© Kiskapu Kft. Minden jog fenntartva

A 2.6-os Linux rendszermag valós idejű és teljesítmény fejlesztései

A Linux rendszermag érzékenységének és valós idejű teljesítményének fejlesztése a jövőben még kedvezőbb fordulatot vehet.

A ***Linux*** rendszermag, minden ***Linux*** terjesztés belső magja, folyamatosan fejlődik. Új módszereket olvaszt magába, növeli a teljesítményét, a skálázhatóságát és használhatóságát. Bár minden rendszermag kiadás újabb alkatrészeket támogat, a nagyobb rendszermag verzióváltások általában komolyabb változásokat jeleznek a rendszermag felépítésében. Ilyen a 2.6-os ***Linux*** rendszermag is. A 2.6-os ***Linux*** rendszermag számos változása jelentősen befolyásolja a ***Linux*** rendszermag teljesítményét az alkalmazott alkatrészektől függetlenül, valamennyi ***Linux*** rendszeren. A 2.6-os rendszermag komoly változásokat hoz a rendszer érzékenységében, jelentősen csökkenti a folyamat és szál-vonatkozású rendszermag többletmunkát valamint a feladat ütemezése és végrehajtása között eltelt időt. Ma már szinte az ipar valamennyi nagyobb ***Linux*** terjesztése a 2003 végén kiadott 2.6 rendszermagon alapul az asztali gépektől a beágyazott rendszerekig. A rendszermag és rendszerteljesítmény rendkívül fontos az olyan koncentrált piacokon mint a beágyazott gépek területe, ahol a magas prioritású folyamatoknak gyakran valós időben kell végrehajtódniuk anélkül, hogy a rendszer félbeszakítaná őket. Ugyanakkor a rendszer teljesítménye és átviteli képessége legalább ilyen fontos a linuxos asztali gépek, és az egyre sikeresebb ***Linux*** vállalati kiszolgálópiacon.

Cikkünkben a valós idejű és rendszerteljesítményt befolyásoló paraméterek természetével foglalkozunk, kiemelve a 2.6 rendszermagba teljesítmény és érzékenység terén bevezetett legfontosabb újításokat. A teljesítmény és az érzékenység továbbra is fontos fejlesztési terület maradt, cikkünkben a Linux teljesítményének és érzékenységének növelését valamint a valós idejű működést célzó néhány aktuális megoldással foglalkozunk. A különféle ***Linux*** rendszermagok és projektek belső, illetve folyamat végrehajtó teljesítményét grafikus teljesítményteszteken jelenítjük meg ahonnan leolvasható, hogyan viselkednek az egyes rendszermagváltozatok különféle terhelési körülmények között.

Lappangás, preemptiv képességek és teljesítmény

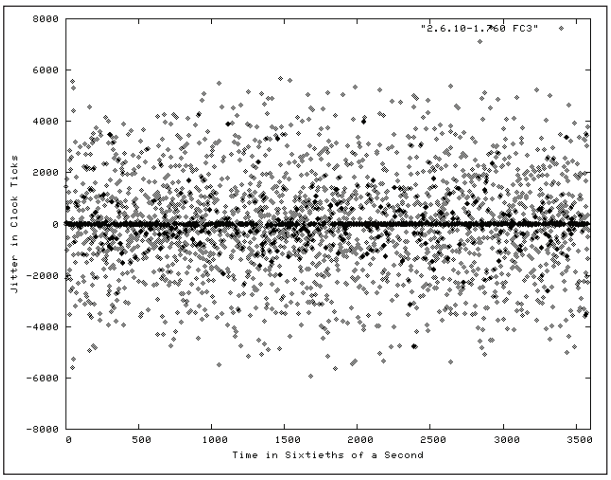
Magasabb teljesítményt gyakorta további és erősebb alkatrészek felhasználásával lehet elérni, azaz komolyabb processzort, nagyobb memóriamennyiséget alkalmazunk. Bár ez egy adatközpontban megfelelő megoldásnak bizonyulhat, jónéhány területen ez nem jelent kielégítő megoldást. A beágyazott ***Linux*** rendszerek például kifejezetten érzékenyek a felhasznált alkatrészek árára. Ezen kívül ha teljesítmény és végrehajtási gondokat memóriavásárlással és más alkatrészekkel próbáljuk megoldani mindössze ideiglenesen elfedjük a problémát. Ahogy a programok igénye növekszik, idővel átlépi a bővített erőforrások határát is, így a probléma ismét a felszínre kerül.

