

Egy számítógépterem „évköznapijai”

Egy számítógéptermet – főleg iskolai környezetben – nem elég csak talpra állítani. Üzemeltetés közben akadnak meglepetések, és folyamatosan fejlődnek a háttérben álló programok is. Ezek tanulságaiból olvasható itt egy kisebb csokorra való.

A QEMU-ről még egyszer

2005 szeptemberében egy *QEMU*-ről szóló cikkben (*Linuxvilág* #56/39. oldal) szó esett arról, hogy bináris formában meglevő régi programjainkat (például *Maple*) el lehet indítani egy emulátorban, például a *QEMU*-ban. Ezt az írást szeretném kiegészíteni: akkoriban a *QEMU* gyorsítását végző *kqemu* kernelmodult (a */dev*-beli bejegyzés nehézkes létrehozása miatt) elég körülményes volt betölteni – erről beszámol a cikk is. Azóta az *udev* rendszernek (és a *kqemu-0.7.2* megjelenésének) köszönhetően ez elegánsabbá vált. Elegendő egy bejegyzés a */etc/udev/permissions.d/50-udev.permissions* fájlba:

```
kqemu:root:root:0666
```

Ekkor a
modprobe kqemu

már minden gond nélkül betölti a gyorsítómodult.

Mennyire törhetetlen az ext3?

A 2005 áprilisában jelent meg a *Linux-sokszorosítás iskolai gépekre* (*Linuxvilág* #51/47. oldal) című írás. Azóta több olyan cikk jelent meg az újságban, amiről örömmel láttam, hogy nagyszerűen kiegészíti ezt (és viszont). Az októberi számban (*Linuxvilág* #51/27. oldal) részletesen olvashattunk a *System Rescue CD*-ről. Magam is ezt használom partíciók kezelésére (pontosabban ezen a *partimage* programot). Fontos

tanulsága volt az elmúlt heteknek, hogy ennek a *LIVE CD*-nek (vagy a winchesterre telepített változatnak) a leállítása előtt nem célszerű az

```
umount -a
```

parancsot kiadni (pedig ez igen csábító lehetőség a szkriptből való partíció-másolás, felcsatolás és a rajta való ténykedés után). Ha a frissen feltöltött partíciót felcsatoltam, frissen odamásoltam egy *menu.lst* fájlra, végül az *umount -a* paranccsal le akartam csatolni „mindent”, akkor újrainduláskor gond volt a *GRUB*-bal (csak annyit írt ki, hogy *GRUB*). S ez nemcsak néhány esetben volt így, hanem 30 gépből mind a 30 gépen így történt. Az *umount -a* valójában nem tudott minden fájlrendszert lecsatolni – hibával leállt, és nem nyúlt hozzá az általam felcsatolt merevlemez-partícióhoz sem, ilyen módon pedig sérült a fájlrendszer. (Ez azért érdekes tanulság, mert korábban az a kép élt bennem, hogy az *umount* utasítás nélkül is szinte mindent túlél az *ext3* fájlrendszer.) Kiderült, hogy ilyenkor explicite érdemes megadni, ha valamelyik partíciót le szeretnénk csatolni.

A pillanat, ami eljött

A 2005 szeptemberi számban jelent meg (*Linuxvilág* #56/51. oldal) *Medve Zoltán* írása *Nagyban kicsit más – Linux az oktatási szférában* címmel. Az 53. oldalon kilátásba helyezi, hogy „már ígérik, hogy a *GDM*-nek is lesz olyan szolgáltatása”, hogy a grafikus felületre való bejelentkezés előtt





elindulhasson egy (lebutított) böngésző, például az üzemeltetéssel kapcsolatos tudnivalókról.

Nos, ez a pillanat most eljött: **UHU 1.2** alatt sikerült ezt megvalósítani **GDM** alól.

Mindössze a `/etc/X11/xdm/Xsetup` konfigurációs fájl végére kellett odairni azt a sort, ami elindítja a böngészőt egy induló **HTML** fájlal. Apró kényelmetlenség ugyan, hogy ha hozzányúlunk a böngészőhöz, akkor nehéz visszakapni a bejelentkezési kurzort. Egyelőre ezt úgy tudom megoldani, hogy kérek egy menüpontot (pl. **Műveletek** – lent), majd a **Mégsem**-re kattintok (úgy, hogy az egér ne legyen a böngésző fölött), és újra ott villog a vékony kurzor a helyén.

Esélyegyenlőséget!

