

Windows migrálás: terminál - a terminál ereje

PCLinuxOS Magazine – 2013. május

Írta: Pete Kelly (critter)

Az előző cikkben bemutattam, hogyan érd el a terminált, hogyan gépelj be néhány egyszerűbb parancsot, eredményükként milyen látványt várhatsz el. Semmi különös, vagy tetszetős, de működtek. Megmutatta, hogy a Linux alatt a terminál használata nem fekete mágia. Nem olyan, csak a komputer zseniknek fenntartott, vagy nagyon bonyolult dolog, hanem valami, amit bárki használhat, aki képes néhány parancsot beírni a billentyűzeten.

Eddig csak két parancsot használtál, az „ls”-t és a „pwd”-t, de biztosan észrevetted, noha ezek furcsa szavak, mégsem misztikus varázsigék a „Roxfort”-ból. Ehelyett egyszerű módja a rendszer utasításának, hogy végrehajtsa meghatározott műveletet néhány fájlra, vagy adaton, illetve adatokat szolgáltatson. Semmi boszorkányság sincs a terminál használatában, de kicsit tanulni kell.

Hogy érezhető előnyöd származzon a terminál használatából, meg kell szoknod a használatát és képesnek kell lenned műveleteket legalább annyira könnyen végrehajtani, mint a grafikus környezetben. Nem kell tökéletesen megtanulnod, csupán ne essen nehezedre a használata. A Linux által használt grafikus rendszer, az X ablakozó nem a szöveg alapú parancssor leváltására készült, hanem annak kiegészítésére, és ez még ma is igaz. Jóllehet nagyon jól elboldogulhatsz kizárólag grafikus felületen, ám mindkét interfész használatával férsz hozzá a Linux operációs rendszer teljes erejéhez és rugalmasságához.

A háttérben a Linux eltér a Windows-tól és egy kis időbe beletelik, amíg hozzászoksz ezekhez az eltérésekhez. Megmutathatom neked mi érhető el és hogyan használd, de rád kell hagyjam annak

eldöntését, hogy milyen messzire mész a lehetőségek feltárásában. Rengeteg hivatkozási forrás van, hogy az itt bemutatott témákban mélyebbre áss. A cikk célja, hogy tisztán lásd mi lehetséges és a „Linux – operációs rendszer” hogyan működik az általában látható grafikus asztal mögött.

Ez ne szegje kedvedet; egyáltalán nem olyan bonyolult, mint ahogy sokszor leírják. A sok száz elérhető parancs közül néhányat alaposan megismeresz, és új szinten kezded uralni a Linux-környezetet. A terminál igazi ereje nem az összes parancs ismeretében rejlik, hanem a használatuk mikéntjében ismeretében. A terminálban teljes ellenőrzésed lehet a parancsok összes jellemzője felett és nem csak egy, a magasabb szintű grafikai alkalmazás tervezője által szükségesnek tartott készlete felett. Az elvárt eredmény elérése érdekében a meglévő parancsok kombinálásával elkészítheted a saját parancsodat. Mindig egynél több mód van valami végrehajtására és semmi sem olyan élvezetes, mint, hogy a „saját módszerrel” csináld. Ebben rejlik a terminál igazi ereje.

A Linux más

Amikor a Gnu/Linux alapját képező Unix operációs rendszert kialakították néhány nagyon fontos, jövőbelátó elvet lefektettek. Néhány hasonlóan fontos koncepciót is bevezettek.

1. Minden eszközt és programot úgy kell megtervezni, hogy csak egy dolgot csináljon, de azt jól.

2. Legyen mód a különféle eszközök együttes alkalmazásának egy adott adatcsomagra. Ez vezetett a „pipe” (csővezeték) bevezetéséhez, hogy lehetővé tegye az adatáramlást a különféle eszközök és szű-

rők között, összetett, átmeneti behatást biztosítva az adatokra. Adatfolyamként, vagy stream-ként ismert.

3. Az eredményként módosított adatfolyamot át kell irányítani egy, vagy több más helyre és a bemeneti adat pedig olvasásra elérhető kell legyen bármilyen kiválasztott forrásból.

