Linux PCMCIA COMO

David Hinds, *dhinds@hyper.stanford.edu*
Traducido por David Limón Romero, *dlr@cuates.pue.upaep.mx*
Original: v2.40, 10 Septiembre 1999

Este documento describe cómo instalar y usar los servicios de las tarjetas PCMCIA con Linux. Las últimas versiones de este documento puede encontrarlas siempre en *ftp://hyper.stanford.edu/pub/pcmcia/doc*. La versión en HTML está en *http://hyper.stanford.edu/HyperNews/get/pcmcia/home.html*

Índice General

1 Información general y requerimientos de hardware

1.1 Introducción

Los Servicios de Tarjeta para Linux son un paquete de soporte completo para PCMCIA o *PC Card*. Incluye un conjunto de módulos cargables en el kernel que implementan una versión de la interface del programa de aplicación de Servicios de Tarjetas, un conjunto de controladores de clientes para tarjetas específicas, y un demonio controlador de tarjetas que responde a los eventos de inserción y extracción de tarjetas, el cual carga y descarga los controladores según sea necesario. Soporta «extracción en caliente» de la mayoría de tarjetas, por lo que pueden ser insertadas y extraídas de forma segura en cualquier momento.

Este software está en continuo desarrollo. Probablemente contenga *bugs*, y debe ser usado con precaución. Haré lo que esté en mi mano para resolver los problemas que me son comunicados, pero si no me los dice, nunca lo sabré. Si usa este código, espero que me envíe sus experiencias, ¡buenas o malas!

Si tiene sugerencias de cómo puede mejorarse este documento, por favor hágamelo saber (*dhinds@hyper.stanford.edu*).

1.2 Licencia y renuncia de responsabilidad

Este documento puede ser reproducido o distribuido en cualquier forma sin mi permiso previo. Las versiones modificadas de este documento, incluyendo traducciones a otros idiomas, pueden ser distribuidos libremente, si son claramente identificados como tales, y siempre que este copyright se incluya intacto.

Este documento puede ser incluído en distribuciones comerciales sin mi previo consentimiento. Aunque no suponga requisito, me gustaría estar informado de su uso. Si pretende incorporar este documento en un trabajo para ser publicado, por favor contacte conmigo para asegurarme que tiene la última versión disponible.

Este documento se proporciona «TAL CUAL», sin garantías expresas o implícitas. Utilice la información en este documento bajo su propio riesgo.

1.3 ¿Cuál es la última versión, y dónde puedo obtenerla?

La versión actual de los Servicios de Tarjetas es la `3.0`, y las actualizaciones menores o reparaciones de *bugs* se numeran `3.0.1`, `3.0.2`, y así sucesivamente.

El código fuente de la última versión está disponible en ***ftp://hyper.stanford.edu*** en el directorio `/pub/pcmcia` como `pcmcia-cs-3.0.?.tar.gz`. Habrá usualmente varias versiones ahí. Por lo general, solo conservo la última versión menor para dar origen a una versión mayor.

Las nuevas versiones pueden contener código relativamente sin probar, así que también conservo la última versión de la última mayor como «colchón» relativamente estable; el retraso actual es `2.9.12`. Vd. decide qué versión es más apropiada, el archivo `CHANGES` mostrará las diferencias más importantes.

ftp://hyper.stanford.edu es replicado en ***ftp://sunsite.unc.edu*** (y todos los servidores réplica de *Sunsite*) en `/pub/Linux/kernel/pcmcia`.

Si no se siente Vd. a gusto compilando controladores, hay controladores precompilados incluidos con las versiones actuales de la mayoría de las distribuciones principales de Linux, incluyendo *Slackware*, *Debian*, *Red Hat*, *Caldera*, *SuSE*, e *Yggdrasil*, entre otros.

1.4 ¿Qué sistemas están soportados?

Este paquete debería correr en la mayoría de portátiles basados en Intel y que sean «Linuxizables». También corre en plataformas basadas en Alpha (*DEC Multia*, por ejemplo). Se programa para hacer al paquete completamente *dual-endian*, así que también soporta plataformas basadas en *PowerPC* (*Apple Powerbooks*, por ejemplo). Los controladores de *sockets* más comunes están soportados. Las bahías de tarjetas PCMCIA para sistemas de escritorio deben funcionar si usan un controlador soportado, y se conectan directamente al bus ISA o PCI, lo opuesto a los adaptadores SCSI-a-PCMCIA o IDE-a-PCMCIA.

Están soportados los siguientes controladores:

- Cirrus Logic PD6710, PD6720, PD6722, PD6729, PD6730, PD6732, PD6832
- Intel i82365sl B, C, y secuencias DF, 82092AA
- O2Micro OZ6729, OZ6730, OZ6832, OZ6833, OZ6836, OZ6860
- Omega Micro 82C092G
- Ricoh RF5C296, RF5C396, RL5C465, RL5C466, RL5C475, RL5C476, RL5C478
- SMC 34C90
- Texas Instruments PCI1130, PCI1131, PCI1210, PCI1220, PCI1221, PCI1250A, PCI1251A, PCI1251B, PCI1450

- Toshiba ToPIC95, ToPIC97 (experimental)
- Vadem VG465, VG468, VG469
- VLSI Technologies 82C146, VCF94365
- VIA VT83C469
- Databook DB86082, DB86082A, DB86084, DB86084A, DB86072, DB86082B

Otros controladores que están registrados como compatibles con el Intel *i82365sl*, funcionarán también como norma general.

El soporte para tarjetas *CardBus* de 32 bits es todavía experimental. Los manejadores previos a la versión `3.0` sólo soportan tarjetas de 16 bits en sockets *CardBus*. Debido al paso tan rápido en el cambio de la tecnología para el hardware de portátiles, aparecen nuevos controladores frecuentemente, y puede producirse cierto estancamiento entre el momento en que aparece un nuevo modelo en el mercado, y el que haya soporte para ese controlador.

Toshiba ha dispuesto parcialmente documentación sobre sus chipsets *ToPIC95* y *ToPIC97*, sin embargo, la información que han liberado no ha sido la realmente adecuada. A pesar de los informes de conflictos, Toshiba no ha hecho algún esfuerzo efectivo para remediar esta situación. Hay problemas serios en el soporte de Linux para los chipsets ToPIC, que no pueden ser resueltos hasta que esté disponible una documentación mejor, o la ayuda adecuada por parte de Toshiba. No recomiendo el uso de portátiles Toshiba por el momento. Para el uso de tarjetas de 16 bits, recomiendo establecer el modo de puente a *PCIC* en la configuración de la BIOS; para tarjetas *CardBus*, la decisión es suya.

El controlador Motorola *6AHC05GA* usado en portátiles Hyundai, no está soportado. El controlador en la *HP Omnibook 600* tampoco.

1.5 ¿Qué tarjetas están soportadas?

La versión actual incluye controladores para una variedad de tarjetas ethernet, para tarjetas módem y puertos serie, varios controladores para adaptadores SCSI, un controlador para tarjetas de unidades ATA/IDE, y controladores para tarjetas de memoria que sólo soportan la mayoría de tarjetas SRAM y algunas tarjetas flash. El archivo `SUPPORTED.CARDS` incluído en cada versión de Servicios de Tarjetas lista todas las tarjetas que se sabe que funcionan al menos en un sistema.

La probabilidad de que funcione una tarjeta que no está en la lista de soportados depende del tipo. Esencialmente todos los módems deberían funcionar con el controlador provisto. Algunas tarjetas de red pueden funcionar si hay versiones OEM de las tarjetas soportadas. Otro tipo de tarjetas de E/S (*frame buffers*, tarjetas de sonido, etc) no funcionarán hasta que alguien escriba los controladores apropiados.

1.6 ¿Cuándo estará soportada mi tarjeta favorita (no soportada)?

Desafortunadamente, no me pagan por escribir controladores para dispositivos, así que si quiere tener un controlador para su tarjeta favorita, probablemente tendrá trabajar un poco. Idealmente, me gustaría trabajar hacia un modelo como el del kernel de Linux, donde yo sea el responsable principalmente del código del núcleo y otros autores puedan contribuir y mantener los controladores para tarjetas específicas. El archivo `SUPPORTED.CARDS` menciona algunas tarjetas para las cuales los controladores están actualmente en progreso. Trataré de ayudar donde pueda, pero tenga en cuenta que depurar controladores de dispositivo del kernel por email no es particularmente efectivo.

Los fabricantes interesados en ayudar a proveer soporte Linux para sus productos pueden contactar conmigo a fin de acordar consultorías.

1.7 Listas de correo y otras fuentes de información

Solía mantener una base de datos y una lista de correo de usuarios de Linux PCMCIA. Recientemente he convertido mi página web para información de Linux PCMCIA en un sitio *HyperNews*, con un conjunto de listas de mensajes de temas de Linux PCMCIA. Hay listas para instalación y configuración, para diferentes tipos de tarjetas, para programar y depurar. La página de información de Linux PCMCIA está en *http://hyper.stanford.edu/HyperNews/get/pcmcia/home.html*. Los usuarios pueden solicitar otificación por email de nuevas respuestas a preguntas particulares, o notificación para todos los mensajes nuevos en una categoría dada. Espero que esto sea un repositorio útil de información, para cuestiones que van más allá del enfoque del COMO.

Hay una lista de Linux dedicada a asuntos de portátiles, la lista `linux-laptop`. Para más información, envíe un mensaje con la palabra `help` a *majordomo@vger.rutgers.edu*. Para suscribirse, envíe un email que contenga el mensaje `subscribe linux-laptop` a la misma dirección. Esta lista de correo puede ser un buen foro de discusión de asuntos de Linux PCMCIA.

La página de Linux Laptop está en *http://www.cs.utexas.edu/users/kharker/linux-laptop* tiene enlaces a muchos sitios que tienen información acerca de la configuración de tipos específicos de portátiles para Linux. Hay también una base de datos para buscar información acerca de configuración de sistemas.

1.8 ¿Por qué no distribuyen binarios?

Para mi, distribuir los binarios puede suponer una molestia importante. Esto es complicado porque algunas características solo pueden ser seleccionadas al momento de compilar, y porque los módulos dependen mucho de contar con una configuración «correcta» del kernel. Así, probablemente necesite distribuir módulos precompilados junto con los kernels correspondientes. Más que esto, la necesidad más grande de los módulos precompilados es cuando se instala Linux en un sistema limpio. Esto típicamente requiere configurar los controladores para que puedan ser utilizados en el proceso de instalación, para una distribución de Linux en particular. Cada distribución de Linux tiene su propia idiosincrasia, y no me resulta factible el proveer discos `boot` y `root` para cada una de las combinaciones de controladores y distribuciones.

PCMCIA forma parte ahora de las principales distribuciones de Linux, incluyendo *RedHat*, *Caldera*, *Slackware*, *Yggdrasil*, *Craftworks* y *Nascent Technology*.

1.9 ¿Por qué el paquete es tan grande?

Bueno, no es realmente tan grande al fin y al cabo. Todos los módulos controladores ocupan alrededor de 500K de espacio en disco. Los programas de utilidades añaden unos 70K, y los scripts en `/etc/pcmcia` son de 50K. Los controladores principales ocupan unos 55K de la memoria del sistema. El demonio `cardmgr` será generalmente intercambiado excepto cuando cuando las tarjetas sean insertadas o extraídas. El tamaño total del paquete es comparable a las implementaciones de servicios de tarjetas de DOS/Windows.

2 Compilación e instalación

2.1 Prerequisitos y configuración del kernel

Antes de empezar, debería pensar si realmente necesita compilar el paquete por sí mismo. Todas las distribuciones comunes de Linux vienen con paquetes de controladores precompilados. Generalmente, sólo necesita instalar los controladores si necesita una característica nueva de los más actuales, o si ha actualizado y/o reconfigurado su kernel de forma que es incompatible con los incluidos en su distribución de Linux. A pesar de que compilar el paquete no es técnicamente difícil, requiere algo de familiaridad general con Linux.

Las siguientes cosas deben estar instaladas en su sistema antes de comenzar:

- El árbol de fuentes del kernel, serie `2.0.*`, `2.1.*`, o `2.2.*`
- Un conjunto apropiado de utilidades de módulos.
- La interface de utilidades `XForms` para X11 (Opcional).

Necesita tener la estructura completa del código fuente del kernel, no sólo una imagen actualizada del kernel. Los módulos de los controladores contienen algunas referencias a los archivos fuentes del kernel. Mientras que Vd. busca compilar un kernel nuevo para eliminar manejadores innecesarios, instalar PCMCIA no requiere que lo haga así.

Los fuentes y parches «estables» actuales del kernel están disponibles en *ftp://sunsite.unc.edu/pub/Linux/kernel/v2.0*, o en *ftp://tsx-11.mit.edu/pub/linux/sources/system/v2.0* Los kernels en desarrollo los puede encontrar en los subdirectorios `v2.1`. las utilidades de módulos actuales puede encontrarlas en la misma ubicación.

En los fuentes del kernel de Linux, el archivo `Documentation/Changes` describe las versiones de todas las clases de otros componentes del sistema que son requeridas por esa versión del kernel. Probablemente quiera revisarlo y verificar que su sistema está actualizado, especialmente si tiene actualizado el kernel. Si está usando un kernel en desarrollo, asegúrese de estar usando la combinación correcta de bibliotecas compartidas y herramientas de módulos.

Cuando configure su kernel, si planea usar una tarjeta ethernet PCMCIA, debe activar el soporte para red, y desactivar los controladores normales de tarjetas de red de Linux, incluyendo `pocket and portable adapters`. Todos los controladores para tarjetas de red PCMCIA están compilados como módulos cargables. Cualquiera de los controladores compilados dentro de su kernel sólo desperdiciará espacio.

Si quiere usar SLIP, PPP o PLIP, necesitará ya sea configurar el kernel con ese soporte activado, o usar la versión de los módulos cargables de esos controladores. Hay una desafortunada deficiencia en el proceso de configuración de los kernels `1.2.X`, en el que no es posible establecer opciones de configuración (como compresión SLIP) para un módulo cargable, así que es probablemente mejor enlazar SLIP dentro del kernel si es que lo necesita.

