QEMU: dobozba zárt univerzum

Iskolánkban nemrég áttértünk az UHU-Linux 1.1-es változatáról az 1.2-esre. Ez egyrészt sok örömteli újdonságot hozott, ám egyben a glibc2.0 támogatás megszűnését is jelentette. Ennek köszönhetően aztán a csak bináris formában rendelkezésre álló Maple 7 programunkat el se lehetett indítani...

Amikor segítséget kértem a *Maplesofttól*, azt a lakonikus választ kaptam, hogy a *Maple 7* program támogatása már megszűnt. („*Please note that Maple 7 is no longer supported by Maplesoft*”.) A program *UHU 1.1* alatt ugyan nagyszerűen működik, viszont pusztán emiatt nem szerettem volna lemondani az annyi újdonságot hozó 1.2-es változatról. Eleinte *chroot*-olt környezettel kísérleteztem, de az operációs rendszer *glibc2.0* helyett *glibc2.1*-re (azaz *LinuxThreads*-ről *Nativ POSIX Thread Library*-re, *NPTL*-re) való átállása olyan mélyreható változásokat hozott, amit nem tudtam egyszerű kézimunkával helyreigazítani. Ekkor a *QEMU*-val próbáltam ki egy olyan lehetőséget, amely tulajdonképpen messze túlmutat az eredeti probléma megoldásán.

A QEMU-ról

Az UHU levelezőlistán kaptam az ötletet, hogy érdemes volna megpróbálkoznom a QEMU-val. Van hivatalos *.uhu* csomag is belőle, de mivel meg szerettem volna őrizni a leírás általánosságot, tehát nem akartam egy adott terjesztéshez kötődni, a cikk megírása során nem ezt használtam. E programnak természetesen vannak kereskedelmi vetélytársai (*Wabi, Virtual PC, VMWare, Serenity Virtual Station, TwoOstwo*), de számomra természetesen fontos szempont volt, hogy szabad szoftvert válasszak – részben a közismert anyagi okok miatt, de azért is, mert ennek lehet jobban a „lelke mélyére nézni”, ami az oktatásban szintén lényeges szempont.

A *QEMU* alkotója *Fabrice Bellard* (a név az ő bejegyzett védjegye), honlapja pedig a *http://fabrice.bellard.free.fr* helyen található. Amint a program neve is sejteti, egy emulátorról van szó: amely az *x86* architektúra mellett néhány más (*ARM*, *PowerPC*, *SPARC*) processzort is képes emulálni. Mindezt dinamikus fordítással, és egészen jó sebességgel.
A felhasználói licenc különböző részekre oszlik. A program lelkét jelentő virtuális processzor törzskönyvtár (*virtual CPU core library*) és a *PC* rendszer-emulátor *LGPL* alá tartozik, míg a felhasználói térben történő (*user mode*) emulátorhasználatra a *GPL* vonatkozik. Ugyanakkor a gyorsítómodul (*kqemu*) szabadalmaztatott termék. Ingyenesen hozzáférhető – ám a karbantartója tudni szeretné, ha valahol közzéteszik. (A modul nevének kezdőbetűje kissé megtévesztő, ugyanis a rendszernek semmi köze a *KDE* környezethez.)
A *QEMU*-nak kétféle futási módja van:

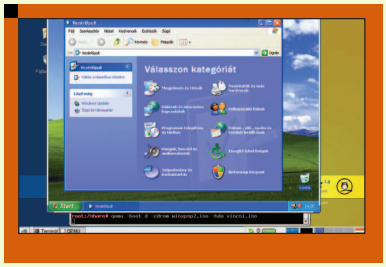
- *Teljes rendszeremuláció*. Ilyenkor a *QEMU* a teljes *PC*-t emulálja, beleértve a processzort és a különböző perifériákat is. Így más operációs rendszerek is futtathatók vele, sőt nyomkövetést is végezhetünk.

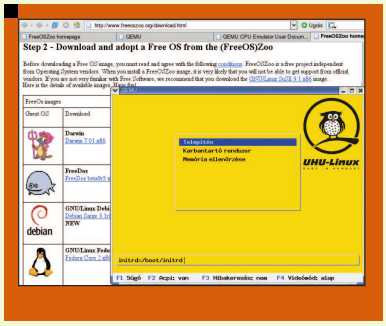
- *Felhasználói térben történő emuláció (csak Linuxon)*. Ilyenkor a *QEMU* a processzoron olyan programot is el tud indítani, amit más processzorhoz fordítottak. Ez segítségére lehet például a *wine* fejlesztőinek és használóinak is.