Ezért igen fontos hogy a ***Linux*** rendszerekben alkatrészkerülő módon, a belső rendszer mag javításával érjünk el teljesítménynövekedést. Cikkünk az ilyen típusú ***Linux*** teljesítménymérésekre koncentrál.

A valós idejű rendszerek azon a rendszerek közé tartoznak, ahol a rendszer helyes működése nem csak a kívánt művelet végrehajtásának sebességétől függ, hanem egy bizonyos időzítési kritériumtól is. Kétféle valós idejű rendszer létezik: a lágy és a kemény. A kemény valós idejű rendszerek esetében a kritikus folyamatoknak egy adott időszeletben kell lefutniuk vagy a teljes rendszer hibásnak tekinthető. Klasszikus példa ilyen rendszerre a számítógép vezérlésű autógyújtásvezérlő rendszer – ha a hengerek nem pontosan a jó időben kapnak gyújtást, az autó nem fog működni. A lágy valósidejű rendszerek azok, ahol az időzítési határidőket át lehet lépni a teljes rendszer összeomlása nélkül; a rendszer helyre tud állni az ideiglenes érzékenység csökkenés után.

Mindkét fajta valós idejű rendszer először a magas prioritású feladatokat hajtja végre egy ismert előrejelezhető időszeleten belül. Ez annyit jelent, hogy az operációs rendszer nem végezhet aránytalan többletmunkát a feladat ütemezés, végrehajtás és kezelés során. Ha a folyamatokkal kapcsolatos többletmunka jelentős mértékben növekszik a feladatok számának növekedésével, akkor a teljes rendszerteljesítmény leromlik ahogy egyre több idő szükséges az ütemezés, váltás és újraütemezés elvégzéséhez. A kiszámíthatóság a valós idejű operációs rendszerek egyik legfontosabb jellemzője. Ha nem tudjuk kiszámítani a rendszer teljesítményigényét egy tetszőleges időpillanatban, nem tudjuk biztosítani, hogy az adott folyamat szükség esetén előre látható lappangással indul el vagy indul újra, illetve hogy az előírt időszeletben befejeződik-e.

A 2.6-os ***Linux*** rendszermagban új típusú ütemező jelent meg, amelynek végrehajtási ideje nem függ az ütemezett



1. ábra A dobozos Fedora Core rendszermag remegési értékei

feladatok számától. Ez a megoldás ***O(1)*** ütemezőként ismert a nagy-***O*** algorimus jelölési rendszerben, ahol az ***O*** az angol ***Order*** szó rövidítése, a zárójeles szám pedig a legrosszabb teljesítmény felső határa az algoritmusban résztvevő elemek függvényében. Az ***O(N)*** tehát annyit tesz, hogy az algoritmus hatékonysága az elemek számától függ, az ***O(1)*** jelentése pedig, hogy az algoritmus, azaz jelen esetben az ütemező futásigénye minden esetben azonos, azaz független a résztvevő elemek számától.

Az operációs rendszernek kiadott feladatvégrehajtási utasítás és a feladat tényleges elindulása között eltelt időt lappangásnak nevezzük. A folyamatvégrehajtás nyilvánvaló módon függ az adott feladat fontossági szintjétől, de azonos fontosságot feltételezve, azt az időt amelyet az operációs rendszer a feladat ütemezésével és futtatás megindításával tölt, a rendszerütemező többletmunkája és a rendszer által végzett egyéb tevékenységek határozzák meg. Amikor feladatot ütemezünk a rendszer futtatási sorába helyezve, a rendszer megvizsgálja, hogy a folyamat magasabb prioritású-e mint az éppen futó folyamat prioritási szintje.

Ha igen, a rendszermag megszakítja az éppen futó folyamatot és átvált az új folyamat környezetére. A rendszermagban az éppen futó folyamat megszakítását és a környezetváltást nevezzük ***preempciónak***.

