

Távoli Linux-indítás

Áttekintjük a távoli Linux-indítás alapjait, a hálózati indítást támogató rendszermaggal kapcsolatos követelményeket és feladatokat.

A „távoli Linux-felügyelet” kifejezés olyan Linux-munkaállomásokra vagy hálózati csomópontokra vonatkozik, melyek nem helyi adathordozóról töltik be a rendszermagot, hanem hálózaton keresztül (leggyakrabban ethernet) kapják meg az indításhoz szükséges utasításokat. Mivel a Linux-rendszermagot és magát az operációs rendszert is könnyű az igényeinknek megfelelően beállítani, a Linux-alapú rendszerek egymástól nagyon eltérő feladatokra is testreszabhatók, kezdve a webkiszolgálóktól a fűrtbe kötött rendszerrekig és az X-kiszolgálóig.

Miért hasznos a távoli felügyelet?

Az első kérdés, ami eszünkbe juthat, hogy miért van szükség egyáltalán arra, hogy Linux-rendszerünket távolról indítsuk el. Végére is egy Linux operációs rendszer helyben történő telepítése nem kíván többet, mint hogy behelyezzük a telepítőkorongot, válaszoljunk néhány kérdésre, majd ígyunk meg egy hosszúkávét, amíg a rendszer települ a gépünkre. Az egyedülálló munkaállomások esetében legalábbis ez a helyzet; mihelyt azonban megnő a fenntartandó és telepítendő munkaállomások száma, főként fűrtelrendezésű vagy kiszolgálófarmok esetén, a helyi adathordozóról történő telepítés már kevésbé ésszerű megoldás. Az olyan tömörített kiszolgálók megjelenésével, mint amilyeneket az RLX Technologies cég is gyárt, a távolról történő rendszerindítás szinte már szükségszerű, mivel ezek a kiszolgálók nem tartalmaznak hajlékonylemezes vagy CD-ROM-meghajtókat. Úgy készítették el őket, hogy operációs rendszerük a helyi, nagy sávsebességű ethernethálózaton keresztül töltődjön be, és távolról lehessen őket felügyelni.

A távoli Linux-rendszerek előnyei

- Központosított, jelenlétet nem igénylő felügyelet: bár sok telepítésnél valaki fizikailag is jelen van, hogy a CD-eket és a hajlékonylemezeket (24/7) kezelje, számos olyan rendszer is létezik, amely mellett hosszabb ideig senki sem tartózkodik. Ha egy programozó egy ilyen rendszeren például távolról szeretne dolgozni, és olyan betöltőállományra lenne szüksége, amely valamelyik helyi adathordozón található, balszerencséjére ezt csak akkor teheti meg, ha valaki a távoli gép mellett tartózkodik, aki az adott adathordozó behelyezéséről gondoskodik.
- Tömör kiszolgálói megoldások: mivel a fűrtbe fűzött és a kiszolgálófarmok esetén a változás egyértelműen a központosított, távoli felügyelet felé mutat, a CD- és a hajlékonylemezes meghajtók kezdenek elavulttá válni. Ezen helyi adathordozóknak már a pusztán jelenléte is bizonyos formai követelményeket támaszt a gépekkel szemben, növelve ezáltal az elérhető legkisebb fizikai méretet. A hálózati csomópontban lévő gépek nagyobb tömörsége érdekében egyes gyártók a CD- és hajlékonylemezes meghajtókat kivonják a forgalomból.
- Sokoldalúság: lehetőség nyílik kijavítani az olyan állományrendszerrel kapcsolatos hibákat, amelyek megakadályozzák,

hogy helyi adathordozóról indítsuk a rendszert. Például futtathatjuk az `fsck` parancsot egy távolról indított gép helyi meghajtóján és kijavíthatjuk az állományrendszer esetleges hibáit.

- **Költségek:** miért fizessünk olyan adathordozóért, amire nincs is szükségünk? Vannak cégek, amelyek merevlemez és egyéb helyi adathordozó nélküli munkaállomásokat árulnak biztonságos terminálkiszolgálói feladatokra. A biztonság itt abban nyilvánul meg, hogy mivel az alkalmazottak a helyi adathordozókhöz nem férnek hozzá, a kényes adatokhoz sokkal nehezebb hozzájutni.
- **Segít kiküszöbölni a különböző változatokból adódó buktatókat:** ha az összes munkaállomást távolról indítjuk egy közös rendszermag-képállomány segítségével, ezzel kiküszöbölhetjük, hogy a helyi adathordozón a rendszermagot állandóan frissíteni kelljen. Elégséges a közös rendszermag-képállományt egyszer frissíteni, ahelyett, hogy a munkaállomások merevlemezein, vagy ami még rosszabb, a rendszerbetöltő hajlékonylemezein kellene a változtatásokat elvégezni.

