

Néhány hasznos terminál tipp

Írta: Paul Arnote (parnote)

OK ... tudom, a Linux grafikus felületek (GUI) **nagyon** jók. Azt is tudom, hogy sok felhasználó van, aki kerüli a Linux parancssorának használatát – minden áron. De a terminál emulátor és a Linux parancssor használata nem csak szórakozás, hanem nagyobb hatalmat és ellenőrzést is ad a rendszered felett. A következőkben néhány általam használt tippet és trükköt ismertetek, amivel jobban irányíthatod a terminálműveleteket.

Terminál bezárva a program fut

Ha a parancssorból indítasz egy programot, akkor a terminál bezárása alpból az onnan indított program leállítását is eredményezi – hacsak nem háttérben futtatással indítod a programot. Persze bármely program futtatható háttérben egyszerűen úgy, hogy egy „&”-et adunk a parancs végéhez, valahogy így:

gcalctool &

De mi van akkor, ha a program már fut és te bezárnád, vagy be kellene zárnod a terminált, a program futását megtartva? Szerencsére a Linux megadja a választás lehetőségét, hogy a parancssorból indított program az (alpból) előtérben fusson, vagy a már elindított programot a háttérbe áthelyezzük.

A futó program háttérben történő futásának eléréséhez a következőt csinálod:

A billentyűzetten a <Ctrl>+Z billentyűket nyomd le (a terminálban ^Z jelenik meg, és a „Stopped [a program neve]” a következő sorban). Ennek hatására a (parancssorból indított) program futása időlegesen leáll.

Ezután a következőket írd be a parancssorba:

```
disown -h %1
bg 1
```

```
[parnote-t42@thinkpad-t42 ~]$ gcalctool
^Z
[1]+  Stopped                  gcalctool
[parnote-t42@thinkpad-t42 ~]$ disown -h %1
[parnote-t42@thinkpad-t42 ~]$ bg 1
[1]+ gcalctool &
[parnote-t42@thinkpad-t42 ~]$
```

Ennek hatására a terminál „kikerül” a program processzének szülője, vagy tulajdonosa státusából és a program az előtérből a háttérfolyamatok közé kerül. Minthogy a háttérbe került, a program tovább fut ugyan, de a háttérfolyamatként. Most már bezárhatod a terminált a program végrehajtásának megszakítása nélkül. Azt is vedd észre, hogy a „bg 1” sor bevitele után a gcalctool-t „gcalctool &”-ként mutatja jelezvén, hogy a program most már a háttérben fut.

Nyiss terminált akárhol

A legtöbb asztali környezet estében a terminált akár indítóból, akár a menüből futásra kiválasztva indítod el, az a képernyő bal felső sarokban nyílik meg. Általában ez megfelel. Azonban mi van akkor, ha a te munkástílusodnak sokkal jobban megfelelne, ha a terminál valahol más helyen nyílna meg?

Szerencsére ez nagyon könnyen megoldható. A legtöbb népszerű terminál programnak megadható, hogy milyen széles (karakter) és magas (sor) legyen, illetve az X és Y koordinátákkal, hogy hol jelenjen meg az ablaka. A legismertebb

terminálprogramoknál – xterm-et kivéve – a formátum a következő:

- -geometry=[szélesség]x[magasság]+[X-koordináta]+[Y-koordináta]

```
$ terminal -geometry=80x20+800+400
```

Egyszerűen illeszd ezt a parancssori kapcsolót a kedvenc terminálot indító parancs végére. Amennyiben a terminált indítóból szoktad meghívni (<Alt>+F2 – ford.), a parancsot ott kell megszerkeszteni. Ha a terminálot általában az alkalmazásindító menüből hívod meg, akkor megszerkesztheted akár a menüelemet (KDE esetén), vagy a desktop fájlt (a /usr/share/applications helyen root-ként) a parancssori kapcsoló beiktatásával.

Emlékezz, azt mondtam, hogy az xterm a kivétel. Ha az xterm a kedvenc terminálalkalmazásod, akkor a parancssori kapcsoló egy kicsit más ... de hasonló lesz. Csak az egyik kötőjelet az elejéről és az egyenlőségjelet távolítsd el. Például, xterm -geometry 80x20+100+100 az xterm ablakát 80 karakter szélesen és 20 sor magasan a 100-as X és 100-as Y koordinátánál nyitja meg.

