

EGY EGYSZERŰ, GYORS ÉS SZABAD ÚJRATELEPÍTŐ ESZKÖZ: UPI

*Nagy Elemér Károly, eknagy@omikk.bme.hu
Marton József Ernő, jmarton@omikk.bme.hu
BME OMIKK*

Számos esetben, de különösen a nyilvános számítógépes termekben fontos, hogy a számítógépeket gyorsan és kevés beavatkozással lehessen újratelepíteni. Ez – hasonló hardverkiépítésű gépek esetén – sok esetben úgy a legegyszerűbb, ha egy „mester” gép beállításait és fájlrendszerait a többire „másoljuk”. Egy ilyen, a BME OMIKK által fejlesztett szabad másolóeszköz a uPi, amely segítségével egy FTP szerverről egy Linux indítólemez segítségével hálózaton keresztül lehet 10-30 perc alatt teljesen újratelepíteni egy tipikus munkaállomást. Az előadás ezt a szoftvert és ennek technikai részleteit mutatja be, a BME OMIKK-os felhasználáson keresztül.

Bevezetés

Nagymértékben hasonló munkaállomások, elsősorban azok szoftvereinek és a szoftverek beállításainak központositott módosítása, illetve frissítése gyakran felmerülő feladat, amelyet jobb híján manuálisan, a változásokat minden gépen egyesével elvégezve oldanak meg, különösen alacsony gépszám esetén. Egy másik, látszólag független probléma különálló gépek teljes tartalmának lementése, elsősorban rendszerleállások utáni helyreállítás céljából. A két esetben közös az egyik lehetséges megoldás, mégpedig az, ha a számítógép egy merevlemezének egy szeletének, azaz egy partíciónak a teljes másolatát (gépképet) egy gépre (egy másikra vagy ugyanarra) visszamásolunk (terítünk). A cikk az első esetre koncentrálna, amikor is több (száz) munkaállomást egy etalon munkaállomás képével terítünk, így elegendő a használt szoftverek biztonsági frissítéseit, újabb verziót, beállításait, friss adatbázisait, stb. egyetlen gépre telepíteni illetve egyetlen gépen beállítani. A uPi egy, teljes egészében szabad szoftvereken alapuló, hálózati környezetbe és a lehető legegyszerűbbre tervezett eszköz erre a célra. Jelen cikk a uPi rendszert mutatja be, elsősorban technikai szempontból.

Követelmények

Bár számos hasonló rendszer létezik a világban, az alábbi követelményeket talán csak a uPi képes egyszerre teljesíteni:

- Legyen olyan egyszerűen használható az eszköz, hogy egy oldal dokumentáció elolvasása után egy ECDL-el rendelkező felhasználó képes legyen használni
- Legyen olyan egyszerű az eszköz, hogy egy hozzáértő rendszergazda egy óra alatt képes legyen azt üzembe helyezni
- Ne igényeljen nem-minimális hardvert (DVD olvasó, PXE-s hálókártya, USB boot támogatás)
- Ne igényeljen különleges hálózati szolgáltatásokat (NFS, TFTP, IPX, DHCP, CIFS, Partimage szerver)
- Támogasson sokféle hardvert és operációs rendszert, ne kösse meg például az alkalmazható operációs rendszert
- Ne tartalmazzon olyan nem-szabad szoftvert, amely döntési kényszert jelenthet (pl. az operációs rendszer újabb verziójának újabb fájlrendszerének támogatás miatt újabb verziót kelljen vásárolni valamelyik komponensből)

- Sokféleképpen lehessen alkalmazni, ne egyetlen integrált rendszer legyen
- Jelentősebb infrastrukturális változtatások nélkül legyen bevezethető
- A teljes üzemeltetési költsége (TCO) legyen alacsony
- Lehessen szerver oldalon szkripteket hozzáadni

A megvalósítás

A megfogalmazott követelményeket kielégítendő egy GNU/Linux¹ alapú rendszert terveztünk, amely a Partimage² szoftvert használja a partíciók képfájlokba mentésére és visszaállítására. A kihívás az optimum megtalálása volt rugalmasság és a minimális hardvert igénylő rendszer (floppy és hálókártya) ellentmondásos követelményei között.

