

Linkelés Linuxban Átírányítások és parancsok láncolása Csatornák

Linux lapok
Varga Tibi 2019

Link

A linkelés célja hogy egy állományt több helyen/néven is elérhetővé tegyünk a fájlrendszerben

\$ ln [kapcsoló] <forrás> <cél>

Az első paraméter a fájl, amihez szeretnénk linket készíteni, a második paraméter a link neve.

Kapcsoló nélkül hard link készítése.

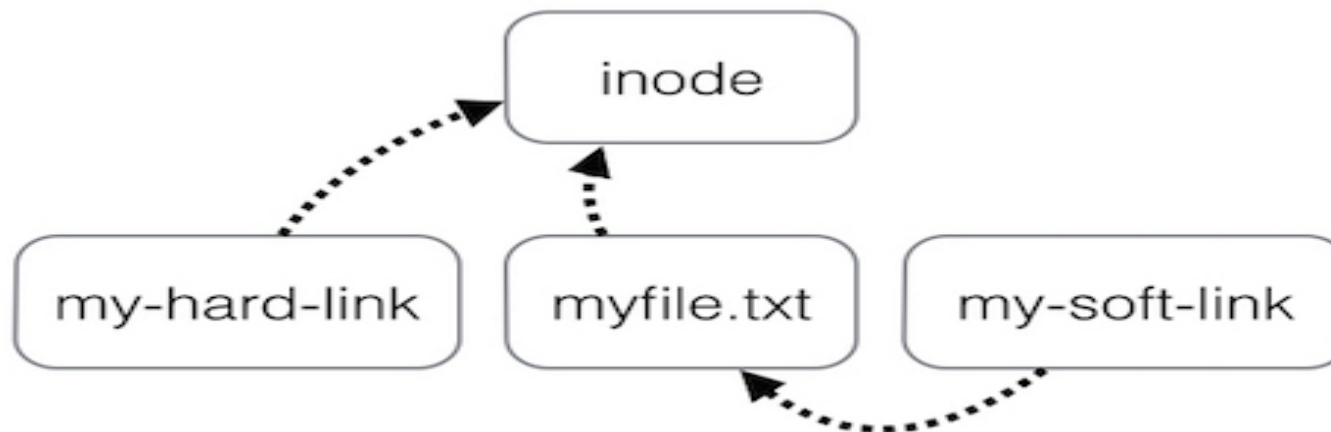
Példa egy link készítésre: **valami.txt** fájlról link készítése a **home/felhasznalo** könyvtarba **valami_masolat.lnk** néven:

ln /home/valami.txt /home/felhasznalo/valami_masolat.lnk

Hard link

\$ ln <forrás> <cél>

- Tulajdonképpen egy fájl több névvel, ugyanarra az inodera mutat, bármelyik változik akkor egyszerre változik az összes
- **Csak azonos partícióban készíthető!**
- eredeti törlése esetén megmarad és **működik a hard link**



Soft vagy symbolic link:

- `$ ln -s <forrás> <cél>`
- hagyományos értelemben egyszerű link (tükör) jön létre, ami az eredeti állományra mutat – **hagyományos link**
- eredeti törlése esetén megmarad a link, **de nem működik többet**

```
tibi@server:~$ cat > feladat1.txt
hello link
^C
tibi@server:~$ ln -s feladat1.txt feladat2.txt
tibi@server:~$
tibi@server:~$ cat < feladat2.txt
hello link
tibi@server:~$ cat >> feladat1.txt
ez itt egy szoft link
^C
tibi@server:~$ cat < feladat1.txt
hello link
ez itt egy szoft link
tibi@server:~$ cat < feladat2.txt
hello link
ez itt egy szoft link
tibi@server:~$ rm feladat1.txt
tibi@server:~$ cat < feladat2.txt
-bash: feladat2.txt: Nincs ilyen fájl vagy könyvtár
```

Symbolic link láthatósága

ls -l listázás parancsnál a tulajdonságok „l” betűvel kezdődnek ebből látható hogy a fájl **symbolic link**

```
root@kali:/home# ls -al
```

```
összesen 16
```

```
drwxr-xr-x  3 root root 4096 nov  15 15:04 .
```

```
drwxr-xr-x 19 root root 4096 szept 23 18:30 ..
```

```
-rw-r--r--  2 root root  21 nov  15 15:00 test1link.lnk
```

```
lrwxrwxrwx  1 root root   9 nov  15 15:04 test1simboliclink.lnk -> test1.txt
```

```
drwxr-xr-x  2 root root 4096 nov  15 15:00 tibi
```

```
root@kali:/home#
```

