



## Mi újság a rendszermag fejlesztése körül

© Kiskapu Kft. Minden jog fenntartva

**A**git jelenség folytatódik. A saját fejlesztését kiszolgáló, kezdetektől villámgyors *git*, és a *Cogito* nevű közkedvelt csomagoló (wrapper) szkriptjei folyamatosan javulnak, és mára már hatalmas lökést adtak a kernel fejlesztésének. Különböző projektek állnak át *BitKeeper*-ről *git*-re. A *Net driver* és a *libata* fejlesztése máris átállt. Ugyanígy tett a *JFS* és *NTFS* projekt is. A stabil *w.x.z.y* kernelfa, melyet *Greg Kroah-Hartman* és *Chris Wright* tartanak karban, szintén nem régen cserélte le a *BitKeeper*-t, *git*-re. Néhány kernelhacker számára a *git* előrelépést jelent a *BitKeeper*hez képest, és sokkal több munkát tudnak vele elvégezni, mint azelőtt. Még maga *Linus Torvalds* is újrarendelje a foltok kezelésének módját abból a célból, hogy azt a *git*-hez hozzáillessze. A *BitKeeper* dokumentációt eltávolították a kernel forrásából és a levelezési listák, mint amilyen a *bk-commits-head*, amelyek eredetileg azt a célt szolgálták, hogy a *BitKeeper* változásait tegyék rajtuk közzé, ma a *git* kernelfolt megjegyzéseket fogadják. Ugyanakkor a *git* nem való mindenkinek. Amikor *Andrew Mortont* megkérdeztük, azt válaszolta, hogy nem áll szándékában a *git*-et saját *-mm* kernel fájának menedzselésére használni, mert saját patch szkriptjei teljesen kielégítik igényeit. *Matt Mackwall* szintén saját villámsebességű verzió követő rendszerével, a *Mercurial*al dolgozik. A *Mercurial* szintén egy kiváló eszköz, és sok szempontból egyenértékű a *git*-tel, kiváltképpen a sebesség területén. Valójában, ahogyan *Linus* rámutatott, mindkettő az alapvető viselkedésük tekintetében azonos. Valóban, mindkettő a verzió követés egy teljesen új aspektusát képviseli, és a projektek ahogyan az *arch* melyek korábban vezetők voltak ezen a téren, most hirtelen

küzdenek azért, hogy felzárkózzanak. *Markus Klotzbuecher* készített egy érdekes új virtuális fájl rendszert, amit *mini\_fo*-nak (*fanout overlay*) neveznek. Mely engedélyezi a felhasználóknak, hogy csak olvasható fájlrendszeren található fájllokba írjanak, azáltal, hogy létrehoz egy írható területet valahol máshol és a felhasználó változásait egy, a csak olvasható terület fölötti rétegben helyezi el. A felhasználó számára mindezt átlátszó módon. Ami eddig csak olvasható adat volt, most írható is tűnik, annak ellenére, hogy a csak olvasható adat sohasem változik. A *mini\_fo* eszközt arra a célja, hogy elérhetővé váljon a szoftver frissítés beágyazott rendszerekben, de egyéb felhasználási területet is találtak a szoftvernek, és még kétségtelenül lesz néhány. *Alan Cox* és *Bartlomiej Zolnierkiewicz*, a két remek *IDE* fejlesztő, bajba kerültek amikor egységesíteni próbálták ügykezeiket. Mialatt *Bartlomiej* a jelenlegi *IDE* karbantartó, *Alan* sok előkészületet tett azért, hogy a régi *IDE* kódot karbantartható állapotba hozza a kimondhatatlan (hulla-szór) rémálomból, amilyen már évek óta volt. Habár *Alan* az utóbbi időben némiképp a kívülről figyelte a *Linux* fejlődését, visszatért, hogy ellenőrizze az *IDE* haladását, és nem tetszését fejezte ki azokkal a változásokkal szemben, amit *Bartlomiej* tett. Úgy tűnik, mintha harag lenne kettejük között, ahogyan ez szokássá vált bármely két fejlesztő között, akik egyszerre, vagy egymás után dolgoztak az *IDE* fejlesztésén. *Bartlomiej* felajánlotta *Alan*-nek, hogy ágtassák el a kódot és készítsék külön a fejlesztést, amennyiben ez őt boldoggá teszi. Azt mondják az *IDE* fejlesztése még mindig ellenségeskedést szülhet a fejlesztők között. Ezért nekünk az *IDE* lemez iparágat magát kell vádolni, amelyik teljesen egyoldalú és megsem-

misített minden lehetséges szabványt, kivételt halmozva kivételre, mindezt elegyítették kereskedelmi titkokkal, és szabadalmaztatott dokumentumokkal, amitől bárki megőrülhetne, még akkor is, ha megkísérli karbantartani az *IDE* kernel kódot. Ez a család, mint *Bartlomiej* és *Alan*, és azok az emberek, akik előttük jöttek, mint *Mark Lord* és *Andre Hedrick*, éppen ezt csinálják hálával tartozunk ezért nagylelkű természetükért. Az *IDE* kód nélkül, legtöbbünk közel sem találná a *Linuxot* ilyen használhatóan. *Benjamin LaHaiser* nemrégiben megpróbálta egyszerűsíteni és jobban karbantarthatóvá tenni a szemaforra alapozott kölcsönös kizárást a teljes kernel architektúrában. A jelenlegi kód komplex, nehezen olvasható, és sok architektúra specifikus részletet tartalmaz. Ezek a kis árnyalatnyi különbségek a támogatott architektúrák számával arányosan nőnek, és érthető az erőfeszítés egy generikus szemafor rendszerre, amely lefordítható, és működtethető egységesen az összes architektúrán. Azonban a szemaforok alacsony szintű műveletek, és a nagy sebességű kód szükségessége vitathatatlan. Pontosan ezért kisebb lassulás, elképzelhető, hogy jelentős mértékű kernel lassulást eredményezne, ezért feltehetően bármelyik kísérlet a kód egységesítésben erős ellenállással találkozna a különböző architektúrák karbantartók részéről. Ez volt az, ahogyan valójában *Benjamin* munkáját fogadták. Habár néhány javítást természetesen készítettek, nem tűnik valószínűnek tehát, hogy a szemafor kód valamikor is igazán általános és egyszerű lesz. A sebesség még mindig elég erős ösztönzőerő.

Zack Brown

Linux Journal 2005. 137. szám



