Chroot

A chroot utasítás segítségével az operációs rendszer egy könyvtárában alakítjuk ki a program futásához szükséges környezetet. Ebben a környezetben a futtatáshoz szükséges összes állományt a forrás fájlrendszer struktúráját követve adjuk meg.

Programok futtatása

Egyszerű shell futtatása

Létrehozzuk a chroot könyvtárat, majd bemásoljuk a `bash` programot. majd `chroot` paramétereként.

```
# mkdir -p /opt/chroot/01-shell-minimal

# type bash
bash: /usr/sbin/bash

# mkdir -p /opt/chroot/01-shell-minimal/usr/bin
cp -a /usr/bin/bash /opt/chroot/01-shell-minimal/usr/bin
```

Futtatjuk a programot `chroot` nélkül. Ellenőrizzük a szülő shell PID-jét, a futtatot program PID-jét, majd kilépés után újra a szülőt.

```
# echo $$
4595

# /opt/chroot/01-shell-minimal/usr/bin/bash
# echo $$
5121
# exit
exit

# echo $$
4595
```

Futtassuk a programot `chroot` segítségével. Hibát kapunk, mert a programhoz tartozó megosztott függvényeket tartalmazó állományok hiányoznak.

```
# chroot /opt/chroot/01-shell-minimal /usr/bin/bash
chroot: failed to run command ‘/usr/bin/bash’: No such file or directory
```

Ellenőrizzük mire van szükség.

```
# ldd /usr/bin/bash
    linux-vdso.so.1 (0x00007f5879c64000)
    libtinfo.so.6 => /lib64/libtinfo.so.6 (0x00007f5879aa9000)
```

```
    libc.so.6 => /lib64/libc.so.6 (0x00007f58798b6000)
    /lib64/ld-linux-x86-64.so.2 (0x00007f5879c66000)
```

A hiányzó állományok listájában három típust különböztetünk meg:

1. a programhoz linkelt állomány 'linux-vdso.so.1 (0x00007f5879c64000)' nincs vele teendő.
2. dinamikusan linkelt állomány amit a program hivatkozás segítségével ér el 'libtinfo.so.6 ⇒ /lib64/libtinfo.so.6 (0x00007f5879aa9000)'.
3. statikusan megadott állomány, amelynek helye nem módosítható '/lib64/ld-linux-x86-64.so.2 (0x00007f5879c66000)'.

Hozzuk létre a könyvtárstruktúrát és másoljuk be az állományokat. A példában a szimbolikus linkeket is létrehozzuk.

```
# mkdir /opt/chroot/01-shell-minimal/usr/lib64

# ln -s usr/bin /opt/chroot/01-shell-minimal/bin

# ln -s usr/lib64 /opt/chroot/01-shell-minimal/lib64

# cp -a /lib64/libtinfo.so.6* /lib64/libc.so.6* /lib64/ld-linux-x86-64.so.2
/opt/chroot/01-shell-minimal/usr/lib64/

# tree /opt/chroot/01-shell-minimal/
/opt/chroot/01-shell-minimal/
├── 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

6 directories, 5 files
```

Futtassuk ismét a bash programot chroot környezetben.

```
# chroot /opt/chroot/01-shell-minimal /usr/bin/bash
bash-5.3# exit
exit
```

Bővített környezet segédprogramokkal

A chroot-ban futtatott bash shell csak a beépített utasításokat képes futtatni (pwd, cd). Minden külső program (ls, cat) hiányzik. A programok futtatásához és ellenőrzéséhez szükséges **minimális** programokat be kell másolni a strukrtúrába.

```
# chroot /opt/chroot/01-shell-minimal /usr/bin/bash
bash-5.3# ls
bash: ls: command not found
bash-5.3# exit
exit
```

Készítsünk másolatot a könyvtárról és bővítsük az alapvető utasításokkal.

```
# cp -a /opt/chroot/01-shell-minimal /opt/chroot/02-shell-extended

# cp -a /usr/bin/cat /usr/bin/id /usr/bin/ls /opt/chroot/02-shell-
extended/usr/bin

# chroot /opt/chroot/02-shell-extended /usr/bin/bash
bash-5.3# ls
ls: error while loading shared libraries: libselinux.so.1: cannot open
shared object file: No such file or directory
```

