

*Ez elméletileg valóban nem sérti a GPL szerződést, végső soron mégis olyan helyzetet teremthet, hogy egy GPL jogú fejlesztést (például a rendszermagot) nem lehet egy másik GPL jogú projektben felhasználni.*

© Kiskapu Kft. Minden jog fenntartva

## Újdonságok a rendszermagfejlesztésben

Az Intel néhány új eseményfigyelő folttal (notification patches) állt elő, amelyek a már létező folyamatindítási leíró kapcsok mellett a folyamatok befejezésekor is nyújtanak leíró kapcsokat (profiling hook). Az Intel műve sokat segíthet az olyan programok továbbfejlesztésekor, mint az Oprofile, Perfmon és a VTune. Bár az ilyen kapcsok felhasználása könnyen hozzásegítheti a forgalmazókat, hogy a GPL szabadalmi megkötéseit megkerüljék, úgy tűnik, hogy az Intel alkotása mentes ettől a gondtól. *Frederic Rossi* továbbra is dolgozik saját asynchronousevent mechanism (AEM) nevű, független eseményjelző kódján. Az AEM segítségével általános célú, a felhasználói térben futó programok is bejegyezhetik magukat a rendszereseményekre és megadhatják a saját eseménykezelő készletüket. A folt fő célja, hogy a rendszermagból gyorsan lehessen adatot visszaadni a felhasználói térbe. Bár az Intel munkája és az AEM közvetlenül nem kapcsolódik egymáshoz, érdekes adalékkal szolgálnak a Linux-fejlesztés irányáról: egyre jobban növekszik az egyes szintek közötti párbeszéd.

A DSI Development csapat által fejlesztett **digsig** nevű önálló modul segítségével még a futtatás előtt ellenőrizhető a futtatható állományokba ágyazott digitális aláírás. Amennyiben a fájl megváltozott, a **digsig** megakadályozza az indítását. Számos alkalmazásnak előnyére válhat ez, például a férgek elleni védelemben vagy a digitális jogok kezelésekor. A Trusted Computing tanúsítvány módszerével ellentétben a **digsig** bizonyítási folyamata mindvégig a géptulajdonos irányítása alatt marad. Nem fordulhat elő például, hogy egy távoli rendszer a gépünkön futó **digsig** modulra támaszkodva ellenőrizze, hogy a távoli rendszer által elfogadott program megadott változatát futtatjuk-e.

Manapság minden AMD Athlon processzorban van egy olyan hiba, amely az utasítás-előlekérésen (prefetch) keresztül bizonyos körülmények közt bizonyos címtartományokon képes memóriakezelési hibákat okozni. Ez nagyon hasonlít a hírhedt f00f Pentium hibához, tekintve, hogy mind a kettő alkatrészhiba, amelyet az operációs rendszernek kell kijavítania. A jó hír az, hogy ez a hiba a legtöbb esetben nem okoz igazán nagy károkat. Ennek a rendellenességnek az előfordulása nagyon ritka, valószínűleg ez az oka annak is, hogy ennyi ideig rejtve tudott maradni. *Andi Kleen* elkészítette a linuxos javítást a 2.4-es és 2.6-os változatokhoz, az AMD fejlesztői pedig jelezték, hogy megoldása együttműködik majd a jövőbeni Athlon nemzedékekkel, amelyekben ezt a hibát már kijavították. Ugyanakkor *Andi* szerint néhány Athlon-változatban, például a K7-ben, nemigen lehet kijavítani ezt a hibát, így aztán az új rendszereken még igen sokáig

szükség lehet erre a hibakerülő megoldásra.

Az SGI egy GPL meghajtót adott ki Altix soros konzoljához, azonban a szerzői jogi megjegyzésben felhívja rá a figyelmet, hogy az SGI a jogot a kódban használt valamennyi szabadalomra fenntartja. Bár a Szabad Szoftver Alapítvány (Free Software Foundation) jelezte az SGI felé, hogy ez elméletileg valóban nem sérti a GPL szerződést, végső soron mégis olyan helyzetet teremthet, hogy egy GPL jogú fejlesztést (például a rendszermagot) nem lehet egy másik GPL jogú projektben felhasználni. Ezek után felmerül a kérdés, hogy ilyen feltételek mellett a kódot egyáltalán be szabad-e engedni a rendszermagba? *Alan Cox* úgy döntött, hogy mind az SGI, mind az FSF oldalán a végére jár a dolognak, és megbizonyosodik afelől, hogy az FSF szerint ez valóban nem tekinthető a GPL szerződés megsértésének, illetve hogy hajlandó-e az SGI módosítani az álláspontját.

*Matthew Wilcox* elkezdte a rendszermagfejlécek két csoportba sorolását: vannak olyanok, amelyeket láthatnak a felhasználói térben futó források, és olyanok is, amelyeket nem. Állandóan gondot okoz, ha a felhasználói alkalmazások rossz rendszermagfejlécekre támaszkodnak. A rendszermag növekedésével és változásával párhuzamosan a fejlécek is változhatnak, s emiatt a felhasználói programok megsérülhetnek. A baj orvoslásához szigorú elkülönítést kell bevezetni. *Matthew* munkája éppen ezt könnyíti meg. Műve másokat is megihletett – többek között *Erik Andersent* –, akik a 2.6-os előkészítéseként megkezdték az alkalmazott elnevezési szabályok felülvizsgálatát.

Megkezdődött a rendszermag kódjának *arch/* könyvtárhoz hasonló, de újfajta rendszerezése. Ez az új rendszer azonban géptípusonkénti szétválasztás helyett fordítóprogramonként különíti el a forrásokat. Amióta az inter „ecc” (a GCC után) a második olyan fordítóprogrammá vált, amelyik sikeresen fel tudja építeni a Linux-rendszermagot, apró hibák kezdtek jelentkezni, így a fejlesztők olyan foltokat kezdtek el küldözgetni, amelyek az egyik vagy a másik fordítóprogram sajátosságaira alapoznak. A nemrégiben lezajlott vita után, amely a GCC által támogatott, de az ecc számára érthetetlen beépített assembly nyelv körül zajlott, *Linus Torvalds* végül úgy döntött, hogy az egyes fordítóprogramokhoz tartozó foltok tárolására fordítónként külön könyvtárakat kell létrehozni. *Sam Ravnborg* és *David Mosberger*, úgy tűnik, elsőként döntött az új rendszer támogatása mellett, és csomagjaik beküldésével megkezdték az átállást.

**Zack Brown**

*Linux Journal 2003. január, 117. szám*