

Задание: необходимо продемонстрировать изоляцию одного и того же приложения (как решено на семинаре - командного интерпретатора) в различных пространствах имен.

`mkdir testfolder1` - создаем новую директорию

`mkdir testfolder1/bin` - создаем в новой директории папку `bin`

`cp /bin/bash testfolder1/bin/` - помещаем туда исполняемый файл командного интерпретатора

`ls testfolder1/bin/` - проверяем, что все получилось

```
user1@ubuntu-server:~$ mkdir testfolder1
user1@ubuntu-server:~$ mkdir testfolder1/bin
user1@ubuntu-server:~$ cp /bin/bash testfolder1/bin/
user1@ubuntu-server:~$ ls testfolder1/bin/
bash
user1@ubuntu-server:~$
```

`ldd /bin/bash` - узнаем, какие библиотеки (зависимости) нужно скопировать

`linux-vdso.so.1 (0x00007ffd4d31c000)`

`libtinfo.so.6 => /lib/x86_64-linux-gnu/libtinfo.so.6 (0x00007fc7ed7b2000)`

`libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007fc7ed7ac000)`

`libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fc7ed5ba000)`

`/lib64/ld-linux-x86-64.so.2 (0x00007fc7ed917000)`

`mkdir testfolder1/lib` - создаем директорию `lib` для библиотек

`mkdir testfolder1/lib64` - создаем директорию `lib64` для библиотек и копируем их в новую файловую систему

```
cp /lib/x86_64-linux-gnu/libtinfo.so.6 (0x00007fc7ed7b2000) testfolder1/lib
cp /lib/x86_64-linux-gnu/libdl.so.2 (0x00007fc7ed7ac000) testfolder1/lib
cp /lib/x86_64-linux-gnu/libc.so.6 (0x00007fc7ed5ba000) testfolder1/lib
cp /lib64/ld-linux-x86-64.so.2 (0x00007fc7ed917000) testfolder1/lib64
```

`$ sudo chroot testfolder1` - проверяем, запустилась ли изолированная оболочка командного интерпретатора `bash`, с корнем, отличным от остальной системы

`bash-5.1# exit` - Ок и выходим

```

user1@ubuntu-server:~$ ldd /bin/bash
        linux-vdso.so.1 (0x00007ffd4d31c000)
        libtinfo.so.6 => /lib/x86_64-linux-gnu/libtinfo.so.6 (0x00007fc7ed7b2000)
        libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007fc7ed7ac000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fc7ed5ba000)
        /lib64/ld-linux-x86-64.so.2 (0x00007fc7ed917000)
user1@ubuntu-server:~$ mkdir testfolder1/lib
user1@ubuntu-server:~$ mkdir testfolder1/lib64
user1@ubuntu-server:~$ cp /lib/x86_64-linux-gnu/libtinfo.so.6 testfolder1/lib
user1@ubuntu-server:~$ cp /lib/x86_64-linux-gnu/libdl.so.2 testfolder1/lib
user1@ubuntu-server:~$ cp /lib/x86_64-linux-gnu/libc.so.6 testfolder1/lib
user1@ubuntu-server:~$ cp /lib64/ld-linux-x86-64.so.2 testfolder1/lib64
user1@ubuntu-server:~$ sudo chroot testfolder1
[sudo] password for user1:
bash-5.0# exit
exit
user1@ubuntu-server:~$ █

```

ls /bin/ls - исполнительный файл ls находится в папки bin

cp /bin/ls testfolder1/bin/ - копируем исполнительный файл ls в изолированную среду (ls показывает содержимое)

ls testfolder1/bin/ - проверяем содержимое

ls testfolder1/lib - проверяем содержимое библиотеки lib

ls testfolder1/lib64 - проверяем содержимое библиотеки lib64

ldd /bin/ls - узнаем, какие новые библиотеки(зависимости) нужно скопировать для ls

```
linux-vdso.so.1 (0x00007ffc7578d000)
```

```
libselinux.so.1 => /lib/x86_64-linux-gnu/libselinux.so.1 (0x00007ff8032ad000)
```

```
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007ff8030bb000) - уже есть
```

```
libpcre2-8.so.0 => /lib/x86_64-linux-gnu/libpcre2-8.so.0 (0x00007ff80302a000)
```

```
libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007ff803024000) - уже есть
```

```
/lib64/ld-linux-x86-64.so.2 (0x00007ff803306000) - уже есть
```

```
libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007ff803001000)
```

