© Kiskapu Kft. Minden jog fenntartva

A LinuxBIOS

Eric megmutatja nekünk, hogy a LinuxBIOS elfogadása és teljesítménye mennyire felkeltette a beágyazott és a fürtözött rendszerek fejlesztőinek figyelmét.

Több mint egy éve dolgozom a Linux NetworX-nél LinuxBIOS-fejlesztőként, és ez idő alatt rengeteg mindent meg kellett tanulnom. Most már bizton kijelenthetem, hogy a Linux rendszermagja igazi profi munka, a C nyelv pedig igazán magas szintű nyelv.

Mi is a LinuxBIOS?

Mikor a mikroprocesszor elindul, szépen sorban elkezdi végrehajtani a ROM-ban található utasításokat. Ezek az utasítások felelősek az alkatrészek felkészítéséért, elsősorban a RAM engedélyezéséért, majd pedig az operációs rendszer betöltéséért. Ennek a rendszernek a felépítése és kezelőfelülete gépről-gépre változó, de alapjában véve mégis mindenhol ugyanaz.

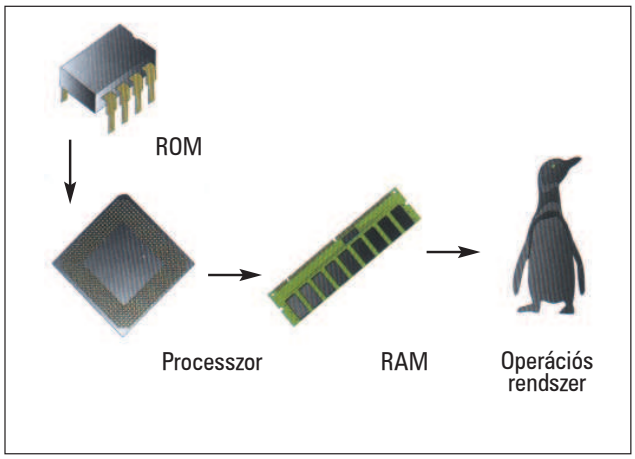
Az Alpha-alapú rendszer esetében a mikroprocesszor beolvassa az egész soros ROM-ot – azaz az SROM-ot – a gyorstárba, majd elkezdi végrehajtani az utasításokat. Az itt található kód feladata, hogy előkészítse a mikroprocesszort, a RAM-ot, és hogy egy flash EEPROM-ból betöltse az SRM-et. Ezt követően a mikroprocesszor végrehajtja az úgynevezett ***palcode***-ot (alapvetően a valódi Alpha-rendszermagot), előkészít még néhány alkatrészt, végül pedig elindítja az operációs rendszert. Mivel a gépi szinten futó program (firmware) két részre van osztva, az SRM frissíthető vagy akár teljesen lecserélhető. Az eredeti Alpha-elképzelés szerint a gépi kódnak operációs rendszerenként különböznie kell (az SRM-szintjén).

Az x86-os processzor 16-bites módban kezdi meg működését, és kezdetben a 16-bites CS regiszter a 64 K-s címterület végére mutat. A 8086-oson ez a cím 0xF000:0xFFF0 == 0xFFFF0, pont 1 MB alatt. 80386 vagy annál újabb mikroprocesszor esetében a cím 0xFFFFFFF0, pont 4 GB alatt. A 286-osnál és későbbi Intel-modelleknél a CS tartalma nem tölthető tetszőlegesen bármikor újra, emiatt a rendszer a ROM-ot a 0xFFFF0000 és 0xF0000 címeken is elérhetővé teszi.