Hiányzik egy újabb megosztott könyvtári függvényeket tartalmazó állomány. Át kell nézni a bemásolt programokat és be kell másolni a hiányzó állományokat.

```
# ldd /usr/bin/cat
    linux-vdso.so.1 (0x00007f6c600ce000)
    libc.so.6 => /lib64/libc.so.6 (0x00007f6c5feb9000)
    /lib64/ld-linux-x86-64.so.2 (0x00007f6c600d0000)
```

A `cat` parancs futtatásához szükséges állományok már rendelkezésre állnak.

```
# ldd /usr/bin/id
    linux-vdso.so.1 (0x00007fa0165b3000)
    libselinux.so.1 => /lib64/libselinux.so.1 (0x00007fa01655f000)
    libc.so.6 => /lib64/libc.so.6 (0x00007fa01636c000)
    libpcre2-8.so.0 => /lib64/libpcre2-8.so.0 (0x00007fa0162bf000)
    /lib64/ld-linux-x86-64.so.2 (0x00007fa0165b5000)

# ldd /usr/bin/ls
    linux-vdso.so.1 (0x00007f53cf8f6000)
    libselinux.so.1 => /lib64/libselinux.so.1 (0x00007f53cf888000)
    libcap.so.2 => /lib64/libcap.so.2 (0x00007f53cf87c000)
    libc.so.6 => /lib64/libc.so.6 (0x00007f53cf689000)
    libpcre2-8.so.0 => /lib64/libpcre2-8.so.0 (0x00007f53cf5dc000)
    /lib64/ld-linux-x86-64.so.2 (0x00007f53cf8f8000)
```

A hiányzó állományok másolását el kell végezni.

```
# cp -a /lib64/libselinux.so.1* /lib64/libcap.so.2* /lib64/libpcre2-8.so.0*
/opt/chroot/02-shell-extended/usr/lib64

# tree /opt/chroot/02-shell-extended
/opt/chroot/02-shell-extended
```

```
├── bin -> usr/bin
├── lib64 -> usr/lib64
└── usr
    ├── bin
    │   ├── bash
    │   ├── cat
    │   ├── id
    │   └── 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

6 directories, 13 files
```

Ellenőrizzük a környezetet

```
# chroot /opt/chroot/02-shell-extended /usr/bin/bash
bash-5.3# ls
bin  lib64  usr
bash-5.3# exit
exit
```

Végezzünk el néhány szépészeti beavatkozást. Létrehozzuk a `/etc` és a `/root` könyvtárakat és bemásoljuk az inicializáló állományokat.

```
# mkdir -p /opt/chroot/02-shell-extended/etc

# mkdir -p /opt/chroot/02-shell-extended/root

# cp -a /etc/skel/.bash* /opt/chroot/02-shell-extended/root/

# cp -a /etc/bashrc /opt/chroot/02-shell-extended/etc/
```

Futtassuk a `bash` utasítást a chroot környezetben. A prompt megváltozik.

```
# chroot /opt/chroot/02-shell-extended /usr/bin/bash
[I have no name!@gepnev /]# id
uid=0 gid=0 groups=0

[I have no name!@gepnev /]# exit
exit
```

Az `I have no name!` üzenet azért jelenik meg, mert a rendszer nem talál bejegyzést a futtató

felhasználóhoz és a csoporthoz. Ez a bejegyzés a `/etc/passwd` és `/etc/group` fájlokban találhatók.

```
# echo 'root::0:0:Super User:/root:/bin/bash' > /opt/chroot/02-shell-
extended/etc/passwd

# echo 'root::0:' > /opt/chroot/02-shell-extended/etc/group

# echo $$
4595

# chroot /opt/chroot/02-shell-extended /usr/bin/bash
[root@gepnev /]# echo $$
9737
[root@gepnev /]# exit
exit

# echo $$
4595
```

NGINX szerver futtatása chroot környezetben

A legegszerűbb megoldásként az operációs rendszer csomagjait használjuk az NGINX szerver környezet kialakításához.