Sajnos, a rendszermag nem mindig használhatja a preemptivitást. Az operációs rendszer magja gyakran igényel kizárólagos erőforrás és belső adatszerkezet elérést, hogy fenntarthassa saját belső szerkezetének sérülésmentességét. A ***Linux*** rendszermag régebbi verzióiban az erőforrásokhoz való kizárólagos hozzáférést gyakran forgózárakkal biztosították. Ez ennyit jelentett, hogy a rendszermag egy rövid ciklusban lépkedett, amíg az adott erőforrás elérhetővé nem vált, vagy amíg használatban volt, ezzel persze megnövelve minden más feladat lappangását amíg a rendszermag be nem fejezte a feladatát.

A rendszermag preemptivitásának felbontása folyamatosan fejlődött az elmúlt nagyobb rendszermag verziókban. Például a beágyazott ***Linux*** és eszköz gyártótó ***TimeSys***-től származó ***GPL 2.4 Linux*** rendszermag, egy korábbi alacsony lappangású ütemezőt és teljesen preemptív rendszermagot tartalmazott. A 2.4-es ***Linux*** rendszermag sorozatban, ***Novell/Ximian***-tól ***Robert Love*** kiadott egy jól ismert rendszermag foltot amely magasabb preemptivitást biztosított és amelyet a hagyományos rendszermag forrásra is alkalmazni lehetett. Más foltok, mint például az 1995 óta belső rendszermagíró ***Molnár Ingo*** alacsony lappangási foltja, tovább fokozták e folt teljesítményét és csökkentették a rendszermag lappangási idejét.

A ***TimeSys*** termékének és az említett foltok kulcselképzelése az volt, hogy a ***forgózárakat mutexekkel (mutual exclusion mechanism, azaz kölcsönös eléréskizárás)*** kell helyettesíteni ahol csak lehet. Ezek ugyanis anélkül biztosították a rendszermag erőforrás és integritás biztonságát, hogy a rendszermagot leállították és várakozásra kényszerítették volna. Ezen úttörő foltok által használt megoldások ma már a 2.6 ***Linux*** rendszermag szerves részét képezik.

Valós idejű megoldások Linux alatt

Három valós idejű linuxos megoldás létezik jelenleg: az ***RTAI Projekt*** által alkalmazott kettős rendszermag modell, amelyet beágyazott ***Linux*** gyártók, például az ***FSMLabs*** termékeiben alkalmaznak; egy beágyazott ***Linux*** gyártó, a ***MontaVista*** szárnyai alatt fejlődő ***Real-time Linux Projekt***; valamint a szabadon elérhető és használható preemptivitási és valós idejű munka, amelyet ***Molnár Ingó*** és más fejlesztők neve fémjelez, s amelyet nyíltan vitatnak meg a ***Linux*** rendszermag levelezőlistán, és amelyre a ***MontaVista Projekt*** is épít. Ezeken az alap rendszermag modelleken felül más kiegészítő projektek, például a robosztus mutexek és a nagy felbontású időzítők is jelentős mértékben hozzájárultak a valósidejű alkalmazások teljes körű támogatásának fejlődéséhez ***Linux*** alatt.

