Службы сетевого доступа к файлам (NFS)

# Служба NFS

## Общая информация

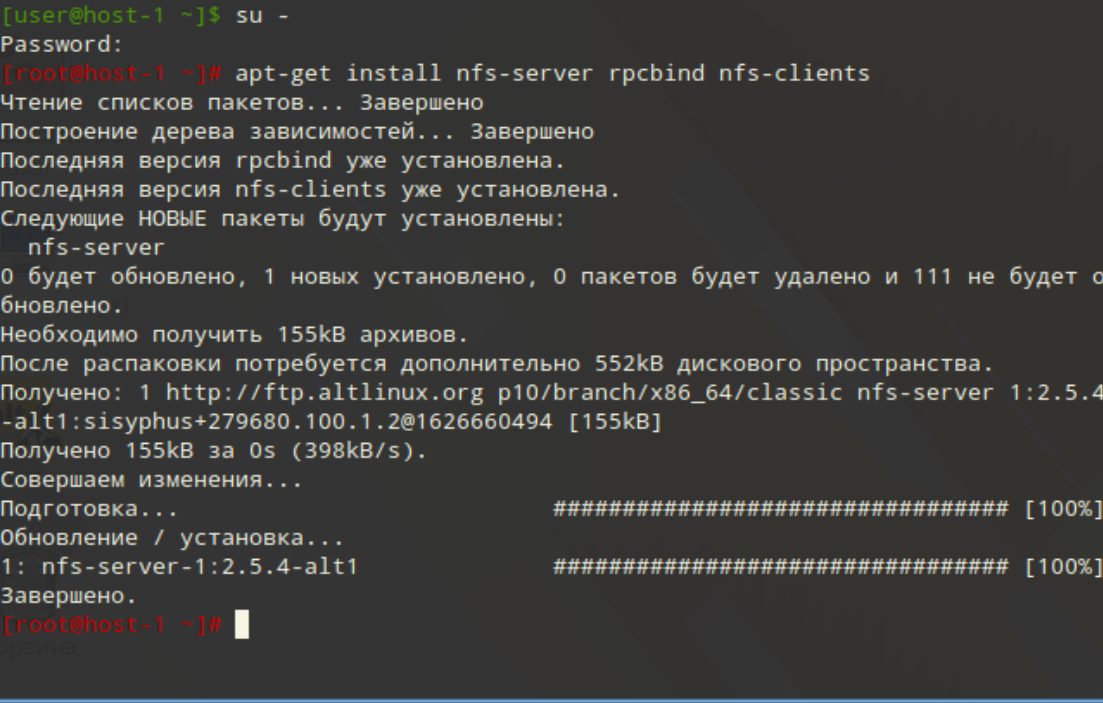
* **Network File System (NFS)** - протокол сетевого доступа к файловым ресурсам,
  + разработан компанией **Sun Microsystems** в 1984
* NFS до версии 4 **не поддерживает шифрование**, для обеспечения защищенной передачи необходимо тунеллирование (SSH, Stunnel, OpenVPN, etc)
* NFS до версии 4 **не поддерживает механизмов аутентификации** пользователей, разграничение доступа по IP/hostname (в доменном окружении умеет аутентифицировать по Kerberos)
* NFS рассчитана на использование **внутри безопасной сети**, рабочим станциям в которой можно доверять, поскольку авторизация доступа к файлам, смонтированным на NFS осуществляется на основании идентификатора пользователя, а подлинность пользователя каждая машина в сети проверяет самостоятельно
* Подразумевает, что ID пользователей и групп одинаковы на клиенте и сервере. В NFSv4 присутствует механизм **idmapping** для управления соответствием ID
* Требуется синхронизация времени

## Службы NFS

* пакет **nfs-server**, основные службы
  + **nfs** - служба nfs-сервера
* пакет **rpcbind**
  + служба **rpcbind** - отображает вызовы RPC в номера портов (TCP/UDP, 111) - необходима для работы NFS, в ОС Альт управляется через **control**
* пакет **nfs-clients**

## NFS-сервер - установка

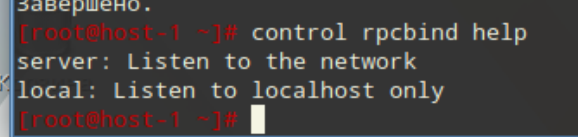
$ apt-get install nfs-server rpcbind nfs-clients



$ control rpcbind help server: Listen to the network

local: Listen to localhost only

$ control rpcbind server



$ systemctl restart rpcbind

$ systemctl enable rpcbind

$ ss -4ltnp | grep rpcbind

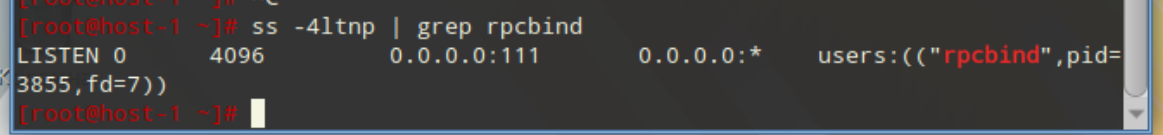
LISTEN 0 128 0.0.0.0:111 0.0.0.0:\*

users:(("rpcbind",pid=20378,fd=7))

$ ss -4lunp | grep rpcbind

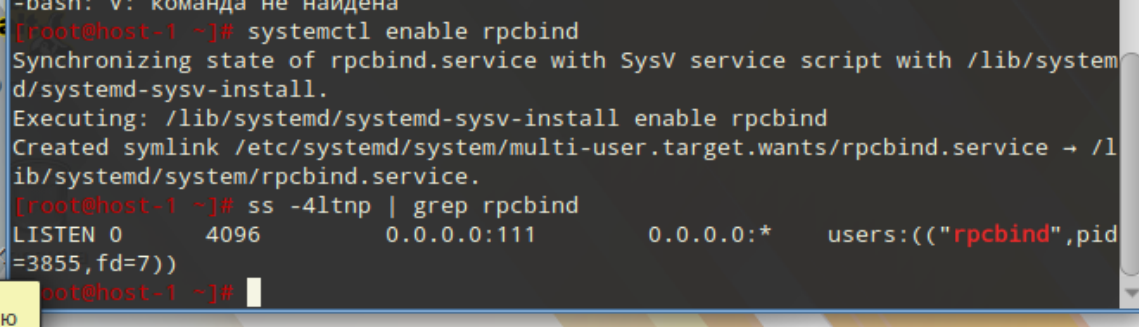
UNCONN 0 0 0.0.0.0:111 0.0.0.0:\*

users:(("rpcbind",pid=20446,fd=6))



