

```
// chroot
mkdir 'new_root' /// новая, пустая корневая директория (контейнер)
/// ПОДКЛЮЧЕНИЕ КОМАНД BASH
whereis bash /// узнать местоположение библиотек, для команды bash
/bin/bash /// через ссылку на /usr/bin/bash
mkdir 'new_root'/bin /// создать в новой корневой директории, директорию для
файловой оболочки
cp bin/bash 'new_root'/bin /// копировать исполняемый файл bash в файловую
оболочку
ldd /bin/bash /// узнать местоположения библиотек, необходимых для запуска bash
libtinfo.so.6 => /lib/x86_64-linux-gnu/libtinfo.so.6 (0x00007fd5dc18e000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fd5dbe00000)
/lib64/ld-linux-x86-64.so.2 (0x00007fd5dc333000)
mkdir 'lib' /// создать директории для размещения в них библиотек
mkdir 'lib64' /// создать директории для размещения в них библиотек
cp /lib/x86_64-linux-gnu/libtinfo.so.6 'new_root'/lib /// копировать, необходимые
для запуска фо, библиотеки
cp /lib/x86_64-linux-gnu/libc.so.6 'new_root'/lib /// копировать, необходимые
для запуска фо, библиотеки
cp /lib64/ld-linux-x86-64.so.2 'new_root'/lib64 /// копировать, необходимые
для запуска фо, библиотеки
/// ПОДКЛЮЧЕНИЕ КОМАНД LS
whereis ls /// узнать местоположение местоположение библиотек, для команды ls
/bin/ls /// через ссылку на /usr/bin/ls
cp bin/ls 'new_root'/bin /// копировать исполняемый файл ls в файловую оболочку
ldd /bin/ls /// узнать местоположения библиотек, необходимых для запуска ls
libselinux.so.1 => /lib/x86_64-linux-gnu/libselinux.so.1 (0x00007f61e4a11000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f61e4600000)
libpcre2-8.so.0 => /lib/x86_64-linux-gnu/libpcre2-8.so.0 (0x00007f61e497a000)
/lib64/ld-linux-x86-64.so.2 (0x00007f61e4a73000)
cp /lib/x86_64-linux-gnu/libselinux.so.1 'new_root'/lib /// копировать,
необходимые для запуска фо, библиотеки
cp /lib/x86_64-linux-gnu/libc.so.6 'new_root'/lib /// копировать,
необходимые для запуска фо, библиотеки
cp /lib/x86_64-linux-gnu/libpcre2-8.so.0 'new_root'/lib /// копировать,
необходимые для запуска фо, библиотеки
cp /lib64/ld-linux-x86-64.so.2 'new_root'/lib64 /// копировать,
необходимые для запуска фо, библиотеки
sudo chroot 'new_root' /bin/bash /// вход в фо, через bash
exit /// выход
```

```
// NAMESPACE network
# через утилиту ip
'В КОНТЕЙНЕРЕ БУДЕТ ОГРАНИЧЕНИЕ НА ИСПОЛЬЗОВАНИЕ СЕТИ'
mkdir 'net_ip' /// 'контейнер'
sudo ip netns add 'net_ip' /// добавить сетевой namespace в контейнер
sudo ip netns exec 'net_ip' bash /// запуск контейнера
ip a /// проверить доступные сетевые интерфейсы -> только один <LOOPBACK> (по
умолчанию)
exit /// выход
sudo ip link add 'veth0' type veth peer name 'veth1' /// в системе: создать и
связать между собой два виртуальных сетевых интерфейса (veth0 и veth1)
ip a /// проверить создание - два последних
sudo ip link set 'veth1' netns net_ip /// в системе: переключить veth1 к
контейнеру
ip a /// проверка что veth1 пропал в основной сети
sudo ip netns exec 'net_ip' bash /// запуск контейнера
ip a /// проверка появления veth1 в контейнере
```

```
exit /// выход
sudo ip addr add '10.0.0.1/24' dev 'veth0' /// в системе: присвоить ip (любой)
хосту veth0
ip a /// проверка появления ip для veth0
sudo ip link set dev 'veth0' up /// в системе: включить сетевой интерфейс veth0
sudo ip netns exec 'net_ip' bash /// запуск контейнера
ip addr add '10.0.0.2/24' dev 'veth1' /// в контейнере: присвоить ip (любой) хосту
veth1
ip link set dev 'veth1' up /// в контейнере: включить сетевой интерфейс veth1
ip a /// проверка включения
exit
ping '10.0.0.2' -c 3 /// проверка сигнала
```

```
// ИЗОЛЯЦИЯ ЧЕРЕЗ unshare
'ИЗОЛЯЦИЯ СРАЗУ ПО НЕСКОЛЬКИМ NAMESPACE-АМ'
man unshare /// мануал
ls -l /proc/$$/ns /// просмотр namespace-ов для текущего процесса ($$ - id
текущего процесса)
cd /proc/$$/ns /// перейти в директорию namespace-ов для текущего процесса
sudo unshare --net --user --fork --pid /bin/bash /// создать дочерний контейнер и
войти в него, добавив в его свои namespace-ы
ls -l /proc/$$/ns /// в контейнере: id namespace-ов -> будут разные
```