Töltsük le a szükséges csomagokat.

```
# dnf download nginx-core nginx-filesystem
Last metadata expiration check: 3:03:15 ago on Tue 14 Apr 2026 06:45:25 AM
CEST.
(1/2): nginx-filesystem-1.20.1-24.0.1.el9_7.2.n  53 kB/s |  10 kB     00:00
(2/2): nginx-core-1.20.1-24.0.1.el9_7.2.x86_64. 1.9 MB/s | 589 kB     00:00
```

Átmeneti könyvtárba csomagoljuk ki az állományokat

```
# rm -rf /tmp/nginx && mkdir /tmp/nginx && cd /tmp/nginx && rpm2cpio
/root/nginx-core*.rpm | cpio -idmu && rpm2cpio /root/nginx-filesystem*.rpm |
cpio -idmu
3287 blokk
3 blokk
```

Hozzuk létre az új NGINX szervert futtató chroot környezetet a kibővített shell futtató környezetből.

```
# cp -a /opt/chroot/02-shell-extended/ /opt/chroot/03-nginx-minimal

# mkdir /opt/chroot/03-nginx-minimal/usr/sbin

# ln -s usr/sbin /opt/chroot/03-nginx-minimal/sbin
```

Másoljuk be a szükséges bináris állományt és a megosztott függvényeket tartalmazó állományokat.

```
# cp -a /tmp/nginx/usr/sbin/nginx /opt/chroot/03-nginx-minimal/usr/sbin/

# ldd /tmp/nginx/usr/sbin/nginx
    linux-vdso.so.1 (0x00007f5ffd906000)
    libcrypt.so.2 => /lib64/libcrypt.so.2 (0x00007f5ffd761000)
    libpcre.so.1 => /lib64/libpcre.so.1 (0x00007f5ffd6e9000)
    libssl.so.3 => /lib64/libssl.so.3 (0x00007f5ffd5f0000)
    libcrypto.so.3 => /lib64/libcrypto.so.3 (0x00007f5ffd000000)
    libz.so.1 => /lib64/libz.so.1 (0x00007f5ffd5d4000)
    libc.so.6 => /lib64/libc.so.6 (0x00007f5ffcc00000)
    /lib64/ld-linux-x86-64.so.2 (0x00007f5ffd908000)

# cp -a /lib64/libcrypt.so.2* /lib64/libpcre.so.1* /lib64/libssl.so.3*
/lib64/libcrypto.so.3* /lib64/libz.so.1* /opt/chroot/03-nginx-
minimal/usr/lib64
```

Teszteljük a program indítását és a megjelenő hibáknál végezzük el a javítási műveleteket.

```
# chroot /opt/chroot/03-nginx-minimal/ /usr/bin/bash

[root@gepnev]# /usr/sbin/nginx
nginx: [alert] could not open error log file: open()
"/var/log/nginx/error.log" failed (2: No such file or directory)
2026/04/14 07:57:03 [emerg] 488951#488951: open() "/etc/nginx/nginx.conf"
failed (2: No such file or directory)

[root@gepnev]# exit
exit
```

Hozzuk létre a könyvtárakat és másoljuk be a megfelelő állományokat és próbáljuk újra.

```
# mkdir /opt/chroot/03-nginx-minimal/etc/nginx

# cp -a /tmp/nginx/etc/nginx/nginx.conf /tmp/nginx/etc/nginx/mime.types
/opt/chroot/03-nginx-minimal/etc/nginx/

# mkdir -p /opt/chroot/03-nginx-minimal/var/log/nginx

# /usr/sbin/nginx
nginx: [emerg] getpwnam("nginx") failed in /etc/nginx/nginx.conf:5
```

A nginx szerver futtatásához szükséges bejegyzés hiányzik a `/etc/passwd` és a `/etc/group` fájlokban.

```
# echo 'nginx::101:101::/var/lib/nginx:/sbin/nologin' >> /opt/chroot/03-
nginx-minimal/etc/passwd

# echo 'nginx::101:' >> /opt/chroot/03-nginx-minimal/etc/group

# mkdir -p  /opt/chroot/03-nginx-minimal/var/lib/nginx/tmp
```

```
# chroot /opt/chroot/03-nginx-minimal/ /usr/bin/bash

# /usr/sbin/nginx
nginx: [emerg] open() "/run/nginx.pid" failed (2: No such file or directory)
nginx: [emerg] open("/dev/null") failed (2: No such file or directory)

# exit
exit
```