$ systemctl start nfs

$ systemctl enable nfs



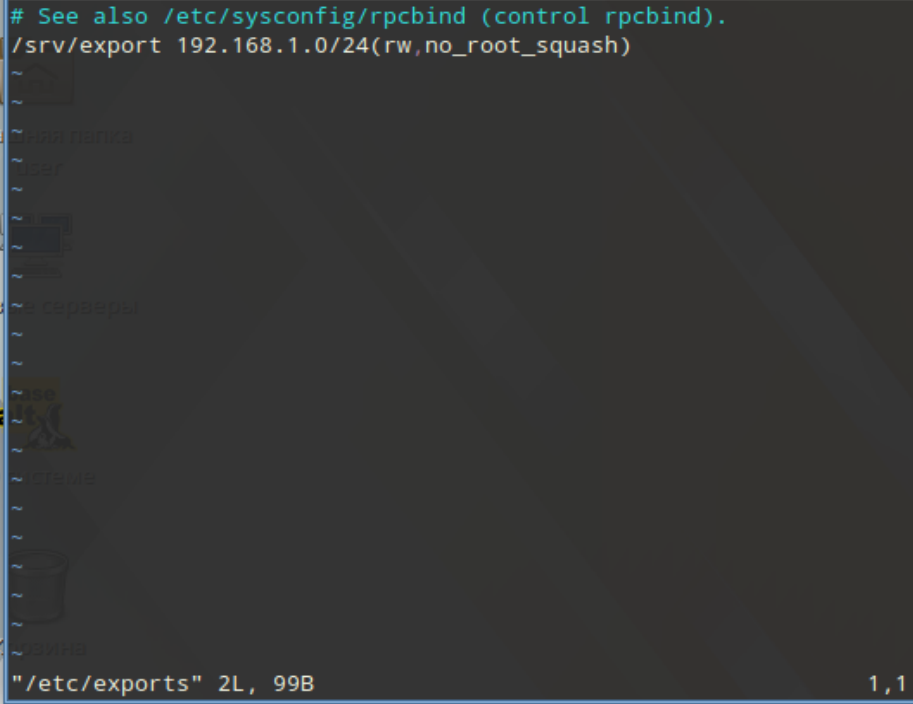
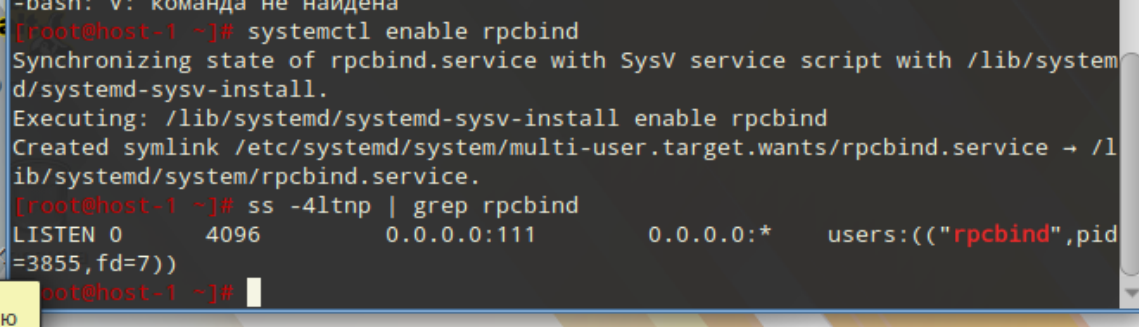
## Экспорт каталогов на сервере (/etc/exports)

<export-folder> <net>(options) <net>(options) . . .

$ mkdir /srv/export

$ echo "/srv/export 192.168.1.0/24(rw,no\_root\_squash)" >> /etc/exports

$ systemctl restart nfs



## Параметры экспорта

* **ro**

**Параметр `ro` при экспорте ресурсов NFS (Network File System) указывает, что ресурс (например, каталог или файл) должен быть доступен только для чтения (read-only). Это означает, что клиенты, подключенные к этому ресурсу, могут только читать данные, но не могут их изменять или записывать.**

**Пример использования параметра `ro` при экспорте ресурса в файле `/etc/exports`:**

**/path/to/share client\_IP(ro)**

**Здесь:**

**- `/path/to/share` - путь к каталогу или файлу, который вы хотите экспортировать.**

**- `client\_IP` - IP-адрес или диапазон IP-адресов клиентов, которым разрешен доступ к ресурсу.**

**- `(ro)` - опция, указывающая, что ресурс доступен только для чтения.**

**После внесения изменений в файл `/etc/exports` необходимо выполнить команду `exportfs -a`, чтобы применить их.**

* **rw**

**Параметр `rw` при экспорте ресурсов NFS (Network File System) указывает, что ресурс (например, каталог или файл) должен быть доступен для чтения и записи (read-write). Это означает, что клиенты, подключенные к этому ресурсу, могут как читать данные, так и изменять их.**

**Пример использования параметра `rw` при экспорте ресурса в файле `/etc/exports`:**

**/path/to/share client\_IP(rw)**

**Здесь:**

**- `/path/to/share` - путь к каталогу или файлу, который вы хотите экспортировать.**

**- `client\_IP` - IP-адрес или диапазон IP-адресов клиентов, которым разрешен доступ к ресурсу.**

**- `(rw)` - опция, указывающая, что ресурс доступен для чтения и записи.**

**После внесения изменений в файл `/etc/exports` необходимо выполнить команду `exportfs -a`, чтобы применить их.**

* **root\_squash** (по-умолчанию)

`root\_squash` - это параметр экспорта файловой системы в NFS (Network File System), который ограничивает привилегии пользователя root на клиентской стороне. Когда этот параметр включен, любые запросы, выполненные с привилегиями root на клиентской машине, будут трактоваться как запросы от анонимного пользователя (обычно `nobody` или `nfsnobody`) на серверной стороне.

Это предостережение от потенциальных уязвимостей безопасности, связанных с предоставлением полного доступа к файлам и каталогам через NFS клиентам, чей root-пользователь может быть различным от root-пользователя сервера.

Пример использования параметра `root\_squash` при экспорте ресурса в файле `/etc/exports`:

/path/to/share client\_IP(rw,root\_squash)

Здесь:

- `/path/to/share` - путь к каталогу или файлу, который вы хотите экспортировать.

- `client\_IP` - IP-адрес или диапазон IP-адресов клиентов, которым разрешен доступ к ресурсу.

- `(rw,root\_squash)` - опции, указывающие, что ресурс доступен для чтения и записи, и что привилегии root на клиентской стороне должны быть подавлены.