4. A hiba- és rendszer információknak elkülönített és párhuzamos kimeneti útvonal szükséges, ami naplózható, jelenthető, vagy az adatkimeneti folyamattal kombinálható.

Ehhez kapcsolódik még, hogy a Linux/Unix mindent fájlként kezel – még az egeredet is – és lassan kezdheted érteni, hogy a Linux más kávéház. A tény, hogy több mint négy évtizede létezik (visszavezetve a Unix-gyökerekre), és sok millió ember, illetve több ezer nagy szervezet használta ez idő alatt, bizonyítja megbízhatóságát és hasznosságát.

Ha használtál Windows parancssort, vagy az MS-DOS-t, akkor feltehetően használtad a helyettesítő karaktereket „?” és „*” a fájlnevekben lévő ismeretlen karakterek helyett és talán még írtál magad is, vagy használtál kötegelt (batch) fájlokat. Ha használtad a Windows új „powershell”-jét, akkor már találkoztál „szabályos kifejezésekkel”, parancshéjjal és magas szintű parancsnnyelvel. A Linux az első naptól kezdve rendelkezik ezekkel és sokkal több egyébbel.

E bevezető cikkek során bemutatok néhány gyakran használt parancsot és azok kombinálásának módját. Kkitérek néhány fontosabb olyan elv magyarázatára, amivel a Windows használata során nem találkoztál.

Indulásként megmutatom, hogy mit lehet kezdeni az egyszerű **ls** parancssal és másik három, négy nagyon egyszerű parancssal.



Légy egyszerű, csináld jól

Az **ls** parancs kilistázza egy (vagy több) könyvtár tartalmát. Ennyi. Az **ls** parancsot erre tervezték és ezt is csinálja. Mindazonáltal ezt nagyon jól csinálja. Bizonyos opciók alkalmazásával beállíthatjuk a kimenetet. Kiterjeszthetjük, vagy korlátozhatjuk a parancs kimenő adatmennyiségét. Megjeleníthetjük különféle formátumokban. Rendezhetjük a kimenetet. Belefoglalhatjuk az alkönyvtárakat, és felvehetünk, vagy kizárhatunk bizonyos típusú fájlokat. Röviden, kérhetjük pontosan azon, és csakis azon adatok megjelenítését, amire szükségünk van.

Az **ls** parancs beírására a terminál a saját home könyvtár tartalmának kilistázásával válaszolt. Nos, csak a tartalom egy részét listázta ki. Néhány fájl és könyvtár rejtve van. Minden rosszindulat nélkül. Ezek a te fájljaid és jogod van látni őket. Azért maradtak rejtve, mert ritkán kell látni. Elrejtésük a normál használat mellett csökkenti káoszt.

A rejtett fájlok neve ponttal kezdődik. Megjelenítésükhöz módosítanunk kell az **ls** (listázás) parancsot egy opció hozzáadásával. Sok **opció** van, és általában egy, vagy két kötőjellel kezdődnek. Amelyiket keressük az „-a”. Az „**ls -a**” „list all” (mind listázása).

```

linux_user : bash
[linux_user@localhost ~]$ ls -a
./          Documents/  .mdk-menu-migrated  .vboxclient-display.pid
./          Downloads/ .Movies/           .vboxclient-draganddrop.pid
.bash_logout .gconf/     .Music/            .vboxclient-seamless.pid
.bash_profile .gnupg/    .Pictures/         .wine/
.bashrc      .gtk-bookmarks .pulse/            .Xauthority
.config/     .gtkrc-2.0 .pulse-cookie     .xbindkeystrc
.dbus/      .kde4/     .Templates/       .xine/
Desktop/    .kderc    .tmp/             .xsession-errors
.dmrc      .local/   .vboxclient-clipboard.pid
[linux_user@localhost ~]$
    
```

Most már láthatod, hogy miért voltak rejtve. Többségük egyáltalán nem jelent számodra semmit – egyelőre!