Átírányítások

parancs > szövegfájl

- A parancs kimenete a konzol helyett a fájlba fog íródni, a fájl addigi tartalma felülírodik.

```
tibi@server:~$ cat > file.txt
ls -l > file.txt
q^C
tibi@server:~$ ls -l > file.txt
tibi@server:~$ cat < file.txt
Összesen 44
-rw-rw-r-- 1 tibi tibi 0 máj 22 15:49 akarmi.txt
drwxr-xr-x 2 tibi tibi 4096 máj 15 08:06 Asztal
drwxr-xr-x 2 tibi tibi 4096 máj 15 08:06 Dokumentumok
lrwxrwxrwx 1 tibi tibi 12 okt 23 20:19 feladat2.txt -> feladat1.txt
-rw-rw-r-- 1 tibi tibi 22 okt 23 20:32 feladat4.txt
-rw-rw-r-- 1 tibi tibi 0 okt 23 20:48 file.txt
drwxrwxr-x 4 tibi tibi 4096 okt 10 11:43 gyakorlas
-rw-r--r-- 1 root root 0 okt 11 20:47 iphone
drwxr-xr-x 4 tibi tibi 4096 okt 10 03:56 Képek
drwxr-xr-x 2 tibi tibi 4096 máj 15 08:06 Letöltések
-rw-rw-r-- 1 tibi tibi 0 okt 10 11:48 masodik
drwxr-xr-x 2 tibi tibi 4096 máj 15 08:06 Nyilvános
drwxr-xr-x 2 tibi tibi 4096 máj 15 08:06 Sablonok
drwxr-xr-x 3 tibi tibi 4096 máj 15 08:14 snap
drwxr-xr-x 2 tibi tibi 4096 máj 15 08:06 Videók
drwxr-xr-x 2 tibi tibi 4096 máj 15 08:06 Zenék
```

\$ parancs >> szövegfájl

- A parancs kimenete a konzol helyett a szövegfájl végéhez fog hozzáfűződni.
- pl.: Dokumentumok \$ ls >> file.txt

\$ parancs < szövegfájl

- A parancs bemeneteként kapja a szövegfájl tartalmát.

Átírányítható a szabványos kimenetet nemcsak fájlokra, hanem eszközökre is!

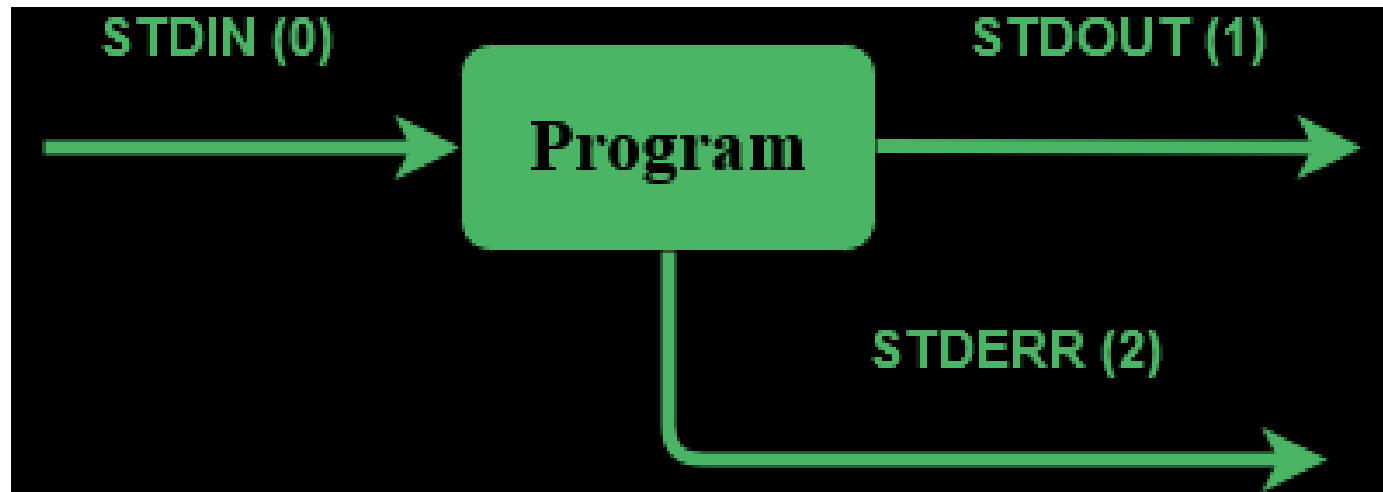
\$ cat music.mp3> / dev / audio

A cat parancs elolvassa a music.mp3 fájlt, és elküldi a kimenetet a / dev / audio címre, amely az audio eszköz. Ha a számítógép hangkonfigurációi megfelelőek, akkor ez a parancs a music.mp3 fájlt lejátssza.

Csatornák (streamek)

3 standard be- és kimeneti csatorna létezik, rendre a 0, 1 és 2 számokkal hivatkozhatóak:

- **Stdin (0)** : bemenet alapértelmezetten a billentyűzet
- **Stdout (1)** : kiment alapértelmezetten a képernyő (terminál)
- **Stderr (2)** : hiba kimenet alapértelmezetten a képernyő (terminál)



Csatornák (streamek)