* **no\_root\_squash**

**`no\_root\_squash` - это противоположность параметра `root\_squash` при экспорте файловой системы в NFS (Network File System). Когда этот параметр включен, клиенты NFS могут использовать привилегии root для доступа к файлам и каталогам на сервере так же, как если бы они выполнялись на сервере.**

**Это означает, что любые запросы, выполненные с привилегиями root на клиентской машине, будут интерпретироваться как запросы от пользователя root на сервере, а не ограничиваться анонимным пользователем (обычно `nobody` или `nfsnobody`).**

**Использование `no\_root\_squash` может быть полезным в случаях, когда клиентам NFS требуется полный доступ к ресурсам на сервере, включая доступ к файлам и каталогам с привилегиями root.**

**Пример использования параметра `no\_root\_squash` при экспорте ресурса в файле `/etc/exports`:**

**/path/to/share client\_IP(rw,no\_root\_squash)**

**Здесь:**

**- `/path/to/share` - путь к каталогу или файлу, который вы хотите экспортировать.**

**- `client\_IP` - IP-адрес или диапазон IP-адресов клиентов, которым разрешен доступ к ресурсу.**

**- `(rw,no\_root\_squash)` - опции, указывающие, что ресурс доступен для чтения и записи, и что привилегии root на клиентской стороне не подавляются.**

* **subtree\_check**

**`subtree\_check` - это параметр, используемый при экспорте файловой системы в NFS (Network File System) для проверки корневого каталога поддерева.**

**Когда параметр `subtree\_check` включен, NFS-сервер проверяет, что каждый запрос клиента NFS находится в пределах экспортированного поддерева файловой системы. Это предотвращает клиентов от обращения к файлам и каталогам за пределами экспортированного поддерева.**

**Использование параметра `subtree\_check` обычно рекомендуется в целях безопасности, чтобы предотвратить возможность доступа к чужим данным на сервере NFS. Однако в некоторых случаях его можно отключить, если известно, что поддерево не содержит конфиденциальных данных и доступ к файлам за его пределами не представляет угрозу.**

**Пример использования параметра `subtree\_check` при экспорте ресурса в файле `/etc/exports`:**

**/path/to/share client\_IP(rw,subtree\_check)**

**Здесь:**

**- `/path/to/share` - путь к каталогу или файлу, который вы хотите экспортировать.**

**- `client\_IP` - IP-адрес или диапазон IP-адресов клиентов, которым разрешен доступ к ресурсу.**

**- `(rw,subtree\_check)` - опции, указывающие, что ресурс доступен для чтения и записи, и что необходимо проверять запросы клиента NFS на соответствие экспортированному поддереву файловой системы.**

* **no\_subtree\_check** (по умолчанию)

`no\_subtree\_check` - это параметр, используемый при экспорте файловой системы в NFS (Network File System), чтобы отключить проверку поддерева.

Когда параметр `no\_subtree\_check` включен, NFS-сервер не будет проверять каждый запрос клиента NFS на нахождение в пределах экспортированного поддерева файловой системы. Это может быть полезно в ситуациях, когда проверка поддерева не требуется или когда ее отключение повышает производительность сервера NFS.

Однако использование этого параметра может представлять определенные риски безопасности, поскольку клиентам будет разрешено обращаться к файлам и каталогам за пределами экспортированного поддерева.

Пример использования параметра `no\_subtree\_check` при экспорте ресурса в файле `/etc/exports`:

/path/to/share client\_IP(rw,no\_subtree\_check)

Здесь:

- `/path/to/share` - путь к каталогу или файлу, который вы хотите экспортировать.

- `client\_IP` - IP-адрес или диапазон IP-адресов клиентов, которым разрешен доступ к ресурсу.

- `(rw,no\_subtree\_check)` - опции, указывающие, что ресурс доступен для чтения и записи, и что проверка поддерева отключена.

* **sync**

**`sync` - это опция, используемая при монтировании файловых систем в Linux, которая указывает на синхронное поведение ввода-вывода.**

**Когда файловая система монтируется с опцией `sync`, операции записи на диск выполняются синхронно, что означает, что система ожидает подтверждения об успешной записи на диск перед тем, как продолжить выполнение следующих операций. Это гарантирует, что данные будут записаны на диск немедленно и без потерь в случае сбоя питания или аварии системы, но это также может привести к снижению производительности, особенно при выполнении больших объемов операций записи.**

**Пример использования опции `sync` при монтировании файловой системы в файле `/etc/fstab`:**

**/dev/sdb1 /mnt/data ext4 defaults,sync 0 2**

**Здесь:**

**- `/dev/sdb1` - устройство, которое будет монтироваться.**

**- `/mnt/data` - точка монтирования, куда будет подключаться устройство.**

**- `ext4` - тип файловой системы.**

**- `defaults,sync` - опции монтирования, включая `sync`, указывающая на синхронное поведение ввода-вывода.**

**- `0` и `2` - параметры для резервного копирования (`dump`) и проверки файловой системы (`fsck`).**

* **async**

**`async` - это опция, используемая при монтировании файловых систем в Linux, которая указывает на асинхронное поведение ввода-вывода.**

**Когда файловая система монтируется с опцией `async`, операции записи на диск выполняются асинхронно, что означает, что система не ожидает подтверждения об успешной записи на диск перед тем, как продолжить выполнение следующих операций. Это может улучшить производительность ввода-вывода, так как операции записи не блокируют выполнение других операций, но при этом существует риск потери данных в случае сбоя питания или аварии системы, так как данные могут оставаться в буферах и не записываться на диск немедленно.**

**Пример использования опции `async` при монтировании файловой системы в файле `/etc/fstab`:**

**/dev/sdb1 /mnt/data ext4 defaults,async 0 2**

**Здесь:**

**- `/dev/sdb1` - устройство, которое будет монтироваться.**

**- `/mnt/data` - точка монтирования, куда будет подключаться устройство.**

**- `ext4` - тип файловой системы.**

**- `defaults,async` - опции монтирования, включая `async`, указывающая на асинхронное поведение ввода-вывода.**

**- `0` и `2` - параметры для резервного копирования (`dump`) и проверки файловой системы (`fsck`).**

* **hide/nohide**

**Опция `hide` и `nohide` используется при монтировании файловых систем в сети (например, через NFS) и определяет видимость файлов в поддиректориях монтируемой файловой системы.**

**- `hide`: Эта опция скрывает содержимое монтируемой директории от клиента, если клиент обращается к этой директории через точку монтирования. Поддиректории будут скрыты от клиента, если они находятся на монтируемом устройстве.**

**- `nohide`: Эта опция отменяет действие `hide` и делает все поддиректории монтируемой директории видимыми для клиента.**

**Пример использования опции `hide` и `nohide` при монтировании NFS в файле `/etc/exports`:**

**/export 192.168.1.0/24(ro,hide)**

**В этом примере:**

**- `/export` - точка монтирования, которая экспортируется для доступа по сети.**

**- `192.168.1.0/24` - диапазон IP-адресов клиентов, которым разрешен доступ.**

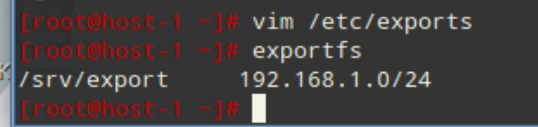
**- `ro` - опция, указывающая, что монтирование происходит только для чтения.**

