

Futó folyamatok kezelése terminálból

PCLinuxOS Magazine – 2013. május

Írta: Antonis Komis (agmg)

A Linux terminálban számos program és parancs érhető el a rendszerfolyamatok kezelésére. E cikkben feltárjuk a folyamat fogalmát és átnézzük azokat a legismertebb parancsokat, amik lehetővé teszik a futó folyamatok („process”-ek) megtekintését, kilövését és a prioritási sorrend megváltoztatását

Mi az a „process” (folyamat)?

A számítógépes program utasítások gyűjteménye; adott feladat végrehajtására való. A program futtatásakor az operációs rendszer RAM-jába (Tetszőleges Elérhető Memória) kerül, hogy a CPU (Központi Feldolgozóegység) hozzáférjen és végrehajthassa. A „process” végrehajtása alatt álló programrész. A programkódot és az aktuális műveletet tartalmazza. Az operációs rendszertől függően a folyamat több százból állhat, amely az utasításokat egyidejűleg hajtják végre.

A „több feladatos” operációs rendszerekben a folyamatok oszthatóknak a processzoron és a rendszer forrásain. A „több feladatos” lehetővé teszi minden egyes processzornak, hogy a feladatok között válthasson anélkül, hogy várni kellene az egyes feladatok befejeződésére. Az operációs rendszer megvalósításának függvényében a váltások végrehajthatók akkor, amikor a feladatok ki- és beviteli műveleteket végeznek, amikor egy feladat jelzi, hogy kapcsolható, vagy hardver-megszakításkor.

```
top - 22:50:10 up 1:31, 3 users, load average: 2.70, 2.61, 2.56
Tasks: 170 total, 1 running, 169 sleeping, 0 stopped, 0 zombie
Cpu(s): 27.4%us, 5.0%sy, 67.4%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.2%si, 0.0%st
Mem: 4019552k total, 3851232k used, 168320k free, 1081484k buffers
Swap: 4088504k total, 8k used, 4088496k free, 1262452k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3606	root	1	19	280m	196m	3032	S	136	5.0	134:46.38	FahCore_a3.exe
1144	root	1	0	380m	128m	86m	S	20	3.3	10:47.86	X
13376	agmg	1	0	315m	29m	21m	S	18	0.8	0:00.55	ksnapshot
3608	agmg	1	0	2964m	92m	37m	S	10	2.4	1:38.33	kwin
3625	agmg	1	0	2971m	108m	39m	S	3	2.8	1:15.63	plasma-desktop
7127	agmg	1	0	347m	34m	19m	S	2	0.9	0:24.34	plugin-containe
9860	agmg	1	0	541m	35m	22m	S	2	0.9	0:07.69	konsole
3896	agmg	1	0	567m	45m	24m	S	1	1.2	2:50.88	transmission-qt
11974	agmg	6	0	11144	1200	852	R	1	0.0	0:04.38	top
2606	agmg	1	0	16828	2036	736	S	0	0.1	0:04.23	dbus-daemon
3027	agmg	1	0	355m	11m	5860	S	0	0.3	0:00.40	klauncher
3031	agmg	1	0	822m	33m	20m	S	0	0.9	0:05.15	kded4
3134	agmg	1	0	443m	21m	11m	S	0	0.6	0:01.43	kglobalaccl
3381	agmg	1	0	521m	22m	13m	S	0	0.6	0:00.92	ksmserver
3636	agmg	1	0	152m	6388	4928	S	0	0.2	0:00.57	akonadi_control
3640	agmg	1	0	1114m	33m	5404	S	0	0.9	0:06.48	mysqld
3815	agmg	1	0	232m	49m	12m	S	0	1.3	0:44.80	net_applet
3848	agmg	1	0	302m	18m	13m	S	0	0.5	0:00.48	kwalletmanager
6645	agmg	1	0	1347m	478m	41m	S	0	12.2	16:38.00	firefox
9025	root	1	0	0	0	0	S	0	0.0	0:11.56	kworker/1:1
12489	root	1	0	0	0	0	S	0	0.0	0:01.46	kworker/0:1
1	root	1	0	4124	704	604	S	0	0.0	0:00.67	init
2	root	1	0	0	0	0	S	0	0.0	0:00.00	kthreadd
3	root	1	0	0	0	0	S	0	0.0	0:00.22	ksoftirqd/0
5	root	1	0	0	0	0	S	0	0.0	0:00.00	kworker/u:0
6	root	RT	0	0	0	0	S	0	0.0	0:00.00	migration/0
7	root	RT	0	0	0	0	S	0	0.0	0:00.00	migration/1
9	root	1	0	0	0	0	S	0	0.0	0:00.24	ksoftirqd/1
11	root	1	-20	0	0	0	S	0	0.0	0:00.00	khelper
12	root	1	0	0	0	0	S	0	0.0	0:00.00	kdevtmpfs
13	root	1	-20	0	0	0	S	0	0.0	0:00.00	netns

top

A **top** program dinamikus, valós idejű képet mutat a futó rendszerről. Megjeleníti a rendszer összefoglaló információit (mint futási idő, cpu-, memória- és swap-használat stb.) és kilistázza a rendszerben a CPU-t leginkább lekötő folyamatokat. A rendszerről megjelenített összefoglaló információk, illetve a megjelenített információk típusa, sorrendje és mérete teljesen testre szabható, és ez a beállítás az újraindítások utánra is megőrizhető. A top programot a PCLinuxOS alpból telepíti és a kedvenc terminálodban a top parancs kiadásával futtatható.

Vessünk rá egy alaposabb pillantást ebben az ablakban.

Első sor: top

```
top - 22:50:10 up 1:31, 3 users, load average: 2.70, 2.61, 2.56
```

Megjelenített információk:

- Pillanatnyi rendszeridő (22:50:10)
- A rendszer futási ideje – az idő, amióta a rendszer él (1:31)
- A bejelentkezett felhasználók száma (3 users)
- A rendszer átlagos terhelése (a 3 hivatkozott érték 3 az utolsó 1, 5 és 15 percre vonatkozik)

Második sor: „task”-ok (feladatok)

```
Tasks: 170 total, 1 running, 169 sleeping, 0 stopped, 0 zombie
```

Megjelenített információk:

- Az összes folyamat száma (170 total)
- A futó folyamatok száma (1 running)
- Az alvó folyamatok száma (169 sleeping)
- Az álló folyamatok száma (0 stopped)
- Zombie folyamatok (0 zombie)

Harmadik sor: cpu(k)