Használj felkiáltójelet!

Nem teljesen szokatlan többszörös parancsokat kiadni, amikor parancssorral dolgozol, de semmi sem olyan unalmas, mint begépelni ugyanazt a parancsot újra, meg újra. Szerencsére van mód egyszerűsíteni. Csak üss egy „!”-et, amit a parancs neve követ és visszaidézi. Ezután csak nyomj egy <Enter>-t a végrehajtáshoz.

```
[parnote-t42@thinkpad-t42 Thunar]$ !mousepad
mousepad uca.xml
```

Például, mondjuk a parancssort használtuk egy adott beállító fájl kedvenc fájl szerkesztőnkben történő megnyitásához. Kedvenc fájl szerkesztőm az Xfce Mouspad-je, ezért azt használom a példában. A „lmousepad” beírása újra előhívja azt a parancsot, amivel a terminálból indítottam a Mouspad-et. Az <Enter>-t lenyomva elindítja.

VALÓBAN használd !!

Miközben az előző rövidítés egy felkiáltó jel után a megismételni szándékozott parancs nevéhez kapcsolódik, a dupla felkiáltójel lehetővé teszi az utolsó parancs előhívását és nincs szükség a parancs nevére a működéshez.

```
$ mousepad /usr/share/applications/gimp.desktop
$ gksu !!
gksu mousepad /usr/share/applications/gimp.desktop
```

Ez akkor nagyon hasznos, amikor hivatkoznod kell az utolsó parancsra egy másik parancsban. Például, mondjuk megnyitottál szerkesztésre egy konfigurációs fájlt normál felhasználóként, de rájössz, hogy rendszergazdaként kell megnyitni azt a fájlt. Egyszerűen beírva, hogy „gksu !!”, vagy „kdesu !!” (KDE esetén), a kdesu-val, vagy gksu-val kapott root jog elnyerése után egyszerűen végrehajtja a parancsot.

Nézd az előzményeket

Egy kimondottan nagy hatású parancs, amit a parancssorban használni lehet, az a **history**. Önmagában végrehajtva felsorolja az összes parancsot, amit a parancssorban addig használtál. Ha rendszeresen használod a parancssort, akkor elég **hosszú** listád kapsz. Minden parancssori tételhez jár egy saját sorszám. Az egyik számítógépemen több mint 1000 sorom van már! Az összes parancsot átlapozni nemcsak unalmas, hanem nagy az esélye annak, hogy átsiklasz a keresett parancs fölött, amikor átlapozod a hosszú listát.

Szerencsére javíthatod a parancs megtalálásának esélyeit. Mondjuk keressük azokat az eseteket, amikor a convert parancsot (az Image Magick parancssori eszköze) használtam. Használjuk a history parancsot a grep parancssal kombinálva.

history | grep convert

```
[parnote-t42@thinkpad-t42 ~]$ history | grep convert
728  convert --help
994  history | grep convert
```

Mivel már tudom, hogy mit keresek, kizárhatom az összes többi parancsot a parancssori előzményekből. Most már csak annyit kell tennem, hogy végrehajtom ismételten a parancsot a „!728” parancs kiadásával, mivel a 728-as szám az, ami a keresett parancshoz a parancssori előzmények között tartozik.

Legutóbb ...

Terminálban, egy parancs legutolsó előfordulásának előhívására a <Ctrl>+R billentyűpáros szolgál. Tudd, hogy ez a billentyűparancs CSAK az adott parancs legutolsó előfordulását hívja vissza.

Vedd észre, hogy a parancssori helyőrző most „reverse-i-search”-re változik és ahogy elkezdesz gépelni (én „mou”-t idézőjelek nélkül írtam), megjelenik az utoljára általam használt mousepad parancs. Az <Enter> lenyomásával újra végrehajtja a parancsot.