**- `hide` - опция, которая скрывает содержимое точки монтирования от клиента.**

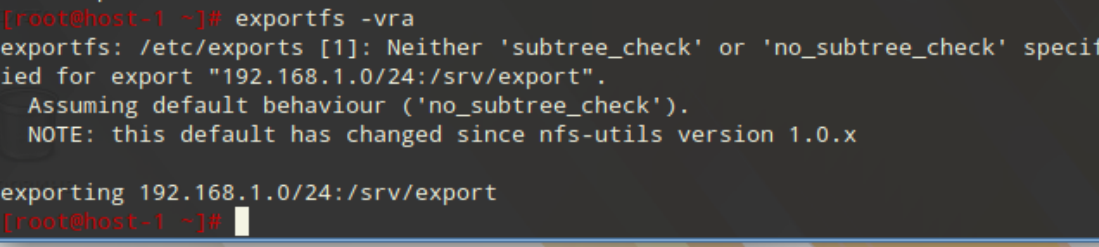
## Просмотр/применение установок

$ exportfs

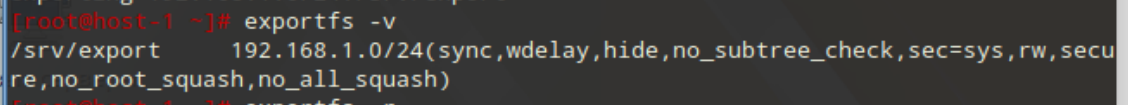
/srv/export 192.168.1.0/24



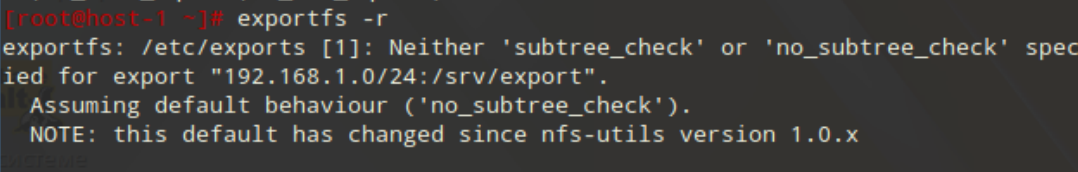
$ exportfs -vra



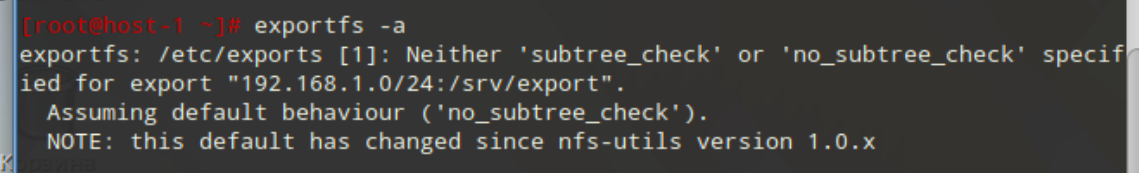
* **-v**

****

* **-r**

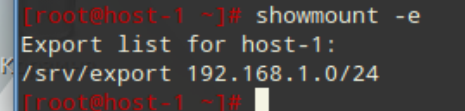
****

* **-a**

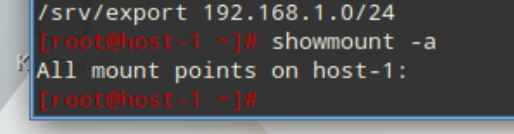
****

## Просмотр разделяемых ресурсов

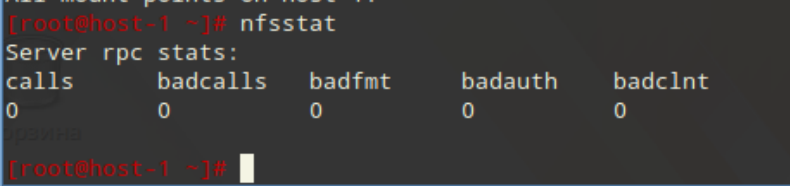
$ showmount -e <server>



$ showmount -a



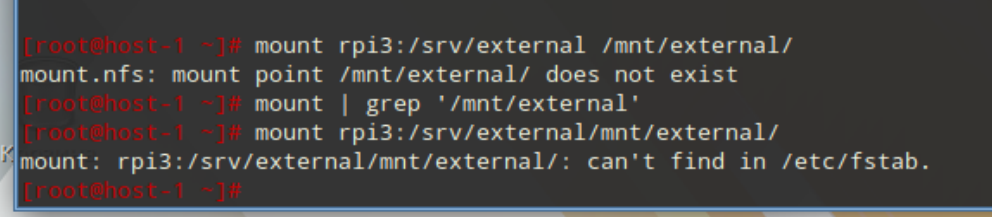
$ nfsstat



## Монтирование с клиента

$ mount rpi3:/srv/external /mnt/external/

$ mount | grep '/mnt/external'



## Автомонтирование через /etc/fstab

server:/myshare /mnt/myshare nfs soft,\_netdev 0 0

* **soft**

**В контексте настройки монтирования файловых систем в NFS (Network File System), опция `soft` определяет, что клиентские запросы на доступ к удаленным ресурсам NFS будут обрабатываться в режиме "мягкого" или "нежесткого" монтирования.**

**В режиме `soft`, если клиентский запрос на доступ к удаленному ресурсу не может быть выполнен, сервер NFS отправит уведомление клиенту о неудаче и вернет ошибку. Однако при этом сама операция не будет повторена, и клиентский процесс, запрашивающий доступ к удаленному ресурсу, может продолжить выполнение.**

**Этот режим может быть полезен в ситуациях, когда немедленное продолжение работы клиента является важным, даже если доступ к удаленному ресурсу временно недоступен. Однако это также может привести к возможной потере данных или некорректному поведению приложений, если удаленный ресурс временно становится недоступным.**

**В отличие от режима `soft`, режим `hard` (жесткое монтирование) будет продолжать пытаться выполнить операции доступа к удаленным ресурсам, даже если они временно недоступны, что может привести к "замораживанию" клиентских процессов в ожидании восстановления доступа к ресурсу.**

* **hard** (по умолчанию)

В контексте настройки монтирования файловых систем в NFS (Network File System), опция `hard` определяет, что клиентские запросы на доступ к удаленным ресурсам NFS будут обрабатываться в режиме "жесткого" монтирования.