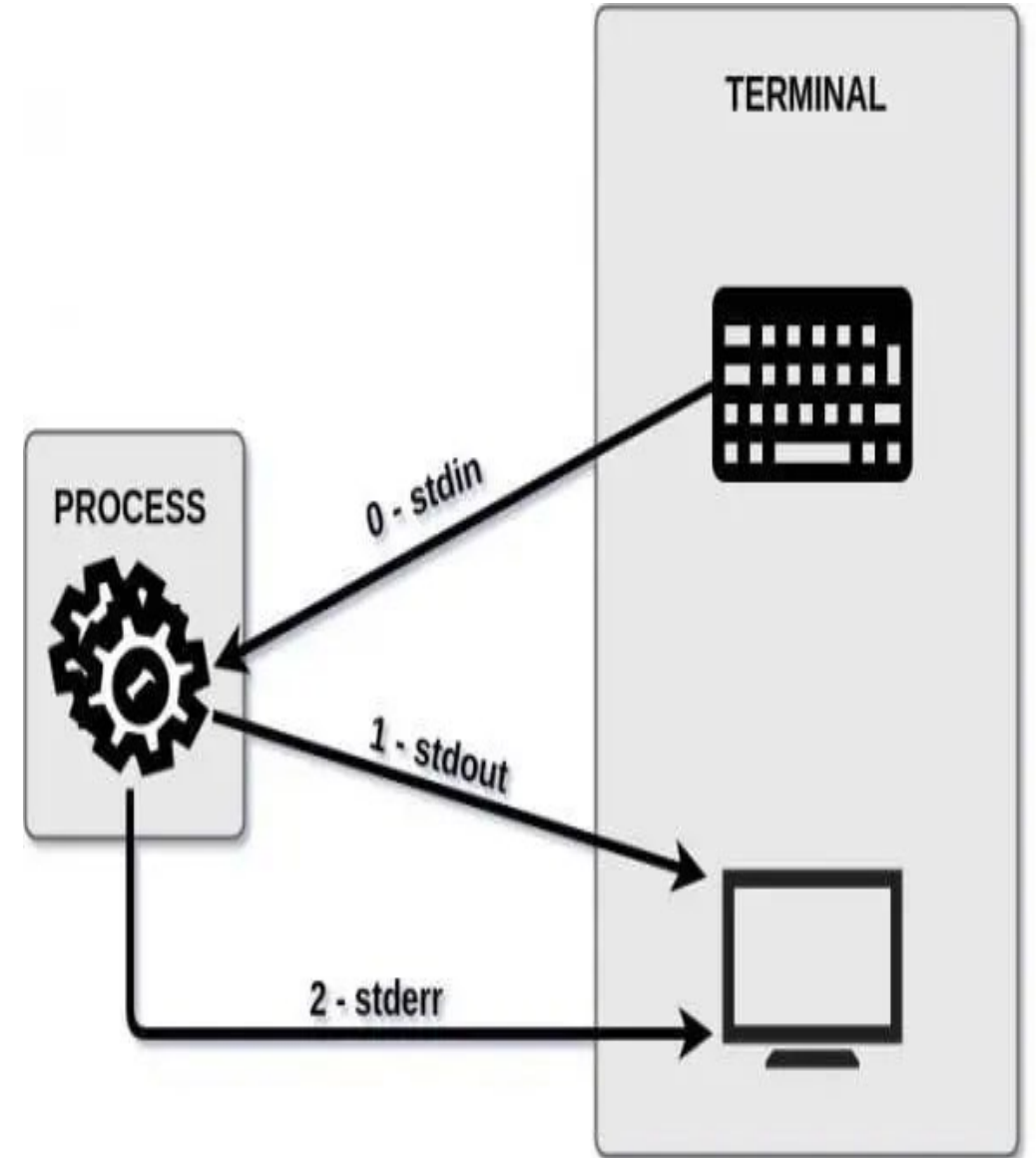
Ezek átirányíthatóak más eszközökre (device) vagy fájlba

2> ÁLLOMÁNY: stderr átirányítása (a megadott fájlba írja a hibaüzeneteket)

&> ÁLLOMÁNY: stdout és stderr átirányítása ugyanabba a fájlba

2>&1: a stderr-t ugyanoda irányítja, ahová a stdout irányítva lett

1>&2: a stdout-ot ugyanoda irányítja, ahová a stderr irányítva lett



Példa:

ls *.log

- A home/tibi könyvtárban a **ls *.log** listázás paranccsal ki akarjuk íratni az összes **.log** fájlt
- Hiba üzenetet kapunk hiszen nincs egy **.log** fájl sem

ls *.log >> err.txt 2>&1

- Ezután ezt a hiba üzenetet íratjuk be az **err.txt** fájlba a **2 >&1** hiba kimenet standard kimenetre irányításával, majd ellenőrizzük az **err.txt** fájl tartalmát

```
root@kali:/home/tibi# ls *.log
ls: '*.log' nem érhető el: Nincs ilyen fájl vagy könyvtár
root@kali:/home/tibi# ls *.log >> err.txt 2>&1
root@kali:/home/tibi# cat < err.txt
ls: '*.log' nem érhető el: Nincs ilyen fájl vagy könyvtár
```

A pipeline - csővezeték

\$ parancs1 | parancs2

- U.n. csővezeték vagy pipeline (!).
- A parancs1 kimenete konzolra kiírás helyett a parancs2 bemenete lesz.

\$ parancs1; parancs2; parancs3

- A parancsok ";"-vel elválasztva egymás után hajtódnak végre balról jobbra.

