

Windows migrálás: terminál - néhány alapparancs

PCLinuxOS Magazine – 2013. június

Írta: Peter Kelly (critter)

A parancsok

Hogy a Linux-parancssor világának megismerésében utadra indítsalak, a következőkben ismertetek néhány, általam gyakrabban használt parancsot. A Linux eszköztára sok parancsot tartalmaz, ezért minden felhasználónak megvan a saját többnyire használt parancskészlete. Íme ötven az enyémből.

apropos, cal, cat, cd, chmod, chown, clear, cp, cut, date, df, dmesg, du, exit, file, find, free, grep, gzip, head, info, kill, less, ln, locate, lp, ls, man, mkdir, mount, mv, nano, passwd, ps, pwd, rm, rmdir, shutdown, sort, su, tac, tail, tar, top, touch, tr, umount, uname, whatis, whereis.

Ez igazából nem sok. Nem feltétlenül a legtöbbször használt Linux-parancsok ezek, de ha tudod őket használni, akkor a többivel sem lesz gondod. E néhány parancs használatához hozzászokva sokkal kényelmesebben fogod érezni magad a terminál használatában. A parancsok közül kilencel már találkoztunk, de referenciaként megismétlem őket.

Segítség kérése

Amikor kezdesz terminált használni, először is segítségre van szükséged. A következő parancsok megadják számodra a segítséget.

A legtöbb parancshoz tartozik felhasználói kézikönyv, amit a következő paranccsal lehet elérni.

man parancsnév pl. man ls (MS = *help*, vagy *parancs* /?)

A fel és le nyilakkal, a lapozás fel és -le gombokkal navigálhatsz. Kilépés a q billentyű lenyomásával. Ha a kézikönyvben foglaltakon túl további információra van szükséged, akkor próbáld ki a parancsok kézikönyvet a következő paranccsal.

info parancsnév

Navigálás mint a **man** parancsnál.

Ha nem ismered a parancs nevét, próbáld ki az **apropos** parancsot némi vonatkozó szöveggel.

Az **apropos cdrom** kilistázza az összes parancsot, aminek a leírásában a cdrom szerepel.

Ha ismered a parancs nevét, de nem vagy biztos a céljában, próbáld ki a

whatis parancsnév hogy rövid leírást kapjál.

Ha tudnod kell, hogy hol található egy parancs és a hozzá tartozó fájlok, akkor

whereis parancsnév megadja az információkat.

Fájlkezelés

Egyszerű felhasználóként beléphetsz és kilistázhatod a legtöbb könyvtár tartalmát, de nem változtathatsz, vagy törölhetsz, kivéve a saját home könyvtáradban. Könyvtárváltáshoz használd a

cd könyvtárnév (MS = *cd*). Például a **cd /etc** elvisz az /etc könyvtárba. A **cd** önmagában a home könyvtárba visz. A **Documents** könyvtárba belépéshez **gépeld be**

cd /home/linux_user/Documents vagy használd a rövidítést **cd ~/Documents**. Emlékszel a hullámvonalra?

Fájlmásoláshoz használd a **cp** (MS = *copy*) parancsot

A **cp fájlnev** cél A cél tartalmazhatja könyvtárakat, az útvonal leírásával, de azoknak a könyvtáraknak már létezniük kell és írási jogosultsággal kell rendelkezze, legalább az útvonal utolsó könyvtárára.

Könyvtárak teljes tartalmának (rekurzív) másolásához a **cp -R** (MS = *xcopy*) kell.

Fájl átnevezésére az **mv régi fájl új fájl** parancs (MS = *rename*, vagy *move*) való. Az új fájlnev opcionálisan tartalmazhat könyvtárcímet, amire írási jogunk van. Fájl törlésére ezt a parancsot használjuk

rm fájlnev (MS = *del*, vagy *erase*). Légy vele óvatos, mert nincs visszaállítás parancs. A parancs PCLinuxOS-es verziója megerősítést kér, mielőtt a sülyesztőbe küldené a fájlt, de ez nincs mindenhol így.

Új könyvtár készítéséhez arra a könyvtárra, ahová elhelyeznéd, írási jogod kell legyen, és használd a következő parancsot

mkdir könyvtárnév (MS = *md*, vagy *mkdir*)

Üres könyvtárat a következő paranccsal törölheted

rmdir könyvtárnév (MS = *deltree*)

Könyvtár és teljes fájl tartalmának eltávolítására az `rm` parancsot `-r` opcióval kell használni. Ha sok fájl tartalmaz a könyvtár, a megerősítés elég fárasztó lehet, de kikapcsolhatod az `-f` (force - erőltet) opcióval.

