

# InitRAMFS, Dracut и the Dracut Emergency Shell

Процесс [запуска Linux](#) проходит через несколько этапов, прежде чем достичь конечной [графической или многопользовательской цели](#). Этап `initramfs` происходит непосредственно перед монтированием корневой файловой системы. Dracut — это инструмент, который используется для управления `initramfs`. Аварийная оболочка `dracut` — это интерактивный режим, который может быть инициирован во время загрузки `initramfs`.

В этой статье будет показано, как использовать команду `dracut` для изменения `initramfs`. Также будут продемонстрированы некоторые основные команды устранения неполадок, которые можно запустить из аварийной оболочки `dracut`.

Оригинал статьи: <https://fedoramagazine.org/initramfs-dracut-and-the-dracut-emergency-shell/>

Руководство [dracut](#): <https://www.kernel.org/pub/linux/utils/boot/dracut/dracut.html>

## InitRAMFS

[Initramfs](#) означает Initial Random - Access Memory File System. В современных системах Linux он обычно хранится в файле в каталоге `/boot`. Версия ядра, для которой он был создан, будет включена в имя файла. Новый `initramfs` генерируется каждый раз при установке нового ядра.

```
$ tree -L 2 /boot
/boot
├── config-4.20.15-200.fc29.x86_64
├── config-4.20.16-200.fc29.x86_64
├── config-5.0.4-200.fc29.x86_64
├── grub2
│   ├── device.map
│   ├── fonts
│   ├── grub.cfg
│   ├── grubenv
│   ├── i386-pc
│   ├── locale
│   └── themes
├── initramfs-4.20.15-200.fc29.x86_64.img
├── initramfs-4.20.16-200.fc29.x86_64.img
├── initramfs-5.0.4-200.fc29.x86_64.img
├── loader
│   └── entries
├── System.map-4.20.15-200.fc29.x86_64
├── System.map-4.20.16-200.fc29.x86_64
├── System.map-5.0.4-200.fc29.x86_64
├── vmlinuz-4.20.15-200.fc29.x86_64
├── vmlinuz-4.20.16-200.fc29.x86_64
└── vmlinuz-5.0.4-200.fc29.x86_64

7 directories, 15 files
$ |
```

По умолчанию Fedora сохраняет предыдущие две версии ядра и связанную с ним `initramfs`. Это значение по умолчанию можно изменить, изменив значение параметра `installonly_limit` в файле `/etc/dnf/dnf.conf`.

Вы можете использовать команду `lsinitrd` для вывода списка содержимого архива `initramfs`:

```
$ lsinitrd /boot/initramfs-5.0.5-200.fc29.x86_64.img \
> | grep -o "/kernel/drivers/gpu/*.xz"
/kernel/drivers/gpu/drm
/kernel/drivers/gpu/drm/drm_kms_helper.ko.xz
/kernel/drivers/gpu/drm/drm.ko.xz
/kernel/drivers/gpu/drm/nouveau
/kernel/drivers/gpu/drm/nouveau/nouveau.ko.xz
/kernel/drivers/gpu/drm/ttm
/kernel/drivers/gpu/drm/ttm/ttm.ko.xz
$ modinfo --field description nouveau
nVidia Riva/TNT/GeForce/Quadro/Tesla/Tegra K1+
$ lspci | grep -o -i "nvidia.*"
NVIDIA Corporation G92 [GeForce 9800 GT] (rev a2)
$ lsinitrd /boot/initramfs-5.0.5-200.fc29.x86_64.img \
> | grep -o "usr/bin/*.x"
usr/bin/awk -> gawk
usr/bin/basename
usr/bin/bash
usr/bin/cat
usr/bin/chown
usr/bin/cp
usr/bin/dmesg
usr/bin/dracut-cmdline
usr/bin/dracut-cmdline-ask
usr/bin/dracut-emergency
usr/bin/dracut-initqueue
usr/bin/dracut-mount
```

На скриншоте выше показано, что мой архив `initramfs` содержит драйвер *nouveau* GPU. Команда `modinfo` сообщает мне, что драйвер *nouveau* поддерживает несколько моделей видеокарт NVIDIA. Команда `lspci` показывает, что в слоте PCI моего компьютера установлена видеокарта NVIDIA GeForce. В архив также включено несколько базовых команд Unix, таких как `cat` и `cp`.

По умолчанию архив `initramfs` включает только те драйверы, которые нужны для вашего конкретного компьютера. Это позволяет уменьшить размер архива и сократить время загрузки компьютера.