A távoli rendszerindítás folyamata

A távoli rendszerindítás utánozza a helyi indítás folyamatát, de a kettő között akad néhány fontos különbség. Anélkül, hogy az egyes feladatokat végrehajtó szolgáltatásokat részleteznénk, az alábbiakban áttekintjük a hálózatról történő rendszerindítás alapvető lépéseit:

1. A hálózati gép elindul vagy újraindul, és felkészül a hálózatról történő rendszerbetöltésre.
2. A hálózati gép szétsugározza a rá jellemző egyedi ethernet MAC-címét, keresve a kiszolgálót.
3. A kiszolgáló, melyet már korábban felkészítettek arra, hogy bizonyos MAC-címeket figyeljen, válaszként az ügyfélgép IP-címét adja vissza. A kiszolgáló egy előre meghatározott IP-tartományból vett IP-címmel válaszol minden olyan kérésre, amely a hálózat fizikai szintjén érkezik.
4. A hálózati csomópontban lévő gép megkapja az IP-címét és felkészíti saját ethernetcsatlóját a TCP-, illetve IP-kapcsolatra.
5. Az imént beállított csatló segítségével a gép most már elküldheti a rendszermagra vonatkozó kérelmet.
6. A kiszolgáló válaszul az ügyfélnek egy hálózati betöltőt küld, amely a hálózati behúzást támogató rendszermagot tölti majd be.
7. A hálózati betöltő – csak olvasható módban – befűzi a saját állományrendszert.
8. A betöltő beolvassa a kiszolgálótól kapott rendszermagot a helyi gép memóriájába és átadja neki a vezérlést.
9. A rendszermag írható, illetve olvasható módban befűzi a saját könyvtárat, a további állományrendszereket és elindítja az `init`-folyamatot.
10. Az `init` elindítja az adott gépre szabott szolgáltatásokért felelős démonokat, ezzel a rendszer teljesen feláll.

A leírásból látjuk, hogy az ügyfél hálózati gépe több ponton is függ a kiszolgáló válaszaitól. A lényeges pontok: a hálózati indítást támogató rendszer, a saját állományrendszer, és az a mód, ahogyan a rendszer, és az IP-adat eljut a kiszolgálótól az ügyfélhez.

Az alapok

Ahhoz, hogy egy hálózati gép távolról indítható legyen, a kiszolgálónak és az ügyfélnek jól meghatározott módon együtt kell működni. Számos alapvető követelménynek szükséges teljesülnie a távoli rendszerindításkor:

- Sokoldalú kiszolgáló: a kiszolgálónak a megfelelő szolgáltatásokat kell futtatnia (ezekről később esik szó), melyek a szükséges tudnivalókkal látják el az ügyfelet.
- Távoli gépindítás: a rendszerindítás megkezdéséhez képesnek kell lennünk, hogy fizikailag elindítsuk vagy újraindítsuk a távoli gépet. Ekkor még nem hagyatkozhatunk az ügyfélén lévő operációs rendszerre, annál is inkább, minthogy olykor éppen az operációs rendszer okozta lefagyás miatt kényyszerülünk újraindítani a gépet.
- A hálózat: az összes hálózati csomópontnak el kell tudni érnie a kiszolgálót, akár közvetlenül, akár közvetve egy átjárón keresztül. Ebben a cikkben csak az ethernethálózatokra térek ki.
- Hálózati indítást támogató rendszer: a rendszer lehet tömörített vagy tömörítetlen, jelölt vagy jelöletlen. A jelölt rendszer, amelyekről az alábbiakban még esik szó, az *Etherboot* nevű eljárás használatkor van szükség. További tudnivalókhöz a forrásokban megadott Etherboot weboldalt is meglátogathatjuk.
- Hálózati betöltő: ez teszi lehetővé a hálózati indítást támogató rendszer, és a kiszolgálóról történő beolvasását az ügyfél memóriájába.
- Az állományrendszer: a régi szép időkben az állományrendszert a hálózaton keresztül, NFS-alapon szolgáltatták. Az olyan eljárásokkal, mint a SYSLINUX, a teljes állományrendszert egy ramlemezre keresztül szolgáltatjuk, amit a rendszerrel együtt küldhetünk el az ügyfélnek.

