

Загрузка системы. Уровни запуска.

Термин уровень выполнения означает режим функционирования операционной системы компьютера, в которой реализована инициализация в стиле ОС Unix System V. Традиционно существуют семь уровней выполнения, пронумерованных от 0 до 6

Для запуска сервиса при старте Linux необходимо убедиться, что существует необходимый сервис в папке `/etc/init.d/`. Пример скрипта для запуска самописного сервиса можно посмотреть в `/etc/init.d/skeleton`. После этого можно добавить сервис при загрузке создав симлинк в нужном уровне запуска `/etc/rcX.d/`, где X может меняться от 0..6

Формат именования скриптов

S[порядковый номер][имя] - имя скрипта для запуска сервиса

K[порядковый номер][имя] - имя скрипта для остановки сервиса

Порядковый номер может меняться от 1..99, чем выше порядковый номер, тем позднее запустится сервис.

Загрузка системы. Уровни запуска. Upstart

Автор: yakim

22.03.2012 12:25 - Обновлено 23.03.2012 06:41

После запуска всех сервисов для указанного уровня инициализации (*runlevel*) выполняется скрипт

/etc/rc.local

, который содержит пользовательские сценарии запуска.

Уровни инициализации в Debian (Ubuntu)

0 - остановка

1 - однопользовательский режим

2-5 - многопользовательский режим

6 - перезагрузка

По умолчанию используется уровень **2**.

P.S. Спасибо сайту proft.me за идею и описание уровней запуска системы.

Upstart

Проект **upstart** создан с целью заменить традиционный способ загрузки пользовательских процессов в **Linux**, а также способ управления ими.

Как и в классическом варианте - **upstart** имеет главный процесс, который отвечает за загрузку и управление демонами. Называется он также как и в классическом варианте - **init**

В **upstart** ввели такое понятие как “задание” (**jobs**). Если во время запуска процесс **init** читает конфигурацию из файла

/etc/inittab

, то процесс

upstart init

читает конфигурацию из файлов каталога

/etc/init/

. Эти файлы и являются

файлами-заданиями

(

jobs

). Каждый файл в каталоге

/etc/init/

отвечает за запуск отдельного демона или сервиса и должен заканчиваться на

.conf

. Также допускается, создавать подкаталоги в которых могут находиться

файлы-задания

.

Синтаксис **файлов-заданий** несложный. В **файлах-заданиях** обязательно содержится

Загрузка системы. Уровни запуска. Upstart

Автор: yakim

22.03.2012 12:25 - Обновлено 23.03.2012 06:41

информация когда задание должно стартовать (определение

start on

), когда останавливаться (

stop on

) и, что запускать (ключевые слова

exec

или

script

).

Итак, когда же задание сработает? Вот здесь и есть первое основное отличие которое решает проблему последовательного запуска скриптов. **upstart** - это подсистема которая реагирует на

co

бытия

. Поэтому после ключевых слов

start on

или

stop on

, обязательно указывается

событие

(

event

), при наступлении которого начинает выполняться запуск или остановка задания.

*Если рассматривать события наиболее часто употребляемые в файлах заданий, то это **st***

artup

,

runlevel

,

stopped

и

started

.

startup

- это самое первое событие, которое распознается подсистемой

upstart

на самом раннем этапе загрузки операционной системы. Событие

runlevel

(с указанием целевого уровня), как следует из названия генерируется при смене уровня запуска. Событие

stopped

генерируется

Загрузка системы. Уровни запуска. Upstart

Автор: yakim

22.03.2012 12:25 - Обновлено 23.03.2012 06:41

после

остановки указанного задания, а

started

после

завершения старта задания. Просмотрев файлы-задания в каталоге /etc/init/

можно найти и другие события.

Таким образом если необходимо создать задание которое будет запускать тестовый демон test на 2-м и 3-м уровне запуска и останавливать на остальных уровнях, достаточно создать файл **test.conf** в котором написать:

```
start on runlevel [23]
```

```
stop on runlevel [!23]
```

```
exec test
```

*Для ручного запуска или остановки заданий в **upstart** предусмотрены команды **start** и **stop***

, расположенные, как правило, в каталоге

/sbin

. Чтобы в

Ubuntu

остановить демон

cron

достаточно ввести команду

sudo stop cron.

Чтобы запустить cron набираем

```
sudo start cron
```

Из других команд управления можно отметить команду `status`, которая показывает в каком состоянии (останова или работы) находится задание (а соответственно и демон).

Также следует отдельно упомянуть команду `initctl`. Эта команда как следует из ее названия (сокращенное от `init control`), позволяет управлять демоном `upstart init`. Команда `initctl` тоже умеет останавливать (`initctl stop`) и запускать (`initctl start`) задания, а также проверять их статусы (`initctl status`). Если нужно посмотреть статус всех заданий есть команда `initctl list`

Upstart просматривает и выполняет только файлы-задания расположенные в каталоге `/etc/init/`, поэтому для того, чтобы выполнялись скрипты из каталогов `/etc/rcX.d/`, необходимо “сказать” об этом `upstart`. Для этой цели в каталоге

Загрузка системы. Уровни запуска. Upstart

Автор: yakim

22.03.2012 12:25 - Обновлено 23.03.2012 06:41

/etc/init/

находится файл-задание

rc.conf

. Именно благодаря этому файлу и происходит просмотр и запуск скриптов для стандартного

init

из каталога

/etc/rcX.d/

Выполняя этот файл-задание, upstart запускает скрипт /etc/init.d/rc с передачей ему уровня запуска (\$RUNLEVEL).

А дальше уже по классической схеме - скрипт rc начинает выполнять скрипты из соответствующего каталога **/etc/rc**
X.d/ **.**

{jcomments on}