Az eredmény a következőkben ismertetett rendszer. Egy FAT-12-es fájlrendszerű lemezt a SYSLINUX³ rendszerbetöltő elhelyezésével teszünk a Linux kernel indítására alkalmassá. Továbbá szükséges egy megfelelően kicsi kernel, valamint egy programgyűjtemény, hogy egy eszköznek használható, erősen korlátozott szolgáltatásokat biztosító operációs rendszer álljon össze.

A BME OMIKK-ban a uPi-val telepítendő számítógépeink mind i386 kompatibilisek, és a bennük található hardverelemek – esetleg 3. féltől származó eszközközrelővel – támogatottak a 2.4.27-es Linux kernelben, így a jelenlegi rendszer erre épül. A feladat jellegéből adódóan a kernelnek az alaplap déli hídját, IDE illetve SATA vezérlőjét, valamint a hálózati interfészt és a kapcsolódó elemeket⁴ kell támogatnia – nem elfeledkezve az olyan elemekről sem, mint pl. a konzol I/O-hoz kapcsolódó eszközök, vagy az ELF bináris fájlok kezelése, különböző fájlrendszerek támogatása. Az uPi disztribúcióban három különböző kernel, konfigurációval együtt megtalálható.

A Linux kernel betöltődése után csatlakoztatja a root fájlrendszert. Jelen esetben ez is a lemezen, tömörítve található, amit kicsomagolva, mint RAM lemezt, csatlakoztat a futó kernel a rendszerhez. Ehhez az ún. *initial ramdisk*⁵ technikát választottuk. Ennek oka a modularitás maximalizálása volt. Így ugyanis lehetőség adódik a konkrét környezet ismeretében (beleértve a hardver, hálózati és a felhasználó által indítási időben megfogalmazott igényeket) lecserezni a root fájlrendszert, így a megfelelő, futási időben betöltendő kernel modulokat illetve programokat tartalmazó operációs rendszer áll rendelkezésre a reguláris *init* folyamat eredményeképpen. Ezért például a SCSI lemezeken futó szoftveres RAID-et és a PCMCIA-s hálókártyát is kernel modulok futási időben történő betöltésével kezelni képes a rendszer.

A kernel szoftverkörnyezete is érdemel néhány szót. A tervezés első körében a meglehetősen kiforrott GNU környezetben gondolkodtunk. Hamar kiderült, hogy nem indokolt az alkalmazása: rengeteg olyan szolgáltatást nyújt, amelyekre egy ilyen rendszerben természetesen nincs szükség⁶. Szerényebb funkciókínálata és sokkal kisebb mérete miatt a BusyBox⁷ környezetet választottuk helyette, amelyet éppen a szerény erőforrású rendszerekhez⁸ terveztek. A BusyBox környezet mellett apróbb komponenseket is beépítettünk, mint például a script-ekből kellemesebben használható *sfdisk*⁹, vagy a Microsoft rendszerek MBR-jával¹⁰ kompatibilis, szabad rendszerbetöltőt elkészítő *ms-sys -s*¹¹ programok.

1 <http://www.gnu.org/gnu/linux-and-gnu.html>

2 <http://www.partimage.org/>

3 <http://syslinux.zytor.com/>

4 Például TCP/IP protokoll stack

5 Az initial ramdisk, rövidebb nevén initrd támogatását külön be kell fordítani a kernelbe a „normál” RAM lemez támogatás mellé

6 Például nemzet- és nyelvfüggő komponensek, stb.

7 <http://www.busybox.net/>

8 Például beágyazott rendszerek, merevlemez nélküli illetve kevés memóriával rendelkező hardver környezetek

9 A util-linux csomag része

10 Master Boot Record, egyszerű esetben azt tartalmazza, hogy melyik partícióról tölthető be operációs rendszer

11 <http://ms-sys.sourceforge.net/>

Az operációs rendszer már rendelkezésre áll, következzen a gépképek elérési technológiájának ismertetése. A képfájlok tárolására FTP szervert választottunk, annak minden tekintetben¹² széles termékcsaládjára miatt. A Partimage részére a képfájlok elhelyezését átlátszó módon, a fájlrendszerbe csatolva kell, hogy biztosítsuk, amennyiben nem fut Partimage szerver a hálózaton¹³. Erre a LUFS¹⁴ rendszert választottuk, amely a fájlrendszerrel kapcsolatos teendők legnagyobb részét a kernelen kívül, hagyományos felhasználói folyamatként (ún. *userspace* programként) oldja meg, megkönnyítve ezzel a fejlesztést és hibakeresést.

