

Если вы видите что-то необычное, просто сообщите мне.

# Simple vs Oneshot - Выбираем тип systemd сервиса

Этот пост довольно подробный, но если вы просто ищите общую информацию когда и какие типы сервисов использовать, читайте под катом.

Когда вы создаете свой `systemd` сервис, выбор типа сервиса может быть довольно сложен. Есть множество доступных и полезных типов сервисов, но этот пост сконцентрирован вокруг разниц между `oneshot` и `simple` простого сервисов. Возможно вас смущает какой из них использовать.

## Время запуска последующей единицы

Это наибольшая разница между `oneshot` и `simple` сервисами, когда стартует слудующая единица. Как указано в `man`: следующая единица простого сервиса стартует сразу же. На картинке ниже можете посмотреть:

## Простой сервис и следующие за НИМ



Напротив же в `oneshot` сервисе, все последующие единицы дождутся завершения сервиса прежде чем они стартанут.

## Oneshot сервис и следующие за НИМ



Давайте рассмотрим простой пример сервиса и последующего за ним:

## simple-test.service

```
[Unit]
```

```
Description=Simple service test
```

```
[Service]
```

```
Type=simple
```

```
ExecStart=/bin/bash -c "echo Simple service - start && sleep 60 && echo Simple service - end"
```

И зависимый сервис:

## dep-simple-test.service

```
[Unit]
```

```
Description=Dependent service
```

```
After=simple-test.service
```

```
Requires=simple-test.service
```

```
[Service]
```

```
ExecStart=/bin/bash -c "echo Dependent service - running"
```

Запуск зависимого сервиса `dep-simple-test.service` запустит `simple-test.service` сначала(из-за `After/Requires` директив), а логи выведут следующее:

```
Jun 19 20:28:16 thstring20200619162314 systemd[1]: Started Simple service test.
Jun 19 20:28:16 thstring20200619162314 systemd[1]: Started Dependent service.
Jun 19 20:28:16 thstring20200619162314 bash[1238]: Simple service - start
Jun 19 20:28:16 thstring20200619162314 bash[1239]: Dependent service - running
Jun 19 20:28:16 thstring20200619162314 systemd[1]: dep-simple-test.service: Succeeded.
Jun 19 20:29:16 thstring20200619162314 bash[1238]: Simple service - end
Jun 19 20:29:16 thstring20200619162314 systemd[1]: simple-test.service: Succeeded.'
```

Простой пример(как и множество дальше) просто используют `sleep` для имитации работы сервиса. Так как `simple-test.service` это просто сервис, сразу за ним следует запуск `dep-simple-test.service`, и можно увидеть как оба сервиса стартуют в одно и то же время.

Но если мы сделаем тоже самое для `oneshot` сервиса, давайте посмотрим как различия выглядят.

## oneshot-test.service

```
[Unit]
```

```
Description=Oneshot service test
```

```
[Service]
```

```
Type=oneshot
```

```
ExecStart=/bin/bash -c "echo Oneshot service - start && sleep 60 && echo Oneshot service - end"
```

# dep-oneshot-test.service

```
[Unit]
Description=Dependent service
After=oneshot-test.service
Requires=oneshot-test.service

[Service]
ExecStart=/bin/bash -c "echo Dependent service - running"
```

Логирование для этих двух единиц(после запуска `dep-oneshot-test.service` ) показывает разницу:

```
Jun 19 20:31:46 thstring20200619162314 systemd[1]: Starting Oneshot service test...
Jun 19 20:31:46 thstring20200619162314 bash[1420]: Oneshot service - start
Jun 19 20:32:46 thstring20200619162314 bash[1420]: Oneshot service - end
Jun 19 20:32:46 thstring20200619162314 systemd[1]: oneshot-test.service: Succeeded.
Jun 19 20:32:46 thstring20200619162314 systemd[1]: Started Oneshot service test.
Jun 19 20:32:46 thstring20200619162314 systemd[1]: Started Dependent service.
Jun 19 20:32:46 thstring20200619162314 bash[1440]: Dependent service - running
Jun 19 20:32:46 thstring20200619162314 systemd[1]: dep-oneshot-test.service: Succeeded.
```

Вы можете видеть как зависимый сервис не запускается пока `oneshot` сервис не завершится.