В режиме `hard`, если клиентский запрос на доступ к удаленному ресурсу не может быть выполнен, сервер NFS будет продолжать пытаться выполнить операцию, пока не получит подтверждение о ее успешном выполнении или не произойдет тайм-аут. В этом случае, если удаленный ресурс становится недоступным, клиентский процесс, запрашивающий доступ к нему, может замереть в ожидании, пока ресурс не станет доступным снова.

Режим `hard` является режимом монтирования по умолчанию для NFS и обычно рекомендуется в ситуациях, когда надежность соединения с сервером NFS критически важна. Он обеспечивает более высокую надежность и целостность данных, поскольку клиенты будут ожидать восстановления соединения с сервером, прежде чем продолжить выполнение операций. Однако это также может привести к "замораживанию" клиентских процессов в случае временной недоступности сервера NFS.

* **bg**

**В контексте настройки монтирования файловых систем в NFS (Network File System), опция `bg` определяет, что клиентские запросы на монтирование удаленного ресурса NFS будут обрабатываться в фоновом режиме.**

**При использовании опции `bg`, если первая попытка монтирования не удалась (например, из-за временной недоступности сервера NFS), клиент будет продолжать попытки монтирования в фоновом режиме, не блокируя работу пользовательского процесса. Это позволяет пользователю продолжать работу, даже если монтирование удаленного ресурса временно не удается.**

**Однако, в случае, если монтирование не удалось и опция `bg` была использована, клиент может не получить уведомления о неудаче монтирования, и монтирование может остаться в незавершенном состоянии. Это может быть полезно в сценариях, когда немедленное монтирование не критично, и пользователь предпочитает не блокировать работу из-за временных сбоев в сети или недоступности сервера NFS.**

* **fg** (по умолчанию)

В контексте настройки монтирования файловых систем в NFS (Network File System), опция `fg` определяет, что клиентские запросы на монтирование удаленного ресурса NFS будут обрабатываться в переднем (прямом) режиме.

При использовании опции `fg`, если первая попытка монтирования не удалась (например, из-за временной недоступности сервера NFS), клиент будет ждать завершения операции монтирования и получит уведомление об успешном или неудачном завершении процесса монтирования.

Режим `fg` является режимом монтирования по умолчанию для NFS. В этом режиме операции монтирования блокируют работу пользователя, пока не завершится попытка монтирования. Это гарантирует, что пользователь будет уведомлен о любых проблемах с монтированием и сможет немедленно реагировать на них.

Выбор между опциями `fg` и `bg` зависит от требований к поведению системы в случае сбоя монтирования и предпочтений пользователей по поводу блокировки работы приложений.

* **intr/nointr**

**В контексте настройки монтирования файловых систем в NFS (Network File System), опции `intr` и `nointr` управляют поведением клиентской системы в случае разрыва соединения с сервером NFS во время выполнения операций ввода-вывода.**

**- `intr`: Если установлена опция `intr`, клиентская система будет разрешать прерывание операций ввода-вывода, ожидающих ответа от сервера NFS, если происходит разрыв соединения. Это означает, что операции ввода-вывода, которые заблокированы на сервере из-за недоступности, могут быть прерваны и завершены, позволяя пользователю продолжать работу с файловой системой.**

**- `nointr`: Если установлена опция `nointr`, клиентская система будет продолжать ожидать ответа от сервера NFS, даже если произошел разрыв соединения. Это означает, что операции ввода-вывода, которые заблокированы на сервере из-за недоступности, будут продолжать ожидать, пока соединение не будет восстановлено, что может заморозить работу пользователя с файловой системой.**

**Выбор между опциями `intr` и `nointr` зависит от предпочтений пользователя и требований к поведению системы в случае временной недоступности сервера NFS. Опция `intr` часто предпочтительна в средах, где важна отзывчивость и пользовательский опыт, так как позволяет избежать замораживания операций из-за временных сбоев в сети.**

* \_**netdev**

**Опция `\_netdev` в контексте настройки монтирования файловых систем в Linux указывает, что монтирование должно быть отложено до момента установления сетевого подключения.**

**Это особенно полезно при монтировании сетевых файловых систем, таких как NFS (Network File System), CIFS (Common Internet File System) или других, когда доступ к ресурсам находится за пределами локальной сети и требует наличия активного сетевого соединения для успешного монтирования.**

**При использовании опции `\_netdev` монтирование будет отложено до тех пор, пока сетевое подключение не станет доступным. Если сеть недоступна при загрузке системы или при запуске процедуры монтирования файловых систем, система будет ждать, пока сетевое подключение не будет установлено, прежде чем продолжить процесс монтирования.**

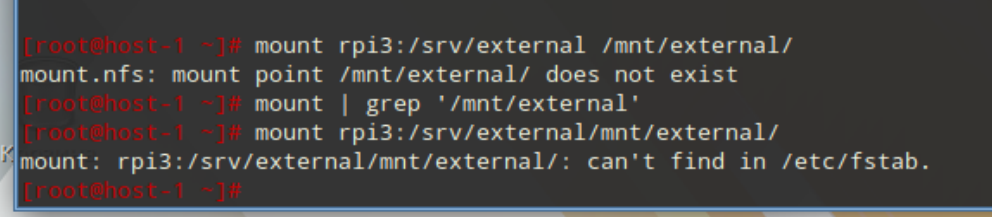
**Использование опции `\_netdev` обеспечивает более надежное и предсказуемое монтирование сетевых файловых систем, так как гарантирует, что они будут доступны только при наличии активного сетевого подключения.**

# NFS 4 - Использование и отличия

* Используется по-умолчанию
* Версии 2 и 3 поддерживаются для совместимости
* Поддерживает шифрование, аутентификацию, в том числе и доменную
* Позволяет через службу **idmapping** управлять отображением имен пользователей на клиенте и сервере
* Использует другой подход в описании разделяемых ресурсов
  + единая иерархия (дерево) ресурсов
  + а не отдельные независимо описанные ресурсы

**Анонимный NFS v4**

$ apt-get install nfs-server



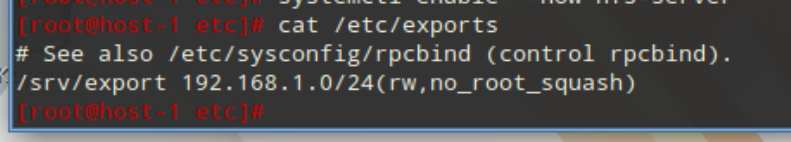
$ systemctl enable --now nfs-server



$ cat /etc/exports

/srv 10.0.0.0/24(ro,no\_subtree\_check,fsid=0)

/srv/backup/fer/pve 10.0.0.0/24(rw,no\_subtree\_check,no\_root\_squash)



