

## Rendszermag-bővölési útmutató

Greg elmagyarázza, hogyan kell használni a BitKeepert vagy a patch és diff programokat együtt.

**A** Linux-magfejlesztés számos vonatkozásban eltér a hagyományos programfejlesztési folyamatoktól. Amikor a rendszermag adott részletén dolgozunk, több területre is figyelmet kell fordítanunk:

1. Figyelnünk kell arra, hogyan fogadják a rendszermag más részei az általunk az adott területen éppen végzett módosításokat.
2. Ne felejtjük el folyamatosan érvényesíteni a változtatásokat a gyors ütemű rendszermagfejlesztés célterületén.
3. Az általunk vagy a mások által végzett módosítások nyomán fellépő összeférhetetlenséget fel kell oldani.
4. Képesnek kell lenni rá, hogy saját módosításainkat mások számára könnyen használható formában tegyük közzé.

Sok éven keresztül foglalkoztam a soros kapuhoz kapcsolódó USB-meghajtók fejlesztésével és karbantartásával, végül az USB-kódok karbantartását a rendszermagba helyeztem át. Eme cikkben a korábbi munkám során használt eszközök működését fogom elmagyarázni, de az újabb eszközökről is szólni fogok, amelyek segítettek a rendszermagfejlesztéssel lépést tartani. Ráadásul ezek tették lehetővé, hogy mindezt kisebb erőfeszítéssel tegyem.

### A patch és a diff program használata

A rendszermaggal kapcsolatos teendők során az egyik leggyakoribb módszer a patch és diff programok használatát foglalja magában. Ezt használhatjuk a magfejlesztéshez valamilyen másfajta forráskódkezelő rendszer helyett. Az egyik módszer lényege abban áll, hogy egyidejűleg két fát használ: az egyik a „tiszta” fa, a másik pedig a „munka” fa. A tiszta fa a Linux-változattal együtt kapott rendszermag, míg a másik a tiszta fával megegyező rendszermagváltozatot alapul, és már a módosításainkat is tartalmazza. Itt érkezik el a pillanat, amikor a patch és diff programokkal kivonatot készíthetünk módosításainkról, hogy ezeket majd át tudjuk vinni a rendszermag frissebb változatába.

A példa kedvéért induljunk el a 2.4.18-as rendszermagtól, amelyet a <http://www.kernel.org/pub/linux/kernel/v2.4/linux-2.4.18.tar.gz> címről letöltöttünk a munkakönyvtárunkba:

```
$ ls
linux-2.4.18.tar.gz
Végezzük el a rendszermag kicsomagolását, azután a linux néven létrejövő könyvtárnak adjunk valami beszédesebb nevet:
$ tar -zxf linux-2.4.18.tar.gz
$ mv linux linux-2.4.18
$ ls
linux-2.4.18 linux-2.4.18.tar.gz
```

Most készítsünk másolatot az adott rendszermagváltozatról, és lássuk el valamilyen más névvel:

```
cp -R /usr/src/linux-2.4.18
/usr/src/linux-2.4.18-greg
$ ls
linux-2.4.18 linux-2.4.18-greg
↳ linux-2.4.18.tar.gz
```

A fejlesztési munkát most már nyugodtan végezhetjük a *linux-2.4.18-greg* könyvtárban, és érintetlenül fogjuk hagyni az eredeti rendszermagkönyvtárat. Amint befejeztük a munkát, a javítófoltot a többi érintett munkatársnak el kell küldenünk. A *Documentation/SubmittingPatches* elnevezésű állományban szereplő leírásból megtudhatjuk, hogy a magfejlesztők milyen formában szeretik a foltokat megkapni vagy éppen küldeni. Amellett ez az állomány ismerteti a *dontdiff* állomány használatát is, amely pontosan ilyen javítófoltok létrehozásában van a segítségünkre. A *dontdiff* az állományoknak azon körét határozza meg, amelyet szeretnénk, ha a diff program nem venne figyelembe.