Fontos: *Légy vele nagyon-nagyon óvatos. Könyörtelen, gyors és végleges. Apró trükk, amit használok, hogy lecserélem az `rm -rf`-et `ls -R`-rel, a törölendő fájlok ellenőrzésére. Ha a kíváncsagságnak megfelel, kiadható az `rm -rf` halálos ítélet. Az `-R` opció beolvassa az összes alkönyvtárat (nagybetűs `-R` kell, mert a `-r` csak megfordítja a listát).*

Készíthetsz egy új üres fájlt a

touch file-name paranccsal.

Gyakran kell hivatkoznod egy fájlra, ami mélyen el van ásva egy alkönyvtárban és ennek egyszerű módja, akár a Windows-ban a hivatkozás készítése. A Linuxban kétfajta hivatkozás van és ezeket linkeként ismerik. Vannak hard linkek és soft, vagy szimbolikus linkek. Most csak a soft linkekre koncentrálnunk. Ezek létrehozására szolgáló parancsok

ln vagy **ln -s** a soft linkekhez (*MS = assign*).

ln -s /home/linux_user/Documents/invoices /November/Acme_inc.txt acme_11

létrehoz egy `acme_11` nevű hivatkozást, ami az elásott fájlra mutat, lehetővé téve könnyebb betöltését, például egy szövegszerkesztőbe. Nagyon hasznos rendszer szkriptek szerkesztésekor, így nem kell emlékezni rá, hol vannak.

Fájl engedélyeinek megváltoztatására a **chmod** parancs való (change mode – a módváltás) (*MS = attrib*). A parancs az új engedélyek megadására számos lehetőséget kínál, ami az eljén zavaró lehet. Egyszerűség kedvéért a következő eljárást kövesd.

Három engedélycsoport van: tulajdonos, csoport és a többiek.

Háromfajta engedély van: olvasás (read), írás (write) és futtatás (execute).

Olvas=4, ír=2 és futtat=1.

Az egyes engedélycsoportokhoz így add hozzá

tulajdonos olvas+ír = 4+2=6

csoport olvas = 4

mindenki más nincs elérés =0

chmod 640 myfile és kész!

Információ

A rendszertől sokféleképpen lehet információt kapni, a következő parancsok megadják a legtöbbet, amire szükség lehet.

ls (*MS = dir*) ahogy azt már láttuk, kilistázza egy könyvtár tartalmát.

pwd (*MS = chdir*) munkakönyvtár kiírása. Ez nyugtázza, hogy a fájlrendszerben hol vagy.

date (*MS = date, vagy time*) Ezt a parancsot önmagában kiadva kiírja a pillanatnyi dátumot és az időt. Ez a parancs használatos a rendszeridő és dátum beállítására, de ezt a PCLinuxOS vezérlőközpontjában sokkal jobban végrehajtható.

Ha naptárra lenne sürgősen szükséged csak gépeld

cal és a hónapra vonatkozó, szépen formázott naptárat kapsz. A **cal -3** hozzáteszi még az előző és a következő hónapot, a **cal -y** a teljes éves naptárat és a **cal 1066** pedig a Hasting-i csata évére vonatkozó történelmi naptárat adja vissza.

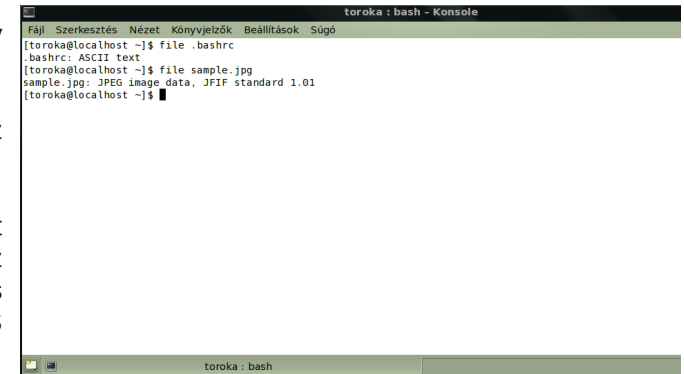
A Linux a rendszerfájlokról adatbázist vezet. Egy rendszerfájl előkerítéséhez használd a

locate fájlnev parancsot. Van egy **fstab** nevű fájl, amit gyakran kell szerkeszteni és általában az `/etc` könyvtárban található. Ha nem emlékszel hol van, **locate fstab** és az megmondja.