Az Alphával ellentétben az x86-os rendszerek a ROM-ban található utasításokhoz egyenként férnek hozzá. Mivel a ROM valójában ISA-eszköz, kezdetben, amikor még nincs semmilyen gyorstár engedélyezve, a benne található kód végrehajtása nagyon lassú, illetve az is ennek a következménye, hogy a lapkakészletnek szinte teljesen fel kell állnia ahhoz, hogy a ROM-hoz hozzáférjünk, mivel az ISA eléréséhez elég hosszú utat kell végigjárnunk a processzortól az északi hídig, a PCI-síntől a déli hídig, és csak ezután érünk az ISA-hoz. Ha ezzel tisztában vagyunk és van egy jó alaplapunk is, nagymértékben megkönnyíti a helyzetünket, ha a rendszer indulásánál működésbe lépő eszközök valamelyikén végzünk hibakeresést.

Az alap PC BIOS feladata közé tartozik: az alkatrészek felkészítése az indulásra, az operációs rendszer betöltése, és az operációs rendszer futtatása közben bizonyos szolgáltatások biztosítása (többnyire a legszükségesebb eszközmeghajtók formájában).

A gépi szinten futó kódot a SPARC és a PowerPC esetében is meghatározták, melyet ***OpenBoot***, ***Open Firmware*** vagy az egykori ***IEEE1275***-ként is ismerünk. A szabványosított ***Forth firmware*** majdnem ugyanott található, ahol az Alpha esetében az SRM. Az ***Open Firmware*** esetében jó néhány egyéni vonással találkozhatunk: több processzoron és kiépítésen működőképes; ***Forth***-alapú bájtkódot használ, így a futtatható programok nem processzorfüggőek; ezenkívül a rendszer indításakor végrehajtódó utasítások legjavát a Forth-alapú bájtkódot átalakítva kapja.



1. ábra A rendszerindítás lépései

Az Itanium/IA64 úgynevezett EFI gépi szinten futó kódot használ, mely inkább felületfüggő, mint az ***Open Firmware***, mivel meghajtói vagy IA32- vagy IA64-kódra épülnek. Felépítését nézve is jóval bonyolultabb annál, az EFI ugyanis IP-vermet és fájlrendszermeghajtókat is tartalmaz. Hasonlóan az ***Open Firmware***-hez a kezdeti alkatrész-készenléti állapot itt sincs meghatározva.

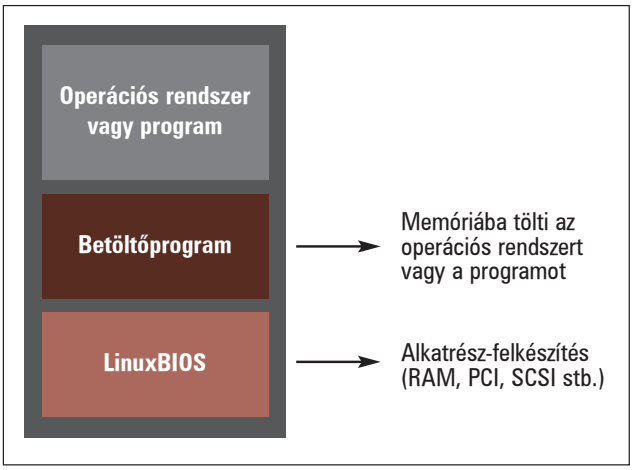
A Linux-rendszermagnak a gépi kóddal szemben támasztott követelményei csekélyek, mivel az eszközöket közvetlenül hajtja meg, a BIOS közreműködése nélkül. Mivel a rendszermag nem használja a BIOS-t, az alkatrészek előkészítése a BIOS részéről egyszerűen felesleges. Ez a megközelítés nem csak a Linuxra jellemző, ugyanis nem ismerek olyan korszerű operációs rendszert, amely ne ezt az irányvonalat követné.

A jelenlegi operációs rendszereknek mindössze egy alapvető rendszerelőkészítő szolgáltatásra van szükségük. A különleges eszközmeghajtók és rendszerjellemzők, amelyeket az ***EFI***, az ***Open Firmware*** vagy akár a PCBIOS biztosít, egyáltalán nem szükségesek, kivéve azokat a részeket, amelyek az operációs rendszer betöltéséhez szükségesek. Mivel ezekre a kódokra nincs szükség, a LinuxBIOS-ban nem is találhatók meg.