```
Cpu(s): 27.4%us, 5.0%sy, 67.4%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.2%si, 0.0%st
```

Megjelenített információk:

- a felhasználói folyamatok CPU-terhelése százalékban (27.4%us);
- a rendszerfolyamatok CPU-terhelése százalékban (5.0%sy);
- a prioritás („nice”) frissítésére fordított CPU-terhelése százalékban (67.4%ni);
- a CPU szabad százaléka (0.0%id);

- az I/O (ki- és beviteli) műveletekre váró folyamatok CPU-terhelése százalékban (0.0%wa);
- a hardveres megszakítások CPU-terhelése százalékban (0.0%hi);
- a szoftveres megszakítások CPU-terhelése százalékban (0.2%si);
- „hypervisor” (a virtuális folyamatok felügyelője) által egyik virtuális gép használatától más feladatokra (pl. másik virtuális gép számára) „ellopott” CPU-idő százaléka. Virtuális gépet nem futtató asztali gép és szerver esetén 0 lesz.

Negyedik sor: memóriahasználat

```
Mem: 4019552k total, 3851232k used, 168320k free, 1081484k buffers
```

Megjelenített információk:

- Teljes rendelkezésre álló memória (4019552k total)
- Felhasznált memória (3851232k used)
- Szabad memória (168320k free)
- Gyorsítótár memória (181484k buffers)

Hasonló információk láthatók az ötödik sorban is, de a swap-re (cserehely).

```
Swap: 4088504k total, 8k used, 4088496k free, 1262452k cached
```

A következő sorok a rendszerben futó folyamatokat listázzák.

Nézzük meg a különböző oszlopok által megadott adatokra:

- **PID**: process azonosítója;
- **User**: a process tulajdonosa;
- **PR**: a process prioritása;
- **NI**: a process nice értéke (később beszélünk róla);
- **VIRT**: a process által használt virtuális memória;
- **RES**: a process által használt fizikai memória;
- **SHR**: a process megosztott memóriája;
- **S**: process státuszát mutatja: S=sleeping (alvó) R=running (futó) Z=zombie („élőhalott”);
- **%CPU**: a process által felhasznált CPU százaléka;
- **%MEM**: a process által használt RAM százaléka;
- **TIME+**: a process teljes aktív ideje;
- **COMMAND**: a process neve;

A top program számos parancssori argumentummal futtatható:

-d A képernyőfrissítések sűrűségét adja meg. Az s interaktív parancssal változtatható meg.

-p Csak az adott azonosítójú programokat figyeli. Ez a jelző ismételhető, de húsz alkalommal használható csak. Az opció nem adható meg interaktív módon, vagy konfigurációs fájlban.

-q Hatására a top azonnal, késlekedés nélkül frissít. Ha rendszergazdai jogokkal

indították, a top a legnagyobb prioritással fog futni.

-S Kumulatív módot jelent, ahol az egyes process-ek az általuk és a „halott” gyermek-folyamataik által felhasznált CPU-idő szerint jelennek meg.

-s A top-ot biztonságos módban futtatja. Kikapcsol minden interaktív parancsot.

-i Hatására a top kizár minden szünetelő, vagy zombie folyamatot.

-C Az összegzett CPU állapotot mutatja az egyes CPU-k helyett. Csak SMP rendszerekben hatásos.

-c A parancs neve helyett a parancssort mutatja.

-H Az összes szálát megjeleníti.

-n ismétlésszám. Ennyi alkalommal frissíti a képernyőt, majd kilép.

-b Kötégtelt mód. Hasznos a top kimenetének más programnak, vagy fájlba küldésekor. E módban a top nem fogad el interaktív parancssori bevitelt. Fut, amíg el nem éri az n-nel megadott ismétlésszámot, vagy ki nem lövik. A kimenet, nem programozható terminálon való megjelenítésre alkalmas szövegfájl.

A top sok egybillentyűs parancsot ismer fel futása közben (interaktív parancsok). A parancssorban az s opció alkalmazása néhányat letilt.

betűköz Azonnali képernyőfrissítés.

^L A képernyő törlése és újrarajzolása.

h vagy **?** Megjeleníti a sűgő képernyőt, a parancsok rövid összegzésével, illetve a kumulatív (gyűjtő) és biztonsági mód állapotát.

k Folyamat kilövés. Kérni fogja a feladat PID-jét és a jelet az elküldésre. Normál kilövéshez 15-ös jel kell. Biztosabb, de eléggé megszakító jellegű kilövéshez küldj 9-est. Az alap signal (jel) 15, SIGTERM. Ez a parancs nem érhető el biztonságos módban.

i A várakozó és zombi process-ek kizárása. Ez egy váltókapcsoló.

I Váltás Solaris (CPU százalék osztva a CPU-k számával) és Irix (CPU százalék kizárólag a felhasznált idő szerint) nézet között. Váltókapcsoló, ami csak az SMP rendszerekben hatásos.

n vagy **#** A megjelenített process-ek számának változtatása. Kér a szám bevitelét. Felülírja a megjelenített process-ek számának ablakmérettől függő automatikus meghatározását. 0-ra annyi process-t mutat, amennyi a képernyőre kifér.

q Kilépés.