Az **ls** egy másik hasznos opciója a „-l”, ami az „**ls**”-nek hosszú lista (long list) készítését írja elő.

```

linux_user : bash
[linux_user@localhost ~]$ ls -l
total 36
drwxr-xr-x 2 linux_user linux_user 4096 Oct 27 07:08 Desktop/
drwx----- 2 linux_user linux_user 4096 Nov 17 05:33 Documents/
drwx----- 2 linux_user linux_user 4096 Jan 19 2010 Downloads/
drwx----- 2 linux_user linux_user 4096 Jan 19 2010 Movies/
drwx----- 2 linux_user linux_user 4096 Jan 19 2010 Music/
drwx----- 2 linux_user linux_user 4096 Jan 19 2010 Pictures/
drwxr-xr-x 2 linux_user linux_user 4096 Nov 4 11:14 Templates/
drwx----- 6 linux_user linux_user 4096 Nov 4 11:14 tmp/
drwxr-xr-x 2 linux_user linux_user 4096 Nov 4 11:14 Videos/
[linux_user@localhost ~]$
    
```

Ez sokkal több információt adott, de kizárta a rejtett fájlokat. Hogy belefoglaljuk az opciókat a következő módon kombinálhatjuk.

Írj be „**ls --help**”-et és megláthatod, hogy mennyivel több opció használhatsz. Az opciók közül néhány számodra érdektelen. Egyszerűen ne foglalkoz velük. Ha nem érted őket, akkor valószínűleg nincs szükséged rájuk. Kísérletezz azokkal, amik érdekelnek, teljesen biztonságos, mivel a parancs csak

a képernyőre ír ki információkat. A legtöbb parancshoz megvan ez az alap súgó.

Adatáramlás és csővezetékek

Amikor begépelted az „**ls --help**”-et, némi

```

linux_user : bash
[linux_user@localhost ~]$ ls -al
total 224
drwx----- 20 linux_user linux_user 4096 Nov 4 13:46 ./
drwxr-xr-x  5 root      root      4096 Nov 4 11:13 ../
-rw-r--r--  1 linux_user linux_user  24 Jan 18 2010 .bash_logout
-rw-r--r--  1 linux_user linux_user 191 Jan 18 2010 .bash_profile
-rw-r--r--  1 linux_user linux_user 458 Oct 27 07:08 .bashrc
drwxr-xr-x  3 linux_user linux_user 4096 Nov 4 11:14 .config/
drwx----- 3 linux_user linux_user 4096 Nov 4 11:14 .dbus/
drwxr-xr-x  2 linux_user linux_user 4096 Oct 27 07:08 Desktop/
-rw-----  1 linux_user linux_user  26 Nov 4 11:14 .dmrc
drwx----- 2 linux_user linux_user 4096 Nov 17 05:30 Documents/
drwx----- 2 linux_user linux_user 4096 Jan 19 2010 Downloads/
drwx----- 4 linux_user linux_user 4096 Nov 4 11:14 .gconf/
drwx----- 3 linux_user linux_user 4096 Nov 4 11:14 .gnupg/
-rw-rw-r--  1 linux_user linux_user  162 Nov 4 11:14 .gtk-bookmarks
-rw-r--r--  1 linux_user linux_user 252 Jun 24 2011 .gtkrc-2.0
[linux_user@localhost ~]$
    
```

