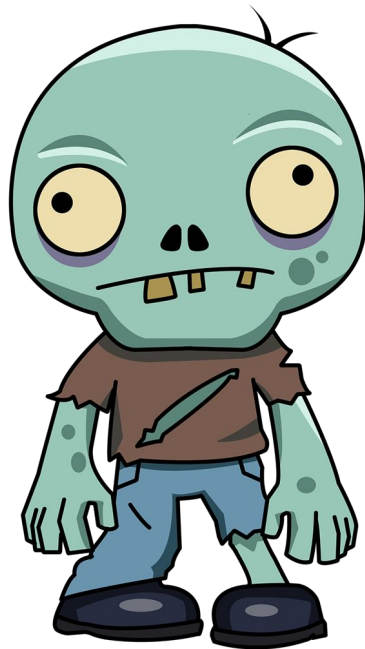






Зомби

exit()



wait()



# Сирота

- процесс оставшийся без родителя
- будет мгновенно усыновлен процессом init (в современных системах systemd)
- явление очень кратковременное (увидеть процесс сироту не получится)





# Как убить зомби?

- А нужно ли...?

- сделать его сиротой
- заставить родителя послать `wait()`