A LinuxBIOS kódja elégséges ahhoz, hogy egy Elfben kódolt programot betöltsön a flash ROM-ból. Ilyen program lehet egy operációs rendszer rendszermagja mint a Linux-rendszermag, de a legtöbb esetben ez mégis valamilyen alkatrészvizsgáló program vagy rendszerbetöltő eljárás (például Memtest86, Etherboot vagy a RedBoot).

© Kiskapu Kft. Minden jog fenntartva

Ha a LinuxBIOS mellé rendszerbetöltőt rendelünk, alkalmas lesz az operációs rendszer betöltésére. A LinuxBIOS eredeti ötlete az volt, hogy a Linux-rendszermagot a ROM-ból töltsük be, és a rendszerbetöltő eljárást arra építsük. Az ***nbc*** nevű rendszerbetöltő program ezt az ötletet valósítja meg: a hálózaton keresztül betölti a Linux-rendszermagot, vagy valamilyen önműködő programot, majd a rendszert a ***kexec*** rendszermagfolt (kernelpatch) segítségével elindítja. Ez a megoldás nagyszerűen működik, ha 512 KB vagy annál nagyobb ROM áll a rendelkezésünkre, viszont a manapság kapható alaplapok többsége sajnos csupán 256 KB ROM-mal rendelkezik. Az x86-os felület esetén szinte teljesen lehetetlen a 360 KB-nál kisebb rendszermag összeállítása.



2. ábra A Linux BIOS betöltése

Ennek az akadálynak a megkerülésére – mely a nem elégséges ROM-méretből fakad – különböző elképzeléseket dolgoztak ki. Közülük az egyik a Tiara, mely teljes gépi kód- és rendszerbetöltő a SiS630-as lapkakészlethez; ilyen még az Etherboot, amelyet átdolgoztak, hogy együttműködjön a LinuxBIOS-szal, valamint a RedBoot, ez azonban még nem teljesen működőképes. Ezenkívül még különböző foltokat készítettek a LinuxBIOS-hoz is, amelyek segítségével ez a nehézség megkerülhető. Az Alpha gépi kódjának olyan programra van szüksége, amely ismeri az alaplapot, amelyen fut, és ez gondokat okozhat. Emiatt az újabb alaplapok támogatása a felmerülő frissítések mennyisége miatt esetenként nagyon nehéz lehet. A LinuxBIOS fejlesztése során megteszünk minden tőlünk telhetőt, hogy ezt a hibát elkerüljük.

A hagyományos x86-os megközelítéssel kezdünk: felkészítjük a Super-IO lapkákat a munkára, beállítjuk a kezdeti értékeket (például a soros kapukat az alapértelmezett címeikre, az IRQ-kat stb.), majd elérhetővé tesszük az IRQ útválasztási táblázatokat, illetve az SMP számára az úgy nevezett ***mp***-táblákat. Hosszú távon egy táblázatban tároljuk az azonnal használható, azonosított lehetőségeket. Ez a programlista tárolja, milyen alkatrészek állnak rendelkezésünkre, illetve beállítja, hogy ezek közül melyeket fogja a számítógép használni, vagy legalábbis megadja, hogy az egyes eszközök milyen erőforrásokat használnak.

A megoldás, melyen dolgozom, egy táblázat karbantartását teszi szükségessé, mely az eszközök egymáshoz és az alaplaphoz való kapcsolódásáról tárol adatokat. Ez a táblázat tartalmazza az összes olyan eszközt, melyet egyetlen azonnal használható (Plug-and-Play) felsorolás sem foglal magában, és elegendő adatot tartalmaz ahhoz, hogy az IRQ-útválasztás megfelelően kezelhetővé váljon. Ráadásul az ötlet úgy tűnik, megfelelően illeszkedik a 2.5-ös rendszermaghoz tervezett eszközfa-struktúrához. Az ACPI látszólag választható megoldást kínál, de meglehetősen PC-centrikusnak tűnik, emellett a feldolgozott byte-kód szükségtelen, sőt veszélyes is lehet.