Most, hogy már nagyvonalakban látjuk, miként zajlik le a távoli rendszerindítás, és tisztában vagyunk vele, hogy az ügyfél és kiszolgáló kapcsolattartásában milyen követelményeknek kell teljesülnie, nézzük meg részleteiben is az egyes szolgáltatásokat!

A gép bekapcsolása távolról

Egy számítógép távolról történő bekapcsolása csak egy kis szelete a távoli alkatrészsztű irányítás lehetőségeinek. Több gyártó is kínál termékei mellé olyan különleges alkatrészeket és programokat, amelyek lehetővé teszik a távolról történő megbízható gépkézelést. Ilyenek például a hőmérsékletfigyelés, a ventilátor- és áramellátás-kezelés, BIOS-frissítés, figyelmeztető jelzések stb. Egy hálózati gép távolról történő indításához ezen alkatrészsztű szolgáltatások közül csupán arra van szükségünk, amely a gép ki- és bekapcsolását lehetővé teszi, bár egy újraindító (reset) szolgáltatás is jól jöhet.

A távoli rendszerindításhoz a munkaállomáson valószínűleg meg kell majd változtatnunk a behúzó adathordozók sorrendjét. A jellemző sorrend általában: hajlékonylemez, CD-ROM, merevlemez. Az esetek nagy többségében a hálózatról való rendszerbeállítás lehetősége vagy nem is szerepel a listában, vagy az összes helyi adathordozó után, a lista legálján talál-

ható. Mivel a legtöbb munkaállomást már a merevlemezre előre telepített működő operációs rendszerrel szállítják, a gép újraindításakor vagy bekapcsolásakor a rendszer a helyi merevlemezről betöltődik.

Ethernet

A valódi hálózati rendszerindításhoz egy PXE-megfelelő ethernetcsatlóra lesz szükségünk. A PXE a Preboot eXecution Environment nevű eljárásra utal, ami része Intel Wired for Management (WfM) elnevezésű kezdeményezésének. Egy PXE-megfelelő ethernetcsatló képes arra, hogy betöltse és futtassa a kiszolgálótól kapott hálózati



betöltőprogramot, mielőtt még áthozná magát a rendszer, és a távoli rendszerindításhoz egy PXE-megfelelő ethernetcsatlóra lesz szükségünk. A PXE a Preboot eXecution Environment nevű eljárásra utal, ami része Intel Wired for Management (WfM) elnevezésű kezdeményezésének. Egy PXE-megfelelő ethernetcsatló képes arra, hogy betöltse és futtassa a kiszolgálótól kapott hálózati

Hajlékonylemez

Bár a cikk a távoli rendszerindításról szól, a hajlékonylemez pedig helyi adathordozó, mégis említést kell tennünk róla, mert a távoli rendszerindításnak létezik egy nagyon fontos kevert típusa. Mivel sok esetben az ethernetcsatló és a gép BIOS-a együttesen nem PXE-megfelelő, létrejött egy nyílt forrású kezdeményezés: az Etherboot. Az Etherboot segítségével rendszerindító hajlékonylemezt készíthetünk, amely egy egyszerű betöltőprogramot és egy ethernet-eszköz-meghajtót tartalmaz. Állítsuk be a rendszerindító adathordozók sorrendjét úgy, hogy a hajlékonylemez legyen az első helyen, majd így indítsuk el a gépet. Az Etherboot-lemezről a meghajtóprogram betöltődik a memóriába, és elkezdődik a MAC-cím szétsugárzása, valamint a kiszolgáló válaszára figyelés. Egy PXE-megfelelő rendszernek a kiszolgáló egy hálózati betöltőprogramot küld válaszként, míg az Etherboot-eljárás esetén egy hálózati indítást támogató jelölt rendszer, és a távoli rendszerindításhoz egy PXE-megfelelő ethernetcsatlóra lesz szükségünk. A PXE a Preboot eXecution Environment nevű eljárásra utal, ami része Intel Wired for Management (WfM) elnevezésű kezdeményezésének. Egy PXE-megfelelő ethernetcsatló képes arra, hogy betöltse és futtassa a kiszolgálótól kapott hálózati

© Kiskapu Kft. Minden jog fenntartva

Melyek az IP-adatok?

Amikor egy ügyfélgép a hálózatról indul el, akár a PXE-megoldást használva, akár hajlékonylemezeről, megkezdi MAC-címének sugárzását olyan kiszolgálót keresve a helyi hálózaton, amelyik a szükséges IP-adatokat majd a rendelkezésére bocsátja. Erre azért van szükség, hogy az ügyfél a megfelelő IP-értékeket az ethernetcsatló rendelkezésére bocsáthassa, és aztán a további párbeszédet már TCP-, illetve IP-alapon folytathassa. Több módja is létezik, hogy a hálózati gépet ellássuk a megfelelő IP-adatokkal: ezek a RARP, a BOOTP és a DHCP nevű módszerek.