Para usar un adaptador token ring PCMCIA, el kernel debe estar configurado con la opción `Token Ring driver support` (`CONFIG_TR`) activada, aunque debe dejar `CONFIG_IBMTR` desactivado.

Si requiere usar un adaptador IDE PCMCIA, su kernel debe estar configurado con la opción `CONFIG_BLK_DEV_IDE_PCMCIA` activada, para los kernels desde `2.0.*` hasta `2.1.*`. Los kernels antiguos no soportan dispositivos IDE extraíbles; los nuevos no requieren una configuración especial.

Si va a usar un adaptador SCSI PCMCIA, debe habilitar `CONFIG_SCSI` cuando configure el kernel. Debe activar también cualquier controlador de alto nivel (disco SCSI, cinta, cdrom, genérico) que espere usar. Debe desactivar todos los controladores de bajo nivel para adaptadores en particular, porque sólo le quitarán espacio.

Si busca modularizar un controlador que se necesita para un dispositivo PCMCIA, debe modificar `/etc/pcmcia/config` para especificar qué módulos necesitan ser cargados para qué tipos de tarjetas. Por ejemplo, si el controlador serie está modularizado, entonces la definición del dispositivo serie debería ser:

```
device "serial_cs"
  class "serial" module "misc/serial", "serial_cs"
```

Este paquete incluye una utilidad llamada `cardinfo` que está basada en X para monitorizar el estado de la tarjeta. Está basada en un *toolkit* de libre distribución, la biblioteca *XForms*. Esta librería está disponible como un paquete separado de la mayoría de distribuciones de Linux. Si desea compilar `cardinfo`, deberá instalar `XForms` y todas las cabeceras y bibliotecas de desarrollo habituales para X antes de configurar el paquete PCMCIA.

2.2 Instalación

He aquí una sinopsis del proceso de instalación:

- Descomprima `pcmcia-cs-3.0.?.tar.gz` en `/usr/src`
- Ejecute `make config` en el nuevo directorio `pcmcia-cs-3.0.?`
- Ejecute `make all`, y luego `make install`.
- Configure el script de inicio y los archivos de opciones en `/etc/pcmcia` para su sistema.

Si planea instalar cualquier controlador que sea una contribución y que no esté incluído en la distribución principal de PCMCIA, descomprima cada uno de ellos en el directorio raíz del árbol PCMCIA. Luego siga las instrucciones normales de compilación. Los controladores extras se compilarán e instalarán automáticamente.

Cuando ejecute `make config`, se le preguntarán algunas opciones de configuración y se comprobará su sistema para verificar que se satisfagan todos los prerequisitos para instalar soporte PCMCIA. En la mayoría de los casos, sólo tendrá que aceptar todas las opciones de configuración que vienen por omisión. Asegúrese de comprobar cuidadosamente la salida de éste comando en caso de que hubiera problemas. Están disponibles las siguientes opciones:

¿Hay un directorio de instalación alternativo?

Si está compilando el paquete para instalarlo en otro equipo, especifique un directorio destino alternativo cuando se le pregunte. Debe ser una ruta absoluta. Todos los archivos serán instalados relativos a este directorio. Entonces estará listo para aplicar `tar` a este directorio y copiarlo a su equipo destino, y desempaquetarlo relativo a su directorio raíz para instalar todo en los lugares apropiados.

¿Necesita indicadores de compilación para depurar?

Vea la sección 7.3 (Primeros auxilios al depurar a bajo nivel) para mayor información acerca de esta opción.

¿Necesita compilar versiones «trusting» de las utilidades de tarjetas?

Algunas de las utilidades de soporte (`cardctl` y `cardinfo`) pueden ser compiladas ya sea de forma `safe` o `trusting`. La forma `safe` previene a los usuarios no-root de modificar configuraciones de tarjetas. La forma `trusting` permite a los usuarios ordinarios ejecutar comandos para suspender y reactivar tarjetas, resetear tarjetas, y cambiar el esquema de configuración actual. La forma configurada por omisión es `safe`.

¿Necesita incluir soporte para tarjetas de 32-bits (*CardBus*)?

Deberá seleccionar esta opción si desea usar tarjetas *CardBus* de 32-bits. No se requiere para tener soporte con puentes *CardBus* si sólo planea usar tarjetas PC de 16-bits.

¿Necesita incluir chequeo de recursos para BIOS PnP?

Esto compila código adicional en el módulo principal PCMCIA para comunicarse con el BIOS PnP de un sistema para obtener información de los recursos que están incluidos en la «placa madre» (puertos serie y paralelos, sonido, etc), para ayudar a prevenir conflictos de recursos. Si se habilita, se crearán algunos archivos extra de recursos bajo `/proc/bus/pccard`, y las herramientas `lspnp` y `setpnp` se pueden usar para visualizar y manipular los dispositivos PnP del BIOS.

¿Cómo configurar opciones específicas del kernel?

Hay algunas opciones de configuración del kernel que afectan a las herramientas PCMCIA. El script de configuración puede deducirlo desde el kernel actual (el caso por omisión y más común). Alternativamente, si está compilando para instalar en otro equipo, puede leer la configuración del árbol de los fuentes del kernel, o cada opción se puede establecer interactivamente.

El script de configuración se puede ejecutar de forma no-interactiva, para compilar automáticamente o para reconfigurar rápidamente después de una actualización del kernel. Algunas opciones adicionales no utilizadas con frecuencia sólo pueden ser establecidas desde la línea de comandos. Ejecutando: `Configure -help` se listarán todas las opciones disponibles.

Al ejecutar `make all` seguido de `make install` compilará y luego instalará los módulos del kernel y los programas de utilidades. Los módulos del kernel serán instalados en `/lib/modules/<version>/pcmcia`. Los programas `cardctl` y `cardmgr` serán instalados en `/sbin`. Si `cardinfo` se compila, será instalado en `/usr/bin/X11`.

Los archivos de configuración serán instalados en el directorio `/etc/pcmcia`. Si está instalando sobre una versión antigua, sus scripts de configuración anteriores se respaldarán antes de ser reemplazados. Los scripts guardados tendrán la extensión de tipo `*.O`.

Si no sabe qué tipo de controlador usa su sistema, puede utilizar la utilidad `probe` en el subdirectorio `cardmgr/` para determinarlo. Hay dos tipos principales: el tipo *Databook TCIC-2* y el tipo compatible con Intel *i82365SL*.

En raras ocasiones, el comando `probe` será incapaz de determinar su tipo de controlador automáticamente. Si tiene un sistema *Halikan NBD 486*, tiene un controlador *TCIC-2* en una localización inusual: necesitará editar `rc.pcmcia` para cargar el módulo `tcic`, y también especificar el parámetro `PCIC_OPTS` a `tcic_base=0x02c0`.

En algunos sistemas que usan controladores Cirrus, incluyendo el *Nec Versa M*, la bios pone el controlador en un estado especial de suspensión al iniciar el sistema. En esos sistemas, el comando `probe` fallará al encontrar cualquier controlador conocido. Si esto pasa, edite `rc.pcmcia` y especifica `PCIC` a `i82365`, y `PCIC_OPTS` a `wakeup=1`.

2.3 Opciones de inicio

El script de inicio de PCMCIA reconoce varios grupos de opciones de inicio, establecidas por medio de variables de entorno. Se pueden separar múltiples opciones por medio de espacios y encerradas en comillas. La colocación de las opciones de inicio depende de la distribución de Linux que se esté usando. Pueden ser colocados directamente en el script de inicio, o pueden mantenerse en un archivo de opciones separado. Revise la sección 2.5 (Notas acerca de distribuciones de Linux específicas) para más detalles. Se pueden establecer las siguientes variables:

PCMCIA

Esta variable especifica si el soporte PCMCIA debe ser iniciado o no. Si está especificado de forma diferente a `yes`, el script de inicio será desactivado.

PCIC

Esto identifica el módulo controlador de *PC Card Interface Controller*. Hay dos opciones: `tcic` o `i82365`. Virtualmente todos los controladores actuales están en el grupo *i82365*. Esta es la única opción obligatoria a establecer.

PCIC_OPTS

Esto especifica las opciones para el módulo PCIC. Algunos controladores tienen características opcionales que pueden o no ser implementadas en un sistema en particular. En algunos casos, es imposible para el socket controlador detectar si esas características están implementadas. Revise la página del manual correspondiente para una descripción completa de las opciones disponibles.

CORE_OPTS

Esto especifica las opciones para el módulo `pcmcia_core`, el cual implementa los servicios principales del controlador PC Card. Es conveniente echar un vistazo a `man pcmcia_core` para más información.

CARDMGR_OPTS

Esto especifica las opciones que se pasarán al demonio `cardmgr`. Revise `man cardmgr` para más información.

SCHEME

Si está activado, el esquema de configuración de PC Card será inicializado a este modo en el momento de arrancar. Revise la sección 4.2 (Un vistazo a los scripts de configuración de PCMCIA) para ver la discusión de esquemas.

Los controladores de sockets de bajo nivel, `tcic` e `i82365`, tienen varios parámetros de sincronización de bus que pueden necesitar ser ajustados para sistemas con velocidades de bus no muy usuales. Los síntomas de los problemas de sincronización incluyen problemas al reconocer las tarjetas, congelamiento bajo carga pesada, tasas de error altas, o rendimiento pobre de dispositivos. Sólo ciertos puentes tienen parámetros de sincronización ajustables, revise la página correspondiente del manual para ver qué opciones existen para su controlador. He aquí un pequeño resumen:

- Los controladores Cirrus tienen muchos parámetros de sincronización. Lo más importante parece ser el indicador `cmd_time`, la cual determina la longitud de los ciclos de bus PCMCIA. En los sistemas rápidos 486 (DX4-100, por ejemplo) parece ser beneficioso el incrementar esto de `6` (por omisión) a `12` o `16`.
- El controlador Cirrus *PD6729 PCI* tiene el indicador `fast_pci`, la cual debe establecerse si la velocidad del bus PCI es mayor a 25 Mhz.
- Para controladores *Vadem VG-468* y *Databook TCIC-2*, el indicador `async_clock` cambia la velocidad relativa del bus PCMCIA y los ciclos de bus del equipo. Activar este indicador añade estados de espera extra a algunas operaciones. Sin embargo, todavía no he sabido de ningún portátil que necesite esto.
- El módulo `pcmcia_core` tiene el parámetro `cis_speed` para cambiar la velocidad de la memoria, la cual se usa para acceder a la *Estructura de Información de Tarjeta* (*Card Information Structure*) (*CIS*) de una tarjeta. En algunos sistemas con pulsos de bus rápidos, incrementar este parámetro (por ejemplo, reducir la velocidad de los accesos a tarjeta) puede resultar beneficioso para quien tenga problemas al reconocerlas.
- Esto no es cuestión de sincronización, pero si Vd. tiene más de un controlador ISA-a-PCMCIA en su sistema o tiene bahías extra en una estación, el módulo `i82365` debe ser cargado con el parámetro `extra_sockets` establecido a `1`. Esto no deberá ser necesario para detección de puentes PCI-a-PCMCIA y PCI-a-CardBus.

He aquí algunas configuraciones de sincronización para algunos sistemas específicos:

- En un *ARM Pentium-90* o en un *Midwest Micro Soundbook Plus*, use `freq_bypass=1 cmd_time=8`.
- En un *Midwest Micro Soundbook Elite*, use `cmd_time=12`.

- En un *Gateway Liberty*, pruebe con `cmd_time=16`.
- En un *Samsung SENS 810*, use `fast_pci=1`.

2.4 Configuración de recursos del sistema

Los servicios de tarjetas deben evitar automáticamente el ocupar puertos e interrupciones que ya estén en uso por otros dispositivos estándar. Intentará así mismo detectar conflictos con dispositivos desconocidos, pero esto no es del todo fiable, y en algunos casos puede que necesite excluir explícitamente recursos para un dispositivo en `/etc/pcmcia/config.opts`.

He aquí algunas configuraciones de recursos para tipos específicos de portátiles. Revíselas con cuidado: pueden darle información necesaria para resolver problemas, pero algunas están (inevitablemente) obsoletas y ciertamente contienen errores. Las correcciones y adiciones serán bienvenidas.

- En un *AMS SoundPro*, excluya la irq 10.
- En algunos modelos *AMS TravelPro 5300*, use el rango de memoria `0xc8000-0xcffff`.
- En un *BMX 486DX2-66*, excluya irq 5, irq 9.
- En un *Chicony NB5*, use el rango de memoria `0xda000-0xdffff`.
- En un *Compaq Presario 1020*, excluya el puerto `0x2f8-0x2ff`, irq 3, irq 5.
- En un *HP Omnibook 4000C*, excluya el puerto `0x300-0x30f`.
- En un *IBM ThinkPad 380*, y posiblemente en las series 385 y 600, excluya el puerto `0x230-0x233`, e irq 5.
- En un *IBM ThinkPad 600*, excluya el puerto `0x2f8-0x2ff`.
- En un *Micron Millenia Transport*, excluya irq 5, irq 9.
- En un *NEC Versa M*, excluya irq 9, y el puerto `0x2e0-2ff`.
- En un *NEC Versa P/75*, excluya irq 5, irq 9.
- En un *NEC Versa S*, excluya irq 9, irq 12.
- En las series *NEC Versa 6000*, excluya el puerto `0x2f8-0x33f`, irq 9, irq 10
- En un *ProStar 9200*, *Altima Virage*, y *Acquiline Hurricane DX4-100*, excluya irq 5, y el puerto `0x330-0x35f`. Puede usar el rango de memoria `0xd8000-0xdffff`.
- En un *Siemens Nixdorf SIMATIC PG 720C*, use el rango de memoria `0xc0000-0xcffff`, y el puerto `0x300-0x3bf`.
- En un *TI TravelMate 5000*, use el rango de memoria `0xd4000-0xdffff`.
- En un *Toshiba T4900 CT*, excluya irq 5, y los puertos `0x2e0-0x2e8`, `0x330-0x338`.
- En un *Twinhead 5100*, HP 4000, *Sharp PC-8700* and PC-8900, excluya irq 9 (sonido), irq 12.
- En un *MPC 800 Series*, excluya irq 5, y el puerto `0x300-0x30f` para el CD-ROM.