Számos érdekessége közül az egyik, ami jól jellemzi az emuláció minőségét: akár önmagán belül is elindítható egy másik *QEMU* emulátor. Ezt még a kereskedelmi vetélytársak mindegyike sem tudja produkálni.

A QEMU előkészítése

A cikk megírásakor forráskódban a *qemu-0.7.0* volt elérhető a fenti honlapon, ezt letöltöttem. Néhány kellemetlen órát okozott (meg néhány tanácstalan levélváltást a „haladó *UHU-Linux*” levelezőlistán), hogy sehogy sem akart lefordulni a forráskód – végül kiderült, hogy első próbálkozásra nem jött le a teljes tömörített fájl, csak a nagyobb része. Érdemes erre figyelni a letöltéskor.
A fordításhoz kell a *texi2html* program is, ez készíti el a *qemu-doc.html* és *qemu-tech.html* fájlokat, amit haszonnal lehet forgatni a továbbiakban. Letöltöttem a *kqemu-0.6.2-1* gyorsító modult is. Kibontás után a *kqemu* könyvtárat (tehát nem a tartalmát) bemásolva a *qemu-0.7.0* könyvtárba, szépen lefordult a program. A keletkezett binárisok a */usr/local/bin*-be kerültek a (rootként





végrehajtott) `make install` után: *qemu*-val kezdődnek neveik, és a különböző processzorok neveivel folytatódnak. Kivétel ez alól a *qemu-img*, ami a hivatalos *.uhu* csomagban *qemu-mkcow* néven szerepel: ez nem emulátor-indító, hanem képfájl készítő program.
A gyorsító modul betöltése disztribúciótól függően eltérően történhet.
A kézi megoldás biztosan működik, root felhasználóként kiadva az alábbi parancsot:

```
modprobe kqemu; rm -f /dev/
➥kqemu; mknod /dev/kqemu c 250
➥0; chmod 666 /dev/kqemu
```

Ellenőrzésként:

```
>modinfo kqemu
vermagic:       2.6.9-19 SMP
➥586 gcc-3.3
```

Ez a kézi töltögetés várhatólag néhány hónapon belül egy kulturálisabb megoldást kap az *UHU-Linux* esetében (és bizonyára más disztribúciókban is). Arról, hogy hogyan készül el az a (lehetőleg nagyméretű) fájl, amit a *QEMU* mint merevlemezt vagy *CD-ROM*-ot kezel, majd később ejtek szót részletesebben. Mindenesetre már itt is fontosnak tartom megemlíteni: minden egyes perifériát mi adhatunk meg az emulátornak egy-egy megfelelően előkészített fájl formájában.

Állatkerti séta

Mielőtt nekiláttam az eredeti célkitűzést jelentő programkörnyezet-kialakítási munkának, körülnéztem a *QEMU* honlapján. Találtam egy érdekes hivatkozást: *„Szabad Operációs Rendszerek Állatkertje" (Free Operating System Zoo):* ➲ *http://www.freeoszoo.org*
Számos (szabad) operációs rendszer képfájlja (*.iso* fájlja) van itt összegyűjtve, kifejezetten a *QEMU*-ban való futtatásra. Maga a *QEMU* is többféle rendszeren futhat (gazdagép, *host*) – lehet *Linux*, *MacOs X* vagy *Microsoft Windows*. A dobozolt (vendég, *guest*) rendszerek pedig számtalanul sokan vannak: a nagyobb *Linux* terjesztések, *BeOS*, *BSD* fajták, *FreeDOS*, *Darwin*, *ReactOS* stb.
Az „állatkert" elnevezés amúgy egyszerre utal a tanulásra és a biztonságra: a futtatott „állatfajta" garantáltan nem fog kárt tenni saját, kényelmesen berendezett világunkban. Ha nagyon akarjuk, az állatokat azért szabad etetni. Elérhető ugyanis a *QEMU*-val az is, hogy a ketrecbe be- vagy onna kijusson egy s más. (Ez nekem ugyebár nagy szerencse, mert különben hogyan futtatnám a Maple-t, vagy hogyan használnám az eredményül kapott fájlokat...)
Persze azért az is sejtethető, hogy az ilyen állapotban tartott élőlények nem olyan sebesen mozognak, mint vadon élő társaik... Pontosabban a sebesség erősen függ a rendelkezésre bocsátott erőforrásoktól (mint ahogy a jobb állatkertek is inkább állatparkok, kellően nagy kifutótérrel az állatok számára). Kimondani is szörnyű, de sokan a pingvint is olyan vadállatnak tekintik, amit inkább rács mögül kell nézegetni. Eme különös embercsoport számára a *Microsoft Windows* alól futtatott *QEMU* lehetőséget ad a többféle *Linux* „dobozban" való futtatására.
Az *UHU-Linux* levelezőlistájának archívumából (*http://lists.uhulinux.hu/arc/halado/2005-06/msg00306.html*) az is kiderül, hogy az „ablakos vadállat"-ot is telepíthetjük és futtathatjuk *QEMU*-ból. Az idézett írás nem kevesebbet állít, mint hogy a *Windows* gyorsabban betöltődhet *QEMU* alól, mint „normál módon". A használt rendszert pedig akár egy *DVD*-re is kiírhatjuk.
A *FreeOsZoo* honlapon nemes egyszerűséggel látszik a *QEMU* futtatásának legegyszerűbb módja:

```
qemu -hda guest_image_name.img
➥-boot c -user-net
```

