

10 hasznos parancs középhaladó Linux felhasználóknak

Minél többet tanulsz a Linuxról, az annál erősebb lesz általad. Néhány parancsot mutatunk be, ami nagyon hasznos lehet a középhaladó Linux felhasználó számára

COLLABORATORS			
----------------------	--	--	--

	<i>TITLE :</i> 10 hasznos parancs középhaladó Linux felhasználóknak		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		2014. január 7.	

Tartalomjegyzék

1. find	1
2. grep	1
3. man	1
4. ps	2
5. kill	2
6. whereis	2
7. service	3
8. alias	3
9. df	4
10. rm	4

A példák listája

1.1. Windows™-ból érkező fájlok végrehajtási jelzőjének eltávolítása	1
5.1. Az összes mc folyamat megszakítása	2
6.1. mc fájljainak keresése	3
7.1. Fut-e a xinetd	3
8.1. df parancs szokásos opciói aliasként	3
10.1. Rendszergazda rémálma	4

Kivonat

Eredeti cikk: [10 Useful Commands For Middle Level Linux Users!](#)

2014. január 4., szombat: A középhaladó **Linux** felhasználó nem olyan felkészült, mint egy szakértő felhasználó, de van jó pár dolog, amit nem használ a rendszerén. Ennek következtében sokszor elég nehéz lesz emlékeznie mindezekre a parancsokra. Íme, a leghasznosabb 10 parancs, amit akkor fogsz használni, ha elérted **Linux**ban ezt a szintet.

Nem tudtam megállni, hogy az egy-egy mondatos ismertető mellett néhány megjegyzést ne tegyek. Az eredeti anyagban ugyanis a szűkszavúság az érthetőség rovására ment – szerintem.

Megjegyzéseim, ehhez hasonlóan, beljebb helyezkednek el, így nyugodtan át lehet lépni, ha valakit nem érdekelne.



Ez a cikk a Linuxon futtatható parancsokról szól!

Az itt leírtak elsőrendűen a **Linuxon** futó parancsokra vonatkozik. Ha más UNIX™-ot használ, **feltétlenül nézzen utána**, a használt rendszeren hogy működik, mert az eltérések beláthatatlan következményekkel járhatnak!

1. find

Ez a program akkor használatos, ha egy adott mappában szükséges fájlokat keresni. A szülő mappában kezd, utána felkeresi az almappákat. A `-name` opció a keresést kisbetű/nagybetű érzékeny módon végzi, míg a `-iname` erre érzéketlen.

Ez a nagyon sokrétű parancs nem csak keresésre, hanem meghatározott keresési szempont alapján meghatározott művelet elvégzésére is alkalmas:

Example 1.1 Windows™-ból érkező fájlok végrehajtási jelzőjének eltávolítása

```
1 #! /bin/sh
2 # [...]
3 cd $1                Az első paraméter egy mappa
4 find . -exec chown -R $2:$3 {} \;    2. és 3. paraméter: felhasználó és csoport ←
5 find . -type f -exec chmod -R 644 {} \;    Jogosultságok beállítása fájlra
6 find . -type d -exec chmod 755 {} \;    Mappák jogosultságai: keresés engedélyezve ←
```

A példa azt mutatja be, hogyan tud a `find` rekurzívan beállítani fájl (és mappa) attribútumokat. A `find` lehetőségeit a kézikönyv oldal (`man find`) bemutatja. Nagyon hasznos példatár lehet: [35 Practical Examples of Linux Find Command](#).



megjegyzés

A közölt példa nem a teljes szkript (ezt jelképezi a `# [...]` sor), ugyanis az elején alapos paraméter ellenőrzések vannak. Ez a tárgy szempontjából itt közömbös.

2. grep

A `grep` parancs használható arra, hogy adott fájlból listázza azokat a sorokat, amelyekben a megadott karakterlánc vagy szó szerepel.

3. man

a „man” parancs a rendszer kézikönyv oldalait adja. Adott parancs on-line dokumentációját mutatja be.

Habár ma már egyre elterjedtebb a HTML-alapú dokumentáció, a jó öreg `man` parancs még mindig nélkülözhetetlen. Sok esetben a kézikönyv oldalt alakították át böngészőben nézhető változatra: [man\(1\) - Linux man page](#). Ugyanitt megnézhető az összes kézikönyv oldal: [Linux man pages](#). A lap alján a keresett parancs első betűjére kattintva annak kézikönyv oldala lesz látható.

Aki szívesebben nézi a grafikus felületen rendszer paranccsal a kézikönyv oldalakat, ismerkedjen meg az `xman` programmal.

4. ps

Ez a folyamat-parancs megmutatja az összes futó folyamat állapotát egyedi azonosító szerint, amit PID-ként (*Process identifier* - *folyamat azonosító*) ismert.

Ez a parancs már egy korábbi írásban is szerepelt, ott részletesebben. Magyar fordítása: [10 rendszer monitorozó parancs Linuxon](#), 5. ismertető.

5. kill

Ez a parancs arra használható, hogy megszakítsunk („megöljünk”) vele egy olyan folyamatot, ami nem válaszol, vagy már nem használjuk. Mindössze a folyamat azonosító vagy PID ismeret szükséges hozzá. A folyamat azonosító megismeréséhez futtatni kell a `ps -A` parancsot `grep`-pel (`ps -A | grep <folyamatneve>`

Általában akkor van szükség a `kill` parancsra, ha egy program önállósítja magát és nem hajlandó kommunikálni. A használathoz szükséges ismerni a megszakítási paramétert is, mert a teljes szintaxis: `kill-<szignálnevevagyszáma>PID`. A leírások általában felsorolják, így a részletesebb megismerésükhöz érdemes felkeresni: [Kill Commands and Signals](#). Az egyes szignálok listázása (természetesen magyarázat nélkül) a `kill-l` paranccsal lehetséges.