r Folyamat újra prioritálása (re-nice). Kéri fogja a feladat PID-számát és a nice-értéket. Pozitív érték megadására a folyamat negatív nice érték irányába megy és csökken a prioritása. Ha rendszergazda futtatja a top-ot, bevihető negatív érték, amire a process a rendes prioritásnál magasabbat kap. Az alap re-nice értéke 10. A parancs biztonságos módban nem érhető el.

S Kumulatív mód váltókapcsoló, a **ps -S** hatására a process CPU-idejébe elhalt leszármazottját is beszámítja. Néhány programnál, pl. fordítók, ami sok önálló feladatra ágazik, normál módban a valóságosnál kisebb terhelést mutat. Mások esetben, pl. héjak és az init, ez a működés megfelelő. Minden esetben próbáld ki a kumulatív módot is, hogy a CPU-használatról alternatív képet is kapszál.

s A frissítések közötti késleltetést változtatja meg. Várja a késleltetési idő megadását másodpercben. Töredék időket mikroszekundumig kezel. 0 bevitelére folyamatosan frissít. Az alapérték 5 mp. Figyeld meg, hogy kis értékek hatására a képernyő szinte olvashatatlanul gyorsan változik és jelentősen megemelkedik a terhelés. A parancs biztonságos módban elérhetetlen.

f, vagy **F** Mezők megjelenítésére, vagy megjelenítésből eltávolítására szolgál.

o, vagy **O** Megváltoztatja a mutatott mezők sorrendjét. További információk lent.

l Váltás az átlagos terhelés és a bekacsolási idő megjelenítése között.

m A memória-információk váltása.

t Váltás a process-ek és a CPU-állapotinformációja között.

c Váltás a parancs neve és a teljes parancssor között.

N Feladatok rendezése a PID szerint (sorba rendezés).

A Feladatok rendezése időrendben (újak elől).

P Feladatok rendezése CPU-használat szerint (alapbeállítás).

M Feladatok rendezése a memóriában maradás mértéke szerint.

T Feladatok rendezése idő, vagy kumulált idő szerint.

W Az aktuális beállítások kiírása a „~/toprc”-be. Javasolt mód top konfigurációs fájl készítésére. A parancs használatával a top következő indításakor a konfiguráció összes változtatását használni fogja (azok kivételével, amik – természetüknél fogva - konfigurációs fájlba nem menthetők).

A **top**-ról sokkal több információt olvashatsz a man oldalain (kézikönyv).

<http://unixhelp.ed.ac.uk/CGI/man-cgi?top>

htop

A **htop** a **top** fejlesztett változata. Lehetővé teszi a lista görgetését mind függőlegesen, mind vízszintesen, hogy megnezhess az összes folyamatot és a parancssort. Emellett végrehajthatsz vele process-függő feladatokat (kill, nice), a process PID-jének beírása nélkül.

Vessünk egy pillantást ennek az ablaknak a felső részére:

The screenshot shows the htop terminal interface. At the top, it displays system statistics: Tasks: 115, 190 thr; 4 running; Load average: 2.63 2.67 2.69; Uptime: 02:26:10; Mem: 1566/3925MB; Swp: 0/3992MB. Below this is a table of running processes with columns for PID, USER, PRI, NI, VIRT, RES, SHR, S, CPU%, MEM%, TIME+, and Command. The first few processes listed are 3606 root, 3630 root, 3688 root, 1144 root, 3608 agmg, 3625 agmg, 19054 agmg, 16617 agmg, 19096 agmg, 7127 agmg, 14028 agmg, 6645 agmg, 3890 agmg, 3900 agmg, 3623 agmg, 3628 agmg, 3729 agmg, 3775 agmg, 3805 agmg, 3831 agmg, 3815 agmg, 3287 agmg, 3472 root, 3650 agmg, 1 root, 109 root, 888 root, 891 root, 910 root, and 933 messagebu.

The screenshot shows the top terminal interface. At the top, it displays system statistics: Tasks: 115, 190 thr; 4 running; Load average: 2.63 2.67 2.69; Uptime: 02:26:10; Mem: 1566/3925MB; Swp: 0/3992MB. Below this is a table of running processes with columns for PID, USER, PRI, NI, VIRT, RES, SHR, S, CPU%, MEM%, TIME+, and Command. The first few processes listed are 1 root, 2 root, 3606 root, 3630 root, 3688 root, 1144 root, 3608 agmg, 3625 agmg, 19054 agmg, 16617 agmg, 19096 agmg, 7127 agmg, 14028 agmg, 6645 agmg, 3890 agmg, 3900 agmg, 3623 agmg, 3628 agmg, 3729 agmg, 3775 agmg, 3805 agmg, 3831 agmg, 3815 agmg, 3287 agmg, 3472 root, 3650 agmg, 1 root, 109 root, 888 root, 891 root, 910 root, and 933 messagebu.

Itt (magonként külön grafikonon) tájékoztatást ad a CPU-, a memória- és swap-használatról, csakúgy mint a futó feladatokról, a rendszerterhelésről és a futási időről. **F2**-vel (setup) teljes mértékben beállítható, hogy itt több vagy kevesebb információ jelenjen meg.

Alatta, a futó folyamatok listája jelenik meg, a **top**-nál korábban látotthoz hasonló módon. Miként azt korábban is említettem, a billentyűzetten található nyilakat

használható a listán belüli függőleges, illetve vízszintes mozgásra. A htop az egérműveleteket is támogatja.

Az ablak lábléce jeleníti meg a **htop** menüparancsait.



F1 lenyomására a **htop** parancsainak teljes listáját megkaphatod:

```
htop 1.0.2 - (C) 2004-2011 Hisham Muhammad
Released under the GNU GPL. See 'man' page for more info.

CPU usage bar: [low-priority/normal/kernel/virtualiz      used%]
Memory bar:   [used/buffers/cache                       used/total]
Swap bar:     [used                                      used/total]
Type and layout of header meters are configurable in the setup screen.

Status: R: running; S: sleeping; T: traced/stopped; Z: zombie; D: disk sleep
Arrows: scroll process list          F5 t: tree view
Digits: incremental PID search      u: show processes of a single user
F3 /: incremental name search       H: hide/show user threads
F4 \: incremental name filtering     K: hide/show kernel threads
Space: tag processes                F: cursor follows process
U: untag all processes              + -: expand/collapse tree
F9 k: kill process/tagged processes P M T: sort by CPU%, MEM%, or TIME
] F7: higher priority (root only)   i: set IO priority
[ F8: lower priority (+ nice)       I: invert sort order
a: set CPU affinity                 F6 >: select sort column
F2 S: setup                          l: list open files with lsof
? F1 h: show this help screen       s: trace syscalls with strace
F10 q: quit

Press any key to return.
```

A **htop** egyebek mellett lehetővé teszi adott process keresését név szerint, a lista rendezését bármely mező szerint (CPU-, memória-használat stb.) és többszörös process-kijelölést kilövésre (kill), vagy a prioritás emelésére, vagy csökkentésére (nice).

A top-hoz hasonlóan a htop is futtatható parancssori argumentumokkal:

- d --delay=**KÉSLELTETÉS** A frissítések késleltetése tizedmásodpercben.
- C --no-color --no-colour Htop indítása fekete-fehér módban.
- h --help A súgó megjelenítése és kilépés.
- p --pid=**PID,PID...** Csak az adott PID-ek megjelenítése.
- s --sort-key=**OSZLOP** Az adott oszlop szerint rendez (a --sort-key=help oszloplistát ad).

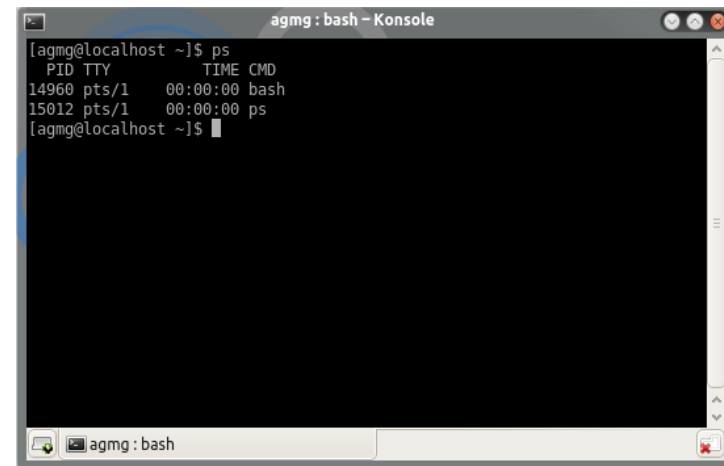
-u --user=**FELHASZN** Csak az adott felhasználó folyamatait mutatja.

-v --version Megjeleníti a verzióinformációkat és kilép.

Ha további információkat akarsz a htop-ról, egyszerűen terminálba írd be, hogy **man htop**, vagy látogasd meg a következő lapot: <http://linux.die.net/man/1/htop>

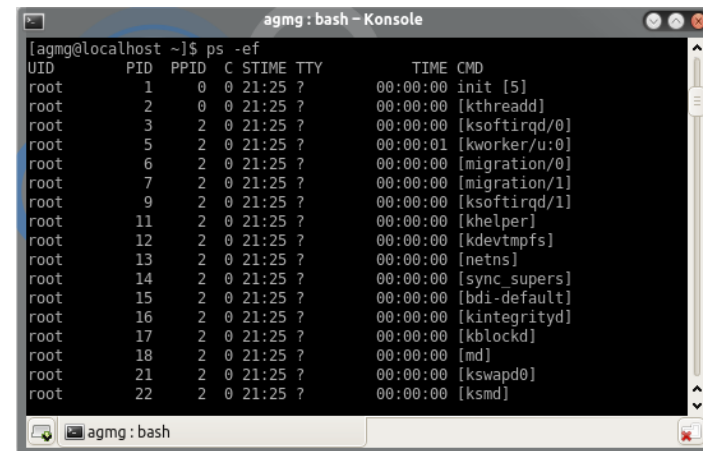
ps

A **ps** parancs, egy másik módja a futó folyamatok megjelenítésének terminálban. A **ps** parancs tipikus formája **ps [opció]**



A **ps** futtatása opciók nélkül az aktuális terminálról jelenít meg információt:

A pillanatnyilag futó process-ről minden információ megjelenítése: **ps -ef**



-e Az összes pillanatnyilag futó folyamatról ad információt.

-f Teljes listát készít.



Könnyebben olvasható listáért, aminek a kimenetét a less parancshoz irányíthatod át, az átlapozás érdekében **ps -ef | less**:

Ha kész vagy, nyomj **q**-t a kilépéshez.

A **ps** parancs opcióinak teljes listája érdekében olvasd el a kézikönyvét: <http://linux.die.net/man/1/ps>

ps tree

A **ps tree** a **ps**-hez hasonló, de a folyamatokat faszerkezet nézetben listázza ki:

A kimenetet a **grep**-en keresztül is átvezetheted, hogy meghatározott process-t keres meg, további parancsok használata nélkül. Például a **ps -ef | grep firefox** parancs, kilistázza az összes Firefox-hoz kapcsolódó folyamatot:

A process megjeleníthető részletes formában az **-l** opcióval:

pstree szintén számos parancssori argumentummal rendelkezik. Nézzünk meg közülük néhányat:

-a A parancssori argumentumok megjelenítése. Ha egy process a parancssorát már kiürítette, zárójelbe kerül. Az **-a** áttételesen kikapcsolja a tömörítést.

```

[agmg@localhost ~]$ ps tree -a
init
├── acpid
├── akonadi_control
│   ├── akonadiserver
│   │   ├── {akonadiserver}
│   │   ├── {akonadiserver}
│   │   ├── {akonadiserver}
│   │   ├── {akonadiserver}
│   │   └── {akonadiserver}
│   └── {akonadi_control}
├── atd -l 1.8
├── auditd
│   └── {auditd}
├── avahi-daemon
│   └── avahi-daemon
├── bluetoothd --udev
├── console-kit-dae --no-daemon
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   └── {console-kit-dae}
├── crond
├── 2*[dbus-daemon]
├── dbus-launch
├── gam_server
├── gconfd-2
├── gpg-agent
├── hald
│   ├── hald-runner
│   │   ├── 2*[hald-addon-acpi]
│   │   ├── 2*[hald-addon-cpuf]
│   │   ├── 2*[hald-addon-gene]
│   │   ├── hald-addon-inpu
│   │   ├── hald-addon-leds
│   │   ├── hald-addon-rfki
│   │   └── {hald-runner}
│   └── {hald}
├── ifplugd
└── kactivitymanage--5*[{kactivitymanage}]
    
```

```

[agmg@localhost ~]$ ps tree -l
init
├── acpid
├── akonadi_control
│   ├── akonadiserver--6*[{akonadiserver}]
│   └── {akonadi_control}
├── atd
├── auditd--{auditd}
├── avahi-daemon--avahi-daemon
├── bluetoothd
├── console-kit-dae--64*[{console-kit-dae}]
├── crond
├── 2*[dbus-daemon]
├── dbus-launch
├── gam_server
├── gconfd-2
├── gpg-agent
├── hald
│   ├── hald-runner--2*[{hald-addon-acpi}]
│   │   ├── hald-addon-cpuf
│   │   ├── 2*[{hald-addon-gene}]
│   │   ├── hald-addon-inpu
│   │   ├── hald-addon-leds
│   │   ├── hald-addon-rfki
│   │   └── {hald-runner}
│   └── {hald}
├── ifplugd
└── kactivitymanage--5*[{kactivitymanage}]
    
```

-l Hosszú sorok megjelenítése. Ha nem konzolra megy, vagy a képernyő szélessége ismeretlen, a sorokat a képernyő szélességéig, vagy 132-ig levágja.

-A ASCII karakterekkel faszerkezetet jelenít meg.

```

[agmg@localhost ~]$ ps tree -A
init
├── akonadi_control
│   ├── akonadiserver--6*[{akonadiserver}]
│   └── {akonadi_control}
├── atd
├── auditd--{auditd}
├── avahi-daemon--avahi-daemon
├── bluetoothd
├── console-kit-dae--64*[{console-kit-dae}]
├── crond
├── 2*[dbus-daemon]
├── dbus-launch
├── gam_server
├── gconfd-2
├── gpg-agent
├── hald
│   ├── hald-runner--2*[{hald-addon-acpi}]
│   │   ├── hald-addon-cpuf
│   │   ├── 2*[{hald-addon-gene}]
│   │   ├── hald-addon-inpu
│   │   ├── hald-addon-leds
│   │   ├── hald-addon-rfki
│   │   └── {hald-runner}
│   └── {hald}
├── ifplugd
└── kactivitymanage--5*[{kactivitymanage}]
    