## Команда Dracut

Команда `dracut` может использоваться для изменения содержимого `initramfs`. Например, если вы собираетесь перенести жесткий диск на новый компьютер, вам может потребоваться временно включить все драйверы в `initramfs`, чтобы убедиться, что операционная система может загрузиться на новом компьютере. Для этого выполните следующую команду:

```
# dracut --force --no-hostonly
```

Параметр *force* сообщает dracut, что перезаписывать существующий архив initramfs можно. Параметр *no-hostonly* переопределяет поведение по умолчанию, включающее только драйверы, которые относятся к работающему в данный момент компьютеру, и заставляет dracut вместо этого включать все драйверы в initramfs.

По умолчанию dracut работает на initramfs для текущего работающего ядра. Вы можете использовать команду *uname* , чтобы отобразить версию ядра Linux, которая у вас запущена в данный момент:

```
$ uname -r  
5.0.5-200.fc29.x86_64
```

После установки и запуска жесткого диска на новом компьютере вы можете повторно запустить команду dracut, чтобы заново сгенерировать initramfs только с теми драйверами, которые необходимы для нового компьютера:

```
# dracut --force
```

Также есть параметры для добавления произвольных драйверов, модулей dracut и файлов в архив initramfs. Вы также можете создать файлы конфигурации для dracut и сохранить их в каталоге `/etc/dracut.conf.d`, чтобы ваши настройки автоматически применялись ко всем новым архивам initramfs, которые генерируются при установке новых ядер. Как всегда, проверяйте страницу `man` для получения подробностей, которые относятся к версии dracut, установленной на вашем компьютере:

```
$ man dracut
```

## Аварийная оболочка Dracut

```

Starting Create Volatile Files and Directories...
Starting File System Check on /dev/...dcf-ecba-42b9-988e-934d2d43a323...
[ OK ] Started Create Volatile Files and Directories.
[ OK ] Reached target System Initialization.
[ OK ] Reached target Basic System.
[ OK ] Started File System Check on /dev/d...66dcf-ecba-42b9-988e-934d2d43a323.
Mounting /sysroot...
[ 10.215405] SGI XFS with ACLs, security attributes, scrub, no debug enabled
[ 10.227823] XFS (sda3): Mounting V5 Filesystem
[ 10.584725] random: crng init done
[ 10.588137] random: 7 urandom warning(s) missed due to ratelimiting
[ 10.663509] XFS (sda3): Ending clean mount
[ OK ] Mounted /sysroot.
[ OK ] Reached target Initrd Root File System.
Starting Reload Configuration from the Real Root...
[ OK ] Started Reload Configuration from the Real Root.
[ OK ] Reached target Initrd File Systems.
[ OK ] Reached target Initrd Default Target.
Starting dracut pre-pivot and cleanup hook...
[ 10.013103] dracut-pre-pivot[426]: Warning: Break before switch_root
Starting Setup Virtual Console...
[ OK ] Started Setup Virtual Console.
Starting Dracut Emergency Shell...

Generating "/run/initramfs/rdsosreport.txt"

Entering emergency mode. Exit the shell to continue.
Type "journalctl" to view system logs.
You might want to save "/run/initramfs/rdsosreport.txt" to a USB stick or /boot
after mounting them and attach it to a bug report.

switch_root:/# |

```

Иногда что-то идет не так на этапе `initramfs` процесса загрузки вашего компьютера. Когда это происходит, вы увидите на экране сообщение «Entering emergency mode» («Вход в аварийный режим»), за которым следует приглашение оболочки. Это дает вам возможность попытаться исправить все вручную и продолжить процесс загрузки.

В качестве несколько надуманного примера предположим, что я случайно удалил важный параметр ядра в конфигурации своего загрузчика:

```
# sed -i 's/ rd.lvm.lv=fedora\/root / /' /boot/grub2/grub.cfg
```

В следующий раз, когда я перезагружу компьютер, он зависнет на несколько минут, пытаясь найти корневой раздел, а затем в конечном итоге сдастся и перейдет в аварийную оболочку.

Из аварийной оболочки я могу ввести `journalctl`, а затем использовать клавишу пробела для перехода вниз по журналам запуска. Ближе к концу журнала я вижу предупреждение, которое гласит: «**/dev/mapper/fedora-root не существует**». Затем я могу использовать команду `ls`, чтобы узнать, что существует:

```
# ls /dev/mapper
control  fedora-swap
```