RARP

A RARP-eljárás (Reverse Address Resolution Protocol) a hálózati csatlókártya egyedi 48-bites ethernetcíméhez (a MAC-címhez) rendel hozzá egy IP-címet. Amikor az ügyfél elindul a hálózatról, MAC-címét szétsugározza a vele azonos fizikai hálózaton lévő összes gépnek. Ezen gépek egyikén futnia kell a `rarpd` démonnak, mely a `/etc/ethers` állományból kiolvassa, hogy az adott 48-bites címhez melyik IP-címet kell hozzárendelni, majd válaszul ezt a vadiúj IP-címet elküldi az ügyfélnek. Miután megkapta az IP-címet, az ügyfél egy TFTP-kérést (Trivial File Transfer Protocol) indít el, hogy megkapja a megfelelő képállományt (erről később még szót ejtünk). Ennek a megoldásnak legnagyobb hátránya, hogy csak a helyi fizikai hálózaton működik, és csak egyetlen adatot szolgáltat: az ügyfél IP-címét.

BOOTP

A BOOTP (Bootstrap Protocol) -eljárás határozott fejlődés a RARP-hoz képest, mivel tartalmaz hálózati átjárótámogatást (a távoli rendszerindítás útvonalválasztón keresztül is működik), és sokkal több tudnivalót szolgáltat az ügyfél számára. A BOOTP az IP-címen kívül olyan adatokat is elküld az ügyfélnek, mint az átjáró (az útvonalválasztó) címe, a kiszolgáló címe, az alhálózati maszk, valamint a rendszerbehúzó állomány (a tulajdonképpeni képállomány). Tartsuk szem előtt, hogy egy adott alkatrész-címhez egy, és csakis egy IP-cím rendelhető hozzá.

A BOOTP legnagyobb hátulütője, hogy az IP-címeket egy az egy kapcsolatban felelteti meg a MAC-címeknek, vagyis egy adott MAC-címet mindig ugyanahhoz az IP-címhez rendeli hozzá. Ha elgondolkozunk azon, hogy milyen követelmények lépnek fel mondjuk olyan esetekben, ha valaki gyakran dolgozik más irodákban, vagy hordozható számítógépével sokat utazik, akkor látjuk, hogy ez az „egy az egyhez” típusú kapcsolat némileg korlátozó. Az ilyen cégek esetében a felhasználók a számítógépeikkel utaznak, és csak néha van szükségük arra, hogy a központi kiszolgálóhoz kapcsolódjanak, például amikor letöltik a leveleiket. Az idő nagyobbik részében IP-címük kihasználatlanul hever – hiszen nem adható ki másnak –, ami nagy pazarlás. Erre a gondra kínál elegáns megoldást a DHCP-eljárás.

DHCP

A DHCP (Dynamic Host Configuration Protocol) a BOOTP ésszerű utódja. A BOOTP valójában már elavultnak számít, a legtöbb helyen felváltotta a DHCP. Az egyik ok, amiért a DHCP népszerűségében felülmúlta a BOOTP-t, az, hogy az előbbi támogatja a dinamikus IP-cím kiosztást, míg az utóbbi csak állandó IP-címeket tud kiosztani (egy MAC-címet mindig ugyanahhoz az IP-címhez rendeli hozzá). A DHCP által támogatott dinamikus címkiosztás lehetővé teszi, hogy egy adott cím több hálózati gépnek is kiosztható legyen (természetesen nem egy időben). A „mozgékony” cégek alkalmazottjainak

példájánál maradványok a következők történének: a gép kapcsolódik a hálózathoz és szétsugározza a MAC-címét. A kiszolgáló, amelyen a `dhcpd` démon fut, már előre elkülönített egy IP-címtartományt a gépek részére, és a tartományban található következő szabad címet egyszerűen hozzárendeli az ügyfélhez. A DHCP az IP-cím kiosztás hosszú élettartamáért is felelős a DHCP-állományon keresztül.

A DHCP számtalan beállítási lehetőséggel bír; részletezésük sajnos túlmutat a cikk keretein. További tudnivalók találhatók a *Ralph Droms* és *Ted Lemon* által írt *The DHCP Handbook* című könyvben (Pearson Higher Education, 1999).