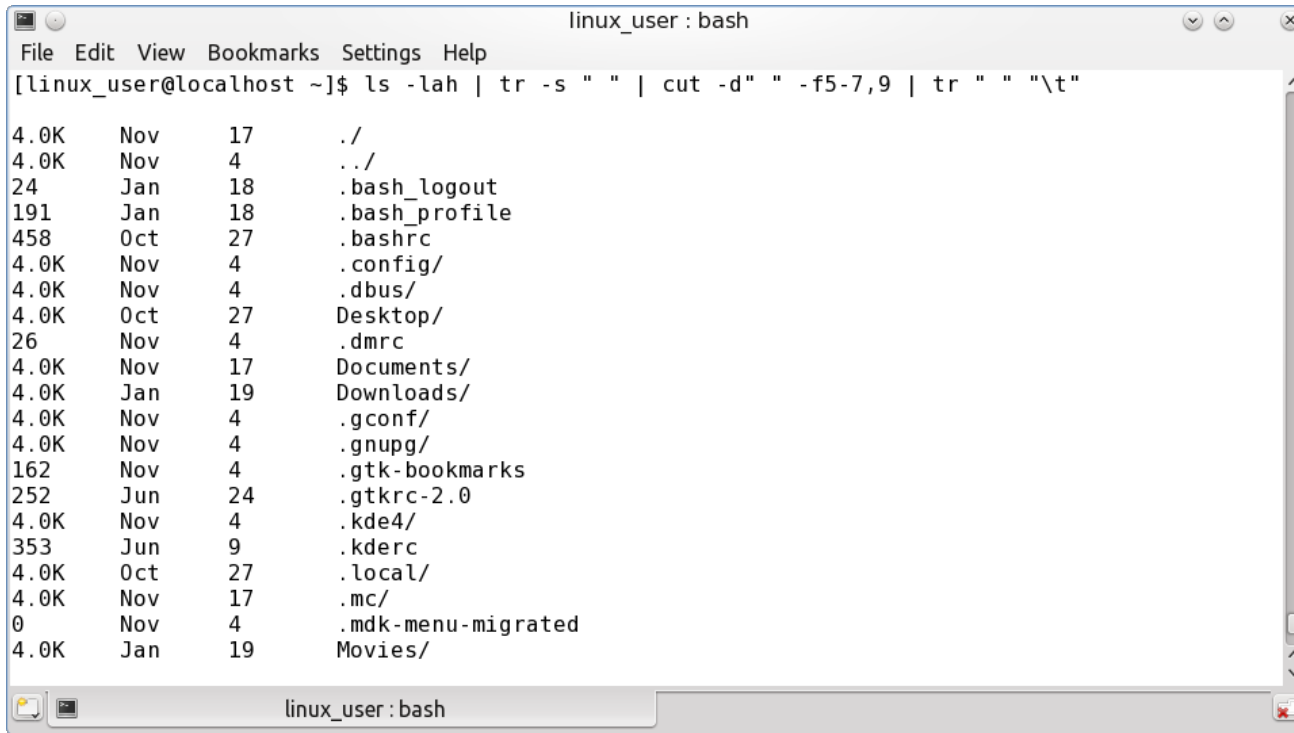
információ lefut a képernyőn, mivel túl sok van belőle. Az Unix szabványos eszköze a terminál képernyőjén szövegfájlok megjelenítésére a „**more**” (több). Ez beolvassa a fájlból az adatokat és megjelenít egyszerre egy képernyőnyit belőle, várva a szóköz billentyű lenyomását. Amikor lenyomod a szóközt, egy újabb oldalnyit, vagy többet ad fel. Sajnos nem volt mód a visszalépésre, hogy újra elolvassd a szöveget, ezért egy új eszközt vezettek be ennek lehetővé tételére. Mi másnak is nevezheték volna el, mint „**less**”-nek (kevesebb) (a **more** modernebb verziói lehetővé teszik a kétirányú görgetést, de több okból is javaslom a **less** használatát. Végül is a „a kevesebb (less) több (more), többé-kevésbé”). Ezen parancsok használatához bevezetjük a függőleges vonalat („|”), vagy „csővezeték” jelet. Az első parancs kimeneti adatai az igéyek szerint áttöltődnek a második parancsba.

ls --help | less

Most már használhatjuk a nyíl és a page-up/page-down (lapozó) billentyűket, hogy az információkat a saját helyükön olvashassuk. Kilépés „q”-val.

Ilyen módon összetett parancsokat lehet képezni, parancsok és szűrők hozzáadásával, hogy a pontosan elvárt eredményt kapjuk. Nézd meg a következő példát.