Néha megkérdezik, hogy az operációs rendszer, amibe bejelentkeztél, milyen kernelt használ (32, 64 bit stb.).

Az **uname -a** megmondja mindezt, ahol az **-a** mindent (all) jelent, a többi opció pedig korlátozza a kimenetet.

A **free** (*MS = mem*) **-m** és **-g** opciókkal megjeleníti az eredményt annak megfelelően MB-ben, vagy GB-ben. A Linux nem fájlkiterjesztés alapján határozza meg a fájl tartalom típusának, de gyakran használja és többnyire a felhasználó kényelmére. A file parancs megmondja, milyen adatok vannak a fájlban.



A **find** egyike azon parancsoknak, amik egyszerre nagyon hatékonyak, nélkülözhetetlenek és a parancssorral ismerkedők számára fárasztóan összetettek. A szintaxisa kicsit eltér a Linux parancsok többségétől, de használata egyszerűsíthető, hogy egy használhatóbb alapeszközünk legyen. A parancs szintaxisa valahogy így néz ki:

find *ahol-keressen* *opciók* *vizsgált-fájljellemző* *keresett-érték*

Ahol-keressen, a keresés kezdő könyvtára.

Az opciókba olyanok tartoznak mint **-maxdepth 1**, hogy a keresés a meghatározott könyvtár legfelsőbb szintjén maradjon.

A *vizsgált-fájljellemző* lehet fájlnev, fájltypus pl. könyvtár, módosítás dátuma, méret, engedélyek, vagy szinte bármi, ami egy fájl jellemző. A keresés finomhangolására többféle kritérium is megadható.

A *keresett érték* lehet fájlnev, vagy részlete, beleértve dzsókereket, mint a * (szabályos kifejezés azok számára, akik értik az ilyen dolgokat). Lehet akár a módosítás, vagy a hozzáférés óta eltelt idő, és lehet minimális, vagy maximális fájl méret.

Ahhoz, hogy a saját és közvetlen alatta lévő könyvtárakban, de nem lejjebb keressen .jpg fájlokat add ki ezt a parancsot

find ~ -maxdepth 2 -name "*.jpg"

Hogy a .JPG kiterjesztésű fájlokat is keresse, kapcsold ki a kis- és nagybetű megkülönböztetést a -name-t állítsd át **-iname**-re.

A **find ~/Documents/ -mtime 0** kilistázza az összes fájlt, ami a legutóbbi 24 órában változott.

A **find / -size +1000M** megmutatja az összes fájlt, aminek a mérete meghaladja az 1000 MB-t (1 GB), mínusz jellel az összes 1000 MB-nél kisebbit listázza – nagyon sok fájl! Ha root-jog nélkül futtatod, egy sor „engedély megtagadva” jelzést kapsz, mivel a parancs sok olyan könyvtárba is be akar lépni, amire nincs elérési jogod. Megkerülhető átirányításnak hívott eljárással, amit a cikk második felében tárgyalunk meg.

A hibajelzéseket elküldhetjük egy **/dev/null**-nak hívott fekete lyukba.

find / -size +100M 2>/dev/null

Rajta szeretnéd tartani a szemed a lemezmeghajtódon, mert problémáid lesznek ha kezdenek megtelni. Egyszerűen

df (MS = *net share*, nem teljesen az, de hasonló) parancs, ami megmutatja a lemez szabad helyét.

Fájl	Szerkesztés	Nézet	Könyvjelzők	Beállítások	Súgó
[toroka@localhost ~]\$ df					
Fájlrendszer	Méret	Fogl.	Szab.	Fo. %	Csatol. pont
/dev/sda1	45G	9,2G	34G	22%	/
/dev/sda4	137G	21G	117G	16%	/home
/dev/sda3	93G	79G	14G	86%	/mnt/DOKU_ASZTAL
/dev/sdd2	30G	7,1G	21G	26%	/media/Toshiba_2
/dev/sdd4	22G	7,5G	13G	37%	/media/Toshiba_3
/dev/sdb5	39G	14G	23G	37%	/media/896c02f0-4a85-4b1b-89f0-bbb5f471104c
/dev/sdc3	29G	791M	27G	3%	/media/3f5949e3-abad-4074-b086-6721a6906884
/dev/sdc2	327G	233G	94G	72%	/media/Tartalek
/dev/sdb3	105G	31G	74G	30%	/media/PCLOS_home
/dev/sdc2	23G	6,8G	15G	32%	/media/38f233fa-b54c-4b28-a065-58c0aff22e3e
/dev/sde1	296G	145G	151G	49%	/media/Linuxnak
/dev/sdd1	181G	108G	73G	60%	/media/Toshiba_1
/dev/sde2	301G	192G	109G	64%	/media/Winnes
[toroka@localhost ~]\$					