Ebben a gyűjteményben voltak jó példák, a számomra megoldást jelentő képfájlt azonban – *Szilágyi Szilveszter* ötletét megfogadva – mégis máshonnan szereztem be: a *Damn Small Linux live CD*-jét választottam, 2.4-es kernellel (és persze glibc2.0-val). Kísérleteztem az 1.1-es *UHU*-val is, de ez valahogy ágyúval verébre való lövöldözésnek tűnt, oly nagynak ígérkezett a képfájl, és oly lassúnak a telepítés. Azért az szívdobogtató látvány volt, amikor a

```
qemu -boot d -hda hda.iso -hdb
➥hdb.iso -cdrom uhu1.1.iso
```

parancs hatására először láttam meg *UHU 1.2*-t futtató gépem „dobozában" a régi világból idecsöppent 1.1-es *UHU-Linux* induló telepítőképernyőjét.

A qemu-img, a képfájl-előállító

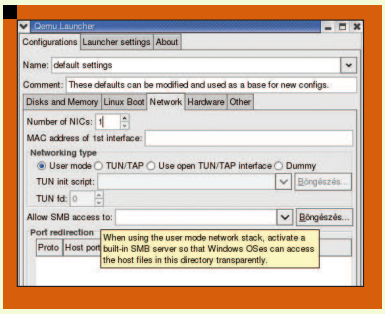
Kiadva a `qemu-img` parancsot, segítséget kapunk a *„QEMU disk image utility"* használatáról.
Egy példa:

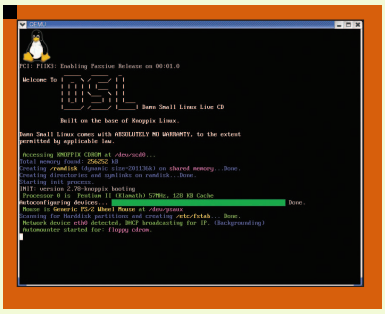
```
qemu-img create hda.iso 100M
```

elkészít egy 100 MB-os image fájlt, ami először üres. (Ne felejtsük le a sor végi `M` betűt!) A típusa többféle lehet és ez természetesen erősen meghatározza a méretet is. Választhatunk tehát `vpc`, `bochs`, `dmg`, `cloop`, `vmdk`, `qcow`, `cow` vagy `raw` formátumot.
Bár a mérete ennek a legnagyobb, sok szempontból az alapértelmezett `raw` az ideális, amelyet aztán a

```
mount -o loop
```

paranccsal könnyedén tudunk fel és lecsatolni, illetve esetleg beletenni valami „eleséget az állatnak".
A `cow` a *copy-on-write* rövidítése, és akkor használható jól, amikor egy csak olvasható képfájlból indulunk ki (*base_image*), és valahol tárolni szeret-



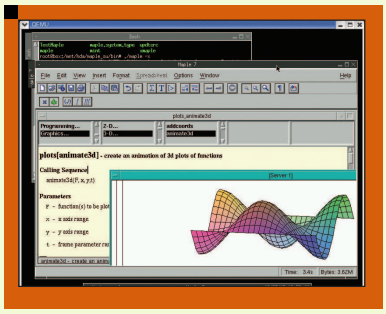


nénk a módosításokat. Gondoljunk csak bele, micsoda nagyszerű lehetőség ez live CD-k testre szabására, megszelídítésére! Erre van kihegyezve a *QEMU* `-snapshot` módja – ennek használatakor minden képfájl írásvédett, és minden változás a */tmp*-ben őrződik. Ezek a változtatások azonban visszaírhatók a képfájlra (*CTRL+Alt+s* paranccsal). A program kezel titkosított vagy tömörített képmásfájlokat is (`qcow` formátum), amiknek a mérete „belülről” (az emulátorból) nézve például 4GB, míg „kívülről” (a gazdagépen mint fájlt tekintve) igazából csak annyi, amennyit a „benti winchesterre” felmásoltunk.

Kiadva a `qemu` parancsot, hosszú listát kapunk a paraméterekről.
Minden megadható, ami csak az ember eszébe juthat: perifériák, nyelv, idő, memória, hálózat, képernyő, grafikus mód, betöltési paraméterek, kernel, initrd. Van ezen kívül számos profi lehetőség, mint például a *gdb* port nyomkövetéshez, eszközök átirányítása... De nem kell kétségbe esni, pilótavizsga nélkül is jól használható mindez, mert jól vannak beállítva az alapértelmezések.
Akit mélyebben érdekel a kérdéskör, annak javaslom a `qemu-launcher` letöltését. Ez egy nagy tudású, *Gtk*, *Gtk::GladeXML* és *Gnome2 PERL* modulokat használó grafikus felület. ***(UHU 1.2 alatt történő elindításához tehát kellenek a perl-gnome2*.uhu csomagok.)*** Aki nem akarja használni, annak is megér egy pillantást, hogy mi minden állítható be (parancssorból is persze) a *QEMU*-ban.

A QEMU bevetés közben

No de térjünk vissza az eredeti kérdéshez: hogyan is fogjuk futtatni a *Maple*-t? Egy olyan kis méretű operációs rendszert kerestem, ami grafikus felületet is ad.
Az internetről is letölthető a ***Damn Small Linux .iso*** fájlja, ami elfér egy névjegykártya méretű CD-n (50MB).



Volt nálam egy ilyen még a tavalyi PHP-konferenciáról, erről elkészítettem az .iso fájlt `dd` paranccsal (`dd if=/dev/cdrom of=dsl.iso`), majd (a fenti `aqemu-img` paranccsal legyártott ***hda.iso*** segítségével) kiadtam a várva várt

```
qemu -boot d -hda hda.iso
➥-cdrom dsl.iso -m 256 -k hu
➥-user-net
```

parancsot.
512 MB RAM van a gépemben, ebből 256-ot szántam az emulátornak (`-m` kapcsoló). Ez nem ment első nekifutásra, de nagyon értelmes és használható hibaüzenetet kaptam, amiből kiderült, milyen paranccsal tudom elérhetővé tenni ezt a nagy memóriaterületet:

```
umount /dev/shm
mount -t tmpfs -o size=272m
➥none /dev/shm
```

A ***QEMU*** annak függvényében állítja be az emulált processzor sebességét, hogy éppen mennyi erőforrás szabad. Ha nem futott más program, 790 MHz is „összejött”, viszont egy ***GIMP*** futtatása mellett (amivel a képernyőképeket begyűjtöttem) néha leesett 20 MHz-re is a sebesség (ami érdekes módon egészen jól használhatónak bizonyult parancssorban).
Aki mélyebben szeretné beleásni magát a ***QEMU*** világába, annak ideális eszköz a ***QEMU monitor*** (***CTRL+Alt+2***), amivel kifinomult módon vezérelhető az emulátor. Használható például kivehető eszközök be- és kicsatolására, naplózására, a ***Virtuális Gép (VM)*** befagyasztására, ami ily módon elmenthető vagy helyreállítható egy korábbi állapotból, valamint egy-egy állapot külső nyomkövetővel való ***VM***-böngészésre. Ebből az állapotból a ***CTRL+Alt+1*** kombinációval lehet visszaváltani a normál nézetre.

Van még egy ***CTRL+Alt+3*** képernyő is, de egyelőre maradjon meglepetés, hogy ennek mi a tartalma.