A rendszermag és a hálózati betöltő átküldése

Miután a hálózati gép megkapta az IP-címét és alkalmassá tette a csatlót a TCP-, illetve IP-alapú párbeszédre, a gép egy BIOS-eljáráson keresztül elküldi a behúzó képállományra vonatkozó kérelmet. Az IP-cím hozzárendelés és a képállomány elküldésének ez a szétválasztása szándékos; így válik lehetővé, hogy különböző kiszolgálók teljesíthessék ezeket a kéréseket. A TFTP (Trivial File Transfer Protocol) pont megfelelő eszköz a képállomány átvitelének lebonyolítására, mivel nagyobb testvérével, az FTP-vel (File Transfer Protocol) ellentétben az állományok letöltéséhez nem igényel felhasználói bejelentkezést. A TFTP kezdetleges biztonsági szolgáltatása abban merül ki, hogy alapértelmezetten csak a kiszolgáló `/tftpboot` könyvtárából küldhet állományokat. Mivel ez a biztonsági megoldás meglehetősen ismert a rendszergazdák körében, ezért a `/tftpboot` könyvtárba csak a nyilvánosságnak szánt állományokat rakják. A `tftp-hda` legfrissebb kiadásába már állományszintű biztonsági eljárást is beépítettek.

Figyeld meg, hogy egy képállomány átviteléről beszélünk – ez lehet jelölt rendszermag (Etherboot-eljárás) vagy hálózati betöltőprogram (PXE-eljárás). Ha az Etherboot-, vagyis a hajlékonylemezes rendszerbehúzási megoldást használjuk, akkor a BOOTP vagy a DHCP egy jelölt rendszermagra mutat. Ha igazi PXE-megoldással élünk, akkor a BOOTP vagy DHCP egy hálózati betöltőprogramra mutat. Ez utóbbi esetben a hálózati betöltő a gép memóriájába töltődik és TFTP-n keresztül egy jelöletlen rendszermag letöltését kezdeményezi. A PXE használatához a TFTP-kiszolgálónak támogatnia kell a `tsize` nevű beállítási lehetőséget (RFC 1784, RFC 2349). A *H. Peter Anvin* által írt `tftp-hpa` program támogatja ezt a lehetőséget. A program letölthető a <http://www.kernel.org/pub/software/network/tftp> címről.

Monolitikus rendszermag

Akár a két lépésben történő PXE-rendszerindítást választjuk (először a hálózati betöltő, majd a rendszermag), akár az egy lépésben lezajló Etherboot-eljárással történő behúzást (csak rendszermag), a végeredmény az lesz, hogy a rendszermag betöltődik az ügyfél memóriájába és megkapja a vezérlést. Talán már felmerült benned, hogy mi is a különbség egy hálózati rendszerindításra készített rendszermag, és egy átlagos, helyi merevlemezeről történő indításhoz készített rendszermag között. Az első, amit el kell döntenünk, hogy modúláris vagy monolitikus rendszermagot építünk-e – az utóbbi esetben nem használhatunk betölthető modulokat. Ha már fordítottál rendszermagot, akkor tudod, hogy a beállítás során a különböző szolgáltatásokat különbözőképpen építhetjük a magba: az `Y` beállítással a szolgáltatás a rendszermag része lesz, `N` beállítással a szolgáltatás nem kerül be a rendszermagba, az `M` beállítással a szolgáltatás csak szükség esetén épül be a rendszermagba. Ha monolitikus rendszermagot építünk, az `M` lehetőséggel nem élhetünk.

A rendszermag jelzőbitjei

Számos jelzőbitet kell bekapcsolnunk, ha monolitikus, hálózati indítást támogató rendszermagot szeretnénk építeni. Először a monolitikus rendszermagot építjük meg, amihez a modulátogatást ki kell kapcsolnunk. A rendszermag `.config` állományában a következőket látjuk:

```
#CONFIG_MODULES is not set
```

ha monolitikus rendszermagot készítünk, ezt a lehetőséget ki kell kapcsolnunk:

```
CONFIG_MODULES=n
```

Az ext2 állományrendszer támogatására szükségünk lesz – ha ezt az állományrendszert fogjuk használni a helyi ügyfélgépen, vagy befűzni egy távoli kiszolgálóról, mint például az NFS saját állományrendszer esetén:

```
CONFIG_EXT2_FS=y
```

Amennyiben távoli saját állományrendszert kívánunk használni, amit NFS-en keresztül csatolunk, akkor az NFS-támogatásra is szükségünk lesz:

```
CONFIG_NFS_FS=y
CONFIG_ROOT_FS=y
```