Hogy megtudd, a helyet hogyan használta fel, add ki a **du** lemezhasználat parancsot argumentum nélkül. Megmutatja, hogy az aktuális meghajtón a fájlok és a könyvtárak mennyit használtak fel. A **du /Documents** az adott könyvtárra ad eredményt. Add hozzá a **-s** opciót, ha csak összefoglalót kérsz. A **du /-t** a gyökérkönyvtárban kiadva a **find** parancshoz hasonlóan hibajelzések tömegét kapod.

Ha boot közben lenyomod az <ESC>-et, akkor információk tömegét látod legördülni, ahogy a kernel beállítja a dolgokat. A kimenet a boot után is él, jegyzi ahogy a kernel elindítja és leállítja a dolgokat, vagy amit fontosnak tart közölni. Ez az információ hasznos lehet, amikor a dolgok nem mennek úgy, ahogy kellene, esetleg egy USB meghajtót nem ismer fel.

A **dmesg** parancs kiírja ezeket az információkat, de a jóval többet mint ami neked kell, miközben te feltehetőleg néhány utolsó sorra vagy kíváncsi.

Használhatunk egy másik szűrő parancsot a kimenet korlátozására.

A **tail** (farok) a fájl utolsó 10 sorát jeleníti meg. A **tail -n20** húsz sort ad vissza. A szűrő használatához ismét a csővezeték-jelet kell alkalmaznunk.

dmesg | tail -n 15

Hasonlóan, ha csak egy fájl elejét szeretnéd látni használd a **head** (fej) parancsot.

dmesg | head -n 20

Egy fájl tartalmának elejétől végéig történő megjelenítésére a már ismert **cat** parancs szolgál, emellett ahogy a **tail** ellentettje a **head**, a **cat**-nek is megvan a párja a **tac**, ami a fájlt fordított sorrendben jeleníti meg, az utolsó sort először. Mi a fenéért akarna valaki ilyet csinálni? Nos, sok Linux-parancs készít log fájlokat a tapasztaltak és a kimenetek dokumentálására. A **tac** parancssal a parancs legutolsó sorait tekinthetjük meg.

Rendszer adminisztrálása

A fájltulajdonlás beállítására a **chown** parancs van, amivel megváltoztathatjuk a fájl mind a felhasználói, mind csoport tulajdonlását. Az **-R** opció lehetővé teszi a rekurziót az alkönyvtárakra, megváltoztatva azokban is a fájlok tulajdonlását.

chown felhasználó:csoport fájl

chown felhasználó fájl

chown :csoport fájl

Megadhatod csak a felhasználót, mind a felhasználót és a csoportot kettősponttal elválasztva, vagy csak a csoportot, egy kettőspontot követően.

Egyszerű felhasználó megváltoztathatja egy fájl csoporttulajdonlását bármely csoportra, amelyiknek tagja, de root-, vagy rendszergazda jog kell a felhasználó, vagy teszőleges csoporttulajdon beállítására.

Néhány disztribúcióban ideiglenes root-jog nyerhető a **sudo** paranccsal. Biztonsági okokból a PCLinuxOS-ben **nincs** ilyen. Itt az **su** (switch user – felhasználóváltás) használatos.

A **su** parancs alpból a root-ra vált, hacsak nincs más felhasználói név megadva. Átválthatsz, átvéve bármelyik felhasználó azonosságát, akinek tudod a jelszavát (emiatt kell a jelszavadat titokban tartani).

A **su guest** a vendég felhasználó jelszavát fogja kérni, majd azt csinálhatsz a vendég felhasználó fájljival és könyvtáraival, amit akarsz, akár törölheted is!

Ezért a **su**-t használd körültekintően, mivel a root felhasználónak megvan az ereje, hogy a rendszer bármelyik, vagy összes fájljával bármit tegyen!

A **su**-nak számos olyan opciója van, amivel tisztában kell lenned.

Opció nélkül a **su** átadja a root-jogokat, de ott maradsz, ahol éppen voltál a fájlrendszerben.