Az **ls** parancs olyan opciókat kapott, hogy készítsen részletes listát az összes fájlról, a méreteket emberi olvasásra alkalmasan. Ez becsővezve a **tr**



```
linux_user : bash
File Edit View Bookmarks Settings Help
[linux_user@localhost ~]$ ls -lah | tr -s " " | cut -d" " -f5-7,9 | tr " " "\t"
4.0K  Nov  17  ./
4.0K  Nov  4   ../
24   Jan  18  .bash_logout
191  Jan  18  .bash_profile
458  Oct  27  .bashrc
4.0K  Nov  4   .config/
4.0K  Nov  4   .dbus/
4.0K  Oct  27  Desktop/
26   Nov  4   .dirc
4.0K  Nov  17  Documents/
4.0K  Jan  19  Downloads/
4.0K  Nov  4   .gconf/
4.0K  Nov  4   .gnupg/
162  Nov  4   .gtk-bookmarks
252  Jun  24  .gtkrc-2.0
4.0K  Nov  4   .kde4/
353  Jun  9   .kderc
4.0K  Oct  27  .local/
4.0K  Nov  17  .mc/
0    Nov  4   .mdk-menu-migrated
4.0K  Jan  19  Movies/
```

parancshoz, hogy a szóközöket vonja össze. Ezután átkerül a cut (vágás) parancshoz, az 5., 6., 7. és 9. oszlop kivételével a többi eltávolítására. Végül ismét visszakerül a tr parancshoz, hogy a szóközöket tabulátorra cserélje. Ez elég komplikált példa, de bemutatja a parancsok láncolásának elvét a célból, hogy pont a kívánt eredményt kapjuk. Tekints el most a részletektől; próbáld csak az alapelvet megérteni.

Átírányítás

A Linux-parancsoknak egy bemenete és egy kimenete van, noha első pillantásra úgy tűnhet, hogy nincs így. A cd parancsnak, ami az aktuális munkakönyvtár másikkra váltására szolgál, úgy tűnik nem rendelkezik ki- és bemenettel. Ám mindkettő van, még ha a menetet nem is használjuk. Az, hogy

ez a be- és kimenet honnan jön, vagy hová megy, tetszőleges. Amikor a be-, vagy a kimenet nincs meghatározva, akkor a parancs az úgynevezett „szabványos bemenet”-et, „stdin”-t, ami általában a billentyűzet, illetve a „szabványos kimenet”-et, „stdout”-ot használja, ami általában a képernyő. Az stdout-ra láttunk példát, az ls parancsot. Amikor a csővezeték jelzését „|” használtuk, akkor a parancs kimenete átkerült egy másik parancshoz.

Használhatjuk még a „<” és a „>” karaktereket is az stdin és stdout megfelelő átírányítására. Ez így működik. Tegyük fel, hogy a Dokumentumok könyvtárunk fájllistáját menteni szeretnénk egy doc-list nevű fájlba ahelyett, hogy képernyőre íránk ki. Ekkor kiadhatjuk az **ls Documents > doc-list** parancsot.

Ha a fájl nem létezik, elkészíti és ezután az ls parancs kimenetét abba írja bele. Ha a fájl létezik,

tartalmát felülírja az új adatokkal. Hogy ezt elkerüljük, két „nagyobb mint...” jelet kell használnunk, vagyis **ls Documents >> doc-list**.

Az új kimenet ekkor a fájlhoz hozzáíródik. Ha rendezni szeretnénk a fájl tartalmát, akkor a fájl betáplálhatjuk a rendezésünket elvégző parancs bemenetére. Nem meglepő, de a parancs neve „sort” (rendezés).

sort < doc-list

Ez rendezi a tartalmat, de a képernyőre írja ki, az eredeti fájlt érintetlenül hagyva. Hogy rendezett fájlt kapjunk egy lépésben, kombináljuk az átírányítást (<, >) és a csővezetékvezést (|) valahogy így:

ls Documents | sort > doc-list

Az **ls** létrehozza a kimenetet, ezt átcsővezetékvezési a **sort** parancsnak, ami ezután rendezi az adatokat és a kimenetet átírányítja a cél fájlba.

Hibaüzenetek

A Linuxban az **stdin** és **stdout** fájlleíróként ismert, egy kicsit olyanok, mint az adathordozó csatornák. Számos fájlleíró lehet aktív, de ezeket, a fájlleíró 0 és 1 néven hivatkozott fájlleírókat az operációs rendszer foglalja le. A harmadik, a rendszer számára lefoglalt fájlleíró a standard error (szabvány hiba) „**stderr**” néven ismert és a 2 számot viseli.