Mivel az ügyfél indítására az Etherboot-eljárást fogjuk használni, be kell kapcsolnunk a hálózati beállítások lehetőségét, valamint legalább annak a hálózati csatoló kártyának (NIC) a támogatását, ami az ügyfélgépben található:

```
CONFIG_NETDEVICES=y
CONFIG_EEEXPRESS_PRO100=y
```

Végül be kell állítani a RARP-, BOOTP- és DHCP-eljárásokon keresztüli IP-címhozzárendelés lehetőségét:

```
CONFIG_IP_PNP_RARP=y
CONFIG_IP_PNP_BOOTP=y
CONFIG_IP_PNP_DHCP=y
```

Ha a saját állományrendszert ramlemezen keresztül szolgáltatjuk, nem pedig NFS-en keresztül (lásd a következő bekezdést), akkor a RAM-állományrendszer lehetőségét engedélyezni kell:

```
CONFIG_BLK_DEV_RAM=y
CONFIG_BLK_DEV_INITRD=y
```

Moduláris rendszermag

Moduláris rendszermag építések az M betűvel választhatjuk ki azokat a modulokat, amelyeket a rendszermag csak szükség esetén tölt be. Mivel a modulok nem állandóan beépítettek a magba, kell valamilyen tárolóhely, ahonnan betöltődnek, ha szükség van rájuk. A Linux rendelkezésünkre bocsát ilyen területet, ez a ramlemez. A ramlemez csak a rendszerbehúzás folyamán él, és csak a modulokat tartalmazza. Használhatnánk távoli saját állományrendszer szolgáltatására is, azonban ennek részletezése kívül esik cikkünk keretein.

A moduláris rendszermag készítésének bemutatásához a MINIX állományrendszer-támogatást modulként adjuk a rendszermaghoz. Így a rendszer kezelni képes a MINIX saját állományrendszert, amit hajlékonylemezzel töltünk be. A modularitás támogatásához legelőször is a modulok támogatását be kell kapcsolnunk, majd a rendszermagot újra kell fordítanunk:

```
CONFIG_MODULES=y
```

Példánkban a MINIX-támogatást dinamikus modulként valósítjuk meg, így ez a beállítás M betűt kap:

```
CONFIG_MINIX_FS=M
```

Miután moduláris rendszermagunkat (bzImage) újrarendeltük és átmásoltuk a `/tftpboot` könyvtárba, keresnünk kell egy módszert a betöltésre. Egy olyan eszköz, mint például a nyílt forrású SYSLINUX képes rá, hogy betöltse a rendszermagot és létrehozza a ramlemez. Egy `pxelinux.cfg` mintaállomány alapbeállításban körülbelül a következőképpen néz ki:

```
DEFAULT bzImage
```

```
APPEND vga=extended initrd=minix.gz
root=/dev/fd0 ro
```

```
PROMPT 1
```

```
TIMEOUT 50
```

A beállítóállományban láthatjuk, hogy a rendszermag neve bzImage, a ramlemezé `minix.gz`, és az ügyfél saját állományrendszer a hajlékonylemezzel lesz betöltve (`/dev/fd0`). Létre kell hoznunk a ramlemez, ami a MINIX-modult (`minix.o`), az `insmod` parancsot (ennek segítségével tölthetjük be a MINIX-modult), a konzoleszközt (`/dev/console`) és a `linuxrc` parancsot tartalmazza, utóbbi a ramlemez használatkor lesz meghívva. (A ramlemez Linux alatti működéséről mindenre kiterjedő leírást olvashatunk a *Werner Almesberger* és *Hans Lerman* által karbantartott `initrd.txt` állományban, mely része a rpm-formátumban terjesztett rendszermagoknak.) Egyéni ramlemezünk létrehozásához, amely a MINIX-modult majd dinamikusan befűzi a rendszermaghoz, illetve a hajlékonylemezzel betölti a MINIX saját állományrendszert, egy állományt kell létrehozunk, és a `dd` parancs segítségével ki kell nulláznunk:

```
dd if=/dev/zero of=minixroot bs=1k count=4096
```

Ezután az állományt egy hurokeszközhöz (loopback) rendeljük `/dev/loop0`:

```
losetup /dev/loop0 minixroot
```

Létrehozzuk az ext2 állományrendszert:

```
mkfs.ext2 /dev/loop0
```

Az eszközt egy szabadon választott csatolási ponthoz fűzzük, például az `/mnt`-hez:

```
mount /dev/loop0 /mnt
```