Ez különösen hasznos, ha pl. használtad az mv parancsot egy fájl más könyvtárba mozgatására és ellenőrizni akard az útvonalat, ahová a fájlt vitted. Szintén nagyon hasznos, amikor nem emlékszel egy korábban, sikeresen végrehajtott parancs pontos szintaxisára.

```
(reverse-i-search)`mou': gksu mousepad /usr/share/applications/gimp.desktop
```

Mozgás könyvtárak között, a könnyebbik út

Mondjuk elég unalmas a cd parancs használata a könyvtárak közötti mozgásra, különösen, ha a célkönyvtár sok szinttel lejjebb van. Minél mélyebben van a célkönyvtár, annál hosszabb az arra irányító parancs és nagyobb az esélye a gépelési hibának is.

```
[parnote-t42@thinkpad-t42 ~]$ cd -
/usr/share/applications
[parnote-t42@thinkpad-t42 applications]$ cd ~
[parnote-t42@thinkpad-t42 ~]$ pushd /usr/share/applications
/usr/share/applications ~
[parnote-t42@thinkpad-t42 applications]$ popd
~
[parnote-t42@thinkpad-t42 ~]$ cd -
/usr/share/applications
[parnote-t42@thinkpad-t42 applications]$ cd -
/home/parnote-t42
[parnote-t42@thinkpad-t42 ~]$ cd-
bash: cd-: command not found
[parnote-t42@thinkpad-t42 ~]$ cd -
/usr/share/applications
[parnote-t42@thinkpad-t42 applications]$ cd -
/home/parnote-t42
[parnote-t42@thinkpad-t42 ~]$
```

Szerencsére vannak rövidítések, amiket használhatsz. Első, a **cd -**, ez váltogat két könyvtár között. Lásd a szóközt a cd és a követő kötőjel között. Második a **cd ~**, ami automatikusan a te /home könyvtáradra visz vissza, függetlenül attól, hogy hol jársz a fájlstruktúrában. Mint a testvérénél, itt is lásd a szóközt a cd és a követő hullámvonal között.

A cd helyett a **pushd** parancsot használva az aktuális könyvtár bekerül a **popd** változóba, majd elvisz téged a meghatározott könyvtárba. Az előző könyvtárhoz visszatéréshez egyszerűen írd be, hogy popd. Sajnos ez csak egyszer használatos. Könnyen túlléphetsz ezen a korlátozáson a **cd -** parancs alkalmazásával. Így ugrálhatsz két könyvtár között amíg csak meg nem unod, örökké ... ha akarsz.

Lődd ki teljesen

Néha találkozol olyan programmal, ami megakad és nem kapcsolható le. Szerencsére van egy Linux parancs kvartett a szolgálatodra várva, hogy megmutathasd a rakoncátlankodó programnak, hogy ki a főnök.

```
[parnote-t42@thinkpad-t42 applications]$ pidof gcalctool
3399
[parnote-t42@thinkpad-t42 applications]$ kill 3399
```

A **kill** **<Tab>** parancs, parancssorból kiadva, felsorolja ezt e négy parancsot. Egyetlen változó, amit elfogad, az a rosszul viselkedő program PID-je (program ID=azonosító). Micsoda? Nem tudod hol találhatod meg a PID-et? Nos, kiadhatod a **top** parancsot terminálban. Az első oszlop sorolja a futó programok PID-jeit. Vagy begépelheted, hogy **pidof** **[programnév]** és a PID-et adja vissza. Ezután csak használd a PID-et a kill parancs parancssori változójaként.

```
[parnote-t42@thinkpad-t42 applications]$ killall gcalctool
[4]- Terminated gcalctool
[5]- Terminated gcalctool
[6]+ Terminated gcalctool
```

A **killall** parancshoz csak a program neve kell, mint parancssori argumentum. Vagyis, ha számos ablakod van az asztalra kitéve, amit egyetlen program irányít – legyen mondjuk a Firefox –, akkor a **killall firefox** parancs bezárja az összes, Firefox által kezelt ablakot. Hasonlóképpen, ha egy program több példányban van megnyitva, az adott program összes példánya bezáródik.

```
[parnote-t42@thinkpad-t42 applications]$ pkill gcalctool
[1]+ Terminated gcalctool
```

```
[parnote-t42@thinkpad-t42 applications]$ pkill lcto
[2]+ Terminated gcalctool
```

A **pkill** parancs a programok bezárására szolgál egyszerűsítés és csak a program nevének egy részletére (vagy a teljes nevére) van szüksége parancssori argumentumként. A korábbi példában a teljes nevet használtam, de pl. a **pkill gcal**, a **pkill tool**, vagy a **pkill lcto** mind bezárja a **gcalctool**-t.

