Régi ismerősök látogatása

A 2.5-ös változat szerint tárgyalt rendszermag API-k közül néhányon kisebb módosításokat hajtottak végre, mielőtt bekerültek volna a 2.6-os rendszermagba. Ezekről a módosított változatokról szeretnék szólni néhány szót.

Ez témában első cikkemet több mint egy évvel ezelőtt írtam, és mivel a Linux fejlesztése gyors ütemben folyik, számos korábban leírt dolog mára elavultnak számít. Ez alkalommal a korábban ismertetett rendszermag API-k módosításait szeretném áttekinteni.

A tty változásai

A tty réteg az egyik legbiztosabb működésű, sokat bizonyított rendszermag-API. A hivatkozások megfelelő számlálásának és a zárolási lehetőségnek a hiánya, valamint a tty-eszközök kiosztásának furcsa módja mind a réteg elöregedésére vezethető vissza. Szerencsére ***Al Viro*** nemrég kipucolta a régről maradt ócskaságokat a 2.5-ös rendszermagsorozatban látott tty-rétegből. Mindeközben természetesen számos dolog megváltozott, és az új tty-illesztőprogramok írásakor erre tekintettel kell lenni.

A Linuxvilág 2002. szeptemberi és novemberi számában (illetve a Linux Journal 2002. augusztusi és októberi számában; a cikkek elérhetők a ➲ www.linuxjournal.com/article/5896 és a ➲ www.linuxjournal.com/article/6226 címen) a tty-rétegről, illetve a `struct tty_driver` adatszerkezet szolgáltatások visszahívóival történő feltöltéséről írtam. Azóta `struct tty_operations` névvel létrejött egy új adatszerkezet, ez tartalmazza az összes szolgáltatás visszahívóját. A `struct tty_driver` továbbra is tartalmazza a régi függvénymutatókat, így ezek átmásolására készült egy új függvény, a `tty_set_operations`. A kettősség hamarosan remélhetőleg megszűnik. A `struct tty_driver` adatszerkezetből jó néhány változó kikerült: a `table`, a `termios`, a `termios_locked` és a `refcount` mezőnek búcsút inthetünk. A tty-réteg most már az összes zárolási és hivatkozásszámlálási igényt kielégíti, megszabadítva az egyes tty-illesztőprogramokat e feladatok kezelésének terhétől. A `magic` és a `num` változót most már nem kell közvetlenül a tty-illesztőprogramnak beállítania. A változóértékek beállítását új függvény, az `alloc_tty_driver` végzi, amit minden tty-illesztőprogramnak meg kell hívnia, ugyanis ez végzi az illesztőprogramok számára a helyfoglalást. Az adott illesztőprogram által támogatott különféle tty-eszközök számát átadott értékként szükséges tudatni ezzel a függvénnyel. A tiny tty illesztőprogram, amelyet korábban vettünk elő példaként, az alábbiak szerint hozza létre a `struct tty_driver` adatszerkezetet:

```
/* a tty illesztőprogram helyfoglalása */
tiny_tty_driver = alloc_tty_driver(TINY_TTY_MINORS);
```

Korábban a tty-illesztőprogram nevének megválasztása is gondokat okozott, ugyanis a rendszermag `devfs` függvénye túlterhelte a `name` mezőt. ***Christoph Hellwig*** munkájának hála, aki a `struct tty_driver` adatszerkezetben létrehozott egy új változót, a `devfs_name`-t. Most már a `name` mező egyszerű, rövid nevet is kaphat, amelyet a tty ***proc*** fájlokban fogunk viszontlátni. A `devfs_name` értékét arra a névre kell állítani, amelyet a `devfs` használ az illesztőprogram eszközcsomópontjának létrehozásakor. A 2.5-ös rendszermagsorozatban a `MOD_INC_USE_COUNT` és a `MOD_DEC_USE_COUNT` makrókat túlságosan csúnyának minősítették, és használatuk a rendszermagban nagyrészt meg is szűnt. Ahhoz, hogy ez megtörténhessen, a modulhivatkozások számlálása az eredeti hívásokhoz képest eggyel magasabb szintre került. Most már lehetséges, hogy a rendszermag az előtt növelje egy-egy modul hivatkozási számlálóját, hogy beugrana magába a modulba. Hasonlóan, amikor a rendszermag végzett a modul használatával, a számláló értékét önműködően csökkenteni tudja.

A modulokra vonatkozó módosítás a tty-rétegben történt, így egyetlen tty-illesztőprogramnak sem kell tartalmaznia a `MOD_*`

2. lista Példa egy „tiocmget” függvényre

```
int
tiny_tiocmget(struct tty_struct *tty,
              struct file *file)
{
    struct tiny_private *tp = tty->private;
    unsigned int msr = tp->msr;
    unsigned int mcr = tp->mcr;
    unsigned int result = 0;

    result = ((mcr & MCR_DTR)     ? TIOCM_DTR: 0)
                                    /* DTR beállítva */
           | ((mcr & MCR_RTS) ? TIOCM_RTS: 0)
                                    /* RTS beállítva */
           | ((msr & MSR_CTS) ? TIOCM_CTS: 0)
                                    /* CTS beállítva */
           | ((msr & MSR_CD)  ? TIOCM_CAR: 0)
                                /* vivőérzékelés beállítva */
           | ((msr & MSR_RI)  ? TIOCM_RI:  0)
                                /* csengetésjelző beállítva */
           | ((msr & MSR_DSR) ? TIOCM_DSR: 0);
                                    /* DSR beállítva */

    return result;
```