# Состояния активации

Состояния активации различных типов сервисов управляют множеством взаимодействия с другими единицами.

|   |   |   |   |
|---|---|---|---|
| Т | Д | В | П |
| и | о | о | о |
| п |   | в | с |
|   |   | р | л |
|   |   | е | е |
|   |   | м |   |
|   |   | я |   |

|                                 |   |  |  |
|---------------------------------|---|--|--|
| S<br>i<br>m<br>p<br>l<br>e      | i<br>n<br>a<br>c<br>t<br>i<br>v<br>e<br>(<br>r<br>(<br>d<br>e<br>a<br>d<br>)<br>)     | a<br>c<br>t<br>i<br>v<br>e<br>(<br>u<br>(<br>n<br>e<br>i<br>n<br>g<br>)<br>) | i<br>n<br>a<br>c<br>t<br>i<br>v<br>e<br>(<br>d<br>e<br>a<br>d<br>) |
| O<br>n<br>e<br>s<br>h<br>o<br>t | i<br>n<br>a<br>c<br>t<br>i<br>v<br>e<br>(<br>d<br>e<br>a<br>d<br>)<br><br>r<br>t<br>) | a<br>c<br>t<br>i<br>v<br>e<br>(<br>n<br>g<br>(<br>s<br>t<br>a<br>r<br>t<br>) | i<br>n<br>a<br>c<br>t<br>i<br>v<br>e<br>(<br>d<br>e<br>a<br>d<br>) |

|   |   |   |   |
|---|---|---|---|
| O | i | a | a |
| n | n | c | c |
| e | a | t | t |
| s | c | i | i |
| h | t | v | v |
| o | i | a | e |
| t | v | t | ( |
| ( | e | i | e |
| R | ( | n | x |
| e | d | g | i |
| m | e | ( | t |
| a | a | s | e |
| i | d | t | d |
| n | ) | a | ) |
| A |   | r |   |
| f |   | t |   |
| t |   | ) |   |
| e |   |   |   |
| r |   |   |   |
| E |   |   |   |
| x |   |   |   |
| i |   |   |   |
| t |   |   |   |
| ) |   |   |   |

Состояние **Во время** различного состояния между `simple` и `oneshot` это причина почему следующая единица ждет завершения `oneshot` сервиса и почему не ждет завершения `simple` сервиса.

## RemainAfterExit (oneshot)

Вы можете заметить такую директиву выше, `RemainAfterExit` меняет поведение `oneshot` сервиса не много. Это просто способ сказать `systemd` что после того как он выходит, он долже держать активное состояние. Для понимания, рассмотрим пример:

# oneshot-remainafterexit.service

[Unit]

Description=Oneshot service test with RemainAfterExit

[Service]

Type=oneshot

RemainAfterExit=yes

ExecStart=/bin/bash -c "echo Oneshot service - start && sleep 60 && echo Oneshot service - end"

Запустив `systemctl status` для этого сервиса во время работы, мы можем увидеть различия:

● oneshot-remainafterexit.service - Oneshot service test with RemainAfterExit

Loaded: loaded (/etc/systemd/system/oneshot-remainafterexit.service; static; vendor preset: enabled)

Active: active (exited) since Fri 2020-06-19 20:55:14 UTC; 7s ago

Process: 1174 ExecStart=/bin/bash -c echo Oneshot service - start && sleep 60 && echo Oneshot service - end  
(code=exited, status=0/SUCCESS)

Main PID: 1174 (code=exited, status=0/SUCCESS)

Jun 19 20:54:14 thstring20200619162314 systemd[1]: Starting Oneshot service test with RemainAfterExit...

Jun 19 20:54:14 thstring20200619162314 bash[1174]: Oneshot service - start

Jun 19 20:55:14 thstring20200619162314 bash[1174]: Oneshot service - end

Jun 19 20:55:14 thstring20200619162314 systemd[1]: Started Oneshot service test with RemainAfterExit.

Заметим что сервис в `active(exited)` состоянии. вместо `inactive(dead)` (который должен быть, в случае если `RemainAfterExit` был отключен). Но если мы это хотим сохранить, что он делает на самом деле? Давайте посмотрим на пример, который использует `ExecStop` директиву.

`ExecStop` запустится когда сервис остановится.