копируем новые библиотеки в папку lib

```
cp /lib/x86_64-linux-gnu/libselinux.so.1 (0x00007ff8032ad000) testfolder1/lib
```

```
cp /lib/x86_64-linux-gnu/libpcre2-8.so.0 (0x00007ff80302a000) testfolder1/lib
```

```
cp /lib/x86_64-linux-gnu/libdl.so.2 (0x00007ff803024000) testfolder1/lib
```

```
cp /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007ff803001000) testfolder1/lib
```

```

user1@ubuntu-server:~$ ls /bin/ls
/bin/ls
user1@ubuntu-server:~$ cp /bin/ls testfolder1/bin/
user1@ubuntu-server:~$ ls testfolder1/bin/
bash ls
user1@ubuntu-server:~$ ls testfolder1/lib
libc.so.6 libdl.so.2 libtinfo.so.6
user1@ubuntu-server:~$ ls testfolder1/lib64
ld-linux-x86-64.so.2
user1@ubuntu-server:~$ ldd /bin/ls
        linux-vdso.so.1 (0x00007ffee67e1000)
        libselinux.so.1 => /lib/x86_64-linux-gnu/libselinux.so.1 (0x00007fd496d75000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fd496b83000)
        libpcr2-8.so.0 => /lib/x86_64-linux-gnu/libpcr2-8.so.0 (0x00007fd496af2000)
        libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007fd496aec000)
        /lib64/ld-linux-x86-64.so.2 (0x00007fd496dce000)
        libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007fd496ac9000)
user1@ubuntu-server:~$ cp /lib/x86_64-linux-gnu/libselinux.so.1 testfolder1/lib
user1@ubuntu-server:~$ cp /lib/x86_64-linux-gnu/libpcr2-8.so.0 testfolder1/lib
user1@ubuntu-server:~$ cp /lib/x86_64-linux-gnu/libpthread.so.0 testfolder1/lib
user1@ubuntu-server:~$ █

```

ls testfolder1/lib - проверяем содержимое библиотеки lib

sudo chroot testfolder1 - переходим в изолированную среду

ls - и проверяем содержимое файловой системы

ls bin/

Ls lib*

ls .. или ls / - при введение этих команд невозможно выйти, так как находимся внутри созданной изолированной среды

```

user1@ubuntu-server:~$ ls testfolder1/lib
libc.so.6 libdl.so.2 libpcr2-8.so.0 libpthread.so.0 libselinux.so.1 libtinfo.so.6
user1@ubuntu-server:~$ sudo chroot testfolder1
[sudo] password for user1:
bash-5.0# ls
bin lib lib64
bash-5.0# ls bin/
bash ls
bash-5.0# ls lib*
lib:
libc.so.6 libdl.so.2 libpcr2-8.so.0 libpthread.so.0 libselinux.so.1 libtinfo.so.6

lib64:
ld-linux-x86-64.so.2
bash-5.0# ls ..
bin lib lib64
bash-5.0# ls /
bin lib lib64
bash-5.0# █

```

Изолируем сетевой интерфейс с помощью утилиты ip

sudo ip netns add testns1 - создаем пространство имен с помощью утилиты ip netns

Ip

sudo ip netns exec testns1 bash - запускаем командную оболочку в созданном пространстве имен

ip nets list

ls

ls /

ping 127.0.0.1

ip link list

ip link set dev lo up - запускаем наш интерфейс


sudo ip link ipolist - и проверяем

```
user1@ubuntu-server:~$ sudo ip netns add testns1
user1@ubuntu-server:~$
user1@ubuntu-server:~$ sudo ip netns exec testns1 bash
root@ubuntu-server:/home/user1# ip netns list
testns1
root@ubuntu-server:/home/user1# ls
testfolder testfolder1 wordpress
root@ubuntu-server:/home/user1# ls /
bin dev home lib32 libx32 media opt root sbin srv sys usr
boot etc lib lib64 lost+found mnt proc run snap swap.img tmp var
root@ubuntu-server:/home/user1# ping 127.0.0.1
ping: connect: Network is unreachable
root@ubuntu-server:/home/user1# ip link list
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
root@ubuntu-server:/home/user1# ip link set dev lo up
root@ubuntu-server:/home/user1# sudo ip link list
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
root@ubuntu-server:/home/user1#
```