Hozzuk létre a hiányzó könyvtárakat és másoljuk be, vagy csatlakoztassuk a speciális `/dev/null` állományt.

```
# mkdir /opt/chroot/03-nginx-minimal/dev

# mkdir /opt/chroot/03-nginx-minimal/run

# cp -a /dev/null /opt/chroot/03-nginx-minimal/dev/

# chroot /opt/chroot/03-nginx-minimal/ /usr/bin/bash

# /usr/sbin/nginx

# exit
exit
```

A program fut, ami ellenőrizhető a futtató környezetből.

```
# ps -ef | grep -v grep | grep nginx
root      493299       1  0 10:14 ?        00:00:00 nginx: master process
/usr/sbin/nginx
101       493300  493299  0 10:14 ?        00:00:00 nginx: worker process
101       493301  493299  0 10:14 ?        00:00:00 nginx: worker process
```

Állítsuk le a megfelelő jelzés segítségével.

```
# kill -2 493299
```

Készítsük el a teljes futtatói környezetet. Létrehozzuk a `/proc` és a `/sys` könyvtárakat. Egyszerűsítjük az nginx szerver konfigurációját '/etc/nginx/nginx.conf'. Létrehozzuk az index.html állományt egyszerű tartalommal.

```
# cp -a /opt/chroot/03-nginx-minimal /opt/chroot/04-nginx-extended/

# cat > /opt/chroot/04-nginx-extended/etc/nginx/nginx.conf <<'EOF'
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

events {
```

```
    worker_connections 1024;
}

http {
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                    '$status $body_bytes_sent "$http_referer" '
                    '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 4096;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    server {
        listen 80;
        listen [::]:80;
        server_name _;
        root /usr/share/nginx/html;
    }
}
EOF

# mkdir -m 0555 /opt/chroot/04-nginx-extended/proc

# mkdir -m 0555 /opt/chroot/04-nginx-extended/sys

# mkdir -p /opt/chroot/04-nginx-extended/usr/share/nginx/html

# echo 'Üdvözöl az NGINX szerver' > /opt/chroot/04-nginx-
extended/usr/share/nginx/html/index.html

# chroot /opt/chroot/04-nginx-extended/ /usr/bin/bash

# /usr/sbin/nginx

# exit
```

Ellenőrizzük a futást.

```
# curl http://127.0.0.1
Üdvözöl az NGINX szerver
```

Állítsuk le a szervert és takarítsuk ki a felesleges állományokat.

```
# ps -ef | grep nginx
```

```
root      498367       1  0 10:34 ?        00:00:00 nginx: master process
/usr/sbin/nginx
101       498368  498367  0 10:34 ?        00:00:00 nginx: worker process
101       498369  498367  0 10:34 ?        00:00:00 nginx: worker process
root      500487  496201  0 10:42 pts/0    00:00:00 grep --color=auto nginx

# kill -2 498367

# ps -ef | grep nginx
root      500579  496201  0 10:43 pts/0    00:00:00 grep --color=auto nginx

# cd /opt/chroot/04-nginx-extended/

# rm -rf root/.bash_history run/nginx.pid var/lib/nginx/tmp/*
var/log/nginx/*
```