2.5 Notas acerca de distribuciones de Linux específicas

Esta sección está incompleta. Todas las correcciones y adiciones serán bienvenidas.

2.5.1 Debian

Debian usa el conjunto de scripts de arranque de tipo System V. El script de inicio PCMCIA está instalado en `/etc/init.d/pcmcia`, y las opciones de inicio se especifican en `/etc/pcmcia.conf`. La configuración de `syslog` de Debian colocará los mensajes del kernel en `/var/log/messages` y los mensajes del demonio `cardmgr` en `/var/log/daemon.log`.

Debian distribuye el sistema PCMCIA en dos paquetes: el paquete `pcmcia-cs` que contiene `cardmgr` y otras herramientas, páginas del manual, y los scripts de configuración; y el paquete `pcmcia-modules` que contiene los módulos controladores del kernel.

2.5.2 Red Hat, Caldera, Mandrake

Estas distribuciones usan la organización de scripts System V. El script de inicio de PCMCIA está instalado en `/etc/rc.d/init.d/pcmcia`, y las opciones de arranque se guardan en `/etc/sysconfig/pcmcia`. Observe que al instalar el paquete de Red Hat puede instalar un archivo de opciones de inicio por omisión que tiene PCMCIA desactivado. Para habilitar PCMCIA, la variable `PCMCIA` debe establecerse en `yes`. La configuración por omisión del `syslogd` de Red Hat grabará todos los mensajes interesantes en `/var/log/messages`.

El paquete PCMCIA de Red Hat contiene un reemplazo para el script de inicio de red, `/etc/pcmcia/network`, el cual se acopla al panel de control de red de Red Hat. Esto es conveniente para el caso donde sólo se usa un adaptador de red, con un conjunto de parámetros de red, pero no tiene la flexibilidad completa del script regular de red PCMCIA. Compilar e instalar una distribución fuente de PCMCIA nueva, sobreescribirá el script de red, rompiendo el enlace con el panel de control. Si prefiere el script de Red Hat, puede evitarlo bien usando únicamente RPMS de Red Hat, o creando `/etc/pcmcia/network.opts` que contenga lo siguiente:

```
if [ -f /etc/sysconfig/network-scripts/ifcfg-eth0 ] ; then
    start_fn () {
        /sbin/ifup $1
    }
    stop_fn () {
        /sbin/ifdown $1
    }
fi
```

Red Hat maneja su distribución de los fuentes de PCMCIA con pocas modificaciones dentro de su SRPM del kernel, en lugar de gestionarlo como un paquete separado.

2.5.3 Slackware

Slackware usa el conjunto de scripts de inicio de BSD. El script de inicio de PCMCIA está instalado en `/etc/rc.d/rc.pcmcia`, y las opciones de inicio se especifican en el mismo `rc.pcmcia`. El script de inicio de PCMCIA se invoca desde `/etc/rc.d/rc.S`.

2.5.4 SuSE

SuSE usa el conjunto de scripts System V, con los scripts de inicio almacenados en `/sbin/init.d`. El script de inicio de PCMCIA está instalado en `/sbin/init.d/pcmcia`, y las opciones de arranque se guardan en `/etc/rc.config`. El script de inicio de SuSE está algo limitado y no permite que las variables de inicio de PCMCIA sean invalidados desde el prompt de inicio de `lilo`.

3 Resolución de problemas de instalación y configuración

Esta sección describe algunos de los errores más comunes del subsistema PCMCIA. Compare sus síntomas con los ejemplos. Esta sección sólo describe fallos generales que no son específicas de un controlador o tipo de tarjeta en particular.

Antes de diagnosticar un problema, debe saber dónde se almacena el registro del sistema (revise la sección 2.5 (Notas acerca de distribuciones de Linux específicas)). Debe estar familiarizado con las herramientas básicas de diagnóstico como `dmesg` y `lsmod`. Preste especial atención al hecho de que muchos componentes de los controladores (incluyendo todos los módulos del kernel) tienen sus propias páginas individuales en el manual.

Intente definir su problema lo más ampliamente posible. Si tiene varias tarjetas, pruebe cada tarjeta de forma aislada, y en diferentes combinaciones. Intente arranques de Linux en frío y arranques en caliente de Windows. Compare el arrancar con tarjetas insertadas, o insertar las tarjetas después de iniciar. Si normalmente usa su portátil ensamblado con una *dockstation*, prúebelo aparte. Algunas veces, dos bahías se comportarán de forma diferente.

Es casi imposible depurar problemas de un controlador cuando se intenta instalar Linux por medio de un dispositivo PCMCIA. En lugar de eso, si puede identificar el problema basándose en los síntomas, los discos de instalación son difíciles de modificar, especialmente sin tener acceso a un sistema Linux ya funcionando. La personalización e instalación de los discos de instalación es completamente dependiente de la distribución de Linux que elija, y más allá del enfoque de este documento. En general, el mejor curso de acción es instalar Linux usando otros medios, obteniendo los controladores más recientes, y depurando el problema entonces, si es que persiste.

3.1 No se cargan los módulos básicos de PCMCIA.

Síntomas:

- Aparecen errores acerca de que la versión del kernel difiere cuando se ejecuta el script de inicio de PCMCIA.
- Después de iniciar, `lsmod` no muestra algún módulo PCMCIA.
- `cardmgr` informa `no pcmcia driver in /proc/devices` en el registro del sistema.

Los módulos del kernel contienen información de la versión, la cual se comprueba con el kernel actual cuando se carga un módulo. El tipo de chequeo depende de la opción del kernel `CONFIG_MODVERSIONS`. Si es falso, entonces el número de versión del kernel se compila dentro de cada módulo y el programa `insmod` comprueba esto para compararlo con el kernel actual. Si `CONFIG_MODVERSIONS` es verdadero, entonces cada símbolo exportado por el kernel tiene un «checksum». Esos códigos se comparan con los códigos correspondientes compilados dentro de un módulo.

La idea de esto fue crear módulos menos dependientes de la versión, porque los checksums sólo pueden cambiar si la interface del kernel cambia, y podría generalmente permanecer a lo largo de actualizaciones menores del kernel. En esencia, los «checksums» se han desactivado para ser mas restrictivos, porque muchas interfaces del kernel dependen de las opciones pasadas al momento de compilarse. También, los checksums han resultado ser jueces excesivamente pesimistas respecto a compatibilidad.

El enfoque práctico de esto es que los módulos del kernel están muy atados a tanto la versión del kernel, como a muchas opciones de configuración del mismo. Generalmente, un grupo de módulos compilados para un kernel `2.0.31` no cargará con otros kernels `2.0.31` a menos que se tome un cuidado especial asegurándose que los dos fueron compilados con configuraciones similares. Esto resulta ser un asunto difícil para la distribución de módulos precompilados del kernel.

Tiene Vd. varias opciones:

- Si obtuvo controladores precompilados como parte de una distribución de Linux, verifique que esté usando el mismo kernel que venía con su distribución, sin modificaciones. Si pretende usar los módulos precompilados que venían con su distribución, deberá permanecer con el mismo kernel que trajera ésta.
- Si ha reconfigurado o actualizado su kernel, probablemente necesitará compilar e instalar el paquete PCMCIA desde cero. Esto se hace fácilmente si ya tiene instalada la estructura fuente del kernel. Revise la sección 2 (Compilación e instalación) para instrucciones más detalladas.
- En algunos casos, las incompatibilidades en otros componentes del sistema pueden prevenir la carga correcta de los módulos del kernel. Si ha actualizado su propio kernel, ponga atención a la sección 2.1 (Requisitos) acerca de utilidades para módulos y `binutils` que se listan en el archivo `Documentation/Changes` del árbol de directorios de los fuentes del kernel.

3.2 Algunos módulos controladores no cargan

Síntomas:

- Los módulos base (`pcmcia_core`, `ds`, `i82365`) cargan correctamente.
- Al insertar una tarjeta, emite un pitido agudo + un pitido grave.
- `cardmgr` informa de errores de versiones diferentes en el registro del sistema.

Algunos de los módulos controladores requieren servicios del kernel que pueden o no estar presentes, dependiendo de la configuración del kernel. Por ejemplo, los controladores de tarjetas SCSI requieren que el kernel sea compilado con soporte SCSI, y los controladores de red requieren un kernel de red. Si a un kernel le falta una característica necesaria, `insmod` puede avisar acerca de símbolos indefinidos y rechazar la carga de un módulo en particular. Note que los mensajes de error de `insmod` no distinguen entre errores por diferencias de versiones y errores por falta de símbolos.

Específicamente:

- El controlador serie `serial_cs` requiere que el soporte en el kernel esté activado con `CONFIG_SERIAL`. Este controlador se debe compilar como módulo.
- El soporte para tarjetas serie multipuerto o tarjetas multifunción que incluyen dispositivos serie o módems, requieren que se active `CONFIG_SERIAL_SHARE_IRQ`.
- Los clientes SCSI requieren que `CONFIG_SCSI` esté activada, junto con las opciones apropiadas para los controladores de alto nivel (`CONFIG_BLK_DEV_SD, CONFIG_BLK_DEV_SR` etc. para kernels `2.1`) que pueden ser compilados como módulos.
- Los controladores de red requieren que se habilite `CONFIG_INET` El soporte para red del kernel no se puede compilar como módulo.
- El cliente token-ring requiere que el kernel se compile con la opción `CONFIG_TR` activada.

Hay dos formas de proceder:

- Recompile el kernel con las características necesarias activadas.

- Si las características han sido compiladas como módulos, entonces modifique /etc/pcmcia/config para precargar esos módulos.

El archivo /etc/pcmcia/config puede especificar qué módulos adicionales necesitan cargarse para un cliente en particular. Por ejemplo, para el controlador serial, uno puede ser:

```
device "serial_cs"
  class "serial" module "misc/serial", "serial_cs"
```

Las rutas hacia los módulos se especifican relativas al nivel más alto del directorio de módulos para la versión actual del kernel; si no se especifica la ruta relativa, entonces la ruta por omisión será hacia el subdirectorio pcmcia.

3.3 fallos en la búsqueda de interrupciones

Síntomas:

- El sistema se congela cuando se cargan los controladores PCMCIA, incluso cuando no hay tarjetas presentes.
- El registro del sistema muestra que el sondeo tuvo éxito, justo antes de que se congele, pero no muestra resultados de las pruebas de interrupciones.

Después de identificar el tipo de controlador, el controlador del socket sondea las interrupciones libres. Este «sondeo» o «tanteo» consiste en programar el controlador para cada interrupción aparentemente libre, generando una interrupción *soft* (suave), para ver si la interrupción puede ser detectada correctamente. En algunos casos, el sondear una interrupción en particular puede interferir con otro dispositivo del sistema.

La razón de este «tanteo» es identificar interrupciones que parezcan estar libres (es decir, aquellas que no están reservadas por otro controlador de dispositivo), ya sea porque no esté conectado físicamente a la controladora, o que esté conectado a otro dispositivo que no tiene un controlador.

En el registro del sistema, un sondeo realizado con éxito tiene este aspecto:

```
Intel PCIC probe:
  TI 1130 CardBus at mem 0x10211000, 2 sockets
  ...
  ISA irqs (scanned) = 5,7,9,10 status change on irq 10
```

Hay dos formas de proceder:

- El sondeo de interrupciones puede estar restringida a una lista de interrupciones utilizando el parámetro irq_list para los controladores. Por ejemplo, irq_list=5,9,10 limitará la búsqueda a tres interrupciones. Todos los dispositivos PCMCIA estarán restringidos a usar esas interrupciones (asumiendo que pasen el tanteo). Puede ser que necesite determinar qué interrupciones son tanteables de forma segura a base de ensayo y error.
- El sondeo de interrupciones puede desactivarse completamente al cargar el controlador del socket con la opción do_scan=0. En este caso, se usará una interrupción por omisión, la cual evita interrupciones ya utilizadas por otros dispositivos.

En cualquier caso, las opciones de tanteo pueden especificarse en el script de inicio de PCMCIA utilizando la definición PCIC_OPTS, por ejemplo:

```
PCIC_OPTS="irq_list=5,9,10"
```

Como podrá notar, `/proc/interrupts` es absolutamente inútil cuando se van a diagnosticar problemas en el sondeo de interrupciones. El tanteo es lo suficientemente sensible como para nunca intentar usar una interrupción que ya está en uso por otro controlador de Linux. Los controladores PCMCIA están ya teniendo en cuenta toda la información de `/proc/interrupts`. Dependiendo del diseño del sistema, un dispositivo inactivo puede todavía ocupar una interrupción y causar problemas si es probado por PCMCIA.

3.4 Fallos en la búsqueda de puertos de E/S

Síntomas:

- El sistema se congela cuando `cardmgr` se inicia por primera vez, incluso cuando no hay tarjetas presentes.
- El registro del sistema muestra un tanteo positivo del controlador del host, incluyendo resultados de sondeos de interrupción, pero no muestra resultados de sondeos de E/S.
- En algunos casos, el tanteo de E/S será positivo, pero avisa de un gran número de de exclusiones aleatorias.

Cuando `cardmgr` procesa los rangos de puertos de E/S listados en `/etc/pcmcia/config.opts`, el kernel tantea esos rangos para detectar los dispositivos latentes que ocupan espacio de E/S pero que no están asociados con un controlador de Linux. El tanteo es de sólo lectura, pero en algunos casos extraños, leer desde un dispositivo puede interferir con una función importante del sistema, resultando en «congelamiento».

La guía de usuario de su sistema debe traer un mapa de los dispositivos del sistema, mostrando sus rangos de E/S y de memoria. Esos pueden ser excluidos explícitamente en `config.opts`.

Por otra parte, si el sondeo no resulta fiable en su sistema, puede ser desactivado estableciendo `CORE_OPTS` a `probe_io=0`. En este caso, deberá ser muy cuidadoso al especificar solamente rangos de puertos genuinamente disponibles en `config.opts`, en lugar de usar las configuraciones por omisión.