$ control rpcbind server

$ systemctl restart rpcbind

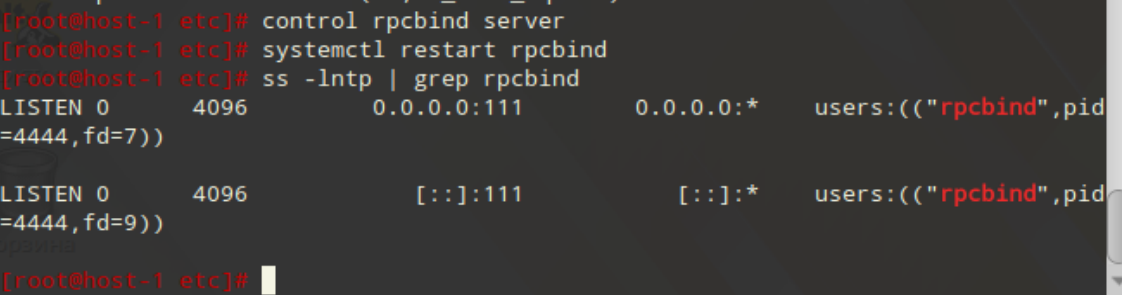
$ ss -lntp | grep rpcbind

LISTEN 0 128 0.0.0.0:111 0.0.0.0:\*

users:(("rpcbind",pid=12704,fd=7))

LISTEN 0 128 [::]:111 [::]:\*

users:(("rpcbind",pid=12704,fd=9))



\

Для настройки сервера и клиентов для работы с различными версиями NFS (NFSv3 и NFSv4) и сравнения их производительности и поддержки функций, вам нужно выполнить следующие шаги:

### Шаг 1: Настройка сервера NFS

1. \*\*Установка необходимых пакетов\*\* на сервере и клиентах:

```bash

sudo apt-get update

sudo apt-get install nfs-kernel-server nfs-common

```

2. \*\*Создание директории для экспорта\*\*:

```bash

sudo mkdir -p /export/nfsv3

sudo mkdir -p /export/nfsv4

```

3. \*\*Настройка прав доступа\*\*:

```bash

sudo chown nobody:nogroup /export/nfsv3

sudo chown nobody:nogroup /export/nfsv4

```

4. \*\*Настройка экспортов\*\* в файле `/etc/exports`:

```bash

sudo nano /etc/exports

```

Добавьте следующие строки для экспорта директорий:

```

/export/nfsv3 192.168.135.122(rw,sync,no\_subtree\_check)

/export/nfsv3 192.168.135.123(rw,sync,no\_subtree\_check)

/export/nfsv4 192.168.135.122(rw,sync,no\_subtree\_check,fsid=0)

/export/nfsv4 192.168.135.123(rw,sync,no\_subtree\_check,fsid=0)

```

5. \*\*Перезапуск службы NFS\*\*:

```bash

sudo exportfs -ra

sudo systemctl restart nfs-kernel-server

```

### Шаг 2: Настройка клиентов NFS

На каждом клиенте (192.168.135.122 и 192.168.135.123) выполните следующие шаги:

1. \*\*Установка необходимых пакетов\*\*:

```bash

sudo apt-get update

sudo apt-get install nfs-common

```

2. \*\*Создание точек монтирования\*\*:

```bash

sudo mkdir -p /mnt/nfsv3

sudo mkdir -p /mnt/nfsv4

```

3. \*\*Монтирование экспортированных директорий\*\*:

Для NFSv3:

```bash

sudo mount -t nfs -o vers=3 192.168.135.121:/export/nfsv3 /mnt/nfsv3

```

Для NFSv4:

```bash

sudo mount -t nfs -o vers=4 192.168.135.121:/export/nfsv4 /mnt/nfsv4

```

### Шаг 3: Тестирование производительности и функций

Для тестирования производительности можно использовать утилиту `dd`, `fio` или другие инструменты производительности.

Пример использования `dd` для тестирования записи и чтения:

1. \*\*Тестирование записи\*\*:

```bash

dd if=/dev/zero of=/mnt/nfsv3/testfile bs=1M count=1024 oflag=direct

dd if=/dev/zero of=/mnt/nfsv4/testfile bs=1M count=1024 oflag=direct

```

2. \*\*Тестирование чтения\*\*:

```bash

dd if=/mnt/nfsv3/testfile of=/dev/null bs=1M count=1024 iflag=direct

dd if=/mnt/nfsv4/testfile of=/dev/null bs=1M count=1024 iflag=direct

```

### Сравнение NFSv3 и NFSv4

#### Производительность

- \*\*NFSv3\*\*: Обычно быстрее в условиях высокой нагрузки из-за своей простоты и меньшего числа дополнительных функций.

- \*\*NFSv4\*\*: Включает улучшения в производительности для определённых рабочих нагрузок, особенно тех, которые выигрывают от кэширования и поддержки агрегации.

#### Поддержка функций

- \*\*NFSv3\*\*:

- Не поддерживает механизм аутентификации и авторизации (например, Kerberos).

- Отсутствие поддержки stateful соединений (состояние не отслеживается).

- Ограниченная поддержка блокировок файлов.

- \*\*NFSv4\*\*:

- Поддерживает улучшенные механизмы безопасности (Kerberos).

- Stateful соединения, что позволяет лучшую синхронизацию и управление состоянием.

- Расширенная поддержка блокировок и кэширования.

- Поддержка ACL (Access Control List).

### Заключение

NFSv4 предоставляет улучшенные возможности безопасности и управления, что делает его предпочтительным выбором для большинства современных приложений, особенно тех, которые требуют усиленных мер безопасности и сложного управления доступом. Однако, в некоторых случаях, когда важна максимальная производительность и простота, NFSv3 может быть предпочтительнее.

Для более детального сравнения можно провести дополнительные тесты с использованием утилит вроде `fio`, которые позволят более точно оценить производительность в различных сценариях использования.

\

sudo mkdir -p /srv/nfs/shared

sudo nano /etc/exports

/srv/nfs/shared 192.168.1.2(rw,sync,no\_subtree\_check)

sudo mkdir -p /mnt/nfs/shared

sudo mount -t nfs -o vers=3 192.168.1.1:/srv/nfs/shared /mnt/nfs/shared

sudo mount -t nfs 192.168.1.1:/srv/nfs/shared /mnt/nfs/shared

mount | grep nfs

Ошибка "rpc.statd is not running" указывает на то, что служба rpc.statd, отвечающая за удаленные блокировки файлов в NFS, не запущена на вашей системе сервера или на клиентской машине.

Чтобы решить эту проблему, у вас есть несколько вариантов:

### Вариант 1: Запуск службы rpc.statd на сервере

1. \*\*Запустите службы на сервере:\*\*

Убедитесь, что на сервере (192.168.1.1) запущены все необходимые службы NFS, включая rpc.statd:

```bash

sudo systemctl start nfs-server rpcbind rpc-statd

```

Чтобы убедиться, что службы запущены и работают без ошибок, используйте:

```bash

sudo systemctl status nfs-server rpcbind rpc-statd

```

Если службы не запущены, запустите их и убедитесь, что они активны и работают корректно.