Создаем два виртуальных порта: один в родительском пространстве имен, второй в дочернем и пропингуем их (налаживаем сеть)

sudo ip link add veth0 type veth peer name veth1 - создадим в родительском пространстве имен

sudo ip link set veth1 netns testns1 - прилинковали созданный интерфейс

 user1@ubuntu-server: ~

```
user1@ubuntu-server:~$ sudo ip link add veth0 type veth peer name veth1
[sudo] password for user1:
user1@ubuntu-server:~$ ip link set veth1 netns testns1
RTNETLINK answers: Operation not permitted
user1@ubuntu-server:~$ sudo ip link set veth1 netns testns1
```

ip a - находим созданный в родительской зоне интерфейс

sudo ip addr add 10.0.0.0/24 dev veth0 - добавляем ему ip-адрес

sudo ip link set dev veth0 up - поднимаем интерфейс


```
12: veth0@veth1: <BROADCAST,MULTICAST,M-DOWN> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether ea:18:e9:2c:91:84 brd ff:ff:ff:ff:ff:ff
user1@ubuntu-server:~$ sudo ip addr add 10.0.0.1/24 dev veth0
user1@ubuntu-server:~$ sudo ip link set dev veth0 up
user1@ubuntu-server:~$
```

ip a - проверяем добавленный адрес интерфейсу

sudo ip netns exec testns1 bash - переходим в новое пространство

ip addr add 10.0.0.2/24 dev veth1 - добавляем ip-адрес второму виртуальному порту

sudo ip link set dev veth1 up - поднимаем интерфейс

```
12: veth0@veth1: <NO-CARRIER,BROADCAST,MULTICAST,UP,M-DOWN> mtu 1500 qdisc noqueue state LOWERLAYERDOWN group default qlen 1000
    link/ether ea:18:e9:2c:91:84 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.1/24 scope global veth0
        valid_lft forever preferred_lft forever
user1@ubuntu-server:~$ sudo ip netns exec testns1 bash
[sudo] password for user1:
root@ubuntu-server:/home/user1# ip addr add 10.0.0.2/24 dev veth1
root@ubuntu-server:/home/user1# sudo ip link set dev veth1 up
root@ubuntu-server:/home/user1# ip a
```

ping 127.0.0.1 - пингуем адрес родительского интерфейса

ping 127.0.0.2 - пингуем виртуальный адрес из родительского интерфейса

```
root@ubuntu-server:/home/user1# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
From 10.0.0.2 icmp_seq=1 Destination Host Unreachable
From 10.0.0.2 icmp_seq=2 Destination Host Unreachable
From 10.0.0.2 icmp_seq=3 Destination Host Unreachable
From 10.0.0.2 icmp_seq=4 Destination Host Unreachable
From 10.0.0.2 icmp_seq=5 Destination Host Unreachable
From 10.0.0.2 icmp_seq=6 Destination Host Unreachable
^C
--- 10.0.0.1 ping statistics ---
8 packets transmitted, 0 received, +6 errors, 100% packet loss, time 7195ms
pipe 4
root@ubuntu-server:/home/user1# exit
exit
user1@ubuntu-server:~$ ping 127.0.0.2
PING 127.0.0.2 (127.0.0.2) 56(84) bytes of data.
64 bytes from 127.0.0.2: icmp_seq=1 ttl=64 time=0.029 ms
64 bytes from 127.0.0.2: icmp_seq=2 ttl=64 time=0.063 ms
64 bytes from 127.0.0.2: icmp_seq=3 ttl=64 time=0.047 ms
64 bytes from 127.0.0.2: icmp_seq=4 ttl=64 time=0.063 ms
64 bytes from 127.0.0.2: icmp_seq=5 ttl=64 time=0.064 ms
^C
--- 127.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4083ms
rtt min/avg/max/mdev = 0.029/0.053/0.064/0.013 ms
user1@ubuntu-server:~$
```

Изолируем пространство имен с помощью утилиты unshare

`sudo unshare --net /bin/bash` - создаем изолированный неименованный namespace (дочерний процесс)

`exit` - при выходе неименованный namespace не сохраняется

```
user1@ubuntu-server:~$ sudo unshare --net /bin/bash
[sudo] password for user1:
root@ubuntu-server:/home/user1# ip a
1: lo: <LOOPBACK> mtu 65536 qdisc noop state DOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
root@ubuntu-server:/home/user1# exit
exit
user1@ubuntu-server:~$
```

`sudo unshare --mount --uts --ipc --net --pid --user /bin/bash` - запускаем изолированные дочерние процессы

```
user1@ubuntu-server:~$ sudo unshare --mount --uts --ipc --net --pid --user /bin/bash
bash: fork: Cannot allocate memory
nobody@ubuntu-server:/home/user1$ ls
```