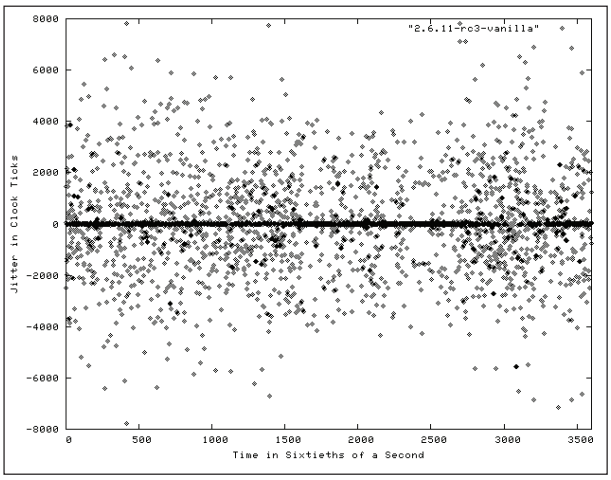
A valós idejű kettős rendszermag elgondolás igen érdekes megközelítése a Linux alatti valós idejű alkalmazáskezelés problémakörének. Ebben az elképzelésben a rendszer tulajdonképpen egy apró, nem ***Linux*** valósidejű rendszermagot futtat, amely viszont a ***Linuxot*** futtatja a lehető legalacsonyabb prioritási szinten. A valósidejű alkalmazások, amelyeket kifejezetten ehhez a nem-linuxos rendszermaghoz írtak a hozzá tartozó valósidejű csatolófelület felhasználásával, ebben a rendszermagban hajtódnak végre, de magasabb szinten mint a ***Linux*** vagy bármely ***Linux*** alkalmazás. Ugyanakkor képesek adatokat cserélni a ***Linux*** alkalmazásokkal. Igaz, műszakilag nagyon érdekes elgondolás, amellyel valósidejű alkalmazásokat futtathatunk ***Linux*** rendszer mellett, azonban teljességben kikerüli az általános ***Linux*** rendszermag preemptivitás és teljesítmény fokozási problémákat. Éppen ezért a ***Linux*** mag fejlődésének szempontjából nem is olyan érdekes.

A ***MontaVista Projektre***, azaz a másik valós idejű ***Linux*** megoldásra igen nagy hatással volt Molnár Ingo és más rendszermagfejlesztők által végzett már meglévő munka, azonban tartalmaz néhány egyéb prototípus foltot, amelyek kizárólag a ***MontaVista*** weblapján érhetőek el. A jelenlegi foltok itt a 2.6.9-es ***Linux*** rendszermaghoz valók (***rc4***). Ezért aztán nem igazán alkalmazhatók probléma mentesen a hivatalos ***Linux*** rendszermag kiadásokra, amelyek írásunk születésekor már a 2.6.1-es verzió felé járnak. Így aztán e projektből származó eredményeket nem tudtuk megmutatni ebben a cikkben.

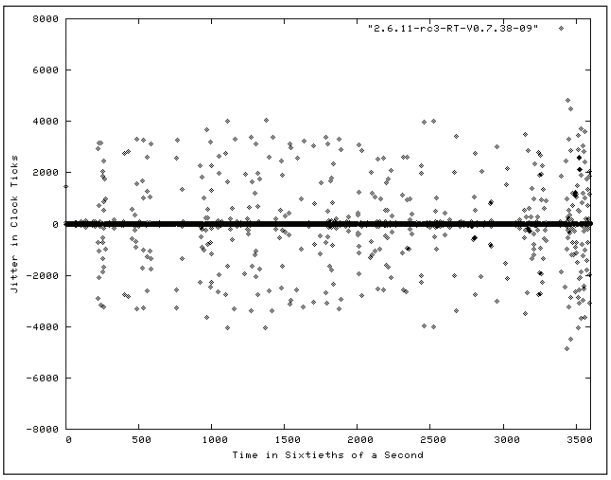
Molnár Ingó az ***O(1) Linux*** ütemező atyja és más fejlesztők munkája az ütemezőn és a preemptivitási képességeken igen komoly lendülettel haladnak. Miközben kibővítik

© Kiskapu Kft. Minden jog fenntartva

© Kiskapu Kft. Minden jog fenntartva



2. ábra Remegési eredmények egyszerű 2.6.11-rc3 Rendszermaggal



3. ábra Remegési eredmények valós-idejű/Preemptív folttal ellátott 2.6.11-rc3 Rendszermag esetében

a ***Linux*** rendszermag lehetőségeit, naprakész foltokat nyújtanak, amelyek a hatékonyabbá varázsolják a rendszerütemezést, csökkentik a lappangási időt és tovább növelik a preemptivitást.
Ezeket a foltokat nagy figyelemmel kísérik számtalan csoport és szervezet fejlesztői, beleértve a ***Raytheont***, olyan beágyazott ***Linux*** fejlesztőket mint a ***TimeSys*** valamint a ***Linux audió*** közösség. A foltok megnövelik a rendszerérzékenységet és minimálisra csökkentik a megszakítások hatását, két részre osztva a megszakításkezelőket: egy közvetlen alkatrész válaszra és egy ütemezhető megszakítás feldolgozó szakaszra. Mint a nevük is mutatja a megszakítások olyan kérelmek, amelyek azonnali figyelmet igényelnek a rendszertől. Az ütemezhető megszakítás kezelési megoldás, vagy ismertebb nevén soft IRQ, a lehető legkisebbre csökkenti az általános rendszer reakcióidő és teljesítmény.
A következő rész ábrái különféle sima rendszermagok teljesítményét érik össze Molnár Ingó és mások által készített ütemezési, preemptivitási foltokkal javított változatok képességeivel. Ezek a foltok naprakészek és teljes ***Linux*** rendszermag fejlesztéseket tartalmaznak melyek közvetlen előnyöket nyújtanak bármely ***Linux*** felhasználónak aki projektjébe szeretné építeni őket.