Hasonlóképp a **pkill fire**, **pkill fox**, **pkill iref** mindegyike bezárja a Firefox böngészőt.

Az **xkill** parancs csak az X Window rendszeren működik és a PCLinuxOS alaptelepítésének nem része (a többi három igen). Ugyanakkor telepíthető a Synaptic segítségével. Végrehajtásakor a kurzor X-re változik. Az X kurzorral egyszerűen kattints a bezárni kívánt program ablakára. A program az X kurzor alatt bezáródik és a kurzor visszaváltozik a rendes formájára.

Heads & Tails (fejrész és farok)

Ha terminálban dolgozol nem kell elhagyni azt a környezetet csak azért, hogy elolvass egy egyszerű szövegfájl – és Linuxban rengeteg ilyen található. A legtöbb Linux beállító fájl nem más, mint egyszerű szövegfájl.

Természetesen a Linux **head** és **tail** parancsairól beszélek. Az egyik (**head**) beolvassa a szövegfájl első részét, miközben a másik (**tail**) beolvassa a szövegfájl utolsó részét anélkül, hogy ahhoz szövegszerkesztőt nyitna, vagy abba töltené be.

```
[parnote-t42@thinkpad-t42 PCLinuxOS.Mag]$ head -n20 ArticleOrderandLinks-JAN2013.txt
##### LINKS #####
##### DONE #####
Table of Contents

##### EDITORIAL #####
##### DONE #####
1. Editor's Welcome

##### PAGE 01 #####
##### DONE #####
2. Keep Tabs On Your HD's Health With GSmartControl
Torrent Freak
Reach Us on the Web
Open Source Initiative
linuxfordummies.org

##### PAGE 02 #####
##### DONE #####
3. ms_meme's Nook: PCLinuxOS Will Shine
[parnote-t42@thinkpad-t42 PCLinuxOS.Mag]$
```

Használatuk nagyon hasonló. Csak add ki a parancsot, amit a megtekinteni kívánt fájl neve követ. Bármelyik parancsot egy szövegfájl nevével együtt kiadva, annak vagy az első 10 sorát (**head** parancs esetén), vagy az utolsó 10 sorát (**tail**

parancs esetén) megjeleníti.

Meg is változtathatod mindkét parancs működését úgy, hogy az alapbeállítás szerinti vacak tíz sor helyett többet mutasson. Egyszerűen a parancs és a megjelenítendő szövegfájl neve közé illeszd be **-n[sorok száma]**. Így, ha az első 20 sorát szeretném látni egy szövegfájlnak, akkor a **head** parancsot úgy adnám ki, hogy **head -n20 mylist.txt**. Hasonlóképpen, ha egy szövegfájl utolsó 20 sorát szeretném látni, akkor a **tail** parancsot **tail -n20 mylist.txt** formában adnám ki. Annyi sort jeleníthetsz meg, amennyit csak akarsz, egyszerűen a megjelenítendő sorok számának megadásával. Ha az egész fájlt megnéznéd csak a szám legyen a fájl sorainak számánál nagyobb (100? 1000?).

Összefoglalás

Bármit is csinálsz, ne szégyelld a Linux parancssorát megismerni. Néhány éve még úgy álltam a Linux parancssorához, mint amit néha kénytelen vagyok használni. Számomra egy nagy fekete lyuk volt. De miután elhatároztam, hogy megtanulom, fellebbent a Linux parancssorról a titokzatosság fátyla és hamarosan felfedeztem, milyen értékes tud lenni. Sok esetben a Linux parancssora sokkal gyorsabban használható, mint a program GUI változata. Emellett a Linux parancssor szórakoztató is lehet.