Létrehozuk a legszükségesebb könyvtárakat:

```
mkdir /mnt/dev /mnt/lib /mnt/sbin
```

Létrehozuk a konzolcsköt:

```
mkdev /mnt/dev/console c 5 1
```

A `minix.o` modul a `/lib` könyvtárba, a `sash` és `insmod` állományokat pedig a `/sbin` könyvtárba másoljuk:

```
cp /usr/src/linux/fs/minix/minix.o /mnt/  
↳ lib/minix.o
```

```
cp /sbin/sash /mnt/sbin/sash
```

```
cp /sbin/insmod /mnt/sbin/insmod/
```

Nézzük meg, hogy az `insmod` parancsnak mely rendszerkönyvtárakra lesz szüksége:

```
ldd /sbin/insmod
```

és ezeket másoljuk be a `/mnt/lib` könyvtárba:

```
cp /lib/libc.so.6 /mnt/lib
```

```
cp /lib/ld-linux.so.2 /mnt/lib
```

Ezután hozzuk létre a `/mnt/linuxrc` parancsállományt, amely a MINIX-modult fogja betölteni:

```
#!/sbin/sash
```

```
/sbin/insmod /lib/minix.o
```

Tegyük végrehajthatóvá az állományt:

```
chmod 777 linuxrc
```

Fűzzük ki a `/mnt` könyvtárat, és válasszuk le a hurokeszközt:

```
umount /mnt
```

```
losetup -d /dev/loop0
```

a `gzip` paranccsal tömörítsük a `minixroot` állományt, és ezzel fel is készültünk a gép indítására:

```
gzip minixroot
```

A fenti példával addig a pontig jutottunk el, amikor a rendszer a saját állományrendszerét a `/dev/fd0` eszköztől megpróbálja betölteni. Ahhoz, hogy ez a példa teljesen működőképes legyen, létre kell hoznunk egy MINIX-állományrendszert a hajlékonylemezen, és az eszközt be kell fűzni a rendszerbe. Legalább az `init`, az `sh` parancsokat, illetve az ezekhez szükséges függvénykönyvtárakat át kell másolnunk a lemezre, és létre kell hoznunk a konzolcsköt.

A távoli NFS saját állomány

Amint az köztudott, az élet a saját állományrendszer (`/`) nélkül értelmetlen. Helyi adathordozóról történő rendszerindításnál

Könyvek

Internetworking with TCP/IP, Volume 1, 4th Edition, szerzője *Douglas E. Comer*. Prentice-Hall, 2000. ISBN 0-13-018380-6.

Linux Core Kernel Commentary, szerzője *Scott Maxwell*. Coriolis Open Press, 1999. ISBN 1-57610-469-9.

Linux in a Nutshell, 3rd Edition, szerzője *Ellen Siever et. al.* O'Reilly & Associates, Inc., 2000. ISBN 0-596-00025-1.

TCP/IP Illustrated, első rész, szerzője *W. Richard Stevens*. Addison-Wesley, 1994. ISBN 0-201-63346-9.

a saját állományrendszer majdnem mindig a helyi merevlemezen található. De honnan kapja meg egy távolról indított gép a saját állományrendszerét? Két választásunk adódik: egy távoli kiszolgálótól NFS-en keresztül befűzzük a sajátot vagy ramlemez használunk. Amennyiben a saját állományrendszer NFS-en keresztül szolgáltatjuk, akkor alapértelmezés szerint a rendszer a `/tftpboot/ip` elérési úton keresi a sajátot, ahol az IP helyén az ügyfélgép IP-címe áll. Ehhez a kiszolgálón kell elindítanunk az NFS-t és exportálnunk kell a `/tftpboot` könyvtárat (vagy a `/tftpboot/ip`-t minden egyes ügyfélgép számára). Ahhoz, hogy az ügyfélgépen elérjünk a bejelentkezési eljárásig (login prompt), a saját állománynak tartalmaznia kell a következőket: az `init` parancsot, a parancshéj végrehajtható állományait, az eszközállományokat (legalább a konzolcsköt), és azokat a dinamikus függvénykönyvtárakat, amelyek az `init` és a héj parancsainak működéséhez szükségesek. Gyors és nem túl elegáns módja a távoli saját állományrendszer létrehozásának az `init`, `sh`, a szükséges függvénykönyvtárak és a konzolcsköt átmásolása:

```
cp /sbin/init /tftpboot/192.168.64.1/sbin/init
```

```
cp /bin/sh /tftpboot/192.168.64.1/bin/sh
```