Teljesítménymérés

2002-ben, a ***Linux Journal*** weblapon megjelentet egy cikk ***Andrew Webber*** tollából ***„Realfeel Test of the Preemptible Kernel Patch”*** címmel. Ez a cikk ***Mark Hahn Realfeel*** nevű nyílt teljesítménymérő programját használta „hogy összehasonlítsa a preemption és érzékenység értékeket az eredeti ***Linux*** 2.4-es és ***Robert Love*** preemptivitási foltjával javított rendszermagot. A Realfeel periodikusan megszakításokat hoz létre, majd a számítógép mért válaszidejét összehasonlítja a rendszer optimális válasz idejével. A megszakításkérés kiadása és annak kezelése közt eltelt idő a lappangás, az előrejelzett és a tényleges lappangás között eltelt idő a remegés (jitter). Ezt a remegés értéket gyakran használják a rendszer válaszidejének mérésére és összehasonlítására. Cikkünk ugyanazt a teljesítménymérő programot használja mint amelyet ***Webber*** cikkében megismerhettünk, ám mérés közben lényegesen nagyobb terhelést tettünk a gépre. Valós idejű operációs rendszerek tesztelése során gyakran alkalmazzák ezt a módszert, hiszen még a nem valós idejű rendszerek is alacsony lappangást mutathatnak amikor terhelés nélkül vagy alacsony terhelés mellett működnek. A következő fejezet rajzai az eredményeket is kicsit másképp ábrázolják, hogy könnyebb legyen elképzelni és összehasonlítani a különféle ***Linux*** rendszerek lappangási idejének különbségeit.

Mérési eredmények

A fejezeteben található eredmények egy közepes teljesítményű ***Pentium*** osztályú rendszeren készültek, amelyet egyetlen 1.7GHz ***AMD Athlon*** processzor hajtott meg, 512MB rendszermemória mellett. A rendszer ***GNOME*** asztalkezelő környezetet és ***Fedora Core 3 Linux*** terjesztésben szokásos rendszerfolyamatokat futtatta, 2004 február 10.-ei legfrissebb foltokkal. A kipróbált rendszerek: eredeti 2.6.10 ***Linux*** rendszermag, a ***2.6.10-1.760_FC3*** rendszermag, amelyet a ***Fedora Core 3*** frissítéssel érhetünk el, eredeti ***2.6.11-rc3*** rendszermag és a ***2.6.11-rc3*** rendszermag ***Molnár Ingo*** legújabb valós idejű preemptivitási foltjával. Minden rendszermagot azonos rendszermag beállításállomány mellett fordítottuk, eltekintve persze az új rendszermagforrásban felbukkanó újabb beállításoktól.
Az olyan több folyamatos rendszerekben mint a ***Linux***, a rendszer soha nem alszik. Bizonyos rendszerfolyamatok, például az ütemező, állandóan futnak. Amennyiben grafikus felületet használunk (GUI) és kezelőfelületeket mint a ***KDE***, a ***GNOME*** vagy akár az egyszerű ***X Window*** rendszer, az ablakkezelők folyamatosan várják a beérkező eseményeket. Mivel az igazi preemptivitást és valós idejű teljesítményre voltunk kíváncsiak, pár alkalmazás elindításával megterheltük a rendszert, miközben az egyes teljesítménymérő tesztek eredményeit gyűjtöttük. Mint korábban említettük, a rendszer ***GNOME***-ot futtatott négy nyitott ***xterm*** mellett – ezek közül egyben futott a ***realfeel*** teljesítménymérő a másikban pedig egy olyan folyamat futott, amely állandóan ***ls*** és ***find*** parancsokat hívogatott a rendszer gyökérpartícióján, végül a maradék kettőben külön forráskönyvtárban található ***Linux 2.6.x*** rendszermagok fordultak, ***clean*** állapotból.