A **su -l** a bejelentkezőhöz visz. Pont úgy mintha éppen bejelentkeztél volna root-ként és a hullámvonal a prompt után most a root könyvtárat **/root** jelenti és nem a te home-odat. A root-jogok visszaadásához és visszatéréshez a halandók közé üss **<CTRL>+d**-t, vagy az **exit** (*MS = exit*) parancsot.

A **su -c** parancs kérni fogja a root-jelszót, végrehajtja a parancsot, majd visszavisz az egyszerű felhasználói állapotodba.

Jelszóváltáshoz használd a **passwd** parancsot. Kéri a jelenlegi jelszavadat, az új jelszavadat és az új jelszó megerősítését. Ha ez utóbbi kettő egyezik,

akkor megváltozik a jelszó. Root-joggal meghatározhatasz felhasználót, akinek a jelszavát akarod megváltoztatni (vagy felfüggeszteni, megakadályozva a felhasználó bejelentkezését).

A parancssorból való rendesen kikapcsoláshoz, vagy a rendszer újraindításához a root-nak van egy **shutdown** (*MS = reboot*) parancsa.

Többfelhasználós rendszerben figyelmeztetned kell a felhasználókat és elegendő időt kell hagynod, hogy menthessék munkájukat ilyen művelet előtt és ez az, amire a parancs való. Az **-r** opció közli a paranccsal, hogy rendszer újrabootolás lesz és a **-h**, hogy rendszerleállítás. Kötelező az idő opció, ami egyfelhasználós rendszer esetén lehet **now** (most) is. Ha nem adsz meg figyelmeztető üzenetet, akkor az alapüzenet megy ki.

shutdown -r now A számítógép azonnali újraindítása.

shutdown -h 18:30 „A Rendszer 18.30-kor leáll, rutin karbantartásra”

shutdown -c Ütemezett leállítás elvetése.

Mindenkor, ha parancsot hajtasz végre, vagy programot futtatsz, a kernel új, önálló process-t (folyamatot) indít el a parancsra. A parancsok gyakran indítanak el al-folyamatokat. Esetenként egy process „megakadhat”, vagy végtelen ciklusba kerülhet, meghíúsítva a bezárására irányuló erőfeszítéseidet. Itt ismernünk kell, hogy a rendszer hogyan hivatkozik az adott process-re, vagyis

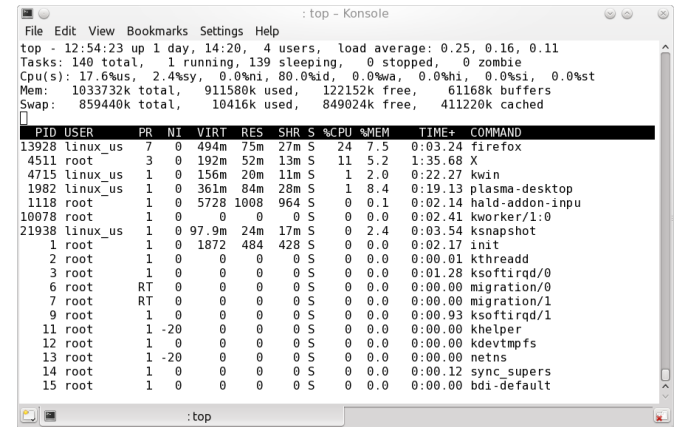
```
File Edit View Bookmarks Settings Help
[linux_user@localhost ~]$ ps, u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
501       7809  0.0  0.1 5160 1732 pts/2    Ss+  12:03   0:00 /bin/bash
501      13111  4.5  7.6 570580 79156 pts/1    Sl   12:44   0:01 firefox
501      13206  0.0  0.0 4604  936 pts/1    R+  12:45   0:00 ps u
501      19717  0.0  0.1 5160 1764 pts/1    Ss   05:32   0:00 /bin/bash

[linux_user@localhost ~]$ kill 13111
[linux_user@localhost ~]$ ps, u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
501       7809  0.0  0.1 5160 1732 pts/2    Ss+  12:03   0:00 /bin/bash
501      13279  0.0  0.0 4604  936 pts/1    R+  12:45   0:00 ps u
501      19717  0.0  0.1 5160 1764 pts/1    Ss   05:32   0:00 /bin/bash

[!]+ Terminated      firefox
[linux_user@localhost ~]$
```

tudnunk kell a process id-t. Erre van a **ps** (*MS = tasklist*). A parancshoz számos opció tartozik. A **ps -u** a te process-eidet listázza. Ha nem mutatná a folyamatot, akkor a **ps ax** valószínűleg fogja, de a lista hosszú lesz. Keresd a PID-oszlop alatti számot.