Kivitelezhető a LinuxBIOS?

A rendszerindító ROM-hardver felépítése az első IBM PC óta rengeteget fejlődött, így manapság szinte minden számítógép helyben frissíthető BIOS-szal rendelkezik, mely hibás frissítés esetén az eredeti állapotába állítható vissza. Ennek következtében általános eljárásnak mondható, hogy az alaplapon a flash ROM-ot egy foglalatban helyezik el. A flashlapka a program számára lehetővé teszi, hogy frissítse a tartalmát, mindemellett a foglalat a lapka cseréjét is biztosítja, ha a frissítés valamilyen okból nem sikerül. Ilyen típusú kiépítéssel már lehetséges egyéni rendszerindító gépszintű programok előállítása. A frissítéshez nincs szükség speciális eszközre, és ha a fejlesztés során elromlik valami, visszaállítható az eredeti állapot.

A jelenlegi PC-kiépítések hátránya, hogy a normál 256 KB-s rendszerindító ROM-ok túlságosan kicsik. Ez a hely elég egy gépi kód számára, de nem elegendően nagy a Linux-rendszermagnak.

A Linux-rendszermag ugyanúgy képes LinuxBIOS-ból futni, mint az általános PCBIOS-ból, ha az átültetést megfelelően elvégezzük el. A LinuxBIOS-t a mai napig három alaplapra is sikeresen átültettem. A legutóbbi alaplapon már nem lehet különbséget tenni, hogy a rendszert a PCBIOS-ból vagy a LinuxBIOS-ból indítottuk-e el. Tehát a meglévő technikai akadályok ellenére úgy tűnik, hogy a LinuxBIOS-t sikerül, illetve már sikerült is átültetnünk újabb rendszerekre.

A fentieken túl az is kulcskérdés, hogy a fejlesztőnek hozzáférése legyen a megfelelő szintű dokumentációhoz. A számítógépgyártók eddig csak nehezen tűntek rábeszélhetőnek arra, hogy támogassák a LinuxBIOS-t, vagy akár arra, hogy megfelelő minőségű dokumentációt biztosítsanak valaki számára, aki végül lekódolná azt. A korlátozott támogatás vagy akár annak teljes hiánya nem minősül új dolognak a szabad programot használók számára, ezt a nehézséget eddig is megoldottuk valahogy, és most sem adhatjuk fel miatta a reményt. Nem szabad elfelejteni, hogy efféle erőfeszítések nélkül nem születnének olyan új gépek, melyeken szabad programot futtathatunk.

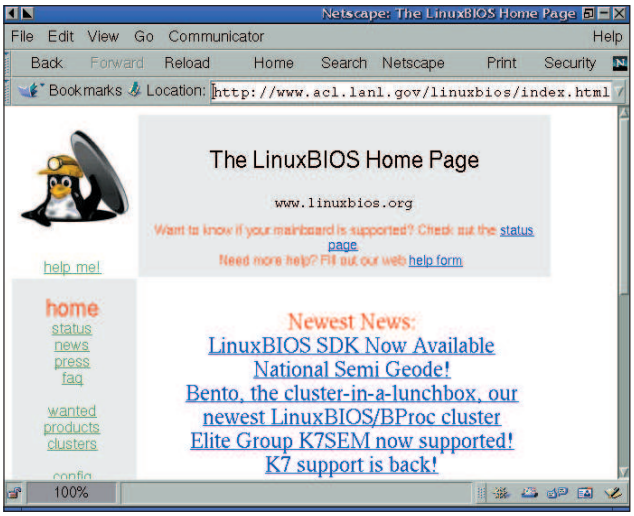
Milyen alkalmazások léteznek a LinuxBIOS-hoz?

Jelenleg két érdekeltségi kör dolgozik a LinuxBIOS-on: az egyik beágyazott rendszereket készít, míg a másik nagyteljesítményű számítógépfürtöket. Ezen alkalmazások számára az ősi x86 gépszintű program elégséges.