## oneshot-execstop.service

[Unit]

Description=Oneshot service test with ExecStop

[Service]

Type=oneshot

RemainAfterExit=no

ExecStart=/bin/bash -c "echo Oneshot service - start && sleep 60 && echo Oneshot service - end"

ExecStop=/bin/bash -c "echo Oneshot service - stop"

В этом сервисе `RemainAfterExit` отключен(это по-умолчанию, но добавлен для наглядности)

● oneshot-execstop.service - Oneshot service test with ExecStop

Loaded: loaded (/etc/systemd/system/oneshot-execstop.service; static; vendor preset: enabled)

Active: inactive (dead)

Jun 19 21:04:10 thstring20200619162314 systemd[1]: Starting Oneshot service test with ExecStop...

Jun 19 21:04:10 thstring20200619162314 bash[1480]: Oneshot service - start

Jun 19 21:05:10 thstring20200619162314 bash[1480]: Oneshot service - end

Jun 19 21:05:10 thstring20200619162314 bash[1604]: Oneshot service - stop

Jun 19 21:05:10 thstring20200619162314 systemd[1]: oneshot-execstop.service: Succeeded.

Jun 19 21:05:10 thstring20200619162314 systemd[1]: Started Oneshot service test with ExecStop.

Теперь видно, что `ExecStop` запускается сразу когда `ExecStart` выполнен, так как сервис перешел в состояние `inactive(dead)`. Теперь взглянем что случится с установленным

`RemainAfterExit`:set:

# oneshot-execstop-remainafterexit.service

[Unit]

Description=Oneshot service test with ExecStop and RemainAfterExit

[Service]

Type=oneshot

RemainAfterExit=yes

ExecStart=/bin/bash -c "echo Oneshot service - start && sleep 60 && echo Oneshot service - end"

ExecStop=/bin/bash -c "echo Oneshot service - stop"

Вывод `systemctl` будет таков:

● oneshot-execstop-remainafterexit.service - Oneshot service test with ExecStop and RemainAfterExit

Loaded: loaded (/etc/systemd/system/oneshot-execstop-remainafterexit.service; static; vendor preset: enabled)

Active: active (exited) since Fri 2020-06-19 21:07:54 UTC; 8s ago

Process: 1708 ExecStart=/bin/bash -c echo Oneshot service - start && sleep 60 && echo Oneshot service - end

```
(code=exited, status=0/SUCCESS)
```

```
Main PID: 1708 (code=exited, status=0/SUCCESS)
```

```
Jun 19 21:06:54 thstring20200619162314 systemd[1]: Starting Oneshot service test with ExecStop and RemainAfterExit...
```

```
Jun 19 21:06:54 thstring20200619162314 bash[1708]: Oneshot service - start
```

```
Jun 19 21:07:54 thstring20200619162314 bash[1708]: Oneshot service - end
```

```
Jun 19 21:07:54 thstring20200619162314 systemd[1]: Started Oneshot service test with ExecStop and RemainAfterExit.
```

Так как сервис до сих пор активен(даже не смотря на то завершился его `ExecStart`), `ExecStop` до сих пор не запущен. Теперь если вы запустите `systemctl stop oneshot-execstop-remainafterexit.service`, посмотрим на вывод:

```
● oneshot-execstop-remainafterexit.service - Oneshot service test with ExecStop and RemainAfterExit
   Loaded: loaded (/etc/systemd/system/oneshot-execstop-remainafterexit.service; static; vendor preset: enabled)
   Active: inactive (dead)
```

```
Jun 19 21:06:54 thstring20200619162314 systemd[1]: Starting Oneshot service test with ExecStop and RemainAfterExit...
```

```
Jun 19 21:06:54 thstring20200619162314 bash[1708]: Oneshot service - start
```

```
Jun 19 21:07:54 thstring20200619162314 bash[1708]: Oneshot service - end
```

```
Jun 19 21:07:54 thstring20200619162314 systemd[1]: Started Oneshot service test with ExecStop and RemainAfterExit.
```

```
Jun 19 21:08:58 thstring20200619162314 systemd[1]: Stopping Oneshot service test with ExecStop and RemainAfterExit...
```

```
Jun 19 21:08:58 thstring20200619162314 bash[1900]: Oneshot service - stop
```

```
Jun 19 21:08:58 thstring20200619162314 systemd[1]: oneshot-execstop-remainafterexit.service: Succeeded.
```

```
Jun 19 21:08:58 thstring20200619162314 systemd[1]: Stopped Oneshot service test with ExecStop and RemainAfterExit.
```

Теперь видно, что `ExecStop` запущен так как сервис теперь неактивен. Это все, конечно, интересно, но `systemctl` не часто останавливает сервис. Вопрос, когда это будет полезно? Смотрим ниже...