3.5 Fallos durante la comprobación de la memoria

Síntomas:

- Los controladores principales cargan correctamente cuando no hay tarjetas presentes, sin errores en el registro del sistema.
- El sistema se congela y/o reinicia tan pronto como se inserte una tarjeta antes de que se escuche algún pitido.

O alternativamente:

- Todas las inserciones de tarjetas generan un pitido agudo seguido de un pitido grave.
- Todas las tarjetas son identificadas como *anonymous memory cards*
- El registro del sistema avisa que varios rangos de memoria han sido excluidos.

Los módulos principales realizan un chequeo de los primeros 16 bits de memoria en el momento en que se inserta la tarjeta. Esta exploración puede interferir potencialmente con otros dispositivos de memoria mapeados. Así mismo, los paquetes de controladores pre-3.0.0 realizan una exploración más agresiva que los controladores más recientes. La ventana de memoria se define en `/etc/pcmcia/config.opts`. La ventana por omisión es grande, así que puede ayudar a restringir la exploración a un rango más reducido. Los rangos razonables para incluir son: `0xd0000-0xdffff`, `0xc0000-0xcffff`, `0xc8000-0xcffff`, o `0xd8000-0xdffff`.

Si tiene controladores PCMCIA DOS o Windows, puede deducir que región de memoria usan esos controladores. Tenga en cuenta que las direcciones de memoria de DOS se especifican normalmente en forma de «segmentos», los cuales dejan el último dígito hexadecimal (así una dirección absoluta de `0xd0000` puede darse como `0xd000`). Asegúrese de añadir el dígito extra de cuando haga los cambios a `config.opts`.

En casos no muy usuales, un fallo en el sondeo de memoria puede indicar un problema de configuración en la sincronización con el controlador. Revise la sección 2.3 (Opciones de inicio) para más información acerca de cómo combatir los problemas comunes de sincronización.

- `cs: warning: no high memory space available!`

Los puentes *CardBus* pueden reservar ventanas de memoria fuera del «agujero de memoria» de 640KB-1MB en la arquitectura de bus ISA. Generalmente es buena idea el configurar puentes *CardBus* para usar ventanas de memoria alta, porque es muy difícil que existan conflictos con otros dispositivos. También, las tarjetas *CardBus* pueden requerir grandes ventanas de memoria, las cuales puede ser difícil o imposible que coincidan en memoria baja. Los servicios de tarjetas preferentemente localizarán las ventanas en memoria alta para puentes *CardBus*, si ambas ventanas de memoria (alta y baja) se definen en `config.opts`. El archivo `config.opts` por omisión ahora incluye una ventana de memoria alta de `0xa0000000-0xa0ffffff`. Si tiene un puente *CardBus* y ha actualizado de una versión de PCMCIA anterior, añada esta ventana de memoria si no está ya definido.

En algunos casos, la ventana de memoria alta por omisión no se utiliza.

En algunos modelos *IBM Thinkpad*, una ventana de `0x60000000-0x60ffffff` funcionará en lugar de la ventana por omisión.

3.6 Fallo al detectar cuando se inserta o se extrae la tarjeta

Síntomas:

- Las tarjetas se detectan y configuran apropiadamente si están presentes al momento de iniciar.
- Los controladores no responden a los eventos de inserción y extracción, ya sea registrando los eventos en el registro del sistema, o emitiendo pitidos.

En muchos casos, el controlador del socket (`i82365` o `tcic`) probará automáticamente y seleccionará la interrupción apropiada para señalar cambios en el estado de la tarjeta. El tanteo automático de interrupciones no funciona con algunos controladores compatibles con Intel, incluyendo los chips Cirrus y los chips usados en IBM Thinkpads. Si un dispositivo está inactivo en el momento del sondeo, su interrupción puede parecer estar disponible. En esos casos, el controlador del socket puede usar una interrupción que es usada por otro dispositivo.

Con los controladores `i82365` y `tcic` la opción `list_irq` puede usarse para limitar las interrupciones que serán tanteadas. Esta lista limita el conjunto de interrupciones que pueden ser utilizadas por las tarjetas PCMCIA así como para monitorizar los cambios en el estado de la tarjeta. La opción `cs_irq` puede usarse también para establecer explícitamente la interrupción que será utilizada para monitorizar dichos cambios.

Si no puede encontrar un número de interrupción que funcione, hay también un estado en modo de búsqueda: ambos, `i82365` y `tcic` aceptarán una opción `poll_interval=100`, para buscar cambios en el estado de la tarjeta una vez por segundo. Esta opción puede usarse también si su sistema tiene un rango corto de interrupciones disponibles para utilizarse con tarjetas PCMCIA. Especialmente para sistemas con más de un controlador de host, hay un pequeño punto para dedicar interrupciones para monitorizar cambios de estado de la tarjeta.

Todas esas opciones deberían establecerse en la línea `PCIC_OPTS=` ya sea en `/etc/rc.d/rc.pcmcia` o en `/etc/sysconfig/pcmcia`, dependiendo de la configuración de su sistema.

3.7 Faltan recursos del sistema

Síntomas:

- Cuando se inserta una tarjeta, es identificada correctamente, pero no puede ser configurada (secuencia de pitidos agudos/graves).
- Aparecen en el registro del sistema alguno de los siguientes mensajes:

```
RequestIO: Resource in use
RequestIRQ: Resource in use
RequestWindow: Resource in use
GetNextTuple: No more items
could not allocate nn IO ports for CardBus socket n
could not allocate nnK memory for CardBus socket n
could not allocate interrupt for CardBus socket n
```

La reserva de interrupciones indica generalmente un problema con el sondeo de interrupciones, véase la sección 3.3 (Fallos en la búsqueda de interrupciones).

En algunos casos, el tanteo parece funcionar, pero únicamente aparecen una o dos interrupciones disponibles. Revise el registro de su sistema para ver si los resultados de la exploración son plausibles. Desactivar el tanteo y seleccionar las interrupciones manualmente puede ayudar.

Si el sondeo de interrupciones no está funcionando adecuadamente, el controlador del socket puede reservar una interrupción para monitorizar inserciones de tarjetas, incluso cuando las interrupciones sean demasiado escasas para esto, constituye una buena idea. En este caso, puede Vd. cambiar el controlador a modo de búsqueda estableciendo `PCIC_OPTS` a `poll_interval=100`. O, si tiene un controlador *CardBus*, intente con `pci_csc=1`, el cual selecciona una interrupción PCI (si está disponible) para cambios de estado en la tarjeta.

La reserva de puertos de E/S no es muy común, pero algunas veces tiene lugar con tarjetas que requieren regiones de espacio de E/S grandes, contiguas y alineadas, o que sólo reconocen pocas posiciones específicas de puertos. Los rangos de puertos de E/S por omisión en `/etc/pcmcia/config.opts` normalmente son suficientes, pero pueden ser extendidos. En casos extraños, la reserva puede indicar que falló el sondeo de puertos de E/S; revise la sección 3.4 (fallos en la búsqueda de puertos de E/S).

La reserva de memoria no es común tampoco con las configuraciones de la ventana de memoria que vienen por omisión en `config.opts`. Las tarjetas *CardBus* pueden requerir regiones de memoria más grandes que las tarjetas típicas de 16-bits. Dado que de que las ventanas de memoria de las tarjetas *CardBus* pueden ser mapeadas a cualquier parte del espacio de la dirección PCI del host (en lugar de sólo mapearlo al «agujero» de 640K-1MB en sistemas PC), es de utilidad especificar ventanas de memoria amplias en la memoria alta, tales como `0xa0000000-0xa0ffffff`.

3.8 Conflicto de recursos entre dos tarjetas

Síntomas:

- Dos tarjetas funcionan bien cuando se usan separadamente.
- Cuando ambas tarjetas se insertan, sólo funciona una.

Esto usualmente indica un conflicto de recursos con un dispositivo del sistema que Linux no conoce. Los dispositivos PCMCIA son configurados dinámicamente, así, por ejemplo, las interrupciones son reservadas conforme se vayan necesitando, en lugar de ser asignadas específicamente a tarjetas o sockets en particular. Dada una lista de recursos que parecen estar disponibles, las tarjetas son recursos asignados en el orden en que son configurados. En este caso, a la tarjeta configurada en último lugar se le está asignando un recurso que en efecto, no está libre.

Revise el registro del sistema para ver qué recursos están usados por la tarjeta que no funciona. Exclúyalos de `/etc/pcmcia/config.opts`, y reinicie el demonio `cardmgr` para recargar la base de datos de recursos.

3.9 No se completa la configuración de dispositivos

Síntomas:

- Cuando se inserta una tarjeta, se escucha un pitido agudo.
- Las inserciones y extracciones posteriores de tarjetas son ignoradas.

Esto indica que la tarjeta fue identificada con éxito, sin embargo, `cardmgr` fue incapaz de completar el proceso de configuración por alguna razón. La más común es que un paso en el script de configuración se ha bloqueado. Un buen ejemplo podría ser el script de red bloqueándose si una tarjeta de red se inserta sin tener presente una conexión a la red.

Para verificar el problema, puede ejecutar manualmente un script de configuración para ver dónde se está bloqueando. Los scripts están en el directorio `/etc/pcmcia`. Toman dos parámetros: un nombre de dispositivo, y una acción. El demonio `cardmgr` graba los comandos de configuración en el registro del sistema. Por ejemplo, si el registro del sistema muestra que el comando `./network start eth0` fue el último comando ejecutado por `cardmgr`, el siguiente comando puede rastrear el script:

```
sh -x /etc/pcmcia/network start eth0
```

4 Uso y características

4.1 Herramientas para configurar y monitorizar dispositivos PCMCIA

Si los módulos son todos cargados correctamente, la salida del comando `lsmod` debería verse como sigue, cuando no hay tarjetas insertadas:

```
Module                  Size  Used by
ds                      5640   2
i82365                 15452   2
pcmcia_core            30012   3  [ds i82365]
```

El registro del sistema deberá también incluir la salida del controlador del socket, describiendo el(los) controlador(es) del host encontrado(s) y el número de sockets detectados.

4.1.1 El demonio de configuración `cardmgr`

El demonio `cardmgr` es responsable de monitorizar los sockets PCMCIA, cargando los controladores cuando se necesita, y corriendo scripts a nivel de usuario en respuesta a las inserciones y extracciones de tarjetas. Graba sus acciones en el registro del sistema, y también usa pitidos para señalar cambios en el estado de las tarjetas. Los tonos de los pitidos indican el éxito o fracaso de un paso de la configuración en particular. Dos pitidos agudos indican que la tarjeta fue identificada y configurada correctamente. Un pitido agudo seguido de un pitido grave indica que la tarjeta fue identificada, pero no pudo ser configurada por alguna razón. Un pitido grave indica que la tarjeta no pudo ser identificada.

`cardmgr` registra información del dispositivo para cada socket en `/var/run/stab`

He aquí el contenido de un ejemplo de `/var/run/stab`:

```
Socket 0: Adaptec APA-1460 SlimSCSI
0       scsi    aha152x_cs      0       sda     8       0
0       scsi    aha152x_cs      1       scd0    11      0
Socket 1: Serial or Modem Card
1       serial  serial_cs       0       ttyS1   5       65
```

Para las líneas que describen dispositivos, el primer campo es el socket, el segundo es la clase del dispositivo, el tercero es nombre del controlador, el cuarto se usa para numerar múltiples dispositivos asociados con el mismo controlador, el quinto es el nombre del dispositivo, y los dos campos finales son los números mayor y menor para este dispositivo (si es aplicable).

El demonio `cardmgr` configura tarjetas basadas en una base de datos de tipos de tarjetas conocidas almacenadas en `/etc/pcmcia/config`. Este archivo describe una variedad de controladores, describe cómo identificar esas tarjetas, y cual(es) controlador(es) pertenecen a cada tarjeta. El formato de este archivo se describe en la página del manual de `pcmcia(5)`.

4.1.2 Las utilidades `cardctl` y `cardinfo`

El comando `cardctl` puede ser usado para comprobar el estado de un socket, o para ver cómo está configurado. También puede ser usado para alterar el estado de configuración de una tarjeta. He aquí un ejemplo de la salida del comando `cardctl config`:

```
Socket 0:
  not configured
Socket 1:
  Vcc = 5.0, Vpp1 = 0.0, Vpp2 = 0.0
  Card type is memory and I/O
  IRQ 3 is dynamic shared, level mode, enabled
  Speaker output is enabled
  Function 0:
    Config register base = 0x0800
      Option = 0x63, status = 0x08
    I/O window 1: 0x0280 to 0x02bf, auto sized
    I/O window 2: 0x02f8 to 0x02ff, 8 bit
```

O `cardctl ident`, para obtener información de la identificación de la tarjeta:

```
Socket 0:
  no product info available
Socket 1:
```

```
product info: "LINKSYS", "PCMLM336", "A", "0040052D6400"
manfid: 0x0143, 0xc0ab
function: 0 (multifunction)
```

Los comandos `cardctl suspend` y `cardctl resume` pueden usarse para desactivar una tarjeta sin descargar sus controladores asociados. El comando `cardctl reset` intenta resetear y reconfigurar una tarjeta. `cardctl insert` y `cardctl eject` emulan las acciones realizadas cuando una tarjeta es insertada o expulsada, incluyendo la carga y descarga de los controladores, y configurando o desactivando los dispositivos.

Si está Vd. corriendo X, `cardinfo` produce de forma gráfica el estado actual de todos los sockets PCMCIA, similar en contenido a `cardctl config`. También proporciona una interfaz gráfica para la mayoría de las otras funciones de `cardctl`.

4.1.3 Inserción y extracción de tarjetas

En teoría, puede insertar y extraer tarjetas PCMCIA en cualquier momento. Sin embargo, es una buena idea no expulsar una tarjeta que está siendo utilizada por algún programa de aplicación. Los kernels anteriores al `1.1.77` solían congelarse cuando las tarjetas serie/módem eran expulsadas, aunque esto parece estar ya solucionado.