Belső hálózatunk webszerverén két kisebb **PHP** programmal próbáltam „esélyegyenlőséget teremteni”, már ami az egyes felhasználók gép mellett töltött idejének mennyiségét illeti. Az egyik egy térképszerű táblázattal mutatja meg, hogy melyik gépre ki jelentkezett be és mikor. Ehhez csak a `/etc/passwd` (és esetleg az csoportinformációt hordozó `/etc/group`) fájl kell

olvasnia a programnak, valamint a szervergépen indított `w` parancs kimenetét kell megfelelő módon kell feldolgozni.

Egy olyan adatfájlal, amely soronként tartalmazza a bejelentkezési nevekhez tartozó fényképek fájlneveit, arckép is megjeleníthető. A bejelentkezés óta eltelt idő alapján átszínezhajjuk az egyes felhasználók arcképe mögötti táblázatcellát, ezzel segítve a gépterem felügyelőjét, hogy a túl régóta jelenlevőket figyelmeztethesse, ha éppen sokan várakoznak.

Tükröm, tükröm, mondd meg nékem...

Nálunk **Samba** alapú hitelesítéssel lépnek be a diákok mind **Linux**, mind **Windows** alól a hálózatra. Bár – némi képp különös módon – ez a program meglehetősen népszerűségnek örvend, előfordul, hogy egy-egy diák elégedetlen a (többnyire diákigazolványból származó) portréjával.

A másik programocská az arcképcseré lehetőségét szolgálja. Ilyenkor természetesen nem automatikusan cserélődik le a kép, nehogy felismerhetetlen portré cserélje le a felismerhetőt, vagy ad abszurdum más login nevével beje-

lentkezve megtréfálják az illetőt. Az érdeklődők a **Linuxvilág** webhelyéről szabadon letölthetik a két említett program forráskódját.

Démonok, klónok, meg minden

Gyakran kerülök abba a helyzetbe, hogy – bármily gondosan gondoltam is át, hogy milyen programokra lesz szükség az év folyamán – kisül, hogy egy-egy informatikai szakkörön vagy tanórán bizony még szükség lenne egyebekre is, amik nincsenek telepítve.

Ilyenkor a „klasszikus” megoldás az, hogy egy gépet helyrehozunk, majd klónozással a többi gépre is felvisszük a változásokat. Ez persze meglehetősen időigényes, és erre általában nincs napközben lehetőség. Főleg akkor nem, amikor tele a gépterem, és szeretnének haladni a témával, csak épp valami hiányzó csomag miatt nem fordul le egy program. Az ilyen esetek miatt eszeltem ki a következő apró cselt.

A rendszerbetöltéskor elindítok a háttérben egy héjprogramot (kvázi „démont”), ami csak annyit tesz, hogy tíz másodpercenként ránéz egy (**Sambával** megosztott) hálózati

1. lista A *menu.lst*

```
default saved          # Fontos!!!

title the old kernel # vagy: az általában futtatott oprendszer
root (hd0,0)
kernel /old_kernel
savedefault

title the new kernel # vagy: a ritkán futtatott oprendszer
root (hd0,0)
kernel /new_kernel
savedefault 0          # Fontos!!!
```

könyvtárban levő fájlra (nevezzük ezt „retesz”-nek). Ha az ott van, akkor lefuttat egy másik (szintén hálózati könyvtárban levő) héjprogramot. Persze mindkét hálózati fájlt csak az adminisztrátor tudja írni vagy törölni (vagy létrehozni). Nyilván először létrehozom a futtatandót, és ha kész, csak akkor érintem meg (touch) a „reteszt”.

A lefuttatott program általában annyit tud, hogy ciklikusan megvizsgálja, megvan-e a */tmp* könyvtárban a „futottam-már” jelzőfájl. Ha nincs, akkor a *-y* opcióval (hogy ne kelljen a felmerülő telepítési kérdésekre manuálisan válaszolni) és a semmibe irányítva a kimenetet és a hibákat, letölt egy-két csomagot (például `apt-get -y install libidn-dev qt-designer>/dev/null 2>&1`), majd létrehozza a „futottam-már” jelzőfájlt a */tmp*-ben.

Amikor 10 másodperc múlva újra megvizsgálja a „futottam-már” fájlt, már ott találja, és nem kísérli meg újra a csomagtelepítést.