# Запуск сервиса при ВЫКЛЮЧЕНИИ

Создавая `oneshot` сервис с `ExecStop` и `RemainAfterExit`, это лучший способ для того, чтобы запустить что-то при включении. Посмотрим как выглядит на практике:

## oneshot-execstop-remainafterexit- install.service

```
[Unit]
Description=Oneshot service test with ExecStop and RemainAfterExit

[Service]
Type=oneshot
RemainAfterExit=yes
ExecStart=/bin/bash -c "echo Oneshot service - start && sleep 60 && echo Oneshot service - end"
ExecStop=/bin/bash -c "echo Oneshot service - stop"

[Install]
WantedBy=multi-user.target
```

Затем запускаем `systemctl enable` чтобы включить сервис. Если мы запустим сервис, или перезагрузимся, то увидим:

```
● oneshot-execstop-remainafterexit-install.service - Oneshot service test with ExecStop and RemainAfterExit
   Loaded: loaded (/etc/systemd/system/oneshot-execstop-remainafterexit-install.service; enabled; vendor
  preset: enabled)
   Active: active (exited) since Fri 2020-06-19 21:14:02 UTC; 5s ago
 Main PID: 366 (code=exited, status=0/SUCCESS)
    Tasks: 0 (limit: 4087)
   Memory: 0B
    CGroup: /system.slice/oneshot-execstop-remainafterexit-install.service
```

```
Jun 19 21:13:02 thstring20200619162314 systemd[1]: Starting Oneshot service test with ExecStop and RemainAfterExit...
Jun 19 21:13:02 thstring20200619162314 bash[366]: Oneshot service - start
Jun 19 21:14:02 thstring20200619162314 bash[366]: Oneshot service - end
Jun 19 21:14:02 thstring20200619162314 systemd[1]: Started Oneshot service test with ExecStop and RemainAfterExit.
```

Как указано выше, наш `ExecStop` не запущен. Теперь перезапускаемся и смотрим на логи:

```
-- Logs begin at Fri 2020-06-19 21:14:50 UTC, end at Fri 2020-06-19 21:18:47 UTC. --
Jun 19 21:14:51 thstring20200619162314 systemd[1]: Starting Oneshot service test with ExecStop and RemainAfterExit...
Jun 19 21:14:51 thstring20200619162314 bash[337]: Oneshot service - start
Jun 19 21:15:51 thstring20200619162314 bash[337]: Oneshot service - end
Jun 19 21:15:51 thstring20200619162314 systemd[1]: Started Oneshot service test with ExecStop and RemainAfterExit.
Jun 19 21:17:48 thstring20200619162314 systemd[1]: Stopping Oneshot service test with ExecStop and RemainAfterExit...
Jun 19 21:17:48 thstring20200619162314 bash[681]: Oneshot service - stop
Jun 19 21:17:49 thstring20200619162314 systemd[1]: oneshot-execstop-remainafterexit-install.service: Succeeded.
Jun 19 21:17:49 thstring20200619162314 systemd[1]: Stopped Oneshot service test with ExecStop and RemainAfterExit.
```

Что будет если, машина была выключена в 9 часов вечера, и это приведет к остановке сервиса, который заставит выключиться машину чуть позже из-за запуска команды из `ExecStop`. Это довольно простой способ для запуска чего-то во время выключения(например процесса очистки). А что еще лучше, это то что у вас нет `ExecStart` с `oneshot` сервисом. Дальше больше.

## Множественные `ExecStarts`

Простой сервис, может только иметь один `Execstart` директиву. Но `oneshot` сервис может иметь один или больше, или вообще не иметь `ExecStart`. Если у вас нет `ExecStart`, тогда необходимо обязательно указать `ExecStop` (так же указать `RemainAfterExit`). Это будет сервис который запускается при выключении, и ни в какое другое время. Он напоминает `oneshot-execstop-remainafterexit-install.service`

но с удаленным `ExecStart`.