```

```

[agmg@localhost ~]$ ps tree -c
init
├── akonadi_control
│   ├── akonadiserver--{akonadiserver}
│   ├── {akonadiserver}
│   ├── {akonadiserver}
│   ├── {akonadiserver}
│   ├── {akonadiserver}
│   └── {akonadiserver}
├── {akonadi_control}
├── atd
├── auditd--{auditd}
├── avahi-daemon--avahi-daemon
├── bluetoothd
├── console-kit-dae
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   ├── {console-kit-dae}
│   └── {console-kit-dae}
├── crond
├── 2*[dbus-daemon]
├── dbus-launch
├── gam_server
├── gconfd-2
├── gpg-agent
├── hald
│   ├── hald-runner--2*[{hald-addon-acpi}]
│   │   ├── hald-addon-cpuf
│   │   ├── 2*[{hald-addon-gene}]
│   │   ├── hald-addon-inpu
│   │   ├── hald-addon-leds
│   │   ├── hald-addon-rfki
│   │   └── {hald-runner}
│   └── {hald}
├── ifplugd
└── kactivitymanage--5*[{kactivitymanage}]
    
```

-c Kikapcsolja az önálló ágak tömörítését (összecsukását). Az ágak kibontása alpból, ha lehet, kikapcsolt (jobbra fent).

-p Mutatja a PID-eket zárójelben, tízes számrendszerben a process neve után. A **-p** áttételesen kikapcsolja a tömörített nézetet (köv. lap, balra fent):

```

[agmg@localhost ~]$ ps tree -p
init(1)
├── acpid(821)
├── akonadi_control(3451)
│   ├── akonadiserver(3453)
│   │   ├── {akonadiserver}(3455)
│   │   ├── {akonadiserver}(3482)
│   │   ├── {akonadiserver}(3483)
│   │   ├── {akonadiserver}(3484)
│   │   ├── {akonadiserver}(3485)
│   │   └── {akonadiserver}(3541)
│   └── {akonadi_control}(3452)
├── atd(1767)
├── auditd(1786)──{auditd}(1787)
├── avahi-daemon(3272)──avahi-daemon(3273)
├── bluetoothd(1254)
└── console-kit-dae(888)
    ├── {console-kit-dae}(889)
    ├── {console-kit-dae}(891)
    ├── {console-kit-dae}(892)
    ├── {console-kit-dae}(893)
    ├── {console-kit-dae}(894)
    ├── {console-kit-dae}(895)
    ├── {console-kit-dae}(897)
    ├── {console-kit-dae}(898)
    ├── {console-kit-dae}(899)
    ├── {console-kit-dae}(900)
    ├── {console-kit-dae}(901)
    └── {console-kit-dae}(902)
    
```

Ha többet akarsz megtudni a **ps tree** parancsról és opcióiról, olvasd el a kézikönyvét: <http://unixhelp.ed.ac.uk/CGI/man-cgi?ps tree+1>

pgrep

A **pgrep** parancs átnézi a rendszer futó folyamatainak táblázatát és kiírja az összes olyan process PID-jét, amelyek a parancssorban megadott feltételekkel megegyezik. Például, ha szeretnéd megtudni a Firefox PID-jét, akkor írd be **pgrep firefox**.

```

[agmg@localhost ~]$ pgrep firefox
3729
[agmg@localhost ~]$
    
```

Amennyiben az azonosító mellett az összes, a feltételnek megfelelő process nevéét is látni szeretnéd, használd az **-l** argumentumot (long output) (jobbra fent):

```

[agmg@localhost ~]$ pgrep -l kde
12 kdevtmpfs
2122 startkde
2906 kdeinit4
3006 kded4
3503 polkit-kde-auth
[agmg@localhost ~]$
    
```

Egy másik, **-l**-el együtt használható hasznos argumentum, az **-f**, ami az összes egyező process PID-je mellett a teljes argumentumlistáját megjeleníti:

```

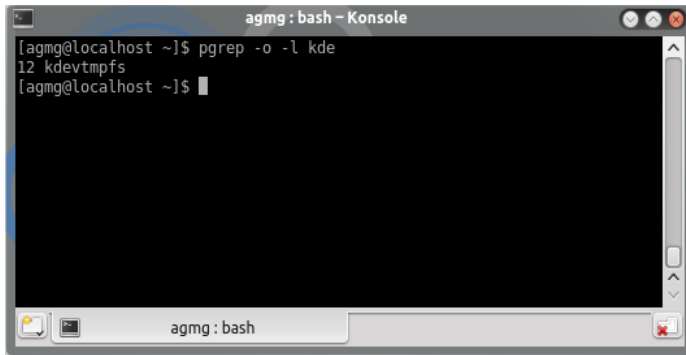
[agmg@localhost ~]$ pgrep -l -f update
3496 /bin/bash /usr/bin/update-notifier-launcher
3604 /bin/bash /usr/bin/update-notifier
3724 yad --title=System update status.. --notification --image=/usr/sh
are/icons/synaptic_ok.png --text=0 updates available; 0 new packages a
vailble.
[agmg@localhost ~]$
    
```

Az **-n** argumentum csak a legutóbb indítottat választja ki az egyező process-ek közül (a példában az **-l** argumentumhoz kapcsolva használtam):

```

[agmg@localhost ~]$ pgrep -n kde
3503
[agmg@localhost ~]$ pgrep -n -l kde
3503 polkit-kde-auth
[agmg@localhost ~]$
    
```

Az **-o** argumentum csak a legrégebbi (legkorábban indítottat) választja ki az egyező process-ek közül (a példában ismét az **-l** argumentumhoz kapcsolva használtam) (következő oldal, balra fent):