Folt létrehozásához a következő parancs használható:

```
$ diff -Naur -X dontdiff \
linux-2.4.18 linux-2.4.18-greg/ > az_en_foltom
Ez a parancs létrehozza az az_en_foltom nevű állományt, amelynek tartalma a 2.4.18-as maghoz képest módosított rendszermag. Ez a folt a többi érintett fél számára már levélben továbbítható.
```

### Friss rendszermagváltozatok

Amint frissebb rendszermag jelenik meg, a működő rendszerben levő módosításokat át kell vinni az új rendszerbe, és a létrehozott foltot meg kell próbálni egy tiszta rendszermagra alkalmazni. A folyamat két lépésből áll:

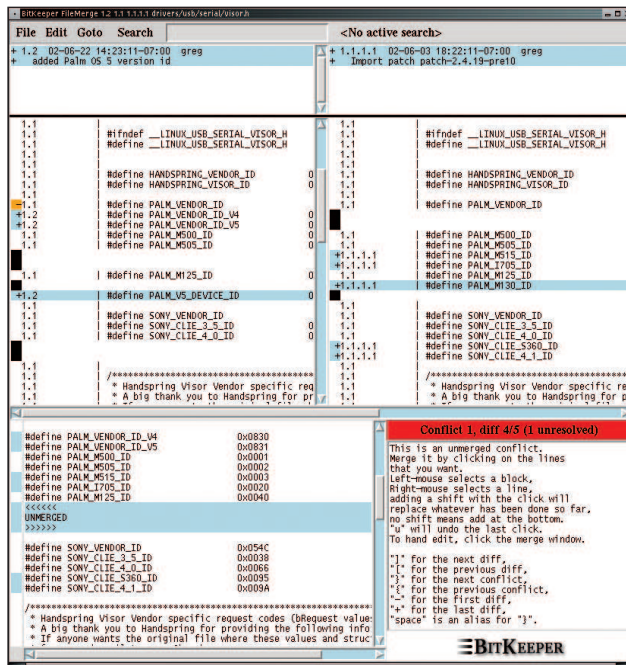
1. Hozzuk létre a javítófoltot, ahogyan azt az előző példában már láttuk.
2. A Kernel.org hivatalos javítófoltját felhasználva növeljük egygel a rendszermag változatszámát:

```
$ cd linux-2.4.18
$ patch -p1 < ../patch-2.4.19
$ cd ..
$ mv linux-2.4.18 linux-2.4.19
```
3. A javítófolt eltávolításával együtt növeljük a munkakönyvtár változatszámát, majd érvényesítjük a frissítést:

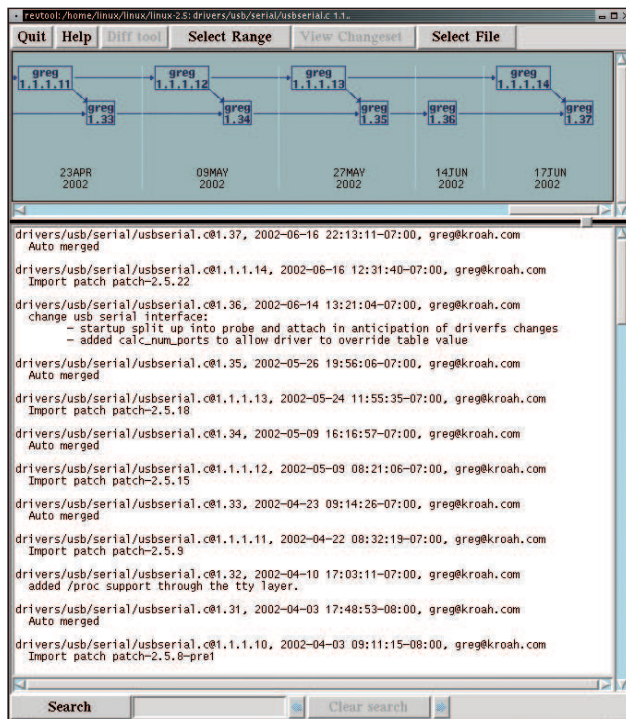
```
$ cd linux-2.4.18-greg
$ patch -p1 -R < ../my_patch
$ patch -p1 < ../patch-2.4.19
$ cd ..
$ mv linux-2.4.18-greg linux-2.4.19-greg
```
4. A hivatalos frissítés után kíséreljük meg saját javítófoltunk telepítését:

```
$ cd linux-2.4.19-greg
$ patch -p1 < ../my_patch
```