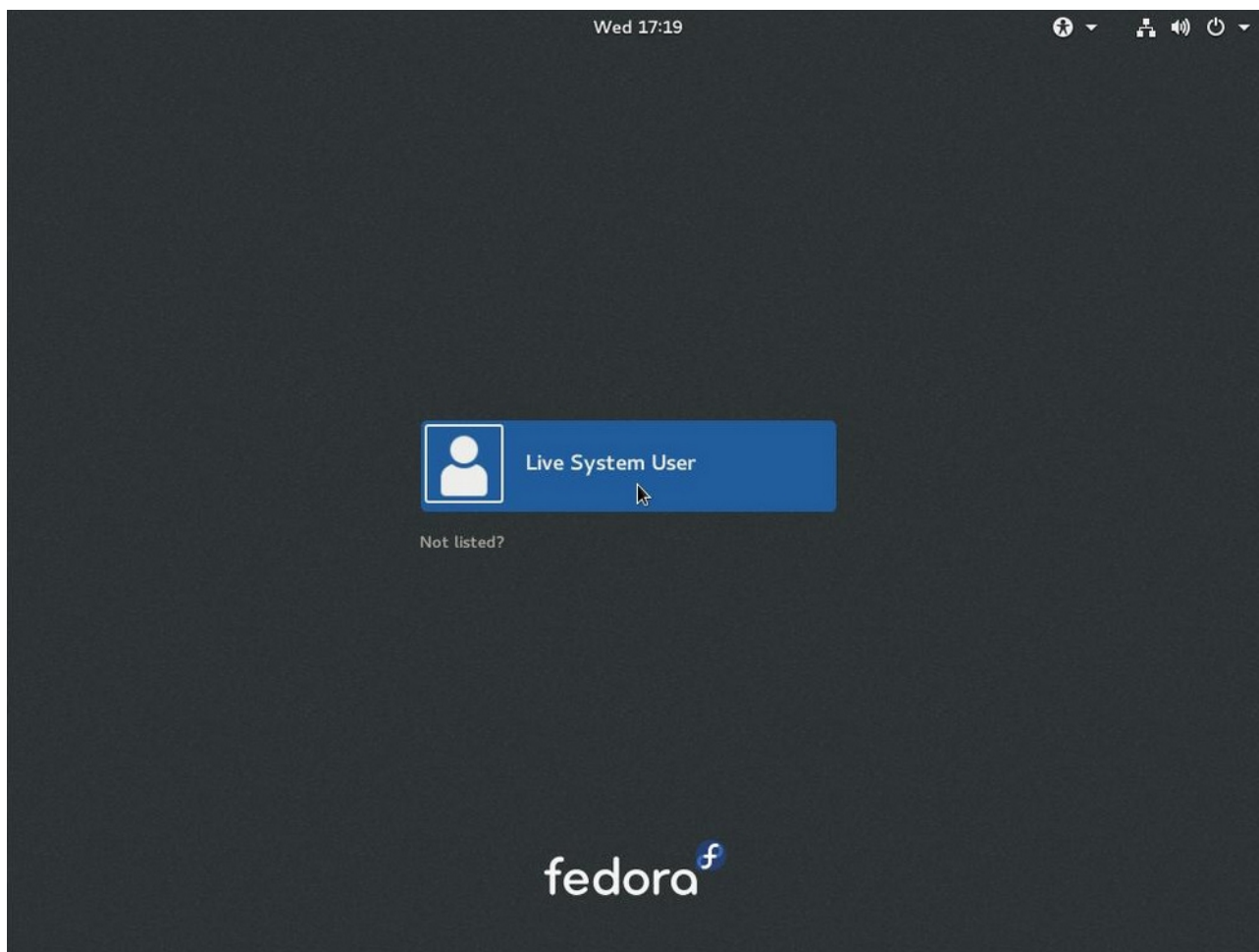
Похоже, том `fedora-root` LVM отсутствует. Давайте посмотрим, что я смогу найти с помощью команды `lv`:



```
# lvm lvscan
ACTIVE          '/dev/fedora/swap' [3.85 GiB] inherit
inactive        '/dev/fedora/home' [22.85 GiB] inherit
inactive        '/dev/fedora/root' [46.80 GiB] inherit
```

Ага! Вот мой корневой раздел. Он просто неактивен. Все, что мне нужно сделать, это активировать его и выйти из аварийной оболочки, чтобы продолжить процесс загрузки:

```
# lvm lvchange -a y fedora/root
# exit
```



Приведенный выше пример демонстрирует только базовую концепцию. Вы можете проверить [раздел устранения неполадок](#) руководства [dracut](#) для получения дополнительных примеров.

Можно получить доступ к аварийной оболочке dracut вручную, добавив параметр *rd.break* в командную строку ядра. Это может быть полезно, если вам нужно получить доступ к файлам до запуска каких-либо системных служб.

Подробную информацию о параметрах ядра, поддерживаемых вашей версией dracut, можно найти на странице руководства *dracut.kernel* :

```
$ man dracut.kernel
```