Az `ldd` parancs megmondja nekünk, hogy az `init`-nek mely dinamikus könyvtárakra van szüksége:

```
ldd /sbin/init
```

```
ldd /bin/sh
```

Ezeket a könyvtárakat másoljuk be a `/tftpboot/ip/lib` könyvtárba. Az eszközök elkészítéséhez használhatjuk az ügyes MAKEDEV programot, amely a MAKEDEV-csomag része:

```
/dev/MAKEDEV -d /tftpboot/129.168.64.1/dev  
↳ console
```

Ha a kiszolgálón a többi szolgáltatás megfelelően működik, akkor az ügyfelet távolról indítva az lefuttatja a távoli saját állományrendszer `init` parancsát, ehhez a szintén távoli konzol használja; elindítja a héjat és várja, hogy megadjunk egy futási szintet (erre azért van szükség, mivel a `/etc/inittab` állomány a távoli saját állományrendszerben nem létezik). Az `s` billentyűt leütve egyfelhasználós üzemmódba jutunk, és a héj készenléti jel (prompt) már a rendelkezésünkre is áll. Egy különleges héjat, a `sash`-t (`standalone shell`) is használhatjuk, amely nagyon hasznos távoli környezetben. Ez annak köszönhető, hogy nem igényel dinamikus befűzött

függvénykönyvtárakat, valamint beépített parancsokat tartalmaz állományrendszerek kezelésére (`mount`, `umount`, `sync`), állományok hozzáféréseinek megváltoztatására (`chmod`, `chgrp`, `chown`), archiválásra (`ar`, `tar`) és egyéb műveletekhez. Az `sh`-héj használata helyett a *sbin/sash* állományt átmosolhatjuk a */tftpboot/ip/sbin/sash* állományba, a rendszermag így a `sash`-héjat fogja elindítani. Egy lecsupaszított *innitab* állományt is felhasználhatunk a `sash` indítására, ahogy ezt az alábbi példa mutatja:

```
id:1:initdefault:
1:1:respawn:/sbin/sash
```

Összegzés

A cikkben szó esett néhány olyan szolgáltatásról és módszerről, melyek segítségével távolról is elindíthatunk egy linuxos gépet. A távoli Linux-felügyelet rendkívül termékeny talaj a további kutatások számára. Ahogy a hálózatok egyre gyorsabbak, egyre több távoli ügyfelet támogatnak, és ahogy a számítógépfürtök mérete egyre nagyobb lesz, és egyre inkább központi felügyeletet igényelnek, úgy fognak a távoli Linux karbantartási módszerek egyre nagyobb szerephez jutni az iparban. A tömörített kiszolgálóeljárások megjelenésével a távoli Linux-felügyelet nemcsak kényelmi szempontokat szolgál, hanem egyenesen nélkülözhetetlenné válik.

Köszönetnyilvánítás

Hálás vagyok a *Vasilios Hoffman (Wesleyan)* kutatásaiért. „N”, ahogy ő szereti hívni magát, bemutatta, hogy a hurokeszközök

hogyan használhatók ramlemezek létrehozására, és hogy távoli rendszerindításokhoz miként hozhatunk létre jól működő moduláris rendszermagot. „N” a Linuxszal kapcsolatos tudnivalók két lábán járó lexikona.



Richard Ferri

az IBM Linux Technológiai Központjának vezető programozója. Olyan Linux-alapú számítógépes fürtök megvalósításán dolgozik, mint a LUI (☞ <http://www.oss.software.ibm.com/lui>) és az OSCAR. New York északi részén él feleségével,

Pattel, három tizenéves fiával, és kétes fajtájú kutyáival.

Kapcsolódó címek

Etherboot-honlap ☞ etherboot.sourceforge.net
Az Etherboot-eljárás a hálózati csatoló beállítására hajlékonylemezt használ, így lehetővé válik a hálózaton keresztüli rendszerindítás.
Távoli Linux-indítás mini-HOWTO
☞ cui.unige.ch/info/pc/remote-boot/howto.html
SYSLINUX-honlap
☞ syslinux.zytor.com – *H. Peter Anvin* kiváló munkája.
A `tftp-hpa` oldala
☞ <http://www.kernel.org/pub/software/network/tftp> – egy újabb kiváló program, szintén *H. Peter Anvintól*.

For geeks only!
Angol nyelvű számítástechnikai és szakkönyvek és magazinok
Kiskapu Kft. 1081 Budapest, Népszínház u. 29.
Telefon: 303-9119, Fax: 303-1619
Nyitva: H-P: 8-18, Kedd: 8-20
www.kiskapu.hu