A Linux parancssora úgy lett programozva, hogy az adatokat az **stdout**-ra küldi, a parancs részét nem képező hiba-, illetve egyéb üzenetek az **stderr**-t célozzák meg kimenetként. Alapból a **stderr** minden üzenetet a terminálablakba ír ki, de ez sokszor zavaró. Valószínűleg nem szeretnéd a hibaüzeneteket az érvényes adatokkal összekeverni, és ha felügyelet nélkül futtatsz hosszú folyamatokat, pl. mentést, valószínűleg szeretnéd a hibaüzeneteket naplófájlba menteni, amiket megnézhetesz a folyamat végén.

Könnyen megoldható az **stderr**, ill. a file descriptor 2 átirányításával. Ha naplózni akarsz a kimenetet, egy ehhez hasonló parancsot használj:

command 2> logfile

command 2> logfile > datafile

Az első a parancs minden hibaüzenetét a logfile-ba írja, miközben a rendes kimenet a képernyőre kerül. A második a hibákat a logfile-ba, az adatokat a datafile-ba küldi. Kimenet nem jelenik meg a képernyőn.

Hogy egyszerűen megszabadulj a hibaüzenetektől, a Linux-nak van egy **/dev/null** nevű különleges eszköze. Noha ha az eszköz (device) általában fizikai, mint a billentyűzet, vagy a hálózati kártya, a Linux-nak van néhány fizikailag meg nem testesülő eszköze. A **/dev/null** is egy ilyen. Minthogy a Linux mindent, az eszközöket is fájlként kezeli, írhatod a **/dev/null**-ra, ami a kapott adatokat visszavonhatatlanul eldobja.

command 2> /dev/null

A héj

Az egész, egy nagyon rugalmas és nagyteljesítményű eszköztárt ad, amit a fájljaiddal és a rendszer fájlokkal végzett munkában használhatsz. A szöveg-alapú környezetet, amiben ezeket az eszközöket használod, egy héjnak nevezett program vezérli, lefordítván minden beírásodat, mielőtt eldöntené, hogyan hajtsa végre. Számos különböző héj érhető el, de a legelterjedtebb a „**bash**”-nek nevezett és a PCLinuxOS-nek az alap shell-je (héja).

A bash erőteljes és barátságos. A sok év alatt sok tulajdonságot adott hozzá, hogy segítsék és gyorsítsák a terminálban végzett munkádat. Az ilyen támogató tulajdonságok egyike a **parancskiegészítés**.

Írd be, hogy „**ls D**”.

Nyomd le a Tab billentyűt. A bash elég okos ahhoz, hogy felismerje az **ls** parancsot írtad be és most megpróbálja bevinni egy fájlt, vagy könyvtár nevét, hogy kilistázza az érintett adatokat. A bash-héj a Tab billentyűre kilistázza az adott helyről látható, összes nagy d-vel kezdődő nevű fájlt és könyvtár nevét. A rendszeremen a Desktop-ot, a Directories-t, a Downloads-ot és a Documents-et listázza ki. Egy további **o** beírásával és a Tab leütésével a lista kettőre szűkül, minthogy a „**Desktop**”-ra nem illik rá. Írj be **c**-t és csak egyetlen lehetőség marad, a „**Documents**” és így a tab ismételt leütésével automatikusan kiegészíti nekem a prompt-nál, készen az Enter lenyomására, ami elkészíti a listámat.

```
File Edit View Bookmarks Settings Help
[linux_user@localhost ~]$ ls D
Desktop/  Documents/ Downloads/
[linux_user@localhost ~]$ ls Do
Documents/ Downloads/
[linux_user@localhost ~]$ ls Documents/
```

Ez egy valódi időmegtakarító, és segíthet, amikor nem vagy biztos egy parancs nevében. Például tudom, hogy van még egy parancs ami **ls**-sel kezdődik, és a gép processzorairól ad tájékoztatást nekem.

