

Администрирование Linux: Инициализация системы. Init, Systemd



Артур
Сагутдинов



Артур Сагутдинов

Начальник IT отдела

ООО «Клинический
институт репродуктивной
медицины»



15+ лет в сфере ИТ



Разрабатываю и внедряю
линуксовую инфраструктуру



[Сисадминский блог](#)



Предисловие

На этом занятии мы поговорим о:

- современных менеджерах управления процессами;
- рассмотрим systemd.

По итогу занятия вы получите представление о менеджерах управления процессами в ОС Linux и научитесь управлять службами.

План занятия

1. [Предисловие](#)
2. [init](#)
3. [systemd](#)
4. [Итоги](#)
5. [Домашнее задание](#)



init

init

init — специальный процесс (демон) управления системой и службами.

Расположение:

`/sbin/init`

Режимы работы init:

- однопользовательский (службы не запускаются);
- многопользовательский (режим запуска по умолчанию);
- сервер (аналогичен многопользовательскому, но без GUI).

Варианты init

- **System V:**
 - Все службы запускаются последовательно.
- **BSD init:**
 - FreeBSD, NetBSD, OpenBSD.
- **systemd:**
 - упрощенный процесс загрузки;
 - параллельный запуск служб;
 - запись событий в системный журнал.

Процесс запуска Init-V

- GRUB загружает и запускает ядро;
- ядро запускает `/sbin/init`;
- `init` разбирает `/etc/inittab` и выполняет сценарий для инициализации системы;
- `init` выполняет скрипт `/etc/rc.d/rc` или `/etc/init.d/rc`;
- скрипты из `/etc/rcn.d` или `/etc/init.d/rcn.d` запускают различные службы.

Уровни запуска Init-V*

0. остановка работы с выключением;
1. (S) — однопользовательский режим;
2. многопользовательский режим без выхода в сеть;
3. многопользовательский режим с сетью, но без запуска X;
4. обычно не применяется;
5. многопользовательский режим с сетью и запуском X (по умолчанию);
6. остановка работы компьютера с перезагрузкой.

* Во всех ОС, кроме Debian

Уровни запуска Init-V*

- S. инициализация компьютера непосредственно после запуска;
- 0. остановка работы компьютера с выключением;
- 1. однопользовательский режим с доступом к сети;
- 2–5. многопользовательский режим с сетью и запуском X (по умолчанию);
- 6. остановка работы компьютера с перезагрузкой.

* В ОС Debian

Определение уровня запуска

Определение уровня запуска:

```
user@user:~$ who -r
```

```
run-level 5  2021-01-21 09:02
```

Определение типа init

- **Systemd** — если в системе есть каталоги `/usr/lib/systemd` и `/etc/systemd`;
- **Upstart** — если каталог `/etc/init` содержит несколько файлов с расширением `.conf`, можно выполнить команду `initctl`;
- **System V init** — если ни один из приведенных вариантов не подходит, однако есть файл `/etc/inittab`.

Также можно выполнить:

```
user@user:~$ file /sbin/init
```



systemd

systemd

user@user:~\$ top

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1006	root	20	0	813384	341216	16196	S	7,3	8,5	1:37.42	Suricata-Main
855	root	20	0	94520	2192	2016	R	0,3	0,1	0:01.44	sdrplay_apiServ
874	redis	20	0	52660	3688	2620	S	0,3	0,1	0:00.74	redis-server
1252	user	20	0	465616	91632	51864	S	0,3	2,3	0:08.23	Xorg
1372	user	20	0	120500	2292	1920	S	0,3	0,1	0:00.83	VBoxClient
1409	user	20	0	2966580	231332	114028	S	0,3	5,7	0:26.11	gnome-shell
11015	user	20	0	43004	3984	3348	R	0,3	0,1	0:00.29	top
1	root	20	0	160392	9676	6728	S	0,0	0,2	0:02.87	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_par_gp
5	root	20	0	0	0	0	I	0,0	0,0	0:00.33	kworker/0:0-ata
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H-kb

user@user:~\$ systemctl status

Модули systemd

Модуль (unit) — описывает запускаемую службу, устройство и т.п.