2. \*\*Повторите попытку монтирования на клиенте:\*\*

После запуска служб на сервере, попробуйте снова выполнить монтирование на клиенте с указанием версии NFS:

```bash

sudo mount -t nfs -o vers=3 192.168.1.1:/srv/nfs/shared /mnt/nfs/shared1

```

### Вариант 2: Использование опции `-o nolock`

Если вы не хотите или не можете запускать rpc.statd на сервере, вы можете использовать опцию `-o nolock` для монтирования NFS с локальными блокировками. Это может быть менее безопасным и менее производительным решением, поэтому используйте его только в случае необходимости:

```bash

sudo mount -t nfs -o vers=3,nolock 192.168.1.1:/srv/nfs/shared /mnt/nfs/shared1

```

### Вариант 3: Настройка NFS без блокировок

Если ваша среда позволяет использовать NFS без блокировок, вы можете настроить сервер NFS таким образом, чтобы он предоставлял NFS без необходимости в удаленных блокировках. Это также можно сделать через опцию `-o nolock`.

### Дополнительные рекомендации

- \*\*Обновите и перезагрузите клиентскую и серверную машины:\*\* Иногда проблемы могут быть вызваны временными сетевыми или конфигурационными проблемами. Попробуйте перезапустить обе машины и повторите попытку монтирования.

- \*\*Проверьте настройки брандмауэра:\*\* Убедитесь, что брандмауэр на сервере и клиенте не блокирует необходимые порты для NFS.

Эти рекомендации помогут вам решить проблему с монтированием NFS с указанием `rpc.statd is not running`. Если проблема остается, проверьте журналы системы (`journalctl`) на обеих машинах для получения дополнительной информации об ошибке.

Да, в Alt Linux монтирование NFS происходит аналогично, как и в других дистрибутивах Linux. Основные шаги для монтирования NFS-директорий остаются такими же, но могут быть некоторые отличия в именах сервисов и пути к конфигурационным файлам. Рассмотрим настройку более подробно для Alt Linux.

### Шаг 1: Убедитесь, что все необходимые службы работают

#### На сервере (192.168.1.1)

Установите и запустите необходимые службы:

```bash

sudo apt-get update

sudo apt-get install nfs-utils

```

Проверьте и запустите службы NFS и RPC:

```bash

sudo systemctl start nfs-server rpcbind

sudo systemctl enable nfs-server rpcbind

```

#### На клиентах (192.168.1.2 и 192.168.1.3)

Установите необходимые пакеты:

```bash

sudo apt-get update

sudo apt-get install nfs-utils

```

Запустите службы RPC:

```bash

sudo systemctl start rpc-statd

sudo systemctl enable rpc-statd

```

### Шаг 2: Настройка и экспорт директорий на сервере

Создайте директорию для экспорта и настройте права:

```bash

sudo mkdir -p /srv/nfs/shared

sudo chown nobody:nogroup /srv/nfs/shared

```

Отредактируйте файл `/etc/exports`:

```bash

sudo nano /etc/exports

```

Добавьте строку для экспорта директории:

```

/srv/nfs/shared 192.168.1.2(rw,sync,no\_subtree\_check) 192.168.1.3(rw,sync,no\_subtree\_check)

```

Примените изменения:

```bash

sudo exportfs -a

```

### Шаг 3: Монтирование директорий на клиентах

#### На клиенте (192.168.1.2) для NFSv3

Убедитесь, что службы RPC работают:

```bash

sudo systemctl start rpc-statd

sudo systemctl enable rpc-statd

```

Создайте точку монтирования и смонтируйте директорию:

```bash

sudo mkdir -p /mnt/nfs/shared\_v3

sudo mount -t nfs -o vers=3 192.168.1.1:/srv/nfs/shared /mnt/nfs/shared\_v3

```

#### На клиенте (192.168.1.3) для NFSv4

Создайте точку монтирования и смонтируйте директорию:

```bash

sudo mkdir -p /mnt/nfs/shared\_v4

sudo mount -t nfs -o vers=4 192.168.1.1:/srv/nfs/shared /mnt/nfs/shared\_v4

```

### Шаг 4: Проверка работы

Проверьте, что директории смонтированы:

```bash

mount | grep nfs

```

Проведите тесты производительности:

#### Тест производительности на клиенте 192.168.1.2 (NFSv3)

```bash

cd /mnt/nfs/shared\_v3

time dd if=/dev/zero of=testfile bs=1M count=100

time dd if=testfile of=/dev/null bs=1M

```

#### Тест производительности на клиенте 192.168.1.3 (NFSv4)

```bash

cd /mnt/nfs/shared\_v4

time dd if=/dev/zero of=testfile bs=1M count=100

time dd if=testfile of=/dev/null bs=1M

```

### Дополнительные настройки

Если необходимо, можно указать дополнительные опции монтирования в `/etc/fstab` для автоматического монтирования при загрузке системы. Например:

#### Для NFSv3

Добавьте в `/etc/fstab` на клиенте 192.168.1.2:

```

192.168.1.1:/srv/nfs/shared /mnt/nfs/shared\_v3 nfs vers=3,rw,sync 0 0

```

#### Для NFSv4

Добавьте в `/etc/fstab` на клиенте 192.168.1.3:

```

192.168.1.1:/srv/nfs/shared /mnt/nfs/shared\_v4 nfs vers=4,rw,sync 0 0

```

Эти шаги помогут вам настроить и проверить работу NFSv3 и NFSv4 на вашем сервере и клиентах в Alt Linux.

Ошибка "Не хватает прав доступа" указывает на то, что у текущего пользователя нет достаточных прав для выполнения операций записи и чтения в указанной директории `/mnt/nfs/shared\_v3`.

Чтобы решить эту проблему, вам нужно изменить права доступа к этой директории. Возможно, вам понадобится выполнить эти команды с правами суперпользователя (`sudo`), особенно если текущий пользователь не обладает достаточными привилегиями.

### Изменение прав доступа

1. \*\*Проверьте текущие права доступа:\*\*

Выполните команду `ls -l` для просмотра текущих прав доступа к директории:

```bash

ls -l /mnt/nfs/shared\_v3

```

Убедитесь, что у вас есть права на запись (`w`) и чтение (`r`) в этой директории.

2. \*\*Измените права доступа с помощью `chmod`:\*\*

Если права доступа не удовлетворяют вашим требованиям, вы можете изменить их с помощью команды `chmod`. Например, чтобы предоставить полные права всем пользователям:

```bash

sudo chmod -R 777 /mnt/nfs/shared\_v3

```

Эта команда предоставит права на чтение, запись и выполнение для всех пользователей на всем содержимом директории `/mnt/nfs/shared\_v3` и её поддиректорий (`-R` означает рекурсивно).