A LinuxBIOS nagy karrier előtt áll a beágyazott alkalmazások terén. Mivel a GPL licenc alatt fejlesztik, a LinuxBIOS teljességgel jogdíjmentes. A LinuxBIOS-nak mindössze 64KB elegendő, nem pazarolja a ROM-ot szükségtelen feladatokra. Mivel a LinuxBIOS modern felépítésű, rendkívül gyorsan indul, még kódoptimalizálás nélkül is.

2001 augusztusában a General Software-nek egy beágyazott rendszeren újraindítás után 0,8 másodperc alatt sikerült eljutnia a LILO-kiírásig. Ez az adott feladathoz képest elfogadhatónak mondható, de a LinuxBIOS esetén egy ilyen jó eredmény teljesen rutinszerű. Ha hidegindítás után a rendszermagot hálózatról töltjük be, a rendszerindulás egy kétprocesszoros kiszolgálófarm esetén mindössze két másodpercet vesz igénybe – a LinuxBIOS hatékonnyá tétele nélkül.

A LinuxBIOS eredményei a SiS-t olyannyira meggyőzték, hogy egy fejlesztőjüket megbízták a LinuxBIOS saját alaplapjaikra való átültetésével, megcélozva a beágyazott rendszereket és jól támogatott felületet hozva létre.
Számítógépfürtök esetén – amelyekkel a Linux NetworX foglalkozik – a LinuxBIOS úgyszintén nagy jövő előtt áll. A soros kapu az alapértelmezett rendszerkonzol, így nincs szükség videóalkatrészre. A soros kapcsolatok egy központi terminálkiszolgálón könnyedén átirányíthatók, hogy távolról hozzáférjünk a gépek rendszerkonzoljaihoz. A soros konzol kezdeti szakasza is előnyökkel jár, például a LinuxBIOS ezen keresztül



képes a különféle alkatrész- vagy egyéb hibák jelentésére. Egy egyszerű BIOS, még ha rendelkezik is soroskonzol-kiegészítéssel, a soros kaput túl későn nyitja meg ahhoz, hogy néhány hibát időben észlelhessen, és rendszerint egy üres CMOS végzetes hatással van a rendszerindításra nézve.
A legtöbb rendszeren a LinuxBIOS lehetővé teszi a hálózaton keresztüli rendszerindítást, ezáltal a DHCP-kiszolgálón keresztül az indítási beállításokon akár egy egyszerű módosítással változtathatunk. Mivel a kód szabad forrású, ha a hálózati tulajdonságok nem megfelelőek, könnyedén módosíthatunk rajtuk. A gyors újraindítás a LinuxBIOS esetén azt jelenti, hogyha valamilyen hibát keresel, és ehhez újra kell indítanod az egyik csomópontot, a gépújraindítási folyamat nem a rendszergazda idejét rabolja.
A LinuxBIOS nyíltsága és Linux-központúsága azt eredményezi, hogy beállításai Linux alól módosíthatók. Így minden, ami a felhasználó parancssorából módosítható, a hálózaton keresztül is megváltoztatható. Azonos gépekből összeállított telepek esetén ezzel rengeteg idő nyerhető, mivel a gépszintű programokon a változtatások központilag elvégezhetők, és nem kell őket külön-külön minden csomópontnál végrehajtani. Nagyszámú gép esetén a számítógéphibák jóval valószínűbbek, mint egyetlen gép esetén. A LinuxBIOS kis gépigényeivel – nem szükséges hajlékonylemez vagy CD-ROM-meghajtó, és merevlemezre sincs feltétlenül szükség – egy kevésbé drága, ugyanakkor jóval megbízhatóbb rendszert kapunk. Minél kevesebb gépet használunk, annál kisebb a kockázata egy esetleges számítógép-meghibásodásnak.
Fürtök esetén a csatlakoztatott számítógépek a LinuxBIOS segítségével beállíthatók, hogy viselkedjenek úgy, mintha egyetlen számítógép lennének, ahelyett, hogy úgy nézzenek ki, mint összekötött számítógép-csomópontok halmaza.