4.1.4 Servicios de Tarjetas y Administración Avanzada de Energía

Los servicios de tarjetas pueden ser compilados con soporte para APM (*Advanced Power Management*) (En castellano: *Administración Avanzada de Energía*), si configuró su kernel con soporte APM. APM está actualmente a cargo de Stephen Rothwell, *Stephen.Rothwell@canb.auug.org.au*. El demonio `apmd` es mantenido por Avery Pennarun, *apenwarr@worldvisions.ca*), con más información disponible en *http://www.worldvisions.ca/~apenwarr/apmd/*. Los módulos PCMCIA serán configurados automáticamente para APM si es detectada una versión compatible en el sistema.

Esté APM configurado o no, puede usar `cardctl suspend` antes de suspender su portátil, y `cardctl resume` después de «despertarlo», para apagar y reactivar sus tarjetas PCMCIA. No funcionará con un módem que esté en uso, porque el controlador serie no puede guardar y restablecer los parámetros operativos del módem.

APM parece ser inestable en algunos sistemas. Si experimenta problemas con APM y PCMCIA en su sistema, intente localizar el problema en un paquete u otro antes de informar de un *bug*.

Algunos controladores, notablemente los controladores PCMCIA SCSI, no pueden recuperarse de un ciclo de suspender/despertar. Cuando se usa una tarjeta PCMCIA SCSI, use siempre `cardctl eject` antes de suspender el sistema.

4.1.5 Apagado del sistema PCMCIA

Para descargar el paquete PCMCIA completo, invoque `rc.pcmcia` con:

```
/etc/rc.d/rc.pcmcia stop
```

Este script tomará algunos segundos para ejecutarse, para darle tiempo a todos los controladores a desactivarse correctamente. Si un dispositivo está en uso actualmente, el proceso de desactivación será incompleto, y puede que algunos módulos del kernel no sean descargados. Para prevenir esto, use `cardctl eject` para desactivar todos los sockets antes de invocar `rc.pcmcia`. El estado de salida del comando `cardctl` indicará si alguno de los sockets no pudo ser desactivado.

4.2 Un vistazo a los scripts de configuración de PCMCIA

Cada dispositivo PCMCIA tiene una «clase» asociada que describe cómo debe ser configurado y manejado. Las clases están asociadas con los controladores de dispositivos en `/etc/pcmcia/config`. Actualmente hay cinco clases de dispositivos de E/S (red, SCSI, cdrom, disco, y serie) y dos clases de dispositivos de memoria (memoria y FTL). Para cada clase, hay dos scripts en `/etc/pcmcia`: un script principal de configuración (por ejemplo, `/etc/pcmcia/scsi` para dispositivos SCSI), y un script de opciones (por ejemplo, `/etc/pcmcia/scsi.options`). El script principal de un dispositivo será invocado para configurarlo cuando se inserte una tarjeta, y para desactivar el dispositivo cuando sea extraída. Para tarjetas con varios dispositivos asociados, el script será invocado para cada dispositivo.

Los scripts de configuración inician al extraer algo de información acerca del dispositivo de `/var/run/stab`. Cada script construye una «dirección de dispositivo», que únicamente describe el dispositivo que ha sido solicitado para configurar, en la variable de shell ADDRESS. Esto es pasado al script `*.opts`, el cual debe proporcionar información acerca de cómo debe ser configurado un dispositivo en esta dirección. Para algunos, la dirección del dispositivo es sólo el número de socket. Para otros, se incluye información extra que puede ser útil para decidir cómo configurar el dispositivo. Por ejemplo, los dispositivos de red pasan su dirección ethernet de hardware como parte de la dirección del dispositivo, así, el script `network.opts` puede usar esto para seleccionar diversas configuraciones.

La primera parte de todas las direcciones de dispositivos es el «esquema» PCMCIA actual. Ese parámetro es usado para soportar múltiples conjuntos de configuraciones de dispositivos basadas en una simple variable externa definida por el usuario. Una uso de los esquemas puede ser el tener un esquema de «casa», y un esquema de «trabajo», el cual puede incluir diferentes conjuntos de parámetros de configuración de red. El esquema actual se selecciona usando el comando `cardctl scheme`. Si no se define un esquema, por omisión se establece el esquema `default`.

Como regla general, cuando se configura Linux para un equipo portátil, los dispositivos PCMCIA deben ser configurados desde los scripts para dispositivos PCMCIA. No intente configurar un dispositivo PCMCIA de la misma forma en que configuraría un dispositivo conectado de forma permanente. No obstante, algunas distribuciones de Linux suministran paquetes PCMCIA que están relacionadas con las herramientas de configuración de dispositivos propios de la misma distribución. En ese caso, alguna de las siguientes secciones puede o no aplicar; idealmente, esto sería documentado por los encargados de la distribución.

4.3 Adaptadores de red PCMCIA

Las interfaces de red tipo ethernet normalmente tienen nombres como `eth0`, `eth1`, y así sucesivamente. Los adaptadores Token-Ring se manejan de forma similar, sin embargo, son llamadas comúnmente `tr0`, `tr1` y así sucesivamente. El comando `ifconfig` se usa para ver o modificar el estado de una interface de red. Una peculiaridad de Linux es que las interfaces de red no tienen archivos de dispositivo correspondientes en `/dev/`, así que no se sorprenda si no los encuentra.

Cuando se detecta una tarjeta ethernet, le será asignado el primer nombre de interface que esté libre, normalmente `eth0`. `cardmgr` ejecutará el script `/etc/pcmcia/network` para configurar la interface, la cual normalmente lee las configuraciones de red de `/etc/pcmcia/network.opts`. Los scripts `network`, y `network.opts` serán ejecutados sólo cuando su tarjeta ethernet esté presente. Si su sistema tiene la facilidad de configuración de red automática, puede o no ser PCMCIA. Consulte la documentación de su distribución de Linux y la sección 2.5 (Notas acerca de distribuciones de Linux específicas) para determinar si los dispositivos de red PCMCIA deben ser configurados con herramientas automáticas, o editando `network.opts`.

La dirección de dispositivo pasada a `network.opts` consiste en cuatro campos separados por comas: el esquema, el número de socket, la instancia de dispositivo, y la dirección ethernet de hardware de la tarjeta, La instancia de dispositivo es usada para numerar dispositivos para tarjetas que tienen varias interfaces de red, así que normalmente será 0. Si tiene varias tarjetas de red usadas para propósitos diferentes, una opción puede ser el configurar las tarjetas basadas en la posición del socket, como en:

```
case "$ADDRESS" in
*,0,*,*)
    # definiciones para tarjeta de red en el socket 0
    ;;
*,1,*,*)
    # definiciones para tarjeta de red en el socket 1
    ;;
esac
```

Alternatívamente, pueden ser configuradas usando su dirección de hardware, como en:

```
case "$ADDRESS" in
*,*,*,00:80:C8:76:00:B1)
    # definiciones para una tarjeta D-Link
    ;;
*,*,*,08:00:5A:44:80:01)
    # definiciones para una tarjeta IBM
esac
```

4.3.1 Parámetros de dispositivos de red

Los siguientes parámetros se pueden definir en `network.opts`:

IF_PORT

Especifica el tipo de transceptor ethernet, para tarjetas que no sean autodetectadas. Consulte `man ifport` para ver los nombres de los transceptores.

PUMP

Una opción booleana (y/n): indica si la dirección IP e información de rutado del host se puede obtener ya sea por BOOTP o DHCP, con el demonio `pump`.

BOOTP

Una opción booleana (y/n): indica si la dirección IP del host y su información de rutado se obtendrán usando el protocolo BOOTP, con `bootpc`.

DHCP

Un opción booleana (y/n): indica si la dirección IP del host y su información de rutado se obtendrán de un servidor DHCP, con `dhcpcd`.

IPADDR

La dirección IP para esta interface.

NETMASK, BROADCAST, NETWORK

Parámetros básicos de red: revise el COMO de red para más información.

GATEWAY

La dirección IP de una máquina pasarela para la subred de este host. Los paquetes con destinos hacia afuera de esta subred serán destinados a dicha pasarela.

DOMAIN

El nombre de dominio de la red local para este host, es usado al crear `/etc/resolv.conf`.

SEARCH

Una lista de búsqueda para búsqueda de nombres, es añadida a `/etc/resolv.conf`. DOMAIN y SEARCH son mutuamente exclusivos: revise `man resolver` para más información.

DNS_1,DNS_2,DNS_3

Nombres de host o direcciones IP para servidores de nombres para esta interface, para ser añadidos a `/etc/resolv.conf`

MOUNTS

Una lista de puntos de montaje NFS para ser montados por esta interface.

IPX_FRAME, IPX_NETNUM

Para redes IPX: el tipo de *frame* y número de red, pasado al comando `ipx_interface`.

Por ejemplo:

```
case "$ADDRESS" in
*,*,*,*)
    IF_PORT="10base2"
    BOOTP="n"
    IPADDR="10.0.0.1"
    NETMASK="255.255.255.0"
    NETWORK="10.0.0.0"
    BROADCAST="10.0.0.255"
    GATEWAY="10.0.0.1"
    DOMAIN="dominio.org"
    DNS_1="dns1.dominio.org"
    ;;
esac
```

Para montar y desmontar automáticamente sistemas de archivos NFS, primero añada todos esos sistemas de archivos a `/etc/fstab`, incluyendo `noauto` en las opciones de montaje. En `network.opts`, liste los puntos de montaje de los sistemas de archivos en la variable MOUNTS. Es especialmente importante usar ya sea `cardctl` o `cardinfo` para apagar una tarjeta de red cuando NFS se encuentre activo. No es posible desmontar limpiamente los sistemas de archivos NFS si una tarjeta de red es símplemente expulsada sin precaución.

En adición a los parámetros usuales de configuración de red, el script `network.opts` puede especificar acciones extra a tomar después de que una interface es configurada, o antes de que se apague la interface. Si `network.opts` define una función de shell llamada `start_fn`, será invocada por el script de red después de que la interface sea configurada, y el nombre de interface se pasará a la función como su primer (y único) argumento. Similarmente, si es definido, `stop_fn` se invocará antes de apagar una interfaz.

El tipo de transceptor se puede seleccionar usando la configuración `IF_PORT`. Esto puede ser, ya sea un valor numérico como en las versiones anteriores de PCMCIA, o una palabra clave que identifique el tipo de transceptor. Todos los controladores de red están configurados por omisión para autodetectar la interface si es posible, o bien, utilizar 10baseT. El comando `ifport` se puede utilizar para comprobar el tipo de transceptor actual. Por ejemplo:

```
# ifport eth0 10base2
#
# ifport eth0
eth0    2 (10base2)
```

El controlador actual (`3.0.10` o posterior) de `3c589` debe autodetectar rápidamente los cambios de transceptor en cualquier momento. Las primeras versiones del controlador 3x589 tenían un algoritmo de autodetección de transceptores algo lento y no muy amistoso. Para esas versiones, el cable de red apropiado debe ser conectado a la tarjeta cuando la tarjeta es configurada, o se puede forzar la autodetección con:

```
ifconfig eth0 down up
```

4.3.2 Comentarios acerca de tarjetas específicas

- Con las tarjetas *IBM CCAE* y *Socket EA*, el tipo de transceptor (10base2, 10baseT, AUI), necesita configurarse cuando el dispositivo de red está configurado. Asegúrese de que el tipo de transceptor que aparece en el registro del sistema concuerda con su conexión.
- Los controladores para tarjetas SMC, Megahertz, Ositech, y 3Com deben autodetectar el tipo de red conectada (10base2 o 10baseT). Establecer el tipo de transceptor cuando se carga el controlador sirve para definir la «primera búsqueda» de la tarjeta.
- La *Farallon EtherWave* actualmente está basada en la *3Com 3c589*, con un transceptor especial. Aunque la EtherWave usa conexiones al estilo 10baseT, su transceptor requiere que la 3c589 sea configurada en modo 10base2.
- Si tiene problemas con un adaptador *IBM CCAE*, *NE4100*, *Thomas Conrad*, o Kingston, pruebe a incrementar el tiempo de acceso con la opción `mem_speed=#` al módulo `pcnet_cs`. Un ejemplo de cómo hacer esto se muestra en el archivo `config.opts`. Pruebe con velocidades por encima de `1000` (en nanosegundos).
- Para el adaptador *New Media Ethernet*, en algunos sistemas, puede ser necesario incrementar el tiempo de acceso al puerto de E/S con la opción `io_speed=#` cuando se cargue el módulo `pcmcia_core`. Edite `CORE_OPTS` en el script de inicio para activar esta opción.
- El soporte multicast en el controlador *New Media Ethernet* está incompleto. El último controlador funcionará con kernels multicast, pero ignorará los paquetes multicast. El modo promiscuo debe funcionar apropiadamente.
- El controlador usado por los controladores token ring IBM y 3Com parecen comportarse bastante mal si las tarjetas no están conectadas cuando son inicializadas. Conecte siempre esas tarjetas a la red antes de activarlas. Si `ifconfig` informa que la dirección de harware como todo `0`, esto debe ser debido a un problema de configuración de la ventana de memoria.
- Algunas tarjetas Linksys, D-Link, e IC-Card 10baseT/10base2 tienen una forma única de seleccionar el tipo de transceptor que no es manejado por los controladores de Linux. Una solución es arrancar DOS y utilizar la utilidad proporcionada por el fabricante para seleccionar el transceptor, haciendo entonces un arranque «en caliente» de Linux. Alternativamente, hay una utilidad Linux para realizar esta función, que está disponible en *ftp://hyper.stanford.edu/pub/pcmcia/extras/dlport.c*.
- Para adaptadores de red inalámbricos WaveLAN, Jean Tourrilhes, *jt@hpl.hp.com*) tiene disponible el *Wireless HOWTO* (Cómo inalámbrico) en *http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/*

4.3.3 Diagnóstico de problemas con adaptadores de red

- ¿Es reconocida su tarjeta como una tarjeta ethernet? Revise el registro del sistema y asegúrese de que `cardmgr` identifique la tarjeta correctamente e inicia uno de los controladores de red. Si no lo hace, su tarjeta puede ser utilizable todavía si es compatible con una tarjeta soportada. Esto es posible hacerlo fácilmente si la tarjeta dice ser *NE2000 compatible*.
- ¿Está configurada la tarjeta apropiadamente? Si está usando una tarjeta soportada, y fue reconocida por `cardmgr`, pero todavía no funciona, pudo ser un conflicto de interrupción o puerto con otro dispositivo. Determine qué recursos está utilizando la tarjeta (en el registro del sistema), e intente de nuevo excluyéndolos en `/etc/pcmcia/config.opts` para forzar a la tarjeta a usar otros.
- Si su tarjeta parece estar configurada adecuadamente, pero a veces se congela, particularmente bajo carga alta, puede ser que necesite intentar cambiar los parámetros de sincronización de su controlador del socket. Revise la sección 2.3 (Opciones de Inicio) para más información.
- Si obtiene mensajes de `network unreachable` cuando intenta acceder a la red, la información especificada en `/etc/pcmcia/network.opts` es incorrecta. Este mensaje es una indicación absolutamente a prueba de tontos de que hay un error de rutado. Por otra parte, las tarjetas mal configuradas normalmente fallarán silenciosamente.
- Para diagnosticar problemas en `/etc/pcmcia/network.opts`, empiece tratando de hacer `ping` a otros sistemas en la misma subred usando sus direcciones IP. Trate entonces de hacer `ping` a su puerta de enlace o «pasarela» (*gateway*), y a máquinas en otras subredes. Debe ser posible hacer `ping` a las máquinas por su nombre si lleva a cabo dichas pruebas con éxito.
- Asegúrese que su problema sea PCMCIA. Puede ser muy útil comprobar si la tarjeta funciona correctamente bajo DOS con los controladores del fabricante. Verifique bien sus modificaciones al script `/etc/pcmcia/network.opts`. Asegúrese que su cable, conector «T», terminador, etc. estén funcionando.