Process bezárásához a **kill PID** parancsot használjuk. Néha egy folyamat ellenállhat a próbálkozásnak és az ennél sokkal erősebb **kill -9 PID**-et kell használni. Nézd meg a következő példát.



A Firefox a 13111 process alatt fut, amíg a **kill** parancs be nem zárja az alkalmazást.

Egy másik parancs, ami lehetővé teszi a folyamatok kezelését a **top** (*MS = mem*). Itt a Firefox id-je 13928.

A process bezárásához nyomj **k**-t és kérni fogja a **PID**-et, hogy leljője:

Írd be **13928** és kérdezi, hogy

kill PID 13928 with signal [15]: Írj y-t (i), hogy bezárad a process-t. A top parancs néha elég összetett lehet, de akad alternatívája, ami sokkal barátságosabb. Ez a **htop**, alpból nincs telepítve, de a tárolókban telepítésre rendelkezésre áll, a **Synaptic** csomagkezelővel. Ajánlom a htop telepítését.

A Windows-tól eltérően a Linux nem használ meghajtókra betűket, pl. **c:**. Minden partíció a fájlrendszeren belül egy adott helyre csatolódik és azután onnan érhető el. Általában zökkenőmentesen kezeli az `/etc/fstab` text fájl segítségével és nem kell foglalkoznod vele. Ha beillesztesz egy USB-meghajtót, akkor az valószínűleg nem lesz az `fstab`-ban benne. A modern rendszerek észlelik és automatikusan csatolják az eszközt (vagy felajánlják a csatolását).

A parancssorral teljesebb ellenőrzésed lehet ezek fölött a **mount**-nak és **umount**-nak (nem elírás) nevezett parancsok fölött. (A parancs `umount` és nem `unmount`. Mondtam, a Linux-osok lusták gépelni.) Hogy használni tudd ismerned kell, a rendszer hogyan hivatkozik az eszközökre. Általában a meghajtó nevével történik mint `/dev/sda1`, ami az első meghajtó (a) első partíciója (1). Ha beillesztesz egy USB-kulcsot, majd kiadod a **dmesg | tail -n 20** parancsot, akkor egy ehhez hasonló kimenetet fogsz látni (az aláhúzásokat én adtam hozzá).

```
usb 5-2.1: ep 0x81 - rounding interval
to 32768 microframes, ep desc says 0
microframes
usb 5-2.1: ep 0x2 - rounding interval
to 32768 microframes, ep desc says 0
microframes
Initializing USB Mass Storage
driver...
scsi8 : usb-storage 5-2.1:1.0
usbcore: registered new interface
driver usb-storage
USB Mass Storage support registered.
usbcore: registered new interface
driver uas
scsi 8:0:0:0: Direct-Access Kingston
DataTraveler G3 PMAP PQ: 0 ANSI: 0 CCS
sd 8:0:0:0: Attached scsi generic sg4
type 0
sd 8:0:0:0: [sdd] 7653312 512-byte
logical blocks: (3.91 GB/3.64 GiB)
sd 8:0:0:0: [sdd] Write Protect is off
sd 8:0:0:0: [sdd] Mode Sense: 23 00 00
00
```

```
sd 8:0:0:0: [sdd] No Caching mode page
present
sd 8:0:0:0: [sdd] Assuming drive
cache: write through
sd 8:0:0:0: [sdd] No Caching mode page
present
sd 8:0:0:0: [sdd] Assuming drive
cache: write through
sdd: sdd1
sd 8:0:0:0: [sdd] No Caching mode page
present
sd 8:0:0:0: [sdd] Assuming drive
cache: write through
sd 8:0:0:0: [sdd] Attached SCSI
removable disk
```

A 8. sor (aláhúzott) mutatja, hogy a rendszer megtalálta a kulcsomat, egy Kingston Data Traveller-t. Most nézz a 17. sorra (szintén aláhúzott). Megmondja, hogy a kernel az eszközt `/dev/sdd`-nek nevezte el és egy partíciója van, a `/dev/sdd1`. Erre az információra van szükségem.

Szükségem van egy helyre, ahová csatolhatom, hogy elérhessem, és a rendszernek van egy külön helye az eltávolítható meghajtókra – az `/mnt`. Ha ide csatolom, nem lesz helyem a továbbiak csatolására, tehát készítenem kell egy könyvtárat ezen belül (vagy alatta). Ehhez root-jogokra van szükségem.

su (írd be a root jelszavát)

mkdir /mnt/usb1 majd végül add ki a **mount** parancsot

mount /dev/sdd1 /mnt/usb1 Olvasata: „csatold ezt az eszközt ide”.

ls /mnt/usb1 Erre a kulcs tartalmát kilistázza a képernyőre.