Az *1. ábra* a ***Realfeel*** teljesítménymérő grafikonján ábrázolja egy dobozos ***Fedora Core*** rendszeren futtatva egy percen keresztül. A rendszer ***2.6.10-1.760_FC3*** rendszermagot futtatott, amely lényegében egy 2.6.10-es rendszermag néhány ***Red Hat*** folttal és fejlesztéssel. A pontok a megszakítás kérés és kezelés közötti remegést (jitter) ábrázolják. Az X tengely a mintaidő 1/60-ad másodpercben. A negatív remegési értékek olyankor fordulnak elő amikor a rendszer gyorsabban válaszolt mint az előrejelzett válaszidő. Amint azt az ábrán láthatjuk, a megszakítási kérelmek legtöbbje a várakozásoknak megfelelő idő alatt készült fel, így eredményképpen egy jól kivehető sörtét vonalat kapunk az Y tengely 0 értéke körül.
A *2. ábra* ugyanezen rendszer ***Realfeel*** programmal mért eredményeit mutatja sima ***2.6.11rc3*** rendszermaggal, amely a következő 2.6.11 rendszermag 3. kiadási jelöltje. Azt eredményeket ismét csak egy perc alatt gyűjtöttük. Amint azt az eredményekből láthatjuk, a ***2.6.11-rc3*** rendszermag jobb teljesítményt nyújtott mint az ***FC3***-as rendszermag, sokkal több helyen tapasztalhatjuk, hogy a megszakítás kérelem és kezelés közötti remegés 0 értékű.
A *3. ábrán* ***Molnár Ingó*** valós-idejű/preemtivitás foltjaival bővített ***2.6.11rc3*** rendszermagot használó, de egyébként változatlan rendszer eredményeit látjuk ***Realfeel*** programmal mérve. Ezeket az eredményeket is egy perc alatt gyűjtöttük össze, az előzőekkel azonos terhelésgenerátor mellett. Amint azt az eredményekből láthatjuk, a valós-idejű/preemptivitás folt esetében lenyűgözően jó remegési eredményeket kaptunk, és viszonylag kevés helyen láthatunk eltéréseket a becsült időintervallum értékekhez képest. Rendszerünkön mindez nagyobb érzékenységet eredményez, a programfutás jóval kiszámíthatóbb mint azt egy sima ***FC3*** vagy eredeti ***2.6.11-rc3*** rendszermag esetében várnánk.

Összefoglalás

A 2.6-os ***Linux*** rendszermag javított ütemezése, ***SMP*** és skálázási fejlesztései nagyobb teljesítményű ***Linux*** rendszereket eredményeztek mint bármikor azelőtt. Jobban ki tudják használni a rendszererőforrásokat valamint kiszámíthatóbban futtatják a rendszermagot és a felhasználói folyamatokat a rendszer igényei szerint. Léteznek további javítások is, ám ezeket jelenleg csak a rendszermag kézi foltozásával használhatjuk, vagy olyan gyártótól kell ***Linux*** terjesztésünket beszerezni, mint a ***TimeSys***, amely már beépítette és teszteli ezeket a nagyteljesítményű foltokat.
Kész csoda, hogy egyáltalán létezik ilyen szabad, nyílt forrású rendszermag és robosztus végrehajtási környezet mint a ***GNU/Linux***. Most, hogy egyéni fejlesztők mellett már cégek is fejlesztik teljesítményét, még fényesebb jövőt jósolhatunk neki. Ezek, és a ***Linux*** egyéb módosításai felvetik és érvéként szolgálhatnak a ***Linux*** mint javasolt operációs rendszer mellett a beágyazott rendszerek, kiszolgálók és asztali gépek területén.

Linux Journal 2005. június, 134. szám

A cikk forrása: ➲ ***www.linuxjournal.com/article/8199***

William von Hagen

© Kiskapu Kft. Minden jog fenntartva