4.4 Dispositivos serie PCMCIA y módems

Los dispositivos serie de Linux son gestionados por medio de los archivos de dispositivo especiales `/dev/ttyS*` y `/dev/cua*`. En los kernels `pre-2.2` los dispositivos `ttyS*` eran para conexiones entrantes, como módems. El uso de dispositivos `cua*` se desaprueba en los kernels actuales, y se puede usar `ttyS*` para todas las aplicaciones. La configuración de un dispositivo serie se puede examinar y modificar con el comando `setserial`.

Cuando se detecta una tarjeta serie o módem, se le asignará el primer slot de dispositivo serie que se encuentre disponible. Este será usualmente `/dev/ttyS1 (cua1)` o `/dev/ttyS2 (cua2)`, dependiendo del número de puertos serie que tenga. El dispositivo `ttyS*` es el que aparecerá en `/var/run/stab`. El script de opciones por omisión para dispositivos serie, `/etc/pcmcia/serial.opts`, enlazará el dispositivo a `/dev/modem` por conveniencia. Para los kernels `pre-2.2`, el enlace se hace al dispositivo `cua*`.

No intente usar `/etc/rc.d/rc.serial` para configurar un módem PCMCIA. Este script sólo debería ser utilizado para configurar dispositivos no extraíbles. Modifique `/etc/pcmcia/serial.opts` si quiere hacer algo especial para configurar su módem. No intente tampoco cambiar las configuraciones de E/S y puerto de un dispositivo serie utilizando `setserial`. Esto podría decir al controlador serie que busque al dispositivo en un lugar diferente, pero no cambiar cómo el hardware de la tarjeta está configurado actualmente. El script de configuración serie le permite especificar otras opciones para `setserial`, así como si se debe añadir una línea a `/etc/inittab` para este puerto.

La dirección del dispositivo pasada a `serial.opts` tiene tres campos separados por comas: el primero es el esquema, el segundo es el número de socket, y el tercero es la instancia del dispositivo. La instancia del dispositivo puede tomar varios valores para tarjetas que soporten múltiples puertos serie, pero para tarjetas de un sólo puerto, siempre será `0`. Si comunmente usa más de un módem, puede especificar diferentes configuraciones basadas en la posición del socket, como en:

```
case "$ADDRESS" in
*,0,*)
    # Opciones para un modem en el socket 0
    LINK=/dev/modem0
    ;;
*,1,*)
    # Opciones para un modem en el socket 1
    LINK=/dev/modem1
    ;;
esac
```

Si un módem PCMCIA ya está configurado cuando Linux arranca, puede ser identificado incorrectamente como un puerto serie ordinario. Esto es inofensivo, sin embargo, cuando los controladores PCMCIA toman el control del módem, se le asignará un slot de dispositivo diferente. Por ello es mejor, ya sea analizar `/var/run/stab` o usar `/dev/modem`, en lugar de indicar que este módulo debe recargarse. Edite la entrada del dispositivo serie, de modo que se lea:

```
device "serial_cs"
  class "serial" module "misc/serial", "serial_cs"
```

4.4.1 Parámetros de dispositivos serie

Los siguientes parámetros se pueden definir en `serial.opts`:

LINK

Especifica una ruta para un enlace simbólico a crear al dispositivo *callout* (para llamar hacia el exterior) (ejemplo, `/dev/cua*` para kernels pre-2.2 o `/dev/ttyS*` para kernels `2.2.x`).

SERIAL_OPTS

Especifica las opciones que se pasan al comando `setserial`.

INITTAB

Si se especifica, se usará para añadir una entrada `inittab` para el dispositivo.

Por ejemplo:

```
case "$ADDRESS" in
*,*,*,*)
    LINK="/dev/modem"
    SERIAL_OPTS=""
    INITTAB="/sbin/getty"
```

4.4.2 Diagnóstico de problemas con dispositivos serie

- ¿Se reconoce su tarjeta como un módem? Revise el registro del sistema y asegúrese que `cardmgr` identifica la tarjeta correctamente e inicia el controlador `serial_cs`. Si no, necesitará añadir una nueva entrada en el fichero `/etc/pcmcia/config` para que pueda ser identificado apropiadamente. Consulte la sección 6.1 (Configuración de tarjetas no reconocidas) para más detalles.

- ¿Es el módem configurado satisfactoriamente por `serial_cs`? Nuevamente, revise el registro del sistema y busque los mensajes del controlador `serial_cs`. Si ve mensajes como `register_serial() failed` debe tener un conflicto de puerto de E/S con otro dispositivo. Otra causa de conflictos tiene lugar cuando el dispositivo es reconocido como una *UART 8250*; la mayoría de módems modernos deben identificarse como UART 16550A. Si piensa que está viendo un conflicto de puertos, edite `/etc/pcmcia/config.opts` y excluya el rango de puertos que fue reservado para el módem.
- ¿Hay un conflicto de interrupciones? Si el registro del sistema se parece normal, pero el módem no funciona, pruebe a cambiar la irq a `0` usando `setserial` y comprobar si el módem funciona. Esto causa que el controlador serie use un modo de búsqueda más bajo en lugar de usar interrupciones. Si esto parece solucionar el problema, es probable que otro dispositivo del sistema esté usando la interrupción seleccionada por `serial_cs`. Deberá añadir una línea a `/etc/pcmcia/config.opts` para excluir esta interrupción.
- Si el módem parece funcionar muy, muy lento, esto es casi un indicador seguro de un conflicto de interrupciones. Asegúrese que su problema sea realmente PCMCIA. Puede ayudarle comprobar si la tarjeta funciona bajo DOS con los controladores del fabricante. Así mismo, evite probar la tarjeta con algo complicado como SLIP o PPP hasta que esté seguro que haga conexiones simples. Si es capaz de establecer «conexiones simples», pero no con SLIP, su problema es más probable que tenga que ver con SLIP, y no con PCMCIA.
- Si obtiene mensajes del kernel indicando que el módulo `serial_cs` no puede cargarse, significa que su kernel no tiene soporte para dispositivo serie. Si ha compilado el controlador serie como módulo, debe modificar `/etc/pcmcia/config` para indicar que el módulo serie debe cargarse antes de `serial_cs`.

4.5 Dispositivos PCMCIA de puerto paralelo

El controlador de puerto paralelo de Linux está estructurado por capas, así que varios tipos de dispositivos de alto nivel pueden compartir el mismo controlador de puerto de bajo nivel. Los dispositivos se gestionan a través de los archivos especiales de dispositivo `/dev/lp*`. La configuración de un dispositivo de impresora puede examinarse y modificarse con el comando `tunelp`.

El módulo `parport_cs` depende de los controladores `parport` y `parport_pc`, los cuales pueden ser compilados dentro del kernel o bien compilados como módulos. La estructura del controlador por capas significa que cualquiera de los controladores paralelos de alto nivel (tales como el controlador `plip`, el controlador de impresora, etc.) deben ser compilados como módulos. Estos controladores sólo reconocen dispositivos de puerto paralelo en el momento de iniciar el módulo, así que pueden cargarse después de que cualquier dispositivo paralelo PC Card sea configurado.

La dirección del dispositivo pasada a `parport.opts` tiene tres campos separados por comas: el primero es el esquema, el segundo es el número de socket, y el tercero es la instancia del dispositivo. La instancia del dispositivo puede tomar varios valores para tarjetas que soportan múltiples puertos paralelos, pero para tarjetas de un solo puerto, siempre será `0`. Si usa habitualmente más de una tarjeta, necesitará especificar diferentes configuraciones basadas en la posición del socket, como en:

```
case "$ADDRESS" in
*,0,*)
    # Opciones para una tarjeta en el socket 0
    LINK=/dev/printer0
    ;;
*,1,*)
    # Opciones para una tarjeta en el socket 1
    LINK=/dev/printer1
```

```
        ;;
    esac
```

Si configura el kernel para cargar el controlador básico de puerto paralelo como módulo, debe editar `/etc/pcmcia/config` para indicar qué módulos necesitan cargarse. Edite la entrada para el dispositivo paralelo de modo que se lea:

```
device "parport_cs"
  class "parport" module "misc/parport", "misc/parport_pc", "parport_cs"
```

4.5.1 Parámetros de dispositivos paralelos

Los siguientes parámetros pueden especificarse en `parport.opts`:

LINK
: Especifica la ruta del enlace simbólico a crear hacia el puerto de impresora.

LP_OPTS
: Especifica las opciones a pasar al comando `tunelp`.

Por ejemplo:

```
case "$ADDRESS" in
*,*,*,*)
    LINK="/dev/printer"
    LP_OPTS=""
```

4.5.2 Diagnóstico de problemas con dispositivos de puertos paralelos

- ¿Hay un conflicto de interrupciones? Si el registro del sistema parece estar bien, pero el puerto no funciona, cambie la irq a `0` usando `tunelp`, y compruebe si las cosas mejoran. Esto cambia el controlador a modo de búsqueda. Si parece solucionar el problema, es probable que otro dispositivo en su sistema esté utilizando la interrupción seleccionada por `parport_cs`. Deberá añadir una línea a `/etc/pcmcia/config.opts` para excluir esta interrupción.
- Si su kernel genera mensajes indicando que el módulo `parport_cs` no puede cargarse, significa que el kernel no tiene soporte para dispositivos paralelos. Si tiene compilado el controlador paralelo como módulo, necesita modificar `/etc/pcmcia/config` para indicar que los módulos `parport` y `parport_pc` deben cargarse antes que `parport_cs`.

4.6 Adaptadores SCSI PCMCIA

Todos los controladores que dan soporte actualmente a tarjetas SCSI PCMCIA son trabajos basados en alguna de las siguientes tarjetas bus ISA: *Qlogic*, *Adaptec AHA-152X*, o *Future Domain TMC-16x0*. Los controladores PCMCIA son compilados enlazando parcialmente código específico PCMCIA (en `qlogic_cs.c`, `toaster_cs.c`, o `fdomain_cs.c`) con el controlador SCSI normal de Linux. Debido a las limitaciones en el modelo del controlador SCSI de Linux, sólo se soporta una tarjeta extraíble por controlador.

Cuando se detecta un nuevo adaptador SCSI, los controladores SCSI sondearán la presencia de dispositivos. Revise el registro del sistema para asegurar que los dispositivos sean detectado apropiadamente. Los nuevos dispositivos SCSI se asignarán a los primeros archivos de dispositivo SCSI disponibles. El primer disco SCSI será `/dev/sda`, la primera cinta SCSI será `/dev/st0`, y el primer CD-ROM será `/dev/scd0`.

En `/var/run/stab` se muestra una lista de los dispositivos conectados a este adaptador, y el script de configuración `/etc/pcmcia/scsi` se llamará una vez para cada dispositivo conectado, ya sea para configurar o apagar ese dispositivo. El script por omisión no toma ninguna acción para configurar dispositivos SCSI, pero desmontará apropiadamente los sistemas de archivos en dispositivos SCSI cuando se extraiga la tarjeta.

Las direcciones de dispositivo que se pasan a `scsi.opts` son complicadas, debido a la variedad de cosas que pueden conectarse a un adaptador SCSI. Las direcciones consisten de de seis o siete campos separados por comas: el esquema actual, el tipo de dispositivo, el número de socket, el canal SCSI, ID, y el número lógico de unidad, y opcionalmente, el número de partición. El tipo de dispositivo será `sd` para discos, `st` para cintas, `sr` para unidades de CD-ROM, y `sg` para dispositivos SCSI genéricos. Para la mayoría de configuraciones, la unidad lógica y el canal SCSI serán `0`. Para unidades de disco con varias particiones, `scsi.opts` se llamará primero para toda la unidad, con direcciones de cinco campos. El script deberá establecer la variable PARTS una lista de particiones. Entonces, `scsi.opts` será llamado para cada partición, con las direcciones más largas, de siete campos.

Si su kernel no tiene un controlador de alto nivel (disco, cinta, etc) para un dispositivo SCSI en particular, entonces no será configurado por los controladores PCMCIA. Como efecto lateral, el nombre del dispositivo en `/var/run/stab` será algo como `sd#nnnn` donde `nnnn` es un número hexadecimal de cuatro dígitos. Esto pasa cuando `cardmgr` no puede traducir una ID de un dispositivo SCSI a su nombre de dispositivo correspondiente en Linux.