Каждый модуль описан в своем файле (unit file):

- `/usr/lib/systemd/system/` — модули из пакетов (Nginx, Apache, MySQL);
- `/run/systemd/system/` — модули, созданные во время работы ОС;
- `/etc/systemd/system/` — модули, созданные пользователем.

Типы модулей

- **Target** — цели;
- **Service** — службы;
- **Socket** — сокеты;
- **Device** — устройства;
- **Mount** — точки монтирования;
- **Automount** — точки монтирования (по запросу);
- **Swap** — монтирование swap;
- **Path** — отслеживает заданный путь;
- **Timer** — проверяет таймер;
- **Snapshot** — записывает состояние модулей;
- **Slice** — управление ресурсами;
- **Scope** — управляет внешними процессами;
- **busname** — управляет DBus.

Типы модулей

- **модули служб** — обычные службы ОС;
- **модули монтирования** — монтируют ФС;
- **целевой модули/цели** — группируют другие модули;

```
user@user:~$ systemd-analyze plot > test.svg
```

```
user@user:~$ sudo systemctl list-dependencies
```

Цели systemd

Цель (target) — нужное состояние системы; ссылка на файл, содержащий зависимости (службы).

➔ Systemd запускает все зависимости из соответствующего target-файла.

➔ Когда все зависимости будут запущены, то система будет работать на соответствующем target-уровне.

```
user@user:~$ systemctl get-default
```

```
graphical.target
```

systemd и runlevels

runlevel **0** : **poweroff**.target / runlevel0.target
runlevel **1** : **rescue**.target / runlevel1.target
runlevel **2** : **multi-user**.target / runlevel2.target
runlevel **3** : **multi-user**.target / runlevel3.target
runlevel **4** : **multi-user**.target / runlevel4.target
runlevel **5** : **graphical**.target / runlevel5.target
runlevel **6** : **reboot**.target / runlevel6.target

systemd и runlevels

runlevels в systemd:

user@user:~\$ ls /lib/systemd/system/runlevel*

```
user@user:~$  
user@user:~$  
user@user:~$ ls /lib/systemd/system/runlevel*  
/lib/systemd/system/runlevel0.target /lib/systemd/system/runlevel4.target  
/lib/systemd/system/runlevel1.target /lib/systemd/system/runlevel5.target  
/lib/systemd/system/runlevel2.target /lib/systemd/system/runlevel6.target  
/lib/systemd/system/runlevel3.target
```