Amennyiben a folt telepítése nem zajlik hibátlanul, oldjuk meg a felmerült összeférhetetlenséget. A folt ezekről a *.rej* és *.orig* kiterjesztésű állományok révén fog tájékoztatást adni, amelyeket összehasonlíthatunk, és a hibákat kedvenc szövegszerkesztőnk segítségével, kézzel javíthatjuk. Ez az összefésülési folyamat lehet a legnehezebb szakasz, ha a forráskódfában magunk is végeztünk olyan módosításokat, amelyek érintik a korábban mások által megváltoztatott területeket. Amennyiben a fent leírt fejlesztési folyamat mellett döntesz,



1. kép Egy jellemző összevágatási nehézség



2. kép A BitKeeper nyomon követi a módosításokat

javaslom, a foltkészítő eszközöket szerezd be a <http://cyberelk.net/tim/patchutils> webhelyről. Ezek a programok ezernyi módon könnyítik meg a szöveges javítófoltok kezelését, és a magfejlesztőknek már sok fáradtságos munkaórát takarítottak meg.

### A forráskódkezelés

A patch és diff programok használatával végzett magfejlesztés általában véve meglehetősen jól működik. Egy idő után azonban az emberek többsége belefárad és elkezdi olyan más

munkamódszer után kutatni, ami nem jár annyi foltözvással meg összevágatással.

Néhány éve fedeztem fel a BitKeepert – amely a <http://www.bitkeeper.com> címen érhető el –, s azóta ezt használom magfejlesztésre. Eredetileg lehetővé tette számomra, hogy nyomon kövessem a rendszermagfán végbemenő külső módosításokat, és hogy a saját módosításaimat minden különösebb erőfeszítés nélkül közzétegyem. Minthogy maga **Linus Torvalds** és **Marcello Tosatti** szintén a BitKeepert használja magfejlesztésre, számomra nem jelent nehéz feladatot, hogy a fő rendszermagfába szánt foltjaimat elküldjem nekik. A BitKeepert magfejlesztő eszközként meglehetősen sokan visszásnak tartják, tekintettel a BitKeeper engedélyezési stratégiájára. A használatba vétel előtt olvasd át a felhasználási szerződést és dönts el, valóban használni kívánod-e. A fejlesztőeszközzel és néhány paranccsal való megismerkedés érdekében végiglépkedhetsz a BitMover honlapján.

A BitKeeperrel végzett munka során rendszermagodat Linus vagy Marcello rendszermagjára építheted, vagy a különböző változatok felhasználásával akár a saját változatodat is létrehozhatod.

Mindazonáltal a BitKeeperrel egy saját rendszerfa építését javaslom, hacsak nem a Linusnak vagy Marcellónak szánt javítófoltok küldését fontolgatod. Ily módon a különböző magfejlesztők által készített változások nem fognak maguk alá temetni és a teendőidre tudsz összpontosítani.

### A két fa

A BitKeeperrel végzett magfejlesztéshez ugyancsak két ismerős fát kell létrehozni (ezt a továbbiakban inkább rendszerraktárnak – repository – nevezném): a „tiszta” fát és a „munka” fát. A kiindulásképpen használt tiszta Bitkeeper-rendszerraktárhoz a saját munkakönyvtárban a Linux-változathoz kapott rendszermagot használod:

```
$ ls
linux-2.4.18.tar.gz
Uncompress this kernel:
$ tar -zxf linux-2.4.18.tar.gz
$ ls
linux linux-2.4.18.tar.gz
```

Ezek után hozd létre a linux-2.4 nevű BitKeeper-feladatot:

```
$ bk setup linux-2.4
```

A BitKeeper most néhány kérdést fog feltenni, azután szerkesztésre megnyit egy állományt – a feladat leírását ebben lehet megadni. Ehhez használhatod a kedvenc szövegszerkesztőt, majd a fájlt mentsd lemezre.