3. lista Példa a tiocmset függvényre

```
int
tiny_tiocmset(struct tty_struct *tty,
              struct file *file,
              unsigned int set, unsigned
              ➥int clear)
{
    struct tiny_private *tp = tty->private;

    if (set & TIOCM_RTS)
        mcr |= MCR_RTS;
    if (set & TIOCM_DTR)
        mcr |= MCR_RTS;
    if (set & TIOCM_LOOP)
        mcr |= MCR_LOOPBACK;

    if (clear & TIOCM_RTS)
        mcr &= ~MCR_RTS;
    if (clear & TIOCM_DTR)
        mcr &= ~MCR_RTS;
    if (clear & TIOCM_LOOP)
        mcr &= ~MCR_LOOPBACK;

/* az új MCR érték beállítása az eszközön */
    tp->mcr = mcr;
    return 0;
}
```

makrókat. Egy `owner` változó ugyanakkor bekerült a `struct tty_driver` adatszerkezetbe, feladata a tty-illesztőprogramot birtokló modul megadása. Az alábbi sor ennek a változónak a helyes megadását szemlélteti:

```
tiny_tty_driver->owner = THIS_MODULE;
```

A tty-mag ebből tudja meg, hogy melyik modul tartozik ehhez a tty-illesztőprogramhoz.
Az 1. listán látható (55. CD Magazin/Api könyvtára), hogy a változások következtében miként módosul a tty-illesztőprogramok helyes kezdeti értékadásának és bejegyzésének módja. A tty-kihívó (callout) eszközöket teljesen eltávolították a rendszermagból. Azok a tty-illesztőprogramok, amelyek a 2.5-ös rendszermagfában megvoltak, és kihívásokat alkalmaztak, a módosításnak megfelelően átalakultak.

Kevesebb ioctl

A tty-adatszerkezet változásaival párhuzamosan néhány tty `ioctl` is kikerült a rendszerből, ezek a következők: `TIOCMGET`, `TIOCMBIS`, `TIOCMBIC` és `TIOCMSET`. Helyüket két új függvényvisszahívó vette át: a `tiocmget` és a `tiocmset`, ezeket a `struct tty_operations` adatszerkezethez adták hozzá. A függvények megadása a következő:

```
int (*tiocmget)(struct tty_struct *tty,
                struct file *file);
int (*tiocmset)(struct tty_struct *tty,
                struct file *file,
                unsigned int set,
                unsigned int clear);
```

A `tiocmget` függvény meghívására akkor kerül sor, amikor a tty-mag vagy a felhasználó tudni szeretné az adott tty-kapu éppen érvényes vonali beállításait. Működése majdnem pontosan olyan, mint a régi `TIOCMGET` `ioctl` hívásé. A vonalállapotot különböző `MSR_*` értékek adják meg, ezeket – a korábbi `ioctl` által az illesztőprogramtól elvárt felhasználói területre való másolás helyett – közvetlenül a függvényhívás adja vissza (lásd a *2. listát*).
A `tiocmset` függvényre akkor van szükség, amikor a tty-mag vagy a felhasználó be szeretné állítani, illetve törölni szeretné a különféle vonali beállítások valamelyikét. Ez a függvény egymaga helyettesíti a `TIOCMBIS`, a `TIOCMBIC` és a `TIOCMSET` `ioctl` hívásokat. A függvény `set` és `clear` változóival szabható meg, hogy melyik vonali beállítást szeretnénk megadni, illetve melyiket akarjuk törölni. Ugyanazt a vonali beállítást nem lehet egyszerre beállítani és törölni is, így a változók feldolgozásának sorrendje érdektelen (lásd a *3. listát*).
A tty-mag módosításai kismértékben az usbserial-magot is érintették. A `tiocmget` és a `tiocmset` függvény újdonságként jelent meg a `struct usb_serial_device_type` adatszerkezetben. Az ezekre a függvényekre irányuló tty-hívásokat a rendszer továbbadja az alacsonyabb szinten futó `usbserial` illesztőprogramoknak – feltéve, hogy az adott `usbserial` illesztőprogram támogatja ezeket a visszahívásokat.

Köszönetnyilvánítás

Szeretném kifejezni hálámat ***Al Viro***, ***Christoph Hellwig*** és ***Russell King*** uraknak, amiért végre nekiláttak a tty-réteg megtisztításának. Odaadásuk révén a tty-réteg méltó része lesz a rendszermagnak. Módosításaik fontos szerepet játszanak a tty-illesztőprogramfelület egyszerűbbé tételében, és az ő révükön az illesztőprogramok készítői a jövőben a rendszermagot érintő párbeszédek kezelése helyett inkább az eszközök jellegzetességeire összpontosíthatnak.

A cikkhez kapcsolódó listák megtalálhatóak az 55. CD Magazin/Api könyvtárában.

Linux Journal 2003. október, 114. szám

Greg Kroah-Hartman (greg@kroah.com)
Jelenleg a Linux-rendszermag különféle illesztőprogram-alrendszereiért felelős. Az IBM-nél dolgozik és a Linux-rendszermaggal kapcsolatos kérdésekkel foglalkozik.