Írd be, hogy „**ls**” és üss Tab-ot. Minthogy ez az első dolog a prompt után, a bash felismeri, hogy parancsot próbálok beírni, és megjeleníti az összes **ls**-sel kezdődő nevű programot, amit csak a rendszer ismer. Ami nekem kell az az „**lscpu**” (gondolhattam volna!). Beírok egy **c**-t, leütöm a Tab-ot és az „**lscpu**” parancs neve a prompt-nál kiteljesedik nekem. Ezután lenyomom az Enter gombot és itt is vannak az információim.

Ha megismételnél egy korábbi parancsot, nem kell újra beírnod. A „fel nyíla” lenyomására az előző parancs megjelenik, készen az átírára, vagy használatra. A fel, vagy a le nyíl lenyomásával végiggörgetheted a korábban begépelte parancsokat.

```
linux_user : bash
File Edit View Bookmarks Settings Help
[linux_user@localhost ~]$ ls
ls                lsb_release      lshal            lsmod            lspcidrake
lsattr            lscpu            lsinitrd         lsnetdrake       lspcmcia
lsbininstall     lsd              lskat            lspci            lspgpot
[linux_user@localhost ~]$ lscpu
Architecture:      i686
CPU(s):            2
Thread(s) per core: 1
Core(s) per socket: 2
CPU socket(s):    1
Vendor ID:         GenuineIntel
CPU family:        6
Model:             42
Stepping:          7
CPU MHz:           3292.524
L1d cache:         32K
L1i cache:         32K
L2d cache:         6144K
[linux_user@localhost ~]$
```

Van még egy másik lehetőség, a visszafelé keresés. Használatához nyomd le a **Ctrl+R** billentyűket, majd kezdjed begépelni a parancsot, amire emlékszel. Gépelés közben a leginkább megfelelő parancsok kerülnek a prompt-hoz. Amikor megtaláltad, amit kerestél, nyomj Enter-t a parancs végrehajtásához, vagy a bal, illetve a jobb nyíl gombot a szerkesztéshez.

Végül, ott vannak a parancssori előzmények. Gépeled be, hogy „**history**” („**his**”, majd Tab a kiegészítéshez). Kapsz egy számozott listát a korábbi parancsaidról. Írj egy felkiáltójelet (!), amit az ismételni szándékozott parancs száma követ, ezután nyomj Enter-t, az azonnali parancsvégrehajtáshoz.

Ha úgy találod, hogy azonos módon használod egy parancsot, azonos opciókkal, akkor készíthetsz hozzá egy ún. „alias”-nak nevezett rövid hívást, hogy spórolj a gépeléssel.

Emberi olvasásra alkalmas fájl mérettel és aszerint rendezett lista készítésére az „**ls -lhS**” parancs szolgál.

Noha ez nem bonyolult, mindig emlékezni fogsz, hogy az **s** nagy? A parancs „**alias lh='ls -lhS'**” elérhetővé teszi az **lh** parancs egyszerű

begépelésével. Sajnos, ezt minden egyes alkalommal meg kell csinálni, ahányszor terminált nyitasz és elindítod a bash héjat. Hogy ezt automatikusan megtörténjen, hozzáadhatod a home könyvtárban lévő **.bashrc** nevezetű rejtett fájlhoz.

Sok szövegszerkesztő van a Linux terminálok részére és a PCLinuxOS-nek egy kiváló, **nano** nevezetű van.

Írd be **nano ~/.bashrc** (itt van az a hullámvonal ismét) és a szerkesztőben a te fájlod nyílik meg. Itt lesz már néhány alias, amiket a PCLinuxOS fejlesztői készítettek és jó hely arra, hogy elhelyezd a sajátodat közéjük, de ez valójában nem számít. Írd be a saját alias-odat egy üres sorba, majd nyomj **Ctrl+X**-et. Válaszolj igennel, amikor a mentést kérdezi. Indítsd újra a terminált és az új parancsod mindig elérhető lesz.

Hogy lásd, a jó emberek, a PCLinuxOS fejlesztői milyen más alias-okat adtak hozzá neked írd be az „**alias**” parancsot.



Screenshot Showcase



Posted by Texstar, April 21, 2013, running e17.

