

Felhasználói Módú Linux

A Felhasználói Módú Linux olyan trükköket tesz lehetővé, mint egy biztonságosan elszigetelt Debian 3.1 futtatása Fedora 4 felett.

A számítógépek programokat futtatnak. A kezdetek irdatlan monstriumaitól kezdve, melyekbe bele volt huzalozva a program, egészen napjaink asztali és hordozható gépeiig minden számítógép valamilyen programot futtat. Neumann János rukkolt ki azzal az elvvel, hogy tároljuk a programokat is ugyanúgy, mint bármely más kis adathalmazt, és a beledrótozott funkciók zártságával ellentétben tegyük lehetővé a számítógépek többcélúvá válását. Hamarosan megszületett az operációs rendszer fogalma: egy olyan programé, amely a közös rendszerszintű részletek számára hoz létre egy elvonatkoztatási réteget, mint amilyen pl. az eszközök kezelése vagy a programok végrehajtása. Nem kellett sok idő, hogy néhány ravasz rendszerprogramozó tudatosítsa, hogy egyetlen processzor látszólag egyetlen pillanatban végrehajthat több feladatot is. Ez adott lendületet az első időosztásos és többfeladatos/többfelhasználós operációs rendszerek kialakulásához. Az összes modern számítógép is ugyanezen a „tárolt program” elven működik. Egy mai számítógép bekapcsolásakor először a BIOS-ban tárolt programokat futtatja (vagy a firmware-t), amely aztán átadja a stafétabotot a többlépcsős betöltéskezelőnek, amely végül betölti az operációs rendszer magját. A rendszermag hajtja végre és állítja be az operációs környezetet, amelyben a rendszer erőforrásai (mint pl. a processzoridő, memória és az eszközök) használhatóak a programok által, melyek a kernel fölött elindulnak. Mindez tárolt programok hosszú láncolata.

A kernel is olyan program, mint bármelyik másik (jóllehet meglehetősen összetett, szép példány). Így hát mi akadályozhatja meg abban, hogy úgy futtassuk a rendszermagot, mint bármely más programot? Végül is semmi. Pontosan erről szól a *Felhasználói Módú Linux (UML)*. A *Linux* kernel általában speciális jogosultságokkal fut, mert szükség van arra, hogy direkt módon el tudja érni a hardvert. A *Felhasználói Módú Linux* lehetőséget ad arra, hogy úgy fordítsuk le a normál *Linux* kernelforrást, hogy egyszerű bináris programként lehessen elindítani a gazda *Linux* kernel felett. Amikor egy kernelt egy másik, „gazda *Linux* kernel” felett indítunk el, akkor igazából (egy vagy több) „vendég” *Linux* rendszert futtatunk, speciális jogosultságok nélkül. (Vannak azért kivételek – néhány programot rendszergazdaként kell telepíteni a *Felhasználói Módú Linux* számára, hogy jól működjenek). Ezek a *Linux* vendégrendszerek teljes operációs rendszerek, melyek egy (szinte) teljesen biztonságos környezetben futnak.

Hogyan kell megcsinálni?

Cikkem hátralevő részében erre adok receptet: hogyan kell lábra állítani egy *UML* rendszert *Linuxot* futtató gazdagépünkön. Ezek után körbejárjuk néhány tulajdonságát és szórakozunk egy kicsit. A bemutatóhoz *Fedora Core 4* gazdagépet használok, *Intel P4*-en 1 GB memóriával – azonban a módszer szinte bármilyen rendszeren és disztribúción működik, amennyiben friss 2.6-os kernel fut rajta, és van legalább 256MB memóriája.

Egy vendég *UML* rendszer olyan, mint bármilyen más *Linux* rendszer. A „*Linux*” a kernelen kívül sajátos gyűjteménye kisebb programoknak, programkönyvtáraknak és megfelelően szervezett fájloknak, melyek lehetővé teszik az operációs rendszer rendeltetésszerű működését. Ezek két külön egységben állnak rendelkezésünkre: a kernel és a fájlrendszer-képmásfájl formájában – ez utóbbi nem más, mint egy virtuális merevlemez-partíció; ezt csatoljuk fel és használjuk majd *UML* rendszerünk gyökér fájlrendszereként. Megválaszthatjuk, hogy előállítjuk-e ezt a két összetevőt, vagy letöltjük az internetről, ahol a népszerűbb disztribúciók számára már előre elkészítették ezeket. A gyorsabb siker érdekében most nem hagyjuk el a járt utat a járatlanért. Mindazonáltal vessünk egy pillantást az *UML Wikire* is; megtudhatjuk, hogy hogyan készíthetjük el fájlrendszereinket saját magunk (lásd a cikkhez tartozó forrásokat). A rendszermagoknak és gyökér fájlrendszereknek számos disztribúcióhoz elkészített változatai elérhetőek; például *Red Hat*, *Fedora Core*, *Debian* és számos más speciális célú képmásfájl vár letöltésre. A bemutatóhoz most *Debian 3.1*-est fogok használni.