A Linux-rendszer az adatokat aszinkron módon írja ki. Azaz az adatok nem kerülnek azonnal az eszköz-

re, ezért fontos, hogy az eszközt korrekten csatoljuk el, különben adatvesztés lehet. A `umount` parancs leválasztás előtt szinkronizálja a meghajtót.

Umount /dev/sdd1 Az eszköz most már biztonságosan eltávolítható.

Parancskimenet vezérlése

Amikor valami parancs kimenetét megkapod, az sosincs a kívánt sorrendben. A kívánalmaknak megfelelő átrendezéshez természetesen egy **sort** (*MS = sort*) (rendez) parancsot használunk.

A **sort** parancs egyike a Linux sok, szűrőnek hívott eszközének. Az adat átkerül egy szűrőhöz, aminek kimenete a megváltoztatott adat. Ez az alapvető parancssori műveletek egyike és egy része a Linux valódi erejének. A szűrő alkalmazásának szokásos módja, hogy az adatot biztosító parancsot egy csővezetéknek nevezett függőlegessel választjuk el egymástól.

A nem túl gyakran használt **tr** parancs arra van, hogy egy karakterkészletet lefordítson egy másikra. Egyik ilyen lehet a váltás kisbetűről nagyra. A parancsot szűrőként használják, az **-s** op-cióját nagyon hasznosnak találom, ami megkeresi az adott karakter többszörös előfordulásait és helyettesíti azokat eggyel. Például **cat szövegfájl | tr -s " "**.

A **cat** elküldi a fájlt a **tr** parancsnak, ami ezután eltávolítja az összes többszörös szóközt. Használd a **"\t"**-t a főleges tabulátorok eltávolítására. Ez az eredeti fájlt nem változtatja meg, csak a parancs kimenetére hat, ahogy az áthalad a szűrőn.

Néhány parancs olyan sok kimenetet produkál, hogy nehéz megtalálni az általad keresett információt. A kimenetet átcsővezetékhez teheted egy lapozóba, mint a **less** (*MS = more*), de akkor is át kell még nézned egy jó csomó értéktelen anyagot. A Linux rendelke-

zik grep nevű paranccsal (a név egy betűszó, de ez lényegtelen), ami megoldhatja ezt a kis problémát.

A **grep** (*MS = find*) parancs arra való, hogy egy szövegfájlban keresse a minta egyezéseit.

A **grep "Once upon a time" ~/Documents/*** átnézi a Documents könyvtár összes fájlját az adott kifejezés összes előfordulásaért.

Ha egy parancs kimenetét elküldjük a **grep**-nek, akkor azt nézi át a kifejezésért.

A **ps ax | grep firefox** sokkal egyszerűbbé teszi a nem válaszoló firefox id-jének megtalálását.

A **-v** opció megfordítja a keresést és azokat a sorokat mutatja, amik nem egyeznek.

Sok Linux-parancs szóköz, tabulátor, vessző és kettőspont tagolású oszlopok formájában adja az adatokat. Az adatok java érdektelen lehet, talán jó lenne levágni és csak a releváns adatokat megjeleníteni? Nos, a **cut** paranccsal megteheted, ami az adatokat karakter sorként, vagy mezőként látja, amiket egy karakter, vagy határoló (alapról Tabulátor) választ el. Az opció nélkül kiadott df parancs kimenete valahogy így fog majd kinézni:

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda1	26G	8.4G	16G	35%	/
/dev/sda5	352G	149G	186G	45%	/home
/dev/sdb6	391G	247G	124G	67%	/share
none	7.8G	124M	7.7G	2%	/tmp
/dev/sdd1	3.7G	613M	3.1G	17%	/mnt/usb

Ezek a mezők láthatóan tabulátorral tagoltak, ami a cut parancs számára alapbeállítás. Ha csak a Filesystem és Mounted oszlopok kellenek nekünk, akkor az **-f** opcióval azonosítva a mezőket kérhetjük a cut-tól az 1. és a 6. mezőt.