Es posible modularizar los controladores SCSI de alto nivel para que puedan cargarse según demanda. Para hacerlo, necesita editar `/etc/pcmcia/config` para decirle a `cardmgr` qué módulos extra necesitan ser cargados cuando sea configurado su adaptador. Por ejemplo:

```
device "aha152x_cs"
  class "scsi" module "scsi/scsi_mod", "scsi/sd_mod", "aha152x_cs"
```

Especificaría que se cargase el módulo principal SCSI y el módulo controlador de disco antes de cargar el módulo controlador PCMCIA normal. El script `Configure` de PCMCIA no detectará automáticamente módulos SCSI modularizados, así que necesitará usar la opción de configuración manual para habilitar el soporte SCSI.

Encienda siempre los dispositivos SCSI antes de encender su portátil, o antes de insertar la tarjeta adaptadora, para que el bus SCSI esté listo cuando el adaptador se configure. También hay que ser muy cuidadoso al expulsar un adaptador SCSI. Asegúrese que todos los dispositivos SCSI asociados sean desmontados y cerrados antes de expulsar la tarjeta. La mejor forma de asegurar esto es usar `cardctl` o `cardinfo` para solicitar que se desactive la tarjeta antes de expulsarla físicamente. Por ahora, todos los dispositivos SCSI deberán encenderse antes de conectar un adaptador SCSI, y deberán permanecer conectados hasta que desconecte el adaptador y/o apague su portátil.

Hay una complicación potencial cuando se usan tarjetas que no se presentan con adaptadores de bus ISA ordinarios. El bus SCSI transporta una señal *termination power* (corriente de terminación) que se necesita para que los terminadores pasivos SCSI ordinarios funcionen apropiadamente. Los adaptadores PCMCIA SCSI no suministran corriente de terminación, así que si se requiere, deberá proporcionarlo el dispositivo externo. Algunos dispositivos externos SCSI deben configurarse para suministrarlo. Otros, como el *Iomega Zip* y el *Syquest EZ*, usan terminadores activos que no dependen de ello. En algunos casos, puede ser necesario usar un bloque terminador especial como el *APS SCSI Sentry 2*, el cual tiene una fuente de alimentación externa. Cuando configure la entrada para el dispositivo SCSI, hágalo teniendo en cuenta si alguno de sus dispositivos requieren o pueden suministrar corriente de terminación o no.

4.6.1 Parámetros de dispositivos SCSI

Los siguientes parámetros pueden ser especificados en `scsi.opts`:

DO_FSTAB

Es una opción booleana (y/n): Especifica si se debe añadir una entrada `/etc/fstab` para este dispositivo.

DO_FSCK

Es una opción booleana (y/n): Especifica si se debe comprobar este dispositivo antes de ser montado, con `fsck -Ta`.

DO_MOUNT

Es una opción booleana (y/n): Especifica si este dispositivo debe montarse automáticamente al momento de insertar la tarjeta.

FSTYPE, OPTS, MOUNTPT

El tipo de sistema de archivos, opciones de montaje, y punto de montaje que se utilizarán para la entrada en `fstab` y/o para montar el dispositivo.

Por ejemplo, un script para configurar una unidad de disco en SCSI ID 3, con dos particiones, y un CD-ROM en SCSI ID 6:

```
case "$ADDRESS" in
*,sd,*,0,3,0)
    # Este dispositivo tiene dos particiones...
    PARTS="1 2"
    ;;
*,sd,*,0,3,0,1)
    # Opciones para la particion 1:
    #  actualizar /etc/fstab, y montar un sistema de archivos ext2 en /usr1
    DO_FSTAB="y" ; DO_FSCK="y" ; DO_MOUNT="y"
    FSTYPE="ext2"
    OPTS=""
    MOUNTPT="/usr1"
    ;;
*,sd,*,0,3,0,2)
    # Opciones para la partición 2:
    #  actualizar /etc/fstab, y montar un sistema de archivos MS-DOS en /usr2
    DO_FSTAB="y" ; DO_FSCK="y" ; DO_MOUNT="y"
    FSTYPE="msdos"
    OPTS=""
    MOUNTPT="/usr2"
    ;;
*,sr,*,0,6,0)
    # Opciones para un CD-ROM en SCSI ID 6
    PARTS=""
    DO_FSTAB="y" ; DO_FSCK="n" ; DO_MOUNT="y"
    FSTYPE="iso9660"
    OPTS="ro"
    MOUNTPT="/cdrom"
    ;;
esac
```

4.6.2 Comentarios acerca de tarjetas específicas

- La tarjeta *Adaptec APA-1480 CardBus* necesita una ventana de puerto de E/S grande (256 puertos contiguos alineados en un límite de 256 puertos). Puede que sea necesario incluir las regiones de los puertos de E/S en `/etc/pcmcia/config.opts` para garantizar que cada ventana pueda encontrarse.
- No está soportado el adaptador *Adaptec APA-460 SlimSCSI*. Esta tarjeta se vendió originalmente bajo el nombre de Trantor, y cuando Adaptec se unió a Trantor, continuaron vendiendo la tarjeta Trantor con etiqueta Adaptec. La *APA-460* no es compatible con ningún controlador de Linux existente.
- He sido informado de la mala interacción entre la tarjeta *New Media Bus Toaster* y un scanner *UMAX Astra 1200s*. Debido a la complejidad del protocolo SCSI, cuando se diagnostican problemas con dispositivos SCSI, es digno de considerar que combinaciones incompatibles como esta pueden existir y no pueden documentarse.

4.6.3 Diagnóstico de problemas con adaptadores SCSI

- Con el controlador `aha152x_cs` (usado por Adaptec, New Media, y algunos más), parece que el soporte SCSI de conexión/reconexión constituye una fuente de problemas frecuentes con dispositivos de cinta. Para desactivar esta «característica», añada lo siguiente a `/etc/pcmcia/config.opts`:

  ```
  module "aha152x_cs" opts "reconnect=0"
  ```

- Con el controlador `aha152x_cs`, ciertos dispositivos parecen requerir un tiempo de espera de inicio más grande, controlado con el parámetro `reset_delay` del módulo. La unidad CDR *Yamaha 4416S* es uno de esos dispositivos. El resultado es que el dispositivo es identificado sin problemas, y luego se congela el sistema. En esos casos, pruebe:

  ```
  module "aha152x_cs" opts "reset_delay=500"
  ```

- Otra fuente potencial de problemas en el sondeo de dispositivos SCSI es el tanteo de LUNs múltiples. Si ve que la detección de un dispositivo es realizada sin problemas, seguida de «timeouts» del bus SCSI cuando se sondea el LUN 1 para ese dispositivo, debe desactivar la opción `CONFIG_SCSI_MULTI_LUN` del kernel.
- Si tiene compilado el soporte SCSI modularmente (`CONFIG_SCSI` es `m`), debe modificar `/etc/pcmcia/config` para cargar los módulos SCSI antes de que se cargue el controlador `*_cs` apropiado.
- Si obtiene mensajes de tipo `aborting command due to timeout` (abortando el comando debido a timeout), cuando se sondea el bus SCSI, es muy probable que tenga un conflicto de interrupciones.
- Si el controlador del host avisa `no SCSI devices found` (no se han encontrado dispositivos SCSI), verifique que el kernel fue compilado con los controladores SCSI de alto nivel apropiados para sus dispositivos (por ejemplo, disco, cinta, CD-ROM, y/o genéricos). Si falta un controlador de alto nivel, los dispositivos de ese tipo se ignorarán.

4.7 Tarjetas de memoria PCMCIA

El controlador `memory_cs` maneja todos los tipos de tarjetas de memoria, y también proporciona acceso directo al espacio de la dirección de memoria PCMCIA para tarjetas que tienen otras funciones. Cuando se carga, crea una combinación de dispositivos de caracteres y de bloques. Revise la página del manual del módulo para ver una descripción completa del esquema de nombres de estos dispositivos. Los dispositivos de bloques se usan para tener acceso a disco (creando y montando sistemas de archivos, etc.). Los dispositivos de caracteres son para lecturas en bruto (que no se procesan) que no se guardan en el buffer y son escritas en posiciones arbitrarias.

La dirección de dispositivo que se pasa a `memory.opts` consiste de dos campos: el esquema, y el número de socket. Las opciones se aplican a la primera partición de memoria común en la tarjeta correspondiente.

Algunas tarjetas de memoria antiguas, y la mayoría de las tarjetas de RAM simple estática, carecen de *Card Information Structure, CIS* (Estructura de Información de Tarjeta), que es el esquema que las tarjetas PCMCIA usan para identificarse a si mismas. Normalmente, `cardmgr` asumirá que una tarjeta que carece de CIS es una tarjeta de memoria simple, y cargará el controlador `memory_cs`. Por tanto, un efecto lateral es que otros tipos de tarjetas pueden detectarse erróneamente como tarjetas de memoria.

El controlador `memory_cs` usa un algoritmo heurístico para determinar la capacidad de esas tarjetas. Este algoritmo no funciona con tarjetas protegidas contra escritura, y puede cometer errores en algunos otros casos. Si una tarjeta se configura de forma errónea, su tamaño puede especificarse explícitamente cuando se haga uso de los comandos `dd` o `mkfs`.

4.7.1 Parámetros de dispositivos de memoria

DO_FSTAB

Es una opción booleana (y/n): Especifica si se debe añadir una entrada `/etc/fstab` para este dispositivo.

DO_FSCK

Es una opción booleana (y/n): Especifica si se debe comprobar este dispositivo antes de ser montado, con `fsck -Ta`.

DO_MOUNT

Es una opción booleana (y/n): Especifica si este dispositivo debe montarse automáticamente en el momento de insertar la tarjeta.

FSTYPE, OPTS, MOUNTPT

El tipo de sistema de archivos, opciones de montaje, y punto de montaje que se utilizarán para la entrada en `fstab` y/o para montar el dispositivo.

He aquí un ejemplo de un script que montará automáticamente las tarjetas de memoria basándose en el socket en que estén insertadas:

```
case "$ADDRESS" in
*,0,0)
    # Montar sistema de archivos, pero no actualizar /etc/fstab
    DO_FSTAB="n" ; DO_FSCK="y" ; DO_MOUNT="y"
    FSTYPE="ext2" ; OPTS=""
    MOUNTPT="/mem0"
    ;;
*,1,0)
    # Montar sistema de archivos, pero no actualizar /etc/fstab
    DO_FSTAB="n" ; DO_FSCK="y" ; DO_MOUNT="y"
    FSTYPE="ext2" ; OPTS=""
    MOUNTPT="/mem1"
    ;;
esac
```

4.7.2 Uso de tarjetas de memoria flash

La dirección de dispositivo que se pasa a `ftl.opts` consiste en tres o cuatro campos: el esquema, el número de socket, el número de región, y opcionalmente, el número de partición. La mayoría de tarjetas flash tienen sólo una región de memoria flash, así que el número de región será generalmente cero siempre.

Para usar una tarjeta de memoria flash como un dispositivo de bloques del tipo de un disco ordinario, primero se crea una partición FTL, o *flash translation layer*, en el dispositivo por medio del comando `ftl_format`. Esta capa oculta los detalles específicos de dispositivo de la programación de la memoria flash y hace que la tarjeta se vea como un simple dispositivo de bloques. Por ejemplo:

```
ftl_format -i /dev/mem0c0c
```

Nótese que este comando accede a la tarjeta por medio de la interface *raw* de la tarjeta de memoria. Una vez formateada, la tarjeta puede tratarse como un dispositivo de bloques ordinario por medio del controlador `ftl_cs`. Por ejemplo:

```
mke2fs /dev/ftl0c0
mount -t ext2 /dev/ftl0c0 /mnt
```

La nomenclatura de dispositivos FTL es difícil. Los números menores de los dispositivos tienen tres partes: el número de tarjeta, el número de región en esa tarjeta, y opcionalmente, la partición dentro de esa región. Una región puede ser tratada como un simple dispositivo de bloques sin tabla de partición (como un disquete), o puede particionarse como un disco duro. El dispositivo `ftl0c0` es la tarjeta `0`, región de memoria común `0`, la región entera. Los dispositivos de `ftl0c0p1` a `ftl0c0p4` son primariamente las particiones de `1` a `4` si la región ha sido particionada.

Hay dos formatos mayores para tarjetas de memoria flash: el estilo *FTL*, y el sistema de archivos *Microsoft Flash*. El formato FTL es generalmente más flexible porque permite que pueda utilizarse cualquier sistema de archivos de alto nivel en una tarjeta flash como si fuera un dispositivo de disco ordinario. El *FFS* es un tipo sistema de archivos completamente diferente. Linux no puede manejar actualmente tarjetas formateadas con *FFS*.

Las tarjetas flash *Intel Series 100* usan el primer bloque flash de 128k para almacenar la información de la configuración de la tarjeta. Para prevenir el borrado accidental de esta información, `ftl_format` automáticamente detectará esto y saltará al primer bloque cuando se cree una partición FTL.

4.8 Tarjetas PCMCIA para unidades ATA/IDE

El soporte para unidades ATA/IDE se basa en el controlador IDE regular del kernel. La parte específica PCMCIA del controlador es `ide_cs`. Asegúrese de usar `cardctl` o `cardinfo` para apagar la tarjeta ATA/IDE antes de expulsarla, porque el controlador no fue programado a prueba de extracción en caliente.

La dirección de dispositivo que se pasa a `ide.opts` consiste de tres o cuatro campos: el esquema actual, el número de socket, el número de serie de la unidad, y un número opcional de partición. El comando `ide_info` puede usarse para obtener el número de serie del dispositivo IDE. Tal y como sucede con los dispositivos SCSI, `ide.opts` se llama primero para el dispositivo entero. Si `ide.opts` retorna una lista de particiones en la variable `PARTS`, el script entonces se llamará para cada partición.

4.8.1 Parámetros para discos ATA/IDE

Los siguientes parámetros se pueden especificar en `ide.opts`:

DO_FSTAB

Es una opción booleana (y/n): Especifica si se debe añadir una entrada `/etc/fstab` para este dispositivo.