A tehenek (COWs) a barátaid

A *Felhasználói Módú Linux* üzemszerű működtetésének van egy sajátos lehetősége, a *Copy-On-Write (COW)*; „írásnál másold”) fájl használata. Az „írásnál másold” egy elterjedt számítástechnikai mechanizmus, amely lehetővé teszi, hogy egy nagy sereg

adat megmaradjon csak-olvasható állapotban, azonban a változtatás előnyeit sem kell feladni: ezek egy másik helyen levő adathalmazban vannak nyilvántartva. A letöltött fájlrendszer-képmásfájl megmaradhat csak-olvasható; az **UML** rendszerünkben elkövetett változtatások csak a **COW** fájlban lesznek rögzítve. Ez arra is lehetőséget nyújt, hogy egyszerre több **UML** rendszert állítsunk lábra ugyanaból a (csak-olvasható) fájlrendszer-képmásfájlból, feltéve, hogy mindegyikhez saját **COW** fájl tartozik. Egy másik előny: ha az **UML** rendszerünk elromlik, egyszerűen törölhető a **COW** fájl, és minden kezdetű előlről. A **COW** fájl tipikus példája a „szellős” (*sparse*) fájloknak; bár a mérete ránézésre (amit az `ls -l` mutat) óriási, tényleges merevlemez-helyet (amit a `du` mutat) csak a nem-üres adatrészek foglalnak.

Az alkotóelemek összegyűjtése

Rakjuk össze a szükséges összetevőket egy frissen, ropogósan elkészített üres könyvtárba. Győződjünk meg arról, hogy van elegendő hely; végül is egy teljes értékű **Debian** telepítést szeretnénk tető alá hozni. Három gigabájt bőségesen elegendő lesz az alaprendszer számára. Töltsük le a **Debian-3.1-x86-root_fs.bz2** fájlt a uml.nagafix.co.uk weblapról, valamint a **2.6.14.3-bs3 UML** kernelt a www.user-mode-linux.org/~blaisorblade/binaries helyről. Végül szerezzük be az **UML** segédprogramok forrását innen: user-mode-linux.sourceforge.net/dl-sf.html. Ha ezek közül valamelyik már nincs ott, igyekszem megadni az új lelőhelyet a cikkhez tartozó források között. Álljon itt egy szkript a fenti művelet-sor megvalósítására. Összegyűjti a szükséges alkotóelemeket és lefordítja az **UML** segédprogramokat, melyek csak forrásban érhetők el. Ha nem lesz szükségünk hálózatra, elhagyható az `uml_utilities` tarlabda beszerzése és lefordítása. A parancsok megvalósíthatóak normál felhasználóként, kivéve az **UML** segédprogramok telepítését, mely rendszergazda jogokat igényel (ezért a `su root`):

```
mkdir /tmp/UML-Demo
cd /tmp/UML-Demo
```

```
wget http://uml.nagafix.co.uk/
↳ Debian-3.1/
-->Debian-3.1-x86-root_fs.bz2
bunzip2 Debian-3.1-x86-
↳ root_fs.bz2
wget http://www.user-mode-
↳ linux.org/~blaisorblade/
↳ binaries/
-->2.6.14.3-bs3/uml-release-
↳ 2.6.14.3-bs3.tar.bz2
tar -xvjf uml-release-2.6.14.3-
↳ bs3.tar.bz2
cp um32-2.6.14-release-mod/
↳ vmlinux-2.6.14.3-bs3 .
wget http://mirror.usermode
↳ linux.org/uml/
-->uml_utilities_
↳ 20040406.tar.bz2
tar -xvjf
↳ uml_utilities_20040406.
↳ tar.bz2
cd tools
make all
su root
make install DESTDIR=/
exit
cd ..
```

