Linux chroot környezet kialakítása

A chroot környezet a minimális konténer futtató környezet bemutatására szolgál. Először a bash shell telepítése és testreszabása történik meg, majd a futtatáshoz szükséges környezetek. Végül a szükséges fájlrendszerek csatlakoztatása.

Létre kell hozni egy könyvtárat. Ez tartalmazza a program futásához szükséges könyvtárakat és fájlokat.

```
# mkdir -p /opt/program
```

A szükséges könyvtárak és szimbolikus hivatkozások létrehozása

```
# mkdir -p /opt/program/usr/bin
# ln -s usr/bin /opt/program/bin
```

A futtatandó shell program másolása a forrásrendszer könyvtárstruktúrájának megfelelően

```
# cp -a /usr/bin/bash /opt/program/usr/bin/bash
```

Chroot működésének tesztelése

```
# chroot /opt/program /usr/bin/bash
chroot: failed to run command ‘/usr/bin/bash’: No such file or directory
```

A program működéséhez szükséges megosztott objektumfájlok listázása.

```
# ldd /usr/bin/bash
    linux-vdso.so.1 (0x00007f65ff555000)
    libtinfo.so.6 => /lib64/libtinfo.so.6 (0x00007f65ff39b000)
    libc.so.6 => /lib64/libc.so.6 (0x00007f65ff1a7000)
    /lib64/ld-linux-x86-64.so.2 (0x00007f65ff557000)
```

A szükséges könyvtárstruktúra létrehozása.

```
# mkdir /opt/program/usr/lib64
# ln -s usr/lib64 /opt/program/lib64
```

Állományok másolása. Fontos megjegyezni, hogy a forrás fájlok időnként szimbolikus hivatkozások, így a hivatkozott állományokat is másolni kell.

```
# cp -a /usr/lib64/libtinfo.so.6* /opt/program/usr/lib64
# cp -a /usr/lib64/libc.so.6* /opt/program/usr/lib64
# cp -a /usr/lib64/ld-linux-x86-64.so.2* /opt/program/usr/lib64
```

A teljes struktúra a következő

```
# tree /opt/program
```

```
/opt/program
├── bin -> usr/bin
├── lib64 -> usr/lib64
└── usr
    ├── bin
    │   └── bash
    └── lib64
        ├── ld-linux-x86-64.so.2
        ├── libc.so.6
        ├── libtinfo.so.6 -> libtinfo.so.6.5
        └── libtinfo.so.6.5
```

Teszteljük a chroot működését

```
# chroot /opt/program /usr/bin/bash
bash-5.3# exit
exit
```

Próbáljuk listázni a chroot környezet tartalmát.

```
# chroot /opt/program /usr/bin/bash
# ls /
bash: ls: command not found
bash-5.3# exit
exit
```

Mivel nincs ls parancs, telepítsük a chroot könyvtárába. Itt is figyeljünk a megosztott objektumfájlokra.

```
# ldd /usr/bin/ls
    linux-vdso.so.1 (0x00007fae7efca000)
    libselinux.so.1 => /lib64/libselinux.so.1 (0x00007fae7ef5d000)
    libcap.so.2 => /lib64/libcap.so.2 (0x00007fae7ef51000)
    libc.so.6 => /lib64/libc.so.6 (0x00007fae7ed5d000)
    libpcre2-8.so.0 => /lib64/libpcre2-8.so.0 (0x00007fae7ecb0000)
    /lib64/ld-linux-x86-64.so.2 (0x00007fae7efcc000)
```

Másolási műveletek.

```
# cp -a /usr/bin/ls /opt/program/usr/bin/ls
# cp -a /usr/lib64/libselinux.so.1* /opt/program/usr/lib64
# cp -a /usr/lib64/libcap.so.2* /opt/program/usr/lib64
# cp -a /usr/lib64/libpcre2-8.so.0* /opt/program/usr/lib64
```

Ellenőrzés.

```
# tree /opt/program

# tree /opt/program
/opt/program
```

```
├── bin -> usr/bin
├── lib64 -> usr/lib64
└── usr
    ├── bin
    │   ├── bash
    │   └── ls
    └── lib64
        ├── ld-linux-x86-64.so.2
        ├── libcap.so.2 -> libcap.so.2.76
        ├── libcap.so.2.76
        ├── libc.so.6
        ├── libpcre2-8.so.0 -> libpcre2-8.so.0.15.0
        ├── libpcre2-8.so.0.15.0
        ├── libselinux.so.1
        ├── libtinfo.so.6 -> libtinfo.so.6.5
        └── libtinfo.so.6.5
```