Milyen alkatrészeket támogat a LinuxBIOS?

A LinuxBIOS jelenlegi állapotában 13 különböző alaplapot támogat. A LinuxBIOS ugyanúgy fut az egyes x86-kiépítéseken, mint Alphán. Az AMD Athlon, az AMD Duron, a Pentium II és Pentium III, az Alpha 211264 CPU-k, az ALI m1631, a Digital Tsunami, az AMD 760, az AMD 760MP, az Intel 440BX, az Intel 440GX, a VIA VT8601, a SiS540, a SiS550, a SiS630 és a SiS730 már mind-mind képesek futtatni a LinuxBIOS-t. Most csak azokat soroltuk fel, melyeket a mostani LinuxBIOS már tartalmazza és működőképesek, de nem említettem a még fejlesztés alatt álló, illetve a LinuxBIOS-ban még nem szereplő átírásokat. Tehát annak ellenére, hogy az alkatrész-támogatás korlátozott, a lista mégis folyamatosan nő. A LinuxBIOS jelen pillanatban nincs egyetlenegy lapkakészlethez, gyártóhoz, vagy processzorkiépítéshez sem kötve.
A számítógép-támogatás minősége változó. A lapkakészlet oldaláról nézve, a SiS-lapkakészletek támogatása a legfejlettebb. Az Intelnek és az AMD-nek egyaránt megvan a maga alapvető politikája lapkakészleteik leírásával kapcsolatban, így mindkét gyártó támogatottsága elég jó. A Via leírásai nyilvánosan sajnos nem hozzáférhetők, így az ő termékeik támogatása meglehetősen nehéz.
A processzoroldalon a Compaq a lényeges részleteket nyilvánossá tette, így az Alphák támogatása kivitelezhető. Ezzel szemben mivel az Alpha-processzorok elég drágák, ráadásul jelen pillanatban a jövőjük is elég kétséges, a támogatásuk nem elsődleges szempont.
A Pentium II és Pentium III processzorok nagyon jó leírással rendelkeznek, kivéve ami a másodszintű (L2) gyorstárukat illeti, de a LinuxBIOS már azt is támogatja. Az AMD Athlon és Duron nem olyan jól támogatott, mivel az AMD nem tett nyilvánossá minden olyan részletet, amelyre a fejlesztésekhez szükség volna.
Az alaplapgyártó oldaláról a támogatás nem feltétlenül szükséges, mivel a legtöbb esetben az alaplapon található eszközökről az első pillanatban megállapítható, hogy mivel van dolgunk.
Az alaplapgyártók alaplapjaikhoz csak egyetlen gépszintű program támogatásában érdekeltek. Mivel a LinuxBIOS egyéb operációs rendszerekhez nem tartalmaz támogatást, különösen a Windowshoz nem, jelenlegi formájában a LinuxBIOS alkalmazásában nem különösebben érdekeltek.

Összegzés

A LinuxBIOS segítségével betekintést nyerünk abba, hogyan működik a gépszintű program, hogyan épül fel, hogyan írták és milyen a felhasználói szerződése. Ahogyan a számítógépek egyre egybevontak, a LinuxBIOS fokozatosan rugalmasabbá válik, és nagyobb lehetőséget biztosít a kód-újrafelhasználásra. Ennek köszönhetően remélhetőleg egyre elterjedtebbé válik majd, és ezáltal még nagyszerűbb LinuxBIOS-t kapunk.

Eric Biederman
(ebiederman@linuxnetworx.com a Linux NetworX-nél dolgozik programfejlesztőként. Elsődleges feladata a LinuxBIOS, valamint segédkezni a számítógéptelepeket karbantartó programok tervezésénél. Ha Eric éppen nem a LinuxBIOS-szal van elfoglalva, szívesen olvas sci-fi-t, vagy játszik a DOSEMU-val, emellett szabadidejét szívesen tölti a Salt Lake City mellett található Wasatch-hegyekben.

© Kiskapu Kft. Minden jog fenntartva