Как сказано выше, `oneshot` сервис может иметь множество `ExecStarts`. Выглядить это буде так:

# oneshot-multiple-execstart.service

```
[Unit]
Description=Oneshot service test with multiple ExecStart

[Service]
Type=oneshot
ExecStart=/bin/bash -c "echo First"
ExecStart=/bin/bash -c "echo Second"
ExecStart=/bin/bash -c "echo Third"
```

Как ожидали, лог будет следующим:

```
-- Logs begin at Mon 2020-06-22 13:24:01 UTC, end at Mon 2020-06-22 13:33:16 UTC. --
Jun 22 13:33:02 thstring20200622092223 systemd[1]: Starting Oneshot service test with multiple ExecStart...
Jun 22 13:33:02 thstring20200622092223 bash[1316]: First
Jun 22 13:33:02 thstring20200622092223 bash[1317]: Second
Jun 22 13:33:02 thstring20200622092223 bash[1318]: Third
Jun 22 13:33:02 thstring20200622092223 systemd[1]: oneshot-multiple-execstart.service: Succeeded.
Jun 22 13:33:02 thstring20200622092223 systemd[1]: Started Oneshot service test with multiple ExecStart.
```

Объединим цепочку в `Execstart` действия, позволит нам создать мощный рабочий процесс прямо внутри `systemd` единицы. Но что будет, если упадет один из `ExecStarts`?

# oneshot-multiple-execstart-failure.service

```
[Unit]
Description=Oneshot service test with multiple ExecStart and failure
```

```
[Service]
Type=oneshot
ExecStart=/bin/bash -c "echo First"
ExecStart=/bin/bash -c "false && echo Second"
ExecStart=/bin/bash -c "echo Third"
```

Пытаясь запустить этот сервис, мы получим следующую ошибку:

```
$ sudo systemctl start oneshot-multiple-execstart-failure.service
Job for oneshot-multiple-execstart-failure.service failed because the control process exited with error code.
See "systemctl status oneshot-multiple-execstart-failure.service" and "journalctl -xe" for details.

$ sudo journalctl -u oneshot-multiple-execstart-failure.service
-- Logs begin at Mon 2020-06-22 13:24:01 UTC, end at Mon 2020-06-22 13:37:16 UTC. --
Jun 22 13:36:53 thstring20200622092223 systemd[1]: Starting Oneshot service test with multiple ExecStart and failure...
Jun 22 13:36:53 thstring20200622092223 bash[1441]: First
Jun 22 13:36:53 thstring20200622092223 systemd[1]: oneshot-multiple-execstart-failure.service: Main process exited, code=exited, status=1/FAILURE
Jun 22 13:36:53 thstring20200622092223 systemd[1]: oneshot-multiple-execstart-failure.service: Failed with result 'exit-code'.
Jun 22 13:36:53 thstring20200622092223 systemd[1]: Failed to start Oneshot service test with multiple ExecStart and failure.
```

Сервис падает и прерывает выполнение. Но, что если вы не хотите чтобы падение остановило сервис на середине? Мы можете добавить `-` символ перед выполнением команды

# oneshot-multiple-execstart-failure-success.service

```
[Unit]
Description=Oneshot service test with multiple ExecStart and failure

[Service]
```

```
Type=oneshot
ExecStart=/bin/bash -c "echo First"
ExecStart=-/bin/bash -c "false && echo Second"
ExecStart=/bin/bash -c "echo Third"
```

Это не очевидно, но отметим, во втором `Execstart` что перед `/bin/bash` стоит `-`. Теперь посмотрим на вывод:

```
-- Logs begin at Mon 2020-06-22 13:24:01 UTC, end at Mon 2020-06-22 13:39:04 UTC. --
Jun 22 13:38:59 thstring20200622092223 systemd[1]: Starting Oneshot service test with multiple ExecStart and failure...
Jun 22 13:38:59 thstring20200622092223 bash[1553]: First
Jun 22 13:38:59 thstring20200622092223 bash[1555]: Third
Jun 22 13:38:59 thstring20200622092223 systemd[1]: oneshot-multiple-execstart-failure-success.service: Succeeded.
Jun 22 13:38:59 thstring20200622092223 systemd[1]: Started Oneshot service test with multiple ExecStart and failure.
```

Второй `ExecStart` упал как и ожидали, но это не уронило в целом сервис или остановило выполнение третьей стадии.

---

Revision #4

Created 14 December 2021 12:46:22 by gasick

Updated 16 April 2023 19:38:12 by gasick