DO_FSCK

Es una opción booleana (y/n): Especifica si se debe comprobar este dispositivo antes de ser montado, con `fsck -Ta`.

DO_MOUNT

Es una opción booleana (y/n): Especifica si este dispositivo debe montarse automáticamente al momento de insertar la tarjeta.

FSTYPE, OPTS, MOUNTPT

El tipo de sistema de archivos, opciones de montaje, y punto de montaje que se utilizarán para la entrada en `fstab` y/o para montar el dispositivo.

He aqui un ejemplo del archivo `ide.opts` para montar la primera partición de cualquier tarjeta ATA/IDE en `/mnt`.

```
case "$ADDRESS" in
*,*,*,1)
    DO_FSTAB="y" ; DO_FSCK="y" ; DO_MOUNT="y"
    FSTYPE="msdos"
    OPTS=""
    MOUNTPT="/mnt"
    ;;
*,*,*)
    PARTS="1"
    ;;
esac
```

4.8.2 Diagnóstico de problemas con adaptadores ATA/IDE

- Algunas unidades IDE violan la especificación PCMCIA al requerir un tiempo mayor para iniciar que el máximo permitido para la configuración de la tarjeta. Desde la versión `3.0.6`, el controlador `ide_cs` automáticamente intentará sondear el dispositivo para darle tiempo de iniciarlos. Con los controladores antiguos, necesita cargar el módulo `pcmcia_core` con:

  ```
  CORE_OPTS="unreset_delay=400"
  ```

- Para usar una unidad de CD-ROM ATA/IDE, el kernel debe compilarse con `CONFIG_BLK_DEV_IDECD` activado. Normalmente será el caso para los kernels estándar, sin embargo es bueno estar enterado por si compila un kernel personalizado.

4.9 Tarjetas multifunción

Se puede compartir una simple interrupción entre varios controladores, como el controlador serie y el controlador ethernet: en efecto: la especificación PCMCIA requiere que todas las funciones de las tarjetas compartan la misma interrupción. Normalmente, todas las funciones de las tarjetas están disponibles sin tener que intercambiar controladores.

El uso simultáneo de dos funciones de tarjetas es algo «difícil» y varios fabricantes de hardware han implementado el compartir interrupciones en sus propias formas incompatibles (y a veces propietarias). Los controladores para algunas tarjetas (Ositech Jack de Diamond, 3Com 3c562, Linksys) soportan de forma apropiada el acceso simultáneo, pero otras (Megahertz en particular) no.

Los kernels antiguos no soportan el compartir interrupciones entre diferentes controladores de dispositivos, así que no es posible para los controladores PCMCIA el configurar esta tarjeta para acceso simultáneo ethernet y módem. Los controladores ethernet y serie se cargan automáticamente. Sin embargo, el controlador ethernet por omisión «posee» la interrupción de la tarjeta. Para usar el módem, puede descargar el controlador ethernet y reconfigurar el puerto serie haciendo algo como:

```
ifconfig eth0 down
rmmod 3c589_cs
setserial /dev/modem autoconfig auto_irq
setserial /dev/modem
```

El segundo `setserial` debe verificar que el puerto ha sido configurado para usar la interrupción que previamente utilizaba el controlador ethernet.

5 Temas avanzados

5.1 Apartado de recursos para dispositivos PCMCIA

En teoría, no debe importar qué interrupción se reserva para cada dispositivo, mientras dos dispositivos no sean configurados para usar la misma interrupción.

En `/etc/pcmcia/config.opts` encontrará un lugar para excluir las interrupciones que son usadas por dispositivos no PCMCIA.

De igual modo, no hay forma de especificar directamente las direcciones de E/S que va a utilizar una tarjeta. El archivo `/etc/pcmcia/config.opts` permite especificar rangos de puertos disponibles para ser usados por una tarjeta cualquiera, o para excluir rangos que causan conflictos con otros dispositivos.

Después de modificar `/etc/pcmcia/config.opts`, puede reiniciar `cardmgr` con `kill -HUP`.

La interrupción que se utiliza para monitorizar el estado de la tarjeta se determina por el módulo controlador de bajo nivel del socket (`i82365` o `tcic`) antes de que `cardmgr` pase a `/etc/pcmcia/config`, así no se ve afectado con los cambios a este archivo. Para establecer esta interrupción, use la opción `cs_irq=` cuando se cargue el controlador del socket, estableciendo la variable `PCIC_OPTS` en `/etc/rc.d/rc.pcmcia`

Todos los controladores de tarjetas tienen un parámetro llamado `irq_list` para especificar qué interrupciones pueden intentar reservar. Dichas opciones deben establecerse en el archivo `/etc/pcmcia/config`. Por ejemplo:

```
device "serial_cs"
  module "serial_cs" opts "irq_list=8,12"
  ...
```

debe especificarse que el controlador serie debe utilizar sólo la irq 8 o la 12. Sin importar las configuraciones de `irq_list`, los Servicios de Tarjetas nunca reservarán una interrupción que ya esté siendo usada por otro dispositivo, o una interrupción que esté excluida en el archivo de configuración.

5.2 Cómo puedo separar configuraciones de los dispositivos para casa y el trabajo?

Esto es bastante fácil con el soporte de «esquemas». Usando dos esquemas de configuración, llamados `casa` y `trabajo`. He aquí un ejemplo del script `network.opts` con configuraciones específicas de esquemas:

```
case "$ADDRESS" in
trabajo,*,*,*)
    # definiciones para la tarjeta de red en el esquema trabajo
    ...
    ;;
casa,*,*,*|default,*,*,*)
    # definiciones para la tarjeta de red en el esquema casa
    ...
    ;;
esac
```

La primera parte de una dirección de dispositivo siempre es la configuración del esquema. En este ejemplo, la segunda cláusula `case` aplicará para ambos esquemas. Así, si un esquema no está establecido por cualquier razón, se tomará por omisión la configuración `casa`.

Ahora, para seleccionar entre dos conjuntos de configuraciones, ejecute:

```
cardctl scheme casa
```

o bien

```
cardctl scheme trabajo
```

El comando `cardctl` hace el equivalente a apagar todas sus tarjetas y luego reiniciarlas. Este comando puede ejecutarse de forma segura estando el sistema PCMCIA cargado o no, pero el comando puede fallar si está usando otros dispositivos PCMCIA en ese momento (incluso si sus configuracion no es explícitamente dependiente de la configuración del esquema).

Para mostrar la configuración del esquema, ejecute:

```
cardctl scheme
```

Por omisión, la configuración del esquema es persistente a través de los inicios del equipo. Esto puede tener efectos no deseados si la red se inicializa para el ambiente equivocado. Opcionalmente, puede establecer el valor inicial del esquema con la opción de inicio `SCHEME`; consulte la sección 2.3 (Opciones de Inicio) para más detalles. También es posible establecer el esquema desde el prompt de inicio de `lilo`. Debido a que `lilo` pasa opciones desconocidas a `init` como variables de entorno, un valor destinado a `SCHEME` (o cualquier otra opción de inicio de PCMCIA) en el prompt de inicio se propagará al script de inicio PCMCIA.

Para ahorrarse tecleo, los esquemas pueden ser especificados en el archivo de configuración de `lilo`. Por ejemplo, puede tener:

```
root = /dev/hda1
read-only
image = /boot/vmlinuz
  label  = casa
  append = "SCHEME=casa"
image = /boot/vmlinuz
  label  = trabajo
  append = "SCHEME=trabajo"
```

Así, al teclear `casa` o `trabajo` en el prompt de inicio arrancará con el esquema PCMCIA apropiado.

5.3 Arranque desde un dispositivo PCMCIA

Tener el sistema de archivos raíz en un dispositivo PCMCIA es algo difícil porque el sistema PCMCIA de Linux no está diseñado para ser enlazado dentro del kernel. Sus componentes principales, los módulos cargables del kernel y el demonio `cardmgr` dependen de un sistema que ya está ejecutándose. La funcionalidad `initrd` del kernel sortea esta limitación permitiendo a Linux iniciar utilizando un disco ram temporal como una imagen raíz mínima, cargar los controladores, y remontar entonces un sistema de archivos raíz diferente. La raíz temporal puede configurar dispositivos PCMCIA y luego remontar un dispositivo PCMCIA como raíz.

La imagen `initrd` de residir en un dispositivo arrancable obligatoriamente; lo que implica no puede tratarse de un dispositivo PCMCIA. Esta es una limitación de BIOS, no del kernel. Aqui es útil distinguir entre dispositivos «arrancables» (es decir, dispositivos desde los que se puede iniciar), y dispositivos *root-able*s (es decir, dispositivos origen, que son montados como raíz). Los dispositivos «arrancables» se determinan por BIOS, y están limitados generalmente a discos flexibles internos y unidades de disco duro. La funcionalidad `initrd` permite disponer de más dispositivos origen, no de más dispositivos «arrancables».

Algunas distribuciones de Linux permitirán la instalación a un dispositivo conectado a un adaptador SCSI PCMCIA, como un efecto lateral involuntario de su soporte para instalar desde unidades de CD-ROM SCSI PCMCIA. Sin embargo, en la actualidad, no hay herramientas de instalación de Linux que soporten el configurar una imagen `initrd` apropiada para iniciar Linux con un sistema de archivos raíz PCMCIA. Configurar un sistema con raíz PCMCIA de este modo requiere que se use otro sistema Linux para crear la imagen `initrd`. Si no tiene otro sistema Linux disponible, una opción podría ser instalar temporalmente una configuración mínima en una unidad no PCMCIA, crear una imagen `initrd`, y luego reinstalar en el dispositivo PCMCIA destino.

El *Linux Bootdisk-HOWTO* contiene información general acerca de la configuración de discos de inicio pero nada específico de `initrd`. El documento principal de `initrd` se incluye con las distribuciones recientes del código fuente del kernel, en `linux/Documentation/initrd.txt`. Antes de empezar, debería leer este documento. Es de utilidad estar familiarizado con `lilo`. El uso de `initrd` también requiere que tenga un kernel compilado con `CONFIG_BLK_DEV_RAM` y `CONFIG_BLK_DEV_INITRD` activados.

Esta es una técnica de configuración avanzada, y requiere un alto nivel de familiaridad con Linux y el sistema PCMCIA. Asegúrese de leer toda la documentación relevante antes de empezar. Las siguientes recetas deberían funcionar, pero las derivaciones de los ejemplos le pondrán rápidamente en un territorio desconocido y «no soportado»; y estará solo.

Este método requiere obligatoriamente que se use una versión del controlador PCMCIA `2.9.5` o posterior. Los paquetes PCMCIA antiguos o los componentes individuales no funcionarán en el contexto `initrd`. No mezcle componentes de diferentes versiones.

5.3.1 El script `pcinitrd`

El script `pcinitrd` crea una imagen básica para iniciar con una partición raíz PCMCIA. La imagen incluye una jerarquía de directorios mínima, algunos archivos de dispositivos, unos cuantos binarios, bibliotecas compartidas, y un conjunto de módulos controladores PCMCIA. Cuando se invoca `pcinitrd`, especifique los módulos controladores que busca que se incluyan en la imagen. Los componentes principales de PCMCIA, `pcmcia_core` y `ds`, se incluyen automáticamente.

Como ejemplo, digamos que su portátil usa un controlador compatible con `i82365`, y quiere iniciar Linux con el sistema de archivos raíz en un disco duro conectado a un adaptador *Adaptec SlimSCSI*. Podría crear una imagen `initrd` apropiada con:

```
pcinitrd -v initrd pcmcia/i82365.o pcmcia/aha152x_cs.o
```

Para personalizar la secuencia de inicio de `initrd`, podría montar la imagen usando el dispositivo *loopback* con un comando como:

```
mount -o loop -t ext2 initrd /mnt
```

y luego editar el script `linuxrc`. Los archivos de configuración se instalarán bajo `/etc` en la imagen, y también puede personalizarse. Consulte la página del manual de `pcinitrd` para mayor información.

5.3.2 Creación de un disquete de inicio `initrd`

Después de crear una imagen con `pcinitrd`, puede crear un disquete de inicio copiando el kernel, la imagen initrd comprimida, y algunos archivos de soporte para `lilo` a un disquete limpio. En el ejemplo siguiente, asumimos que el dispositivo raíz PCMCIA deseado es `/dev/sda1`:

```
mke2fs /dev/fd0
mount /dev/fd0 /mnt
mkdir /mnt/etc /mnt/boot /mnt/dev
cp -a /dev/fd0 /dev/sda1 /mnt/dev
cp [kernel-image] /mnt/vmlinuz
cp /boot/boot.b /mnt/boot/boot.b
gzip < [initrd-image] > /mnt/initrd
```

Genere un fichero `/mnt/etc/lilo.conf` que contenga:

```
boot=/dev/fd0
compact
image=/vmlinuz
    label=linux
    initrd=/initrd
    read-only
    root=/dev/sda1
```

Finalmente, invoque a lilo con:

```
lilo -r /mnt
```

Cuando `lilo` es invocado con `-r`, realiza todas las acciones tomando como directorio raíz el especificado. La razón para crear los archivos de dispositivo bajo `/mnt/dev` es que `lilo` no podrá usar esos archivos en `/dev` cuando se ejecute con este directorio raíz alternativo.

5.3.3 Instalación de una imagen `initrd` en una unidad no-Linux

Un uso común de la funcionalidad `initrd` puede darse en sistemas donde el disco duro interno está dedicado a otro sistema operativo. El kernel de Linux y la imagen `initrd` pueden ponerse en una partición no-Linux, y `lilo` o `LOADLIN` pueden configurarse para iniciar Linux desde esas imágenes.

Asumiendo que tiene un kernel que se ha configurado para el dispositivo raíz apropiado, y una imagen `initrd` creada en otro sistema, la forma más fácil de iniciar Linux es utilizando `LOADLIN`, como:

```
LOADLIN <kernel> initrd=<imagen-initrd>
```

Una vez que pueda iniciar Linux en su máquina destino, puede instalar `lilo` para permitir que Linux se inicie directamente. Por ejemplo, digamos que `/dev/hