

Урок 1. Механизмы пространства имен

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

```
mkdir chroot_example
cd chroot_example
mkdir bin
mkdir lib/x86_64-linux-gnu/
mkdir lib64
cp $(which ls) ~/Документы/containerization/chroot_example/bin/
cp $(which bash) ~/Документы/containerization/chroot_example/bin/
cp $(which ip) ~/Документы/containerization/chroot_example/bin/
ldd $(which ls)
ldd $(which bash)
ldd $(which ip)
cp /lib/x86_64-linux-gnu/libtinfo.so.6 ~/Документы/containerization/chroot_example/lib/x86_64-
linux-gnu/
cp /lib/x86_64-linux-gnu/libc.so.6 ~/Документы/containerization/chroot_example/lib/x86_64-
linux-gnu/
cp /lib64/ld-linux-x86-64.so.2 ~/Документы/containerization/chroot_example/lib64/
cp /lib/x86_64-linux-gnu/libselinux.so.1
~/Документы/containerization/chroot_example/lib/x86_64-linux-gnu/
cp /lib/x86_64-linux-gnu/libc.so.6 ~/Документы/containerization/chroot_example/lib/x86_64-
linux-gnu/
cp /lib/x86_64-linux-gnu/libpcre2-8.so.0
~/Документы/containerization/chroot_example/lib/x86_64-linux-gnu/
cp /lib/x86_64-linux-gnu/libdl.so.2 ~/Документы/containerization/chroot_example/lib/x86_64-
linux-gnu/
cp /lib64/ld-linux-x86-64.so.2 ~/Документы/containerization/chroot_example/lib64/
cp /lib/x86_64-linux-gnu/libelf.so.1 ~/Документы/containerization/chroot_example/lib/x86_64-
linux-gnu/
cp /lib/x86_64-linux-gnu/libmnl.so.0 ~/Документы/containerization/chroot_example/lib/x86_64-
linux-gnu/
cp /lib/x86_64-linux-gnu/libbsd.so.0 ~/Документы/containerization/chroot_example/lib/x86_64-
linux-gnu/
cp /lib/x86_64-linux-gnu/libcap.so.2 ~/Документы/containerization/chroot_example/lib/x86_64-
linux-gnu/
cp /lib/x86_64-linux-gnu/libc.so.6 ~/Документы/containerization/chroot_example/lib/x86_64-
linux-gnu/
cp /lib/x86_64-linux-gnu/libz.so.1 ~/Документы/containerization/chroot_example/lib/x86_64-
linux-gnu/
cp /lib/x86_64-linux-gnu/libmd.so.0 ~/Документы/containerization/chroot_example/lib/x86_64-
linux-gnu/
sudo chroot .
```

Типы пространств имен:

Для создания системного пространства имен используется команда unshare с ключом -U:
unshare -U

Для создания пространства имен процессов используется команда unshare с ключом -p:
unshare -p

Для создания пространства имен сетевых интерфейсов используется команда unshare с ключом -n:

```
unshare -n
```

Для создания пространства имен монтирования файловых систем используется команда unshare с ключом -m:

```
unshare -m
```

Для создания пространства имен устройств используется команда unshare с ключом -r:

```
unshare -r
```

Для создания пространства имен IPC используется команда unshare с ключом -i:

```
unshare -i
```

1. Использование контейнеров Docker:

```
docker run --name bash-container1 -it bash
```

```
docker run --name bash-container2 -it bash
```

2. Использование утилиты chroot:

```
mkdir /chroot-dir1
```

```
mkdir /chroot-dir1/bin
```

```
cp /bin/bash /chroot-dir1/bin/
```

```
chroot /chroot-dir1 /bin/bash
```

```
mkdir /chroot-dir2
```

```
mkdir /chroot-dir2/bin
```

```
cp /bin/bash /chroot-dir2/bin/
```

```
chroot /chroot-dir2 /bin/bash
```

3. Использование утилиты systemd-nspawn:

```
systemd-nspawn -D /var/lib/machines/machine1
```

```
systemd-nspawn -D /var/lib/machines/machine2
```

4. Использование утилиты LXC (Linux Containers):

```
lxc launch ubuntu:18.04 bash-container1
```

```
lxc launch ubuntu:18.04 bash-container2
```

#Назначить IP-адрес интерфейсу в пространстве имен, можно выполнить следующую команду:

```
ip netns exec <namespace> ip addr add <ip_address>/<netmask> dev <interface>
```

#Добавить маршрут в пространство имен, можно использовать команду route:

```
ip netns exec <namespace> route add -net <network_address>/<netmask> gw <gateway_address>
```