Egy **linux-2.4** nevű könyvtár fog létrejönni, ide fogja elhelyezni a feladatot. Az eredeti rendszermagváltozatot hozzuk át egy új rendszerraktárba:

```
$ ls
linux linux-2.4 linux-2.4.18.tar.gz
```

```
$ bk import -tplain linux linux-2.4
```

Ez a művelet el fog tartani egy darabig. Amint a BitKeeper végzett az fájlok átvételével, javaslok ennek a pontnak a rendszermag változatszámával történő megjelölését. Ez később a különböző rendszermagváltozatok fellelését meg fogja könnyíteni:

```
$ cd linux-2.4
$ bk tag LINUX_2.4.18
```

Készíts hasonmást erről a rendszerraktárról, vagyis a tiszta rendszerfáról, azaz klónozd, hogy később módosítani tudd:

```
$ bk clone linux-2.4 greg-2.4
```

A rendszermagon végzett összes munkát a **greg-2.4** nevű

© Kiskapu Kft. Minden jog fenntartva

könyvtárban fogjuk elvégezni. A bk-klónnál használhatjuk a -l lehetőséget, amely nagy lemezrészt használ fel, cserébe a meta-adatállományokhoz tartozó befűzések létrehozása következtében gyorsabban fog futni. Amint megtörténik egy állomány frissítése, a régi befűzést a BitKeeper megszünteti, és ahol csak szükséges, újat hoz létre. Ha nagyszámú különböző rendszerraktárt hozol létre ugyanazon a lemezen, ajánlott ennek a lehetőségnek a használata.

Amint befejeztük a változásokészletek létrehozását, beépítve a fejlesztési folyamat során végzett összes módosítást – erről a BitKeeper oktató oldalai részletesen szólnak –, létre szeretnénk hozni a változtatásainkat felsorakoztató foltot. Ez egy a *greg-2.4* könyvtárból kiadott egyszerű paranccsal tehető meg:

```
$ bk export -tpatch -rLINUX_2.4.18..+ -h \
> ../az_en_foltom
```

Ez a parancs létre fog hozni a jelzett rendszermagtól – LINUX\_2.4.18 – a legfrissebb változásokészletekig terjedő foltot, amelyet mentünk lemezre *az\_en\_foltom* néven. Ez a folt már bármilyen címre továbbítható levélben, pontosan ugyanúgy, ahogy a diff által készített folt. Biztosan észre fogod venni, hogy ennek az elkészítése sokkal rövidebb ideig tartott, mint a folt létrehozása a patch és diff programok használatát feltételező előző módszerrel.

## Új rendszermagváltozatok

Amint megjelenik egy új rendszermagváltozat működő rendszerből, a módosításokat nyilván át szeretnéd vinni az új rendszermagba. Itt mutatkozik meg a BitKeeper igazi ereje a hagyományos patch-diff-módszerrel szemben. Először térjünk vissza az eredeti tiszta fához, és hozzuk át az új foltot:

```
$ ls
greg-2.4 linux-2.4 patch-2.4.19
```

```
$ cd linux-2.4
$ bk import -tpatch -SLINUX_2.4.19
➔ ../patch-2.4.19
```

Ha a BitKeeper egyes állományokat újonnan létrehozottként vagy töröltként jelölt meg, azok valójában még csak át lettek nevezve vagy át lettek helyezve a fában. Ekkor a BitKeeper egy grafikus felülettel ellátott eszközt bocsát a rendelkezésedre, amellyel kiderítheted, hogy tulajdonképpen mely állományok jöttek most létre, melyek kaptak új elnevezést, és mik kerültek törlésre.