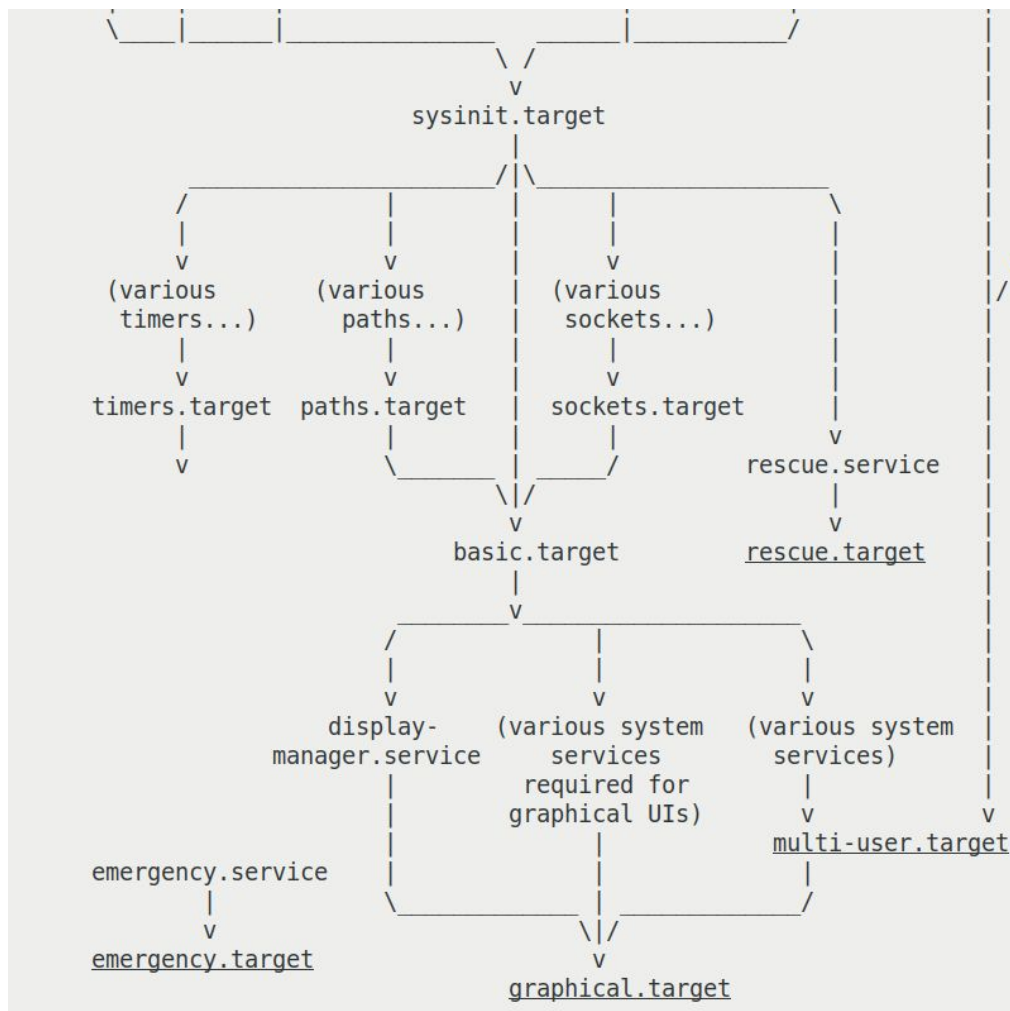
targets в systemd:

user@user:~\$ ls /lib/systemd/system/*.target

Иерархия целей

user@user:~\$

man **bootup**



Пример цели

Graphical.target (default)

```
user@user:~$ cat /lib/systemd/system/graphical.target
# SPDX-License-Identifier: LGPL-2.1+
#
# This file is part of systemd.
#
# systemd is free software; you can redistribute it and/or modify it
# under the terms of the GNU Lesser General Public License as published by
# the Free Software Foundation; either version 2.1 of the License, or
# (at your option) any later version.

[Unit]
Description=Graphical Interface
Documentation=man:systemd.special(7)
Requires=multi-user.target
Wants=display-manager.service
Conflicts=rescue.service rescue.target
After=multi-user.target rescue.service rescue.target display-manager.service
AllowIsolate=yes
```

Модули служб

Просмотр всех служб:

```
user@user:~$ systemctl --all -t service
```

Просмотр выбранной службы (ssh):

```
user@user:~$ systemctl status sshd.service
```

Пример модуля

cat /etc/systemd/system/multi-user.target.wants/ssh.service

```
[Unit]
Description=OpenBSD Secure Shell server
After=network.target auditd.service
ConditionPathExists=!/etc/ssh/sshd_not_to_be_run

[Service]
EnvironmentFile=-/etc/default/ssh
ExecStartPre=/usr/sbin/sshd -t
ExecStart=/usr/sbin/sshd -D $SSH_OPTS
ExecReload=/usr/sbin/sshd -t
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartPreventExitStatus=255
Type=notify
RuntimeDirectory=sshd
RuntimeDirectoryMode=0755

[Install]
WantedBy=multi-user.target
Alias=sshd.service
```


Раздел Unit

[Unit]

Description=OpenBSD Secure Shell server — описание службы

After=network.target auditd.service — запуск после (after) этой службы

ConditionPathExists=!/etc/ssh/sshd_not_to_be_run — условная
ЗАВИСИМОСТЬ

Раздел Service

```
[Service]
EnvironmentFile=-/etc/default/ssh
ExecStartPre=/usr/sbin/sshd -t
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/usr/sbin/sshd -t
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartPreventExitStatus=255
Type=notify
RuntimeDirectory=sshd
RuntimeDirectoryMode=0755
```

Раздел install

[Install]