3. \*\*Повторите команды `dd` с правами суперпользователя:\*\*

После изменения прав доступа, попробуйте выполнить вашу команду `dd` снова, используя `sudo`, если это необходимо:

```bash

sudo dd if=/dev/zero of=/mnt/nfs/shared\_v3/testfile bs=1M count=100

sudo dd if=/mnt/nfs/shared\_v3/testfile of=/dev/null bs=1M

```

### Проверка времени выполнения

Чтобы измерить время выполнения операций `dd`, вы можете использовать команду `time`, как вы указали:

```bash

time sudo dd if=/dev/zero of=/mnt/nfs/shared\_v3/testfile bs=1M count=100

time sudo dd if=/mnt/nfs/shared\_v3/testfile of=/dev/null bs=1M

```

Эти команды измерят время, затраченное на запись и чтение файла `testfile` в директории `/mnt/nfs/shared\_v3`.

### Заключение

Убедитесь, что ваш пользователь или суперпользователь имеют необходимые права доступа к директории `/mnt/nfs/shared\_v3`, чтобы успешно выполнить операции записи и чтения файлов. Если у вас остались дополнительные вопросы или проблемы, не стесняйтесь обращаться!

cd /mnt/nfs/shared\_v3

time dd if=/dev/zero of=testfile bs=1M count=100

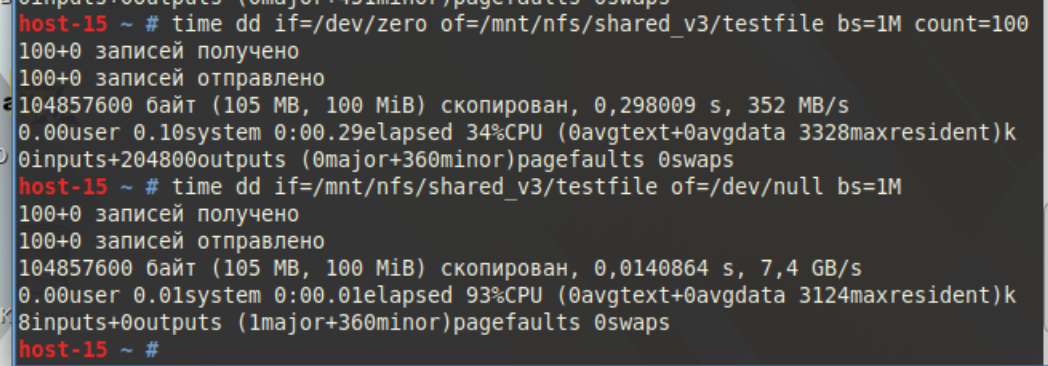
time dd if=testfile of=/dev/null bs=1M

cd /mnt/nfs/shared\_v4

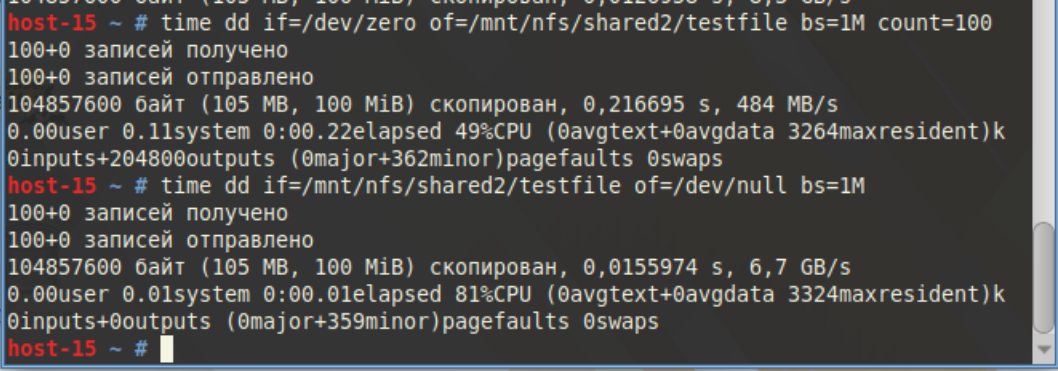
time dd if=/dev/zero of=testfile bs=1M count=100

time dd if=testfile of=/dev/null bs=1M

Nfs3



Nfs4



Команды `dd` в UNIX-подобных системах используются для копирования и преобразования файлов и потоков данных. Давайте разберемся, что делают данные команды и как они работают:

### 1. `time dd if=/dev/zero of=testfile bs=1M count=100`

Эта команда делает следующее:

- `dd`: утилита для копирования данных.

- `if=/dev/zero`: указывает `dd` использовать `/dev/zero` как источник данных. Файл `/dev/zero` в UNIX-подобных системах предоставляет бесконечный поток нулевых байтов.

- `of=testfile`: указывает `dd` записать данные в файл `testfile`.

- `bs=1M`: определяет размер блока данных для чтения и записи как 1 мегабайт (1M).

- `count=100`: определяет количество блоков данных размером `bs` для записи, в данном случае - 100 блоков по 1 мегабайту, т.е. общий размер файла будет 100 мегабайт.

Итак, команда `dd` в этом случае создает файл `testfile`, содержащий 100 мегабайт нулевых байтов.

### 2. `time dd if=testfile of=/dev/null bs=1M`

Эта команда делает следующее:

- `dd`: опять же, утилита для копирования данных.

- `if=testfile`: указывает `dd` использовать файл `testfile` как источник данных.

- `of=/dev/null`: указывает `dd` отправлять данные в `/dev/null`, специальное устройство в UNIX-подобных системах, которое игнорирует все данные, направленные в него.

- `bs=1M`: определяет размер блока данных для чтения и записи как 1 мегабайт (1M).

Итак, команда `dd` в этом случае читает данные из файла `testfile` и отправляет их в `/dev/null`, что просто измеряет скорость чтения данных из файла без фактического сохранения или обработки этих данных.

### `time` перед командами

Команда `time` используется для измерения времени, затраченного на выполнение следующей за ней команды. Она выводит три значения:

- `real`: общее время выполнения команды от начала до конца.

- `user`: время, затраченное на выполнение пользовательских процессов.

- `sys`: время, затраченное на выполнение системных процессов.

Например:

```

real 0m1.234s

user 0m0.456s

sys 0m0.789s

```

Эти значения позволяют оценить эффективность работы команды.

### Заключение

Таким образом, команды `dd` с параметром `time` позволяют создавать файлы с заданным содержимым и измерять производительность операций чтения и записи на уровне блоков данных. Это полезно для тестирования производительности хранилищ и файловых систем, а также для создания тестовых данных.