Most pedig térjünk vissza a munkarendszerraktárhoz, és hozzuk át bele a módosításokat:

```
$ cd ../greg-2.4
$ bk pull
```

A BitKeeper a 2.4.18-as és 2.4.19-es változatú rendszermagok módosításait összeválogatja a munkarendszerraktárba. Abban az esetben, ha a saját módosításaid valamilyen összefésülési nehézségbe ütköznek az új rendszermagban, a program tájékoztatni fog róla, és meg fogja kérdezni, hogy mit kívánsz tenni. Az ehhez hasonló összeütközések megoldásához a nagy segítséget jelentő háromutas összeválogató eszköz használatát javaslom. Ez az eszköz megmutatja az eredeti fájlt a saját magad által végzett változtatásokkal, illetve a folt vagy mások által véghez vitt módosításokkal. Ezt követően lehetővé teszi, hogy eldöntsd, melyik változtatást szeretnéd elfogadni, esetleg a fájlt kézzel szerkeszteni, netán mindkét módosítást érvényesíteni. A 1. kép egy a fő rendszermagban végzett változtatással összeütközésbe került állományt szemléltet, amelynek módosítását jómagam hajtottam végre.

Amint végeztél az összeférhetlenségek feloldásával – ami

ugye nem volt sokkal könnyebb, mint a .rej kiterjesztésű állományok átvizsgálása –, máris folytathatod a munkát a frissített rendszermagon. Hadd mondjam el ismét, hogy az összes általunk végzett módosítást a *greg-2.4* könyvtárban kiadott alábbi paranccsal egyetlen foltba lehet foglalni:

```
$ bk export -tpatch -rLINUX_2.4.19..+ -h \
> ../my_patch
```

## A BitKeeper további előnyei

A BitKeeper lehetővé teszi az adott fájlban végzett összes módosítás időrendbeli, egyszerű áttekintését. Látható például, hogy a módosítást az egyik rendszermagfolt végezte-e vagy maga a fejlesztő. A 2. képen látható példában a rendszerraktárban tárolt *drivers/usb/serial/usbserial.c* állomány időbeli változásait kísérhetjük figyelemmel. Ezzel az eszközzel megfigyelhetjük, milyen változások zajlottak le, de még azt is, hogy egy-egy változatban mely programsorok lettek módosítva. A magfejlesztéshez használt BitKeeper egyik roppant nagy előnye, hogy igen nagy teljesítményű változatkezelő rendszer, és lehetővé teszi, hogy ugyanazon a programkódrészletben több más fejlesztővel együtt egyidejűleg dolgozz. Lehetővé teheted, mások számára, hogy munkafádból kivonatot készítsenek, vagy a munkafád tárolására akár önálló kiszolgálógépet is telepíthetsz. A hasznos példák végett keresd fel a BitKeeper oktatóoldalait, amelyekből megtudhatod, hogyan kell telepíteni, és milyen módon kell a fejlesztési ciklust működtetni.

## Összefoglalás

A Linux-magfejlesztés két különböző útját mutattam be: az egyik a patch és diff programok használatát jelenti, a másik a BitKeeperét. Személy szerint úgy találom, hogy a BitKeeperrel több időt tudtam a tényleges fejlesztési munkára fordítani, és kevesebbet kellett a módosítások bogarászásával töltenem. Továbbá nem veszítettem el a fonalat a 2.2, 2.4 és 2.5 jelű rendszermagokba épített linuxos USB- és gyors csatlakoztatású (Hot Plug) PCI egységek meghajtóprogramjaival kapcsolatban sem.

*Linux Journal* 2002. szeptember, 101. szám



Greg Kroah-Hartman

jelenleg a Linux-rendszerek USB és gyors csatlakoztatású (PCI Hot Plug) egységeinek rendszermagba épített meghajtóprogramjainak fejlesztője. Az IBM-nél dolgozik, és rendszer-maggal kapcsolatos kérdésekkel foglalkozik.

A greg@kroah.com címen lehet vele kapcsolatba lépni.

## Kapcsolódó címek

Val Henson és Jeff Garzik „BitKeeper for Kernel Developers” (BitKeeper rendszermagfejlesztők számára), 2002, Ottawa Linux Symposium Proceedings.

A BitKeeper oktatóoldalai a Világhálón

➔ <http://www.bitkeeper.com/Test.html>

A BK-műveletről pingvinstílusban a 2.5 rendszermagban, a *Documentation/BK-usage/bk-kernel-howto.txt* állományban olvashatsz.

Külső változások kezelése a BitKeeper használatával

➔ <http://www.bitkeeper.com/tracking.ps>