WantedBy=multi-user.target

Alias=sshd.service

Вызов: `systemctl enable sshd.service`

выполняет

`ln -s '/usr/lib/systemd/system/sshd.service'`

`'/etc/systemd/system/multi-user.target.wants/sshd.service'`

systemctl

- systemctl `list-units` — список модулей;
- systemctl `list-units --type=service` — список модулей-служб;
- systemctl `status <модуль>` — состояние выбранного модуля;
- systemctl `enable\disable <модуль>` — разрешить/запретить модуль;
- systemctl `start\stop\restart <модуль>` — запустить/остановить/... модуль;
- systemctl `daemon-reload` — перезапуск конфигурации systemd.

Передача параметров в службу

Иногда требуется передать службе дополнительные параметры. Это можно выполнить либо через файл, например:

```
sudo nano /etc/my.conf
```

```
param1=-p1
```

```
param2=-p2
```

Выполним изменения в unit-файле, раздел [service]:

```
EnvironmentFile=/etc/my.conf
```

```
ExecStart=/usr/bin/test_service $param1 $param2
```

Либо указать параметры без файла в разделе [service]:

```
ExecStart=/tmp/test.py $params
```

Шаблоны служб

Шаблон - специальный unit-файл, позволяющий systemd работать с несколькими экземплярами unit.

Для вызова шаблона использует специальный формат:

`<имя_службы>@<аргумент>.service`

Пример:

```
ExecStart=/usr/sbin/openvpn --daemon --writepid  
/var/run/openvpn/%i.pid --cd /etc/openvpn/ --config %i.conf
```

Вызов:

```
sudo systemctl start openvpn@config1.service
```

Шаблоны служб: пример

Создадим файл:

`/etc/systemd/system/test_service@.service`

Содержащий

`[Unit]`

`Description=Test service %I`

`[Service]`

`Type=simple`

`ExecStart=/bin/sleep 120 %I`

Шаблоны служб: результат работы

```
user@user:~$ sudo nano /etc/systemd/system/test_service@.service
[sudo] password for user:
user@user:~$ sudo systemctl daemon-reload
user@user:~$ sudo systemctl start test_service@1
user@user:~$ ps aux|grep sleep
root      2211  0.0  0.0   8088    596 ?        Ss   11:59   0:00 /bin/sleep 120 1
user      2256  0.0  0.0   9048    732 pts/0    S+   12:00   0:00 grep --color=auto sleep
user@user:~$ sudo systemctl start test_service@10
user@user:~$ sudo systemctl start test_service@50
user@user:~$ ps aux|grep sleep
root      2211  0.0  0.0   8088    596 ?        Ss   11:59   0:00 /bin/sleep 120 1
root      2260  0.0  0.0   8088    588 ?        Ss   12:00   0:00 /bin/sleep 120 10
root      2265  0.0  0.0   8088    588 ?        Ss   12:00   0:00 /bin/sleep 120 50
user      2267  0.0  0.0   9048    740 pts/0    S+   12:00   0:00 grep --color=auto sleep
user@user:~$
```

journalctl

Журнал — база данных, в которой хранятся сообщения ядра и служб, начиная с загрузки и заканчивая завершением работы.

```
user@user:~$ journalctl
```

Настройки журнала:

```
user@user:~$ nano /etc/systemd/journald.conf
```

journalctl

- `user@user:~$ journalctl -u=sshd` — сообщения для модуля ssh;
- `user@user:~$ journalctl -b 0 -u ssh` — ...только в текущем сеансе;
- `user@user:~$ journalctl --list-boots` — загрузки;
- `user@user:~$ journalctl -n 100 /usr/sbin/sshd` — показать внешний лог;
- `user@user:~$ journalctl --since=yesterday --until=now` — временной период.



Итоги

Итоги

Сегодня мы:

- рассмотрели как устроен менеджер процессов `systemd`;
- познакомились с управлением службами;
- познакомились со службой журнала.

Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте в чате учебной группы и/или в разделе “Вопросы по заданию”.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

⌘ нетология

**Задавайте вопросы и
пишите отзыв о лекции!**

Артур Сагутдинов