

## Mi újság a rendszermag fejlesztése körül

© Kiskapu Kft. Minden jog fenntartva

■ **John W. Linville** bejelentette, hogy létrehozott egy *git* tárat az általa fejlesztett *Fedora Core* kernel számára. Az ezen a helyen található anyag alapja mindig hivatalos *Fedora* rendszermag, amit **Linville** kiegészít a *kernel.org*-on található hálózati foltokkal. A cél egyrészt az, hogy a *Fedora* felhasználók is hozzáférhessenek a legújabb eszközök meghajtóihoz, amelyekre másként várniuk kellene, amíg bekerülnek a hivatalos kiadásba, másrészt pedig egy tesztelői közösség összekovácsolása, amely különösen a hálózati eszközök kipróbálására koncentrál majd.

**Eric Dean Moore** elvállalta a korábban a támogatott eszközök listáján nem szereplő *LSI Logic MPT Fusion SCSI* meghajtó karbantartását. Ez a meghajtó korábban egységesen támogatta a *SCSI* és *Fibre Channel* eszközöket, nemrég azonban szétvált több olyan meghajtóra, amelyek csak egy-egy felületet támogatnak. Az egészben az igazán ironikus momentum az, hogy bár ezt a váltást maguk a fejlesztők javasolták, az *MPT Fusion* melletti legfőbb érv, illetve hírnevének alapja egészen idáig az volt, hogy egyetlen felületen keresztül támogatja a *SCSI*-t és *Fiber Channel*-t.

**Alessandro Di Marco Sluggard** néven megkezdte egy olyan speciális fájlrendszer fejlesztését, amely a kernelforrás letöltését hivatott egyszerűbbé tenni. Ahogy a rendszermag növekszik, a forrás letöltése egyre több és több időt és helyet igényel. **Sluggard** erre egy viszonylag egyszerű megoldást talált ki: az *rsync* segítségével egyszerűsíti a letöltés folyamatát. A helyi gépen a felhasználó egy speciális fájlrendszert olvas, amely transzparens módon követi a forrásfájlok változásait, és a hozzáférés előtt automatikusan frissíti, amit kell. Ezzel a módszerrel elmondása szerint neki körülbelül 200 MB helyet sikerült spórolnia, de szerinte ennél többre is van esély attól függően,

hogy milyen szolgáltatásokat akar valaki belefördíteni a rendszermagba. **Andrey Volkov** elkészítette az *ST M41T85* valósídejű órajelchipet támogató kódot. Munkájához **Mark A. Greer** *ST M41T100* chiphez készült meghajtóját használta fel alapként. Ami azt illeti, egyesek éppen azért azt javasolták, hogy a két kódot egyesíteni kellene. Ugyanakkor a két hardver közötti számos apró eltérés miatt egyelőre nem világos, hogy melyik megközelítés a gazdaságosabb, célszerű volna ugyanis elkerülni a rengeteg *#if* ág megjelenését a forrásban. Akár az is kiderülhet tehát, hogy a két kód egyesítése nem hogy nem egyszerűsíti a helyzetet, hanem kifejezetten bonyolítja. Mindazonáltal mind **Andrey** mind **Mark** egyetért az egyesítéssel, ha ez tűnik a helyes útnak. Mivel a *Sun* nyilvánossá tette a *ZFS* fájlrendszer forráskódját, **Tarkan Erimer** nemrég azt a kérdést kezdte feszegetni, vajon azt jelenti-e ez, hogy valamikor a jövőben *Linux* alatt is meg lehet valósítani a *ZFS* támogatását. Sajnos azonban ez nem tűnik túl valószínűnek, mivel a *Sun* a *CDDL* licenst használta, amely megengedi, hogy az így kibocsátott kódot zárt forráskódú alkalmazásokhoz is linkelni lehessen. Ez pedig összeegyeztethetetlen a *GPL* elveivel. Hacsak a *Sun* úgy nem dönt, hogy mind *CDDL*, mind *GPL* alatt kibocsátja a kódot, a *ZFS* támogatását legálisan nem lehet beilleszteni a hivatalos kernelfába. Annak persze továbbra sincs akadálya, hogy valaki a *Sun* kódja nélkül teljesen újraírja ennek a fájlrendszernek a kódját. Ha valakinek netán kétségei lettek volna afelől, mit is gondol **Linus Torvalds** a *CVS* rendszerről, nos az megnyugodhat. Amikor nemrég valaki azt javasolta, hogy a rendszermag dokumentációjába meg kellene emlékezni a jelenleg létező *CVS* táraokról is, **Linus** a következőket válaszolta: „*Semmi értelmét nem látom. A CVS*

*úgy rossz, ahogy van, és bárki, aki ezzel tartja karban az általa fejlesztett kódot, fel kell készüljön azokra a nehézségekre, amelyek a kernelfával való egyesítés során majd felmerülnek. És ez nem holmi elmélkedés, hanem gyakorlati tapasztalat, amivel nekünk már évekkkel ezelőtt tisztába kellett jönnünk, amikor hasonló dologgal próbálkoztunk. Éppen ezért nem akarok látni egyetlen külső CVS fát sem, és nem is vagyok hajlandó ezekkel semmilyen szinten foglalkozni.*” Hát ennyi. A *Linux* fejlesztése során alkalmazott filozófia egyik alaptétele az, hogy a rendszernek nem kell egyetlen szabványhoz sem vakon alkalmazkodnia csupán azért, mert az a szabvány létezik. A szabványoknak értelmeseknek kell lenniük, ha pedig a *Linux* nem követ egy amúgy jól átgondolt előírást, akkor az ahhoz való visszatérésnek valamiféle gyakorlati haszonnal kell szolgálnia. Ez a dilemma a közelmúltban is felmerült, amikor **Matthew Wilcox** megpróbálta kijavítani a *NO\_IRQ* definícióját, hogy az egyezzen a *PCI* szabvánnyal. **Linus** rámutatott, hogy ezt a szabványt maguk a hardvergyártók is évek óta figyelmen kívül hagyják, így a *NO\_IRQ* definíciójának megváltoztatása számos jelenleg jól működő meghajtó működéséptelenné válásához vezethet. Ami azt illeti, a *Linux* kernel sokkal áttekinthetőbb lenne, ha a különböző architektúrákat legalább nagyjából fedésbe hoznák, illetve ha az összes meghajtót sikerülne úgy módosítani, hogy tükrözze ezt a változást. Ez lenne a leghelyesebb. Ezzel együtt **Linus** elutasította a javaslatot, mondván igaz ugyan, hogy ez eredeti felépítés hibás, a kijavítása azonban olyan mennyiségű járulékos problémát vonna maga után, ami megkérdőjelezi az egész akció értelmét.

*Linux Journal* 2006. 144. szám

Zack Brown