

Mi újság a rendszermag fejlesztése körül

© Kiskapu Kft. Minden jog fenntartva

■ **Linus Torvalds** összeállított egy olyan *git* tárat a 2.6-os rendszermag fejlesztéséhez, amely az összes anyagot tartalmazza egészen a *BitKeeper* bevezetéséig visszamenőleg. A mag fejlesztése továbbra is egy új fában folyik, de mostantól a régi fa is hozzáférhető a tájékozódás végett, illetve bármilyen más célból, ami a fejlesztés során felmerülhet. Igaz ugyan, hogy ezek a foltok nem most először tűnnek fel egy *git* tárbán, viszont ez az első olyan alkalom, amikor egy ilyen, minden régebbi információt, és az elmúlt 3 „*git*-es hónap” termését tartalmazó tár mérete már elfogadhatónak tekinthető. A *git* első bejelentése óta tapasztalható fejlődése amúgy egyenesen elképesztő, és immár az 1.0-ás változat megjelenése is a küszöbön van. Bár a fejlesztők többsége valószínűleg ki fogja várni, amíg az eddig használt kedvenc változatkezelő rendszerének összes szolgáltatása elérhető lesz a *git*-ben is, magának a *git*-nek a fejlődése szempontjából ezek az új képességek tulajdonképpen már csak a habot jelentik a tortán. Azok az alapfunkciók ugyanis, amelyek lehetővé teszik hogy a rendszer robusztus módon támogassa az elosztott fejlesztést, kezdettől fogva megvannak. Úgy fest, a *RelayFS* hamarosan bekerül a fő kernelfába. Ez a modul régóta jelen van az **Andrew Morton** által fenntartott *-mm* fában, és az elmúlt hónapokban egyáltalán nem kellett az alapfunkcióit foltozni. Amit nyújt, azt eddig számos felhasználó találta már hasznosnak a legkülönbözőbb projektekben, és a kernelbe való bevezetésével kapcsolatos egyetlen érv tulajdonképpen csak az volt, hogy a *DebugFS* hasonló funkciókat lát el. Utóbbival igazából nincs is semmi gond, hacsak az nem, hogy egy ilyen nevű fájlrendszert senki nem akar semmi másra használni, csak nyomkövetésre. Így aztán **Andrew** nemrég kijelentette, hogy a *RelayFS* kódját be fogja nyújtani **Linus Torvalds**nak, különösen hogy a fejlesztők maguk is elérkezettnek látják az időt erre a lépésre. **Timothy R. Chavez** és mások elkészítettek egy olyan, az auditálási képességeket kiegészítő foltot a *VFS*-hez (*Virtual*

Filesystem), ami lehetővé teszi, hogy egy fájlrendszer-objektum adatait annak neve és neve alapján kérdezhessük le. A *VFS* jelenlegi megvalósításában ezt lehetetlen megoldani. Amikor **Timothy** először előállt az ötlettel, a kernelfejlesztők részéről némi ellenállással találkozott, mondván az *inotify* szolgáltatást pontosan ezért találták ki, és egészen hasonló lehetőségeket nyújt. A **Timothy** által kezdeményezett projekt indulásakor ugyanakkor az *inotify* fejlesztése még meglehetősen az elején járt, és csak egy külső foltként volt elérhető. Ennek megfelelően több értelme volt az auditálással kapcsolatos szolgáltatást egy külön, önálló kódrészletként megvalósítani. Most, hogy az *inotify* legalább már az *-mm* fába bekerült, kicsit jobban védhető a korábbi álláspont, miszerint talán mégis célszerűbb ezt használni. Ugyanakkor **Timothy**, és a folt más fejlesztői, egyetemben olyan korábbi kritikusokkal mint **Greg Kroah-Hartman** kidolgoztak egy tervet, amelynek lényege az, hogy a két rendszer közös szolgáltatásait megvalósítják egy absztrakt felületként, amit aztán mindkét projekt közösen használhat. Az azonban, hogy ez a nagy egyesítés mikorra készül el, egyelőre nyitott kérdés. **Adrian Bunk** „eltávolítandóként” jelölt meg néhány az *OSS*-ben található meghajtott. Annak eldöntése, hogy mi menjen, és mi maradjon soha nem könnyű. A cél nyilván az, hogy megmaradjon a támogatás minden létező, forgalomban vagy használatban levő hardverelemhez, így **Adriannak** minden egyes döntés előtt meg kell vizsgálnia, hogy létezik-e a megfelelő *ALSA* meghajtó, és az valóban megfelelően működik-e. Mármint a dolog hátulütője az, hogy ezt gyakorta csak olyasvalaki segítségével lehet megtenni, aki rendelkezik az érintett, esetenként kőkorszaki eszközzel, az ilyen emberek felkutatása pedig nem éppen egyszerű. A munka dandárját természetesen **Adrian** végzi, ő az, aki levélben megkeresi a potenciális segítőköt, és begyűjti tőlük a megfelelő kérdésekre adott válaszokat. Az ilyen „háztáji tisztogatás”, mint amit ő végez gyakorta igen hálátlan feladat, sőt sok-

szor mindenféle ellenállásba is ütközik azok részéről, akik kedvenc hardverük támogatásának elvesztésétől tartanak. Köszönettel tartozunk tehát **Adriannak** és a hozzá hasonlóknak azért, hogy minden héten órákat áldoznak arra, hogy „tisztán és szárazon” tartsák a kernelt.

A különböző *WSL* (*Wireless Security Lock*) „kütyük” egyre nagyobb támogatást élveznek *Linux* alatt is. Ezek a készülékek érzékelik, ha egy vezeték nélküli eszköz egy bizonyos ponttól túlságosan távolra került, és ilyenkor aktiválnak bizonyos védelmi funkciókat, például zárolják a monitort. **Brian Schau**, aki élete első kernelmodulját készíti, éppen egy ilyen eszközhöz ír meghajtott. Igaz ugyan, hogy egy *Bluetooth* telefon is képes betölteni hasonló funkciót, meg abban is van valami, hogy egy a felhasználói térben futó alkalmazás jobban megfeleljen erre a célra, mint **Brian** kernelmodulja, a projektnek mégis van értelme, hiszen egy létező hardverelemhez nyújt támogatást. A jelek szerint tehát, így vagy úgy, de a *WSL* egyszer csak támogatott lesz *Linux* alatt is.

Andrew Morton fűzött némi magyarázatot ahhoz a felzéséghez, hogy egyes felhasználók miért használnak újabban inkább cserefájlt (*swap*) cserepartíció helyett. A 2.6-os kernel esetében szerinte gyakorlatilag nincs különbség a két megoldás között. Akár a teljesítményt, akár a megbízhatóságot nézzük, a cserefájl és a cserepartíció ugyanazt nyújtja, egy apró kivétellel: ha a cserefájlt erősen töredeztet (fragmentált), az a teljesítmény rovására megy. De mivel a cserefájl fragmentáltságának mértéke nem változik a használat során, ezt a problémát egész egyszerűen kiküszöbölhetjük úgy, hogy a telepítéskor létrehozunk egy egybefüggőt. Ugyanakkor a 2.4-es kernelnél még a cserepartíció a győztes, mert ennek a használata nem igényli néhány olyan memóriaterület lefoglalását, amelyek a cserefájl kezeléséhez szükségesek.

Zack Brown

Linux Journal 2005. 139. szám