Két lehetőségre külön is felhívnam a figyelmet:

- a `killall` paranccsal minden megadott nevű folyamat megszakítható:

Example 5.1 Az összes mc folyamat megszakítása

```
killall mc
```

- Ma már szinte teljesen átvette a szerepet a grafikus felület. „Lefagyott” program kilövéséhez nem feltétlenül szükséges terminált/konzolt nyitni és megkeresni a folyamat azonosítót, általában használ, ha az egérmutatót az adott program ablakába mozgatjuk, itt `<Ctrl>+<Alt>+<Esc>` billentyű-kombinációt nyomunk. Megjelenik egy halálfej, most az `<Enter>` megnyomására megszakad a program futása. Ha meggondoltuk magunkat `<Esc>` lenyomásával a megszakítást szakítjuk meg :-).

6. whereis

Ha szükséges egy program binárisa, forrása, kézikönyv oldala helyének ismerete, használjuk a `whereis` programot.

A program a keresést a szokásos **Linux** útvonalakon (esetleg a `PATH` környezeti változóban megadott helyeken keres. Ha nem adunk meg hozzá opciót, akkor minden megfelelő fájlnevet kiír, például

Example 6.1 mc fájljainak keresése

```
$ whereis mc
mc: /usr/bin/mc /etc/mc /usr/lib64/mc /usr/share/mc /usr/share/man/man1/mc ↔
    .1.bz2
```

A program használatához elegendő a kézikönyv oldalát megismerni (`manwhereis`).

7. service

Ez a program egy adott szolgáltatás indítása, leállítása vagy újra indítása felügyeletére való. Nem kell a rendszert újraindítani a szolgáltatás leállításához, indításához, vagy újraindításához.

Ez a lakonikus megfogalmazás mindent elmond. A program kézikönyv oldala (`manservice`) sem sokkal többet ír. Az egyes futási szintek kezeléséhez, egy program, szolgáltatás betöltéskor való beállítása vagy éppen kikapcsolása a különböző futási szinteken a `chkconfig` programmal lehetséges.

Example 7.1 Fut-e a `xinetd`

```
# service xinetd status
A(z) xinetd fut (PID = 3500) ...
```

Ha nem létező szolgáltatást adunk meg, a hibüzenet mellett egy rövid segítséget is kapunk (mintha `service-h` parancsot adtunk volna ki).

8. alias

Ez a shell beépített parancsa, ami arra alkalmas, hogy egy rendszeresen használt hosszú parancsok új nevet adjunk.

Most a MINT community egy lapját kapcsolom ide: [Use alias to create shortcuts for commands](#). Még kevés angol tudással is érthető, mit kíván szemléltetni.

Ha az összes definiált alias-t szeretnénk látni, a parancsot paraméterek nélkül kell kiadni: `alias`, ha egy konkrét alias tartalmára vagyunk kíváncsiak, akkor ezt kell paraméternek adni, de utána már nem szabad írni semmit. Például:

Example 8.1 `df` parancs szokásos opciói aliasként

```
$ alias df
alias df='df -h -x supermount'
```

9. df

Ezzel a programmal lehet a lemez kihasználtságot megállapítani. Nagyon hasznos parancs mind a felhasználó, mind a rendszer adminisztrátor számára.

Ha particionált a lemez, külön kiírja a partíciók foglaltsági adatait. Az előzőekben bemutatott aliasban (`df -h -x supermount`) az egyes opciók jelentése:

- `-h`: „human readable”, azaz könnyen érthető érték megjelölést használ (pl. Gb)
- `-x`: az ezt követő felsorolás azokat a típusokat tartalmazza, amelyeket ki kell hagyni a kijelzendők közül; jelen esetben a „supermount”

10. rm

Ez a parancs arra szolgál, hogy teljes fájlokat és mappákat tudjunk a rendszerből törölni.

Elég veszélyes parancs, ugyanis a **Linux**ban alapértelmezetten nincsen undelete lehetőség. éppen ezért szokták azt az alias definiálni, ami tartalmazza a `-i` kapcsolót, mert így minden egyes törlendő fájlra rákérdez (`-i`: interactive=always - mindig kérdezzen rá). Ezt ellensúlyozni lehet a `-f` kapcsolóval (`-f`: kényszerített mód, ha nem találja a törlendő fájlt, nem küld hibüzenetet, emellett jó néhány kapcsoló, így pl. a `-i` hatását is felülírja).



FONTOS! FONTOS! FONTOS! FONTOS! FONTOS

Nehogy kipróbáld a következő példában szereplő parancsot!!! Következménye ugyanis a teljes rendszer kitörlése!!!

Example 10.1 Rendszergazda rémálma

```
# rm -rf /*
```

Ezzel a rendszerrel a továbbiakban „nem lesz gond”. A parancs ugyanis kérdés nélkül (`f`) rekurzívan (`r`) kitöröl minden fájlt, kezdve a gyökérmappával (`/`).

Ez a parancs magát egy könyvtárat nem töröl ki, csak a tartalmát. Üres könyvtár törlésére az `rmdir` parancs szolgál.