Futtassuk a bash programot a chroot környezetben és listázzuk a fájlokat.

```
# chroot /opt/program /usr/bin/bash
bash-5.3# ls -l /
total 4
lrwxrwxrwx. 1 0 0    7 Dec  1 07:13 bin -> usr/bin
lrwxrwxrwx. 1 0 0    9 Dec  1 07:20 lib64 -> usr/lib64
drwxr-xr-x. 4 0 0 4096 Dec  1 07:20 usr
```

Az ls parancs közvetlenül hívható, nem szükséges a shell. Így működnek a konténerek is. Ott az entrypoint lesz a program ami a konténer indításakor automatikusan lefut.

```
# chroot /opt/program /usr/bin/ls -l /
total 4
lrwxrwxrwx. 1 0 0    7 Dec  1 07:13 bin -> usr/bin
lrwxrwxrwx. 1 0 0    9 Dec  1 07:20 lib64 -> usr/lib64
drwxr-xr-x. 4 0 0 4096 Dec  1 07:20 usr
```

Ha elhagyjuk a chroot második paraméterében a futtatandó programot, akkor automatikusan a /bin/bash indul.

```
# chroot /opt/program /bin/bash
bash-5.3# exit
exit

# chroot /opt/program
bash-5.3# exit
exit
```

Programok egy részének futtatásához szükség van a /proc, a /sys és néha a /dev könyvtárakra. Ezek a könyvtárak csatlakoztathatók többféle módszerrel. A könyvtárakat előre létre kell hozni a chroot környezetben.

```
# mkdir /opt/program/dev
# mkdir /opt/program/proc
# mkdir /opt/program/sys
```

A /proc fájlrendszer kezelése

A /proc fájlrendszer csatlakoztatása (1. módszer)

```
# mount -t proc proc /opt/program/proc

# chroot /opt/program /usr/bin/ls /proc
1      18846  21    2439  37    5     7369  9817       kpagecgroup
10     18950  2119  2444  3716  50    744   9818       kpagecount
...
```

A /proc fájlrendszer leválasztása.

```
# umount /opt/program/proc
```

A /proc fájlrendszer csatlakoztatása (2. módszer)

```
# mount --bind /proc /opt/program/proc

# chroot /opt/program /usr/bin/ls /proc
1      18846  21    2439  37    5     7369  9817       kpagecgroup
10     18950  2119  2444  3716  50    744   9818       kpagecount
...
```

A /proc fájlrendszer leválasztása.

```
# umount /opt/program/proc
```

A /sys fájlrendszer kezelése

A /sys fájlrendszer csatlakoztatása (1. módszer)

```
# mount -t sysfs none /opt/program/sys

# chroot /opt/program /usr/bin/ls /sys
block  class  devices   fs        kernel  power
bus    dev    firmware  hypervisor  module
```

A /sys fájlrendszer leválasztása.

```
# umount /opt/program/sys
```

A /sys fájlrendszer csatlakoztatása (2. módszer)

```
# mount --bind /sys /opt/program/sys

# chroot /opt/program /usr/bin/ls /sys
block  class  devices   fs        kernel  power
bus    dev    firmware  hypervisor  module
```

A /sys fájlrendszer leválasztása.

```
# umount /opt/program/proc
```

A /sys fájlrendszer kezelése

A /dev fájlrendszer csatlakoztatása

```
# mount --bind /dev /opt/program/dev

# chroot /opt/program /usr/bin/ls /dev
autofs       log        sgx_vepc  tty34  ttyS10   userfaultfd
block        loop-control    shm      tty35  ttyS11   v4l
...
```

A /dev fájlrendszer leválasztása.

```
# umount /opt/program/dev
```

Az elkészült chroot környezet kiegészíthető további állományokkal, programokkal.

```
# tree /opt/program/
/opt/program/
├── bin -> usr/bin
├── dev
├── lib64 -> usr/lib64
├── proc
├── sys
└── usr
    ├── bin
    │   ├── bash
    │   └── ls
    └── lib64
        ├── ld-linux-x86-64.so.2
        ├── libcap.so.2 -> libcap.so.2.76
        ├── libcap.so.2.76
        ├── libc.so.6
        ├── libpcre2-8.so.0 -> libpcre2-8.so.0.15.0
        ├── libpcre2-8.so.0.15.0
        ├── libselinux.so.1
        ├── libtinfo.so.6 -> libtinfo.so.6.5
        └── libtinfo.so.6.5
```

From:
https://wiki.r-l.hu/ - **Reverse-Logic wiki**

Permanent link:
https://wiki.r-l.hu/doku.php?id=kubernetes:chroot&rev=1764577905

Last update: **2025/12/01 08:31**