A **ps | cut -f 1,6** sajnos nem fog működni esetünkben, mert a határoló nem tabulátor, amiként mi gondoltuk, hanem a kimenetet szóközők töltik ki. Használhatjuk a **tr -s** parancsot a fölösleges szóközők eltávolítására és ezután a **cut -f** opcióját az elválasztó meghatározására.

df | tr -s ' ' | cut -d ' ' -f 1,6

filesystem	Mounted
/dev/sda1	/
/dev/sda5	/home
/dev/sdb6	/share
none	/tmp
/dev/sdd1	/mnt/usb

Vegyes parancsok

A terminál képernyője néha túlzsúfolt és idegesítő lehet a futó folyamatoktól.

Add ki a **clear** (tisztít) parancsot és üres képernyővel indíthatsz újra. (*MS = cls*)

A terminál-parancsok listája nem lehet teljes szövegszerkesztő nélkül. A szövegfájlok annyira fontosak a Linux működésében, hogy százával vannak parancsok azok manipulálására. Ezt a szöveget megírni és többször szerkeszteni kellett. Számos szövegszerkesztő érhető el Linux alatt, beleértve a különösen hatásos, de rettentően barátságtalan **vi** editort. A PCLinuxOS egy **nano**-nak (*MS = edit*) hívott, nagyon szép, egyszerű parancssori szövegszerkesztővel érkezik. Ha még nincs telepítve a te változatodban, nyisd meg a Synaptic-ot és add a rendszerhez.

Írd be **nano fájlnev** és ha az van, betölti. Ha nincs, akkor egy üres fájl nyit, készen arra, hogy tartalmat adj hozzá. A képernyő alján látható a menü, ahol a ^ azt jelenti, hogy nyomd le a <CTRL>-t és nyomd meg az azt követő betűt, hogy az adott műveletet végrehajtsd. <CTRL>-g némi segítséget ad fel.

Ha ki kell nyomtatnod egy fájlt és a nyomtatód telepítve van, akkor **lp az-én-fájlom** (*MS = print*) kinyomtatja. Ilyen egyszerű.

A Windows-felhasználók többsége ismeri a bolti **Winzip**, vagy **pkzip** tömörítő programokat. A Linux parancssori változata a **gzip**.

A **gzip az-én-fájlom** becsomagolja az alapbeállítások szerint a fájlt és a kiterjesztése alpról **.gz** lesz.

A **gzip -d az-én-fájlom.gz** az alapbeállítások szerint kicsomagolja a fájlt.

A **gzip** elfogad 1 és 9 közötti számokat opcióként, -1 gyorsabb és -9 agresszívabb tömörítésre utasít.

A **tar** a múltból való parancs, de még mindig nagyon hasznos. A **tar**, a **tape archiv** (szalagos archiválás) betűszava. Amikor a fájlokat mágneses szalagokra írták ki, fájlgyűjteményeket raktak össze későbbi visszaállításra. Ez még most is hasznos, noha a fájlokat többnyire „tar”-olják és tömörítik, majd a „tarball” más fájlrendszerekre, vagy az internetre kerül át.

Az alap kiterjesztése (az emberek számára) a **.tar**. A **tar** parancs most képes tömöríteni a fájlokat számos tömörítési eljárás használatával. Ezt általában olyan fájlkiterjesztés jelzi, mint a **.tar.gz**, vagy hasonló.

tar -cf file1 file2 létrehoz egy „tarball”-t a file1 és file2-ből.

A **tar -czf docs.tar.gz ~/Documents/** egy zip-pelt fájlt hoz létre a Documents könyvtár összes fájljából.

A **tar -tf docs.tar.gz** kilistázza az archívumban lévő fájlokat.

A **tar -xzf docs.tar.gz** kicsomagolja az archívum fájljait.

Ennek a parancsnak olyan sok opciója van, hogy vagy tényleg el kell olvasnod a leírását, vagy a használatban ezeknél az egyszerű példáknál maradsz, ami feltehetően minden, amire szükséged lehet.



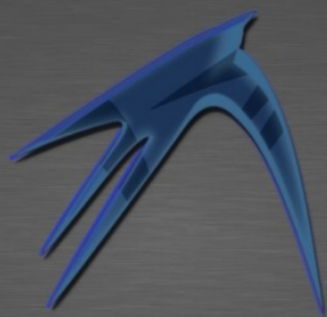
Open Source Initiative

Want To Help?

Would you like to help with the PCLinuxOS Magazine? Opportunities abound. So get involved!

You can write articles, help edit articles, serve as a "technical advisor" to insure articles are correct, create artwork, or help with the magazine's layout.

Join us on our [Google Group mailing list](#).



Lightweight X11
desktop environment



Screenshot Showcase



Posted by francesco bat, May 16, 2013, running KDE.