Természetesen Partimage szerver használatára is lehetőség van, ez azonban nem szükséges, így Linux szerver nélkül is használható a uPi, az FTP szerverre – a letöltendő programok miatt – mindenképpen szükség van.

Következzen a keretrendszer konfigurációja a konkrét képterítési feladathoz. Egyetlen konfigurációs állomány szerkesztésével definiálhatjuk a hálózati környezetet (ideértve az FTP szerver elérési pontjait is, IP cím vagy DHCP megadását, stb.), valamint a feladatspecifikus szoftvereket. Az így kapott BusyBox/Linux operációs rendszerünk indulásának utolsó lépésében végrehajtandó script elkészítésével olyan, teljesen automatikusan működő rendszerre változtathatjuk a uPi-t, ami fölött teljes kontrollunk van – ezt a felhasznált komponensek szabad szoftver mivolta is garantálja. Másképp fogalmazva, egy szöveges konfigurációs fájl szerkesztése után, amiben beállítjuk például a szerver IP címét, egyetlen, "make floppy0" paranccsal elkészíthetjük a lemezt, vagy beállíthatjuk azokat a programokat, amelyek a rendszer betöltése után automatikusan elindulnak.

Természetesen lehetőség van a rendszert sokkal többre is használni, a rendszer ugyanis már a Partimage programot is az FTP szerverről tölti le, és éppúgy letölthet más programokat (például perl-t és gawk-ot) is, illetve shell fájlokat is, azaz az újratervezett gépen például MAC címtől függő módon módosíthatja a fájlokat, például beállíthatja a hálózati nyomtatókat – bizonyos korlátok között.

Bár a speciális igényekhez konfigurált uPi lemez elkészítéséhez célszerű egy GNU/Linux környezetet használni, a kész lemezek használatához még erre sincs szükség, csupán egy FTP szervere és egy számítógépre.

A uPi, mint diagnosztikai keretrendszer

Az eddigiekben a uPi egy kitüntetett oldalát vizsgáltuk: azt, amelyik életre hívta a rendszert. Az elkészült termék azonban több, mint egy képterítő rendszer. A konfigurálhatósága miatt például diagnosztikai rendszerként is felfogható. Munkaállomások hardverének automatikus tesztelésére, esetleg hálózati problémák felderítésére összeállított verzió is könnyedén kialakítható – csupán az utolsó lépésben a kiegészítő szoftverek és beállítások megfelelő gyűjteményét (netcat + memtest86) kell összeállítani. Természetesen a rendszer a példához hasonlóan a leltár ellenőrzésétől a napi inkrementális mentés készítéséig nagyon sok mindenre használható.

Elérhetőség

A uPi lemezkészítő csomag, a dokumentáció és kész lemezek a méltán világhírű nyílt forrástárházban, a SourceForge-on, a <https://www.sourceforge.net/projects/upidisk/> címen elérhetőek.

Teljesítmény

A BME OMIKK-ban az FTP szerver egy 666 MHz-es Pentium3 gépen futó Windows 2000 Professional-on futó FileZilla szerver, amely IDE lemezekről dolgozik, a 3GB-os, tömörítve 1,5GB-os gépkép elkészítése mégis csak 15 perc, a telepítése pedig csak 8 perc.

¹² Gyakorlatilag minden operációs rendszerre létezik néhány, így nem kötelezzük el magunkat operációs rendszer, és implicit tárolási alrendszer mellett ezzel a választással.

¹³ Ha fut, akkor a gépképek titkosított mozgatása is megoldott

¹⁴ Linux Userland Filesystem – <http://sourceforge.net/projects/lufs>