Nyilván hasonlóképp bármi más parancs is lefuttatható. Kiadható például egy `halt` parancs, ami megspórolja a gép kikapcsolásához tartozó egyéb fázisokat, amik a diákok késlekedését próbálják lefaragni. Főleg kisebb diákjaim igen szeretik azt a megoldást (tanóra végén), hogy az arra kiszemelt ember létrehoz a(z írható) hálózati könyvtárban egy „kikapcs” fájlt. Persze előzőleg én már elindítottam egy ciklust, ami néhány másodpercenként ránéz erre a fájlra, és ha megvan, akkor adja ki a `-` mindenkire ható `-halt` parancsot.

Ami elromolhat, az el is romlik

Nem is gondolná az ember, hányféle módon lehet elrontani ezt az egyszerű folyamatot. Például fájl helyett könyvtárat hoznak létre, vagy nagy betűvel hozzák létre: „Kikapcs”, vagy lecsatolják a könyvtárat stb. A tanári gépen célszerű előtte kilőni ezt a „démont”, nehogy az is áldozatul essen a kikapcsolási hadjáratnak. Érdeemes arra is vigyázni, nehogy olyan állapotban maradjanak a hálózati fájlok, hogy a legközelebbi óra elején a betöltés után egyből kezdetét vegye az össznépi kikapcsolás (mint ahogy már jártam).

Ha egy kicsit több időnk van (és esetleg egy-két program telepítésénél vagy fájlok másolásánál komolyabb változtatást eszközöltünk), akkor érdemes újraklonozni a gépeket. *Koblínger Egmont* megoldása igen elegáns: egy gépet helyrehozunk, majd *MD5*-vizsgálattal megállapítjuk, mennyiben tér el a klónozendó partíció (amit már használtak a felhasználók) a vágóytól – ezután csak a különbséget kell gépenként feltölteni.

Ennél kicsit lassabb egy másik módszer, ami arra alapoz, hogy azt a merevlemezt, amelyikről másolunk, mint egyik szűk keresztmetszetet kiküszöböljük: a partíciók másolásának elkezdése előtt egy héjprogram ciklusa másodpercenként figyel egy (hálózaton található) fájl meglétét. Mikor minden gép kész a másolásra, csak akkor hozom létre ezt a fájlt, és ekkor kezdődik a *partimage* futtatása. Ekkor nyilván túlnyomó részt a gyorsítótárból tudja adagolni az információt a szervergép. Sokkal gyorsabban végzek így egy

gépterem telepítésével, mintha sorban indítanám (10-20 másodperc különbséggel) az egyes gépeken a klónozást. Persze a csomagok ütközése sokat lassít, de azért egy 5 GB-os partíció 30 gépre való másolása 15 perc alatt lezajlik így.

A GRUB képességeiről

Szép lehetőség a *GRUB* azon opciója, hogy megadható, hogy legközelebb melyik menüpontnak megfelelő operációs rendszer induljon. Így nézhet ki a *menu.lst* fájl (1. lista).

Amikor olyan operációs rendszert (vagy új kernelt) szeretnénk indítani, amit csak egyszer kell futtatni (például a klónozáshoz szükséges *System Rescue CD* merevlemezen levő változatát), akkor előzőleg kiadjuk a

```
grub-set-default 1
```

parancsot.

Ekkor a másik opció töltődik be, de betöltéskor már át is állítja a legközelebb betöltendő menüpontot a másikra. Lehetőség van a `fallback` opció használatára is, ha több bizonytalan betöltendő menüpontunk is van...

Lehet távoli fájlrendszeren levő kernelt is betölteni az új *GRUB*-bal (sőt, a *menu.lst*-t is el lehet itt helyezni).

A http://www.gnu.org/software/grub/manual/html_node/helyen_vagy_a_GRUB_felhasznalok_kozossége_által_szerkesztett <http://www.autistici.org/grub/moin.cgi/FrontPage> wiki oldalon hasznos részleteket találhatunk ezzel kapcsolatban. Fellapozhatjuk persze a *Linuxvilág* 43. és 45. számát is, melyben *Komáromi Zoltán* írt a(z akkori) *GRUB*-ról, s akitől ezeket a hivatkozásokat és ötleteket is kaptam. Sok türelmet és sikert kívánok minden linuxos gépterem karbantartójának!



Szabó Zoltán

(szz@freemail.hu)
Három gyermekével és feleségével Pan-nonhalmán él. Tíz éve kísérletezik a Linux-szal. Matematikát és informatikát tanít, diákotthonban keseríti a rábí-zottak életét. Szívégye a PHP és a PostgreSQL.