A vadállat etetése

Először még megformázatlanul vár bennünket a ***hda***-nak szánt ***hda.iso***. Az egeret a ***QEMU*** felületére vihetjük, és rákattinthatunk. Ekkor az egér már a „ketrecben” mozog, ahonnan aztán ***CTRL+Alt*** kombinációval csalogathatjuk ki újra (ez olvasható is az ablak fejlécén, nehogy elfelejtse valaki).
CTRL+Alt+F kombinációval lehet teljes képernyőt kérni (vagy a QEMU indításakor a `-full-screen` kapcsolóval).
Kérjünk egy terminálablakot a „ketrecben”, lehetőleg root-ként. Majd:

```
mke2fs /dev/hda
```

paranccsal megformázható (és esetleg a `tune2fs -j -c0 /dev/hda` segítségével ez ***ext3*** típusúvá is alakítható) a ***hda***-ként jelzett munkaterületünk. Tévedés ne essék: ez ***nem*** az eredeti gép ***/dev/hda***-ja (még akkor sem, ha az `mke2fs` történetesen arról panaszkodik, hogy ugyan már mi ütött belénk, hogy egy teljes merevlemezt szeretnénk megformázni, nem pedig csak egy partíciót?!). Előzőleg önmagunk megnyugtatásaként ki is próbálhatjuk például a `cfdisk` segítségével, hogy valóban akkora-e az a bizonyos ***hda***, amekkorát szerettünk volna.
Ezek után próbaként felcsatolható a frissen formázott félig virtuális merevlemezünk egy egyszerű `mount /mnt/hda` paranccsal (a ***/etc/fstab***-ból már tudja a ***Damn Small Linux***, hogy ezt hová kell csatolni). Adjunk ki egy `touch ittjartam` parancsot, hogy később akárki láthassa, hogy valóban ez az a terület, amihez hozzáférünk innen is, onnan is.
Első körben csak ennyi történt a „ketrecben”. Szépen, kulturáltan lehet kikapcsolást kérni a grafikus felületről – azaz a ***QEMU*** egyelőre befejezheti futását. (Ez elérhető a ***CTRL+Alt+x*** paranccsal is.) Ezek után beadjuk az állatnak az eledelt.
Felcsatolva eredeti operációs rendszerünk valamelyik könyvtárába a ***hda.iso*** fájlt (`-o loop` kapcsolóval) – és feltéve persze, hogy az kellően nagy méretű – belemásolható a ***maple_su (maple single user)*** könyvtár, ahol mindaz megtalálható, amire szükségünk lesz. (Remélhetőleg látszik az ***ittjartam*** fájl is, ami most már törölhető.)
Ezután `umount`-tal lecsatolva a ***hda.iso***-t, indulhat a végjáték. Ismét adjuk ki a

```
qemu -boot d -hda hda.iso
➥-cdrom dsl.iso -m 256 -k hu
➥-user-net
```

parancsot (vagy inkább írjunk egy kis parancsfájlt, amiben ezt eltároljuk).
A grafikus felületen root-ként felcsatolhatjuk a ***hda***-nkat (mint az előbb).

S lám, a vadállatból kezes bárány vált...

A `/mnt/hda/maple_su/bin/maple` paranccsal karakteres felületet kapunk, ami mondjuk a 888! pontos kiszámolására, a ? első 1000 jegyének kiírására vagy egy harmadfokú egyenlet analitikus megoldására teljesen elegendő.
Az `-x` kapcsolóval azonban elénk tárul a csodálatos világ háromdimenziós (akár mozgó, `animate3d`) ábrákkal. Egy-egy szép matematika dolgozat számára el is menthető, exportálható a kimeneti képernyő, vagy annak egy-egy ábrája (sokféle formátumban), amit szintén a ***hda*** területünkön tárolhatunk, és ha elhagytuk a ketrecet, akkor a jól megszokott operációs rendszerünkben használhatjuk `-o loop` kapcsolóval felcsatolva a ***hda.iso*** fájlt.
Egyetlen képfájlban is konzerválható a ***QEMU***-ban futtatandó rendszer: a 2005. júliusi ***Linuxvilág „Készítsünk Live CD-t!”*** című cikkének segítségével (38. o) tetszőleges kiindulási ***Live CD***-ből elkészíthető egy saját (például ***Maple***-t is tartalmazó) ***.iso*** fájl.
Ily módon ezt az alkalmazást a későbbiekben is jól tudja majd használni bárki, akinek szüksége lesz rá – még akkor is, ha már nem is emlékszik senki arra, hogy valaha használtunk ***glibc2.0***-t.
Sok sikert a ***QEMU*** megismeréséhez!

Szabó Zoltán
Három gyermekével és feleségével Pannonhalmán él. Tíz éve kísérletezik a Linuxszal. Matematikát és informatikát tanít, diákotthonban keseríti a rábízottak életét. Szívügye a PHP és a PostgreSQL.
(szz@freemail.hu)