Hozzuk létre a végleges könyvtárat és tegyük hozzá a konténer működéshez szükséges állományokat.

```
# cp -a /opt/chroot/04-nginx-extended /opt/chroot/05-nginx-final

# ln -s /dev/stdout /opt/chroot/05-nginx-final/var/log/nginx/access.log

# ln -s /dev/stderr /opt/chroot/05-nginx-final/var/log/nginx/error.log

# tree /opt/chroot/05-nginx-final/
/opt/chroot/05-nginx-final/
├── bin -> usr/bin
├── dev
│   └── null
├── etc
│   ├── bashrc
│   ├── group
│   ├── nginx
│   │   ├── mime.types
│   │   └── nginx.conf
│   └── passwd
├── lib64 -> usr/lib64
├── proc
├── root
├── run
├── sbin -> usr/sbin
├── sys
├── usr
│   ├── bin
│   │   ├── bash
│   │   ├── cat
│   │   ├── id
│   │   └── ls
│   ├── lib64
│   │   ├── ld-linux-x86-64.so.2
│   │   ├── libcap.so.2 -> libcap.so.2.48
```

```
│   │   ├── libcap.so.2.48
│   │   ├── libcrypto.so.3 -> libcrypto.so.3.5.1
│   │   ├── libcrypto.so.3.5.1
│   │   ├── libcrypt.so.2 -> libcrypt.so.2.0.0
│   │   ├── libcrypt.so.2.0.0
│   │   ├── libc.so.6
│   │   ├── libpcre2-8.so.0 -> libpcre2-8.so.0.11.0
│   │   ├── libpcre2-8.so.0.11.0
│   │   ├── libpcre.so.1 -> libpcre.so.1.2.12
│   │   ├── libpcre.so.1.2.12
│   │   ├── libselinux.so.1
│   │   ├── libssl.so.3 -> libssl.so.3.5.1
│   │   ├── libssl.so.3.5.1
│   │   ├── libtinfo.so.6 -> libtinfo.so.6.2
│   │   ├── libtinfo.so.6.2
│   │   ├── libz.so.1 -> libz.so.1.2.11
│   │   └── libz.so.1.2.11
│   ├── sbin
│   │   └── nginx
│   └── share
│       └── nginx
│           └── html
│               └── index.html
└── var
    ├── lib
    │   └── nginx
    │       └── tmp
    └── log
        └── nginx
            ├── access.log -> /dev/stdout
            └── error.log -> /dev/stderr

23 directories, 33 files
```

Konténer image létrehozása

Létrehozzuk a konténer készítéséhez szükséges könyvtárat. A `tar` program segítségével a végleges fájlrendszer struktúrát összecsomagoljuk.

```
# mkdir /opt/nginx-kontener

# cd /opt/chroot/05-nginx-final && tar -czf /opt/nginx-
kontener/rootfs.tar.gz .

# ls /opt/nginx-kontener/
rootfs.tar.gz

# cat > /opt/nginx-kontener/Dockerfile <<'EOF'
FROM scratch
```

```
ADD rootfs.tar.gz /

CMD ["/usr/sbin/nginx", "-g", "daemon off;"]
EOF
```

Futtassuk a build folyamatot.

```
# cd /opt/nginx-kontener/

# podman build -t custom-nginx:1.20 .
STEP 1/3: FROM scratch
STEP 2/3: ADD rootfs.tar.gz /
--> 24bfe43d6b1e
STEP 3/3: CMD ["/usr/sbin/nginx", "-g", "daemon off;"]
COMMIT custom-nginx:1.20
--> 39f8c1b752ea
Successfully tagged localhost/custom-nginx:1.20
39f8c1b752ea12cb5fa8331a9a1ea87b27941e0b45b21696eeb111582f123d51

# podman image ls
REPOSITORY                          TAG         IMAGE ID      CREATED
SIZE
localhost/custom-nginx              1.20        39f8c1b752ea  24 seconds ago
15.5 MB
```

Indítsuk és teszteljük a konténert.

```
# podman run -d --rm --name nginx-test -p 80:80 custom-nginx:1.20
28d56790d6c596cdd1a192fc7253c909f624d3aed9d7f96ac917d41ea4cedd7c

# podman ps
CONTAINER ID  IMAGE                           COMMAND               CREATED
STATUS             PORTS                 NAMES
28d56790d6c5  localhost/custom-nginx:1.20  /usr/sbin/nginx -...  About a
minute ago  Up About a minute  0.0.0.0:80->80/tcp  nginx-test

# curl http://127.0.0.1
Üdvözöl az NGINX szerver
```

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

Permanent link:
https://wiki.r-l.hu/doku.php?id=podman:chroot

Last update: **2026/04/14 09:29**