Ezzel megvannak a szükséges összetevők, táncra készen állunk. Minden Linux rendszernek van egy kernel parancssora. A legtöbb rendszerben ezt a betöltéskezelő hívja meg (**GRUB**, **LILO** stb.). Ebben az esetben viszont mi magunk fogalmazzuk meg a parancssort, amivel utasítjuk a rendszert a **Debian** gyökér fájlrendszer képmásfájl (és a hozzá tartozó, **Debian1.cow** nevű **COW** fájl) használatára – e kettő együttesen hozza létre a fájlrendszer gyökerét. Az aktuális terminálunk lesz a vendég **UML** rendszer konzolja:

```
cd /tmp/UML-Demo
./vmlinux-2.6.14.3-bs3 ubd0=
↳ Debian1.cow,Debian-3.1-x86-
↳ root_fs root=/dev/ubda
```

Miután lefutott a parancs, ismerős **Linux** kernelüzeneteket látunk megjelenni, végül ott villog a **Debian** bejelentkezési parancssor végén a kurzor. Beléphetünk **root**-ként (nincs jelszó), és kémleljünk kicsit körbe – magam az **1. Lista** szerint tettem meg ezt.

Elég dögös, nem igaz? Íme a legszemélyesebb **Debian 3.1** homokozónk, melyben kedvünkre építhetünk vagy

rombolhatunk. El lehet tekinteni a **hwclock**-ra és a **tty0**-ra vonatkozó figyelmeztetésektől; ezek a legtöbb **UML** rendszerben megjelennek, hiszen néhány hardverjellemzőt nem támogat az **UML** rendszermag. Ezen a ponton átállítható a **root** jelszó is, ha szükséges.

Lássuk a hálózati kapcsolat felhúzását. Szükség van két statikus **IP** címre, mégpedig az alagút két vége számára, melyet a korábban lefordított **UML** segédprogramjaink által fogunk létrehozni. Példámban a 192.168.1.100-at és a 192.168.1.101-et használom, de bármi más is megfelel, ami illeszkedik a helyi hálózat adottságaihoz. Az induláshoz az alábbi módon indítsuk el **Debian UML** rendszerünket:

```
cd /tmp/UML-Demo
./vmlinux-2.6.14.3-bs3
↳ ubd0=Debian1.cow,
↳ Debian-3.1-x86-root_fs
↳ root=/dev/ubda eth0=
↳ tuntap, , ,192.168.1.100
```

Ha **Debian** vendégrendszerünk lábra állt, jelentkezzünk be újból **root**-ként, és módosítsuk a hálózati konfigurációt.

Kezdjük a `/etc/network/interfaces` fájl szerkesztésével, melynek csak ennyit kell tartalmaznia:

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
address 192.168.1.101
netmask 255.255.255.0
gateway 192.168.1.1
```

A választásunknak megfelelő gépnevet írjuk be a `/etc/hostname` fájlba, végül másoljuk a gazdagép `/etc/resolv.conf` névfeloldási beállításait **Debian** vendégrendszerünk azonos helyére.

Indítsuk újra a vendégrendszert. Miután újra betöltődött, tudnunk kell pingelni őt a helyi hálózat bármely pontjáról.

Lenne néhány javaslatom az újonnan hálózatba kötött **Debian** rendszerünkre vonatkozóan. Először is telepítsük az **OpenSSH**-t, majd frissítsük minden telepített csomagot. Ezt az alábbi paranccsal tudjuk

1. Lista Debian 3.1 UML vendégrendszer betöltése

```
Debian GNU/Linux testing/unstable (none) tty0
(none) login: root <ENTER>
Linux (none) 2.6.14.3-bs3 #7 Fri Dec 16 17:47:00 CET 2005 i686
GNU/Linux
The programs included with the Debian GNU/Linux system
are free software;
the exact distribution terms for each program
are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY,
to the extent
permitted by applicable law.
```

```
(none):~# ps -ef <ENTER>
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	22:56	?	00:00:00	init [2]
root	2	1	0	22:56	?	00:00:00	[ksoftirqd/0]
root	3	1	0	22:56	?	00:00:00	[watchdog/0]
root	4	1	0	22:56	?	00:00:00	[events/0]
root	5	1	0	22:56	?	00:00:00	[khelper]
root	6	1	0	22:56	?	00:00:00	[kthread]
root	7	6	0	22:56	?	00:00:00	[kblockd/0]
root	8	6	0	22:56	?	00:00:00	[pdflush]
root	9	6	0	22:56	?	00:00:00	[pdflush]
root	11	6	0	22:56	?	00:00:00	[aio/0]
root	10	1	0	22:56	?	00:00:00	[kswapd0]
root	12	1	0	22:56	?	00:00:00	[kjournald]
root	299	1	0	22:56	?	00:00:00	/sbin/syslogd
root	305	1	0	22:56	?	00:00:00	/sbin/klogd
root	343	1	0	22:56	?	00:00:00	/usr/sbin/exim4 -bd -q30m
root	348	1	0	22:56	?	00:00:00	/usr/sbin/inetd
daemon	361	1	0	22:56	?	00:00:00	/usr/sbin/atd
root	364	1	0	22:56	?	00:00:00	/usr/sbin/cron
root	379	1	0	22:56	tty0	00:00:00	/bin/login --
root	380	379	0	22:56	tty0	00:00:00	-bash
root	384	380	0	22:57	tty0	00:00:00	ps -ef

```
(none):~# df -h <ENTER>
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/ubda	1008M	264M	694M	28%	/
tmpfs	768M	0	768M	0%	/tmp
tmpfs	14M	0	14M	0%	/dev/shm

```
(none):~# halt <ENTER>
```

```
Broadcast message from root (tty0) (Sun Jan 15 22:57:17 2006):
The system is going down for system halt NOW!
```

megtenni (válaszolva a menetközben felmerülő egyszerű kérdésekre):

```
apt-get update
apt-get install openssh-server
apt-get upgrade
```

Innentől a lehetőségek szinte korlátlanok. Bármely hálózati szolgáltatás vagy alkalmazás futtatható eme *Debian UML* vendégrendszer alól. Egyetlen gépen többféle kernelváltozat vagy disztribúció alatt

ki lehet próbálni alkalmazásokat. A fájlrendszer képmásfájl és a *COW* fájl fel lehet tölteni egy pendrive-ra, melynek révén stabil, számítógép-független fejlesztési környezetben dolgozhatunk akárhol. A *Felhasználói Módú Linux* könnyűvé és fájdalommentessé teszi olyan rendszerváltoztatások kipróbálását, melyek egyébként a betölthetetlenségig porba döntenék a rendszert.

A pusztítás öröme...

Rendben, ezek után már mindenki tudja, hogy már a burookban is erre vágyott tulajdonképpen. Íme, itt a lehetőség. Indítsunk el egy különálló *Debian UML* vendégrendszert, és engedjük el egy `rm -rf /.` parancsot. Ha hozzám hasonló érzésvilágú a kedves Olvasó, akkor valószínűleg remegő ujjakkal üti majd le ezt a néhány billentyűt. Indítsuk tehát el újdonsült *Debian* vendégrendszerünket az alábbi paranccsal (közben tudatosítva azt, hogy most egy másik, beszédes nevű *COW* fájl használunk, nem pedig azt, amin az imént véres verejtékkel beállítottuk a hálózati paramétereiket):

```
cd /tmp/UML-Demo
./vmlinux-2.6.14.3-bs3 ubd0=
↳ DangerDanger.cow,Debian-3.1-
↳ x86-root_fs
-->root=/dev/ubda
```

Miután halálraítélt barátunk betöltődött sorsába, essünk túl az elkerülhetetlenen. Győződjünk meg arról kétszer(háromszor), hogy ezt *Debian* vendégrendszerünk terminálablakában gépeljük be (2. Lista), nem pedig a gazdagépen. Elég rendesen elintéztük. Még egy `halt` parancsot sem bírnak kiadni, mivel maga ez a parancs is megsemmisült. Egy másik ablakból löjjük ki a rendszert ezzel a paranccsal:

```
killall -9 vmlinux-2.6.14.3-bs3
```

Most pedig nézzük meg, mi történne, ha újra be szeretnénk tölteni a rendszert a fenti paranccsal (3. Lista). Elvérzett. Mindenesetre tanuljunk a leckéből, és *ne* tegyünk ilyet igazi

2. *Lista* A rombolás örömszerző jellegű, főleg, ha csak próbaképpen tesszük

```
Debian GNU/Linux
↳ testing/unstable (none)
↳ tty0
(none) login: root
Linux (none) 2.6.14.3-bs3 #7
↳ Fri Dec 16 17:47:00 CET
↳ 2005 i686 GNU/Linux
The programs included with
↳ the Debian GNU/Linux system
↳ are free software;
the exact distribution terms
↳ for each program are
↳ described in the
individual files in
↳ /usr/share/doc/*/copyright.
Debian GNU/Linux comes with
↳ ABSOLUTELY NO WARRANTY, to
↳ the extent
permitted by applicable law.

(none):~# rm -rf /
rm: cannot remove
↳ `//proc/meminfo': Operation
↳ not permitted
rm: cannot remove
↳ `//proc/uptime': Operation
↳ not permitted
(... Many warnings about
↳ read-only filesystems
↳ omitted ...)
rm: cannot remove
↳ `//proc/loadavg': Operation
↳ not permitted
rm: `//proc/self' changed
↳ dev/ino: Operation not
↳ permitted
(none):~# df -k
-bash: df: command not found
(none):~# ps -ef
-bash: ps: command not found
(none):~# halt
-bash: /sbin/halt: No such
↳ file or directory
(none):~#
```

rendszerben. Szerencsénkre ez csak egy *UML* vendégrendszer *COW* fájjal, így egyszerűen törölhetjük a *DangerDanger.cow* fájlt (*jelentése: VeszélyVeszély*), és vendégrendszerünk vissza is állt eredeti egészségi állapotába.

3. *Lista* Egy igen beteg Debian *UML* vendégrendszer

```
(... boot messages omitted ...)
EXT3-fs: INFO: recovery required on readonly filesystem.
EXT3-fs: write access will be enabled during recovery.
kjournald starting. Commit interval 5 seconds
EXT3-fs: ubda: orphan cleanup on readonly fs
EXT3-fs: ubda: 66 orphan inodes deleted
EXT3-fs: recovery complete.
EXT3-fs: mounted filesystem with ordered data mode.
VFS: Mounted root (ext3 filesystem) readonly.
warning: unable to open an initial console.
Kernel panic - not syncing: No init found. Try passing init=
↳ option to kernel.
EIP: 0073:[<a01c6691>] CPU: 0 Not tainted ESP: 007b:b7f3afac
↳ EFLAGS: 00000282
Not tainted
EAX: 00000000 EBX: 000012eb ECX: 00000013 EDX: 000012eb
ESI: 000012e8 EDI: 00000000 EBP: b7f3afb8 DS: 007b ES: 007b
a10afb80: [<a0032d2a>] show_regs+0x21a/0x230
a10afbb0: [<a0016c8c>] panic_exit+0x2c/0x50
a10afbc0: [<a004a275>] notifier_call_chain+0x25/0x40
a10afbe0: [<a0037501>] panic+0x71/0x100
a10afc00: [<a000e2c0>] init+0x100/0x170
a10afc20: [<a002bf59>] run_kernel_thread+0x39/0x50
a10afce0: [<a001c3d4>] new_thread_handler+0xc4/0x120
a10afd20: [<b7f3b420>] 0xb7f3b420
```

Még néhány megjegyzés a *COW* fájlokkal kapcsolatban

A *uml_moo* segédprogram az *UML* segédprogramok egyike. Segítségével megtehetjük azt, hogy egy fájlrendszer-képmásfájlt és a hozzá tartozó *COW* fájlt beolvasunk, és egy olyan fájlrendszer-képmásfájlt generálunk, mely e kettő egybeolvasztásával születik (ahogy azt a vendégrendszer is látja futás közben). Ez egyszerűvé teszi a működő, jól beállított vendég-fájlrendszer képmásfájlok sokszorosítását.

Összefoglalás

A *Felhasználói Módú Linux* jó szórakozás, de van néhány igazi alkalmazása is. Ki lehet vele próbálni ismeretlen vagy megbízhatatlan alkalmazásokat, miközben a nem a teljes gazdagép lesz a veszélyeztetett terület, hanem csak a *COW* fájl. Virtuális hálózatkat is létrehozhatunk az egyszerre elindított *UML* rendszerekkel, így minimális ráfordítással lehet „laboratóriumi körülményeket” teremteni, ahol ki lehet próbálni mindazokat

az eszement linuxos trükköket, amelyeket félne az ember „rendes” számítógépeken elkövetni.

Linux Journal 2006., 145. szám

Matthew E. Hoskins vezető UNIX rendszeradminisztrátor a New Jersey Technológiai Intézetben, ahol a testületi adminisztratív feladatok jó részét ő látja el. Szeret azzal kísérletezni, hogy egészen eltérő rendszerek és szoftvereket bír együttműködéssel, többnyire egy vékony Perl ragasztótérrel (amit a helyiek „MattGlue” néven szoktak emlegetni; glue=enyv). Amikor nem számítógép-rendszerekkel trükközik, akkor a konyhában tesz gyaganezt. Matt a Hivatásos Újságírók Társaságának a tagja. Szomjúhozza az Olvasók visszajelzését – elérhető a matt@njit.edu e-mail címen.

KAPCSOLÓDÓ CÍMEK

↳ www.linuxjournal.com/article/8883