```
agmg : bash - Konsole
[agmg@localhost ~]$ pgrep -o -l kde
12 kdevtmpfs
[agmg@localhost ~]$
```

Továbbiakat a `pgrep` parancsról és opcióiról, a kézikönyvében találhatsz: <http://www.lehman.cuny.edu/cgi-bin/man-cgi?pgrep+1>

killall, killall, kill

Az fenti parancsokkal a folyamatoknak bármilyen jel (signal) küldhető, mindegyikkel másképpen, de általában bezárásra (terminate) (TERM jelzés küldése – SIGTERM – 15-ös érték), vagy egy lefagyott kilövésére (KILL jelzés küldése – SIGKILL – 9-es érték) használt. E parancsok alaphelyzetben TERM jelzést küldenek a megadott folyamatnak.

A Linux jelzéseivel kapcsolatos további információkért látogasd meg a <http://man7.org/linux/man-pages/man7/signal.7.html> lapot.

A pkill használata

A `pkill` a folyamatot a nevét használva lövi ki. Például, ha kilőnéd a Firefox process-t, egyszerűen írd be `pkill firefox`.

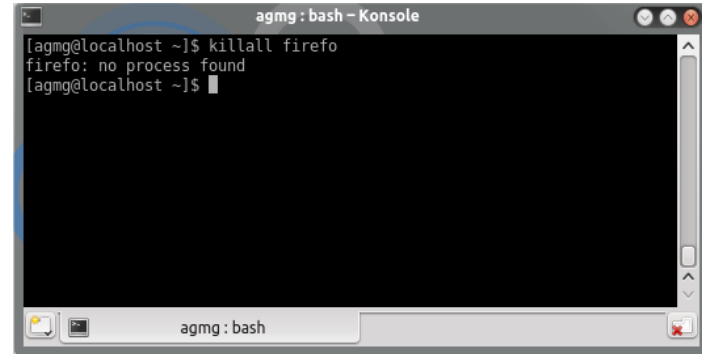
Jó ha tudod, a `pkill` kilő minden olyan folyamatot, aminek a nevében a „firefox” szerepel. Hogy lásd, ezek mely process-ek, használd a korábban ismertetett `pgrep -l firefox` parancsot.

A `pkill` a process-eknek alapbeállításból a TERM (befejez) jelzést küldi. Megváltoztathatod, ha a parancsot a `-signal` argumentummal küldöd. A „signal” meghatározható akár névvel (a vezető SIG-gel, vagy anélkül), akár számmal. Például, ha kilőnéd erővel a Firefox-ot, írhatod

`pkill -9 firefox` vagy
`kill -SIGKILL firefox` vagy
`pkill -KILL firefox`.

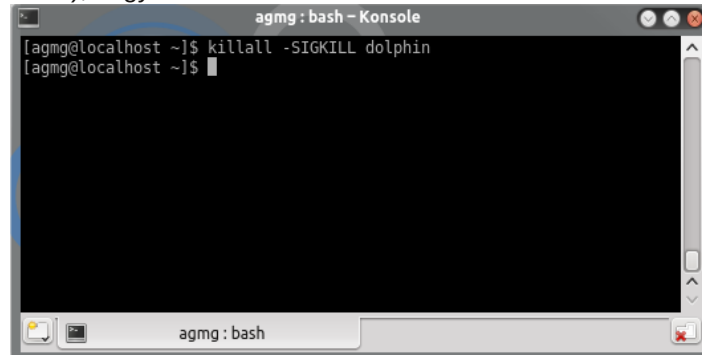
A killall használata

A `killall` a `pkill`-hez hasonló módon működik, de alpból a process pontos nevét várja el. Tehát, ha írod, hogy `killall firefox`, a következő hibát láthatod:



```
agmg : bash - Konsole
[agmg@localhost ~]$ killall firefox
firefox: no process found
[agmg@localhost ~]$
```

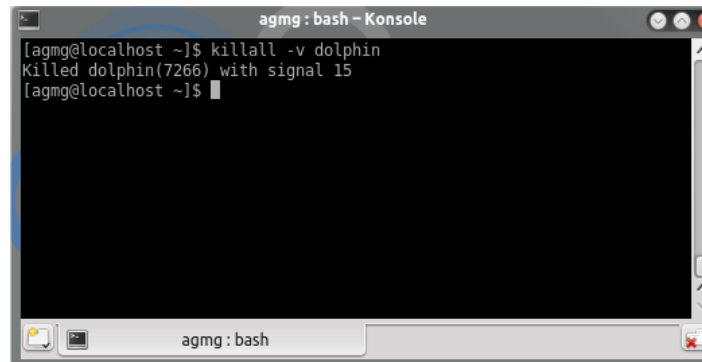
Alpból a `killall` is TERM jelzést ad ki. Megváltoztathatod az `-s` argumentum használatával, ami (mint a `pkill` esetében) lehet a jelzés neve (bevezető SIG-gel, vagy anélkül), vagy az értéke:



```
agmg : bash - Konsole
[agmg@localhost ~]$ killall -SIGKILL dolphin
[agmg@localhost ~]$
```

A -v argumentum

használata utasítja a `killall`-t, jelezze, ha a jelzés sikerrel ment ki:

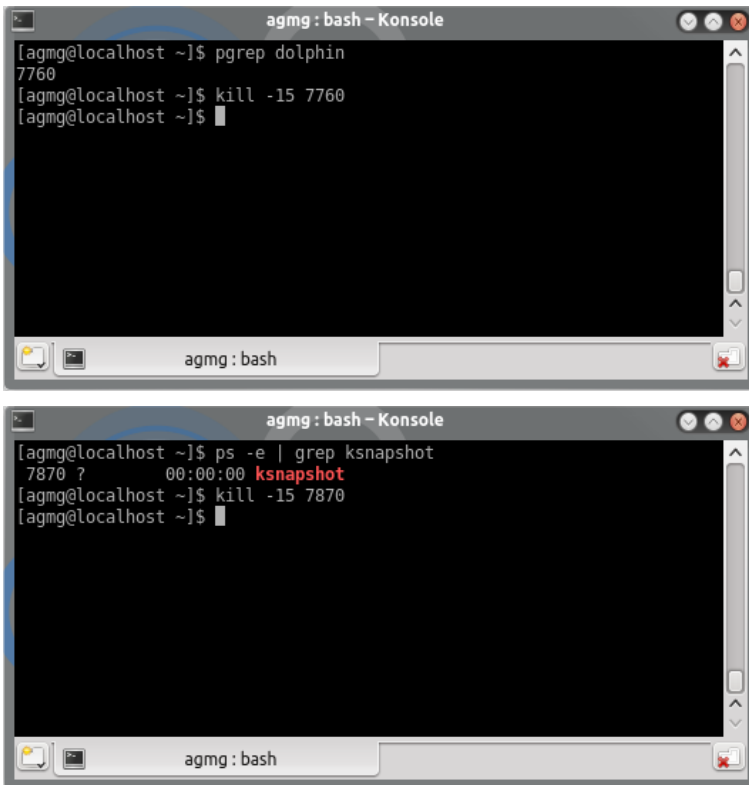


```
agmg : bash - Konsole
[agmg@localhost ~]$ killall -v dolphin
Killed dolphin(7266) with signal 15
[agmg@localhost ~]$
```


A **killall** és opcióival kapcsolatos további információk a <http://linux.die.net/man/1/killall> lapon.

A kill használata

A **kill** a PID-et használva küld jelzést a process-nek. Az alapbeállítás itt is a TERM, de a **pkill**-hez, vagy a **kill**-hez hasonlóan megváltoztatható (**-s** argumentummal). Ez a legkényemtelenebb módszer, mert a parancs használata előtt ismerni kell a PID-et. Használható a korábban tárgyalt **top**, **htop**, **ps**, vagy **pgrep** parancs bármelyike a kívánt folyamat PID-jének megállapítására.



```
agmg : bash - Konsole
[agmg@localhost ~]$ pgrep dolphin
7760
[agmg@localhost ~]$ kill -15 7760
[agmg@localhost ~]$

agmg : bash - Konsole
[agmg@localhost ~]$ ps -e | grep ksnapshot
7870 ?        00:00:00 ksnapshot
[agmg@localhost ~]$ kill -15 7870
[agmg@localhost ~]$
```

Ismét csak, ha többet akarsz megtudni a kill-ről, ne habozd elolvasni a kézikönyvét a: <http://unixhelp.ed.ac.uk/CGI/man-cgi?kill> lapon

A nice és a renice

A parancsok bármelyike használható a folyamatok prioritásának megváltoztatására. De mi is a process prioritása?

A folyamatok esetében a prioritás a processzor idejének kezelésére vonatkozik. Technikai leírás helyett, hadd adjak egy példát. Mondjuk, hogy az Interneten szörfölsz, miközben a kedvenc mp3 fájljaidat hallgatod. Nagyszerű alkalom, hogy összerakd a üdülésről felvett videóidat a kedvenc videó szerkesztőddel. Miközben szörfölsz, zenét hallgatsz és videó kódolsz, a böngésződ lelassul, vagy rövidebb szünetek lesznek az mp3 lejátszásban. Ez azért van, mert a harmadik feladat, a videó kódolása túl sokat köt le a rendszered erőforrásaiból, a többi feladatot takaréklángra rakva. Ha nem bánod, hogy a videó néhány órával később lesz kész, megmondhatod a processzornak, hogy fordítson kevesebb figyelmet erre a feladatra és többet a böngészésre, illetve a zenehallgatásra. (Ez csak egy alkalmazási példa, de remélem érted a lényegét.)

A Linuxban a CPU-nak beállíthatunk, a feladatokra vonatkozó alapelveket. Az alapelvek neve **nicess**, vagy **nice value** (nice érték). A Linux nice-ok értékskálája -20-tól 19-ig terjed. A kisebb szám magasabb prioritást ad a feladatnak. Amennyiben a nice értéke olyan magasabb szám, mint a 19, a feladat prioritása a legalacsonyabbra kerül és a CPU csak akkor dolgozik a folyamaton, amikor módjában áll. Az nice alapértéke 0.

A skála használatával a CPU forrásai sokkal pontosabban eloszthatók. Az alacsonyabb prioritású, kevésbé fontos programok nice értéke magasabb, míg a démonokhoz és szolgáltatások egyaránt magas prioritású programoknak beállíthatók, hogy több figyelmet kapjanak a CPU-tól. Megadható az is, hogy meghatározott felhasználó összes process-e magasabb nice értéket kapjon, ezzel korlátozva a lehetőségét a számítógép alapszolgáltatásainak lelassítására.

A **nice** utasítással egy parancs előzetesen módosított prioritással futtatható. Másrészt a **renice** parancs egy futó folyamat prioritásának megváltoztatására való. A process módosított prioritása csak az adott futására érvényes. Leállítás után és újraindításkor az alapbeállítás szerinti prioritását visszakapja.

A nice parancs szintaxisa: **nice [OPCIÓN] [PARANCS [ARG]...]**

-n, --adjustment=ADJUST A prioritás növelése az ADJUST értékével, ami alapból 10. A tartomány -20 (magasabb prioritás) és 19 (alacsonyabb) között van.

--help A súgó megjelenítése és kilépés.

--version A verzióinformáció megjelenítése és kilépés.

Példa:

Firefox futtatása alacsonyabb prioritással: **nice -n 5 firefox**

Firefox futtatása magasabb prioritással: **nice -n -6 firefox**

A renice parancs szintaxisa:

renice *prioritás* **[[*-p*] pid ...]** **[[*-g*] pgrp ...]** **[[*-u*] felhasznál...]**

renice -n növelés **[[*-p*] pid ...]** **[[*-g*] pgrp ...]** **[[*-u*] felhasznál...]**

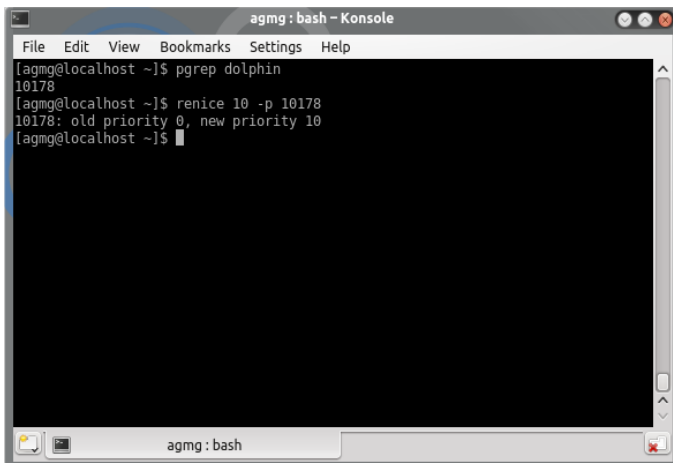
A **renice** parancs megváltoztatja egy vagy több futó folyamat időzítési prioritását. Az azonosító paramétereket process, process csoport, vagy felhasználó azonosítóként, illetve felhasználónévként értelmezi. Egy process csoportra alkalmazott **renice** hatására a process csoport időzítési prioritása megváltozik. A felhasználóra kiadott **renice**-ra az általa indított folyamatok időzítési prioritása megváltozik. Alapból a process-ek kijelölése a process azonosítóval (PID) történik.

A következő opciók érhetők el:

- g** Az azonosító paramétereket process PID-ként értelmezi.
- n** Hatására az argumentumot nem abszolút értéként, hanem a process(-ek) pillanatnyi prioritását adott értékkel módosítóként értelmezi.
- u** Hatására az azonosító paramétereket a felhasználó neveként, vagy azonosítójaként értelmezi.
- p** Az azonosító értelmezését visszaállítja (az alapbeállítás szerinti) PID-re.

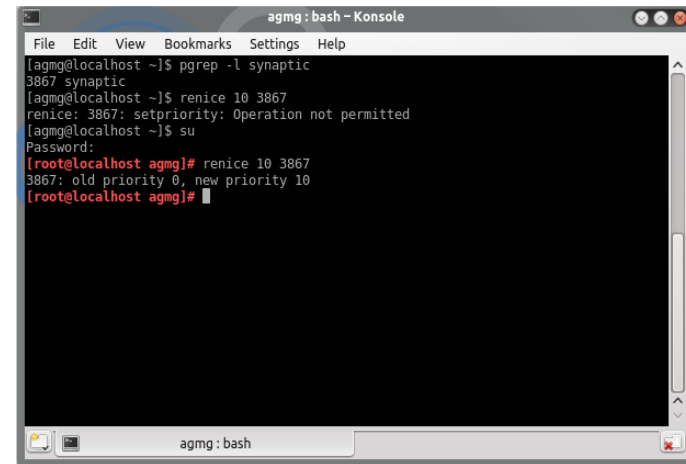
Néhány példa:

Mondjuk, szeretnénk a Dolphin fájlkezelő prioritását csökkenteni. Először meg kell állapítanunk az azonosítóját (PID). (A **pgrep**-et használtam a példában):



```
agmg@localhost ~]$ pgrep dolphin
10178
[agmg@localhost ~]$ renice 10 -p 10178
10178: old priority 0, new priority 10
[agmg@localhost ~]$
```

Ha egy adott process prioritását, ami felett nem rendelkezel **növelni** szeretnéd, „operation not permitted” (nem engedélyezett művelet) üzenetet kapsz. Csak rendszergazda tudja bármely process prioritását megváltoztatni. A normál felhasználó csak a saját process-ei prioritását tudja változtathatja meg és csak növelheti a „nice értéket” 0 és PRIO_MAX (20) tartományban.



```
agmg@localhost ~]$ pgrep -l synaptic
3867 synaptic
[agmg@localhost ~]$ renice 10 3867
renice: 3867: setpriority: Operation not permitted
[agmg@localhost ~]$ su
Password:
[root@localhost agmg]# renice 10 3867
3867: old priority 0, new priority 10
[root@localhost agmg]#
```

A két parancsral kapcsolatos további információkért tanulmányozhatod a kézikönyveket:

Nice kézikönyv, terminálban **man nice** vagy menj ide: <http://www.manpagez.com/man/1/nice/>

Renice kézikönyvét itt találhatod meg: <http://www.manpagez.com/man/8/renice/>

xkill

Az **xkill** olyan parancs, ami az X szerverrel bezárhatja a klienskapcsolatokat. Nagyon veszélyes, de hasznos parancs olyan programok bezárására, amik nem kívánt ablakokat tesznek ki a felhasználó képernyőjére. A PCLinuxOS alapból nem telepíti az **xkill**-t. Telepítéséhez nyisd meg a Synaptic csomagkezelőt és keress rá az **xkill**-re.

Az **xkill** parancssori paraméter nélküli kiadása hatására egy különleges kurzort (halálfej) jelenik meg, amivel kiválaszthatod a kilövendő ablakot. Kattints a bezárandó alkalmazáson határozottan, vagy a jobb egérgombbal válaszd ki azt az X window/alkalmazás-t, aminek a kliensét kilőnéd. Ha nem egy gyöker ablakra kattintasz, a szerver az adott ablakot létrehozó klienssel a kapcsolatát bezárja. Ha meggondolnád magad, jobb egérgombbal kattintva elvetheted a műveletet.

Az xkill-t az alábbi parancssori argumentumokkal használhatod:

-display *képernyő neve* Meghatározza, hogy melyik X szerverhez kapcsolódjon.

-id *forrás* Ez az opció megadja a forrást, aminek a létrehozóját kell kilőni. Ha nincs meghatározva, akkor az X szerver egy speciális kurzort jelenít meg, amivel kiválasztható a kilőni kívánt ablak.

-button *szám* Ez az opció meghatározza, hogy a mutató eszköz melyik gombja jelöli ki a kilöendő ablakot. Ha „any” szó szerepel, akkor bármelyik gomb használható. Az alapbeállítás a billentyűkiosztás első gombját (ami általában a balszélső) használja.

-all Az opció jelzi, hogy a képernyőn az összes felsőszintű ablak kliense kilöendő. Az xkill kérni fogja a nem-gyöker ablak kiválasztását, az egyes gombokon számos kilövési móddal. Az opció használta erősen ellenjavallott.

-frame Az opció jelzi az xkill-nek, hogy tekintsen el a felsőszintű kliensablakok felkutatásának szokásos módjától (amit általában ablakkezelővel kötnek össze), és egyszerűen értelmezze úgy, hogy a gyöker közvetlen leszármazottját akarod kilőni.

```
agmg@localhost ~]$ xwininfo
xwininfo: Please select the window about which you
would like information by clicking the
mouse in that window.
xwininfo: Window id: 0x340001d "agmg : xwininfo - Konsole"
Absolute upper-left X: 393
Absolute upper-left Y: 181
Relative upper-left X: 0
Relative upper-left Y: 0
Width: 561
Height: 538
Depth: 32
Visual: 0xa5
Visual Class: TrueColor
Border width: 0
Class: InputOutput
Colormap: 0x3400001 (not installed)
Bit Gravity State: NorthWestGravity
Window Gravity State: NorthWestGravity
Backing Store State: NotUseful
Save Under State: no
Map State: IsViewable
Override Redirect State: no
Corners: +393+181 -412+181 -412-49 +393-49
-geometry 561x538+385-41
agmg@localhost ~]$
```

Az ablak azonosítója egyszerűen kideríthető az **xwininfo** parancs futtatásával. Az **xwininfo** eszköz alapbeállításból nincs telepítve, de a Synaptic csomagkezelő segítségével telepíthető.

Amikor terminálban beírod: **xwininfo**, az egérkurzor keresztre vált, lehetővé téve annak az ablaknak a kiválasztását, amivel kapcsolatban információkat kérsz.

<-----

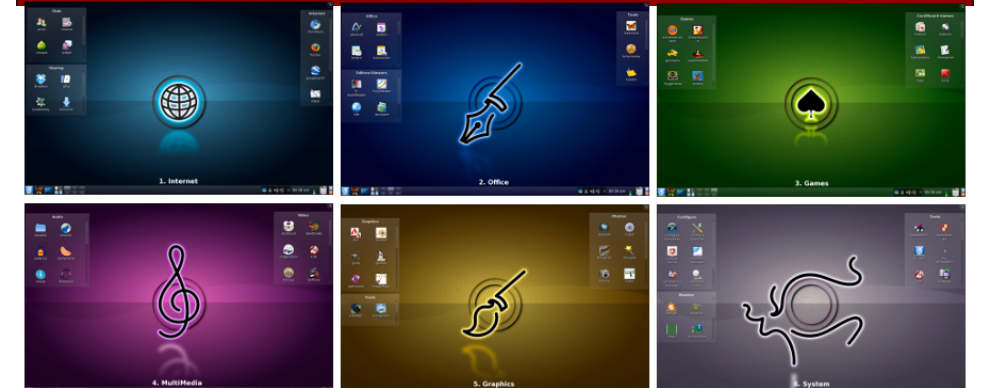
Nekünk a „Window ID”-re (ablakazonosító) van szükségünk. Ennek alapján a kívánt ablakot kilövő parancs ilyen lesz:

xkill -id 0x340001d

Az **xkill** kézikönyve terminálban a **man xkill** begépelésével, akár a következő lap meglátogatásával érheted el: <http://www.manpagez.com/man/1/xkill/>



PCLinuxOS Full Monty ...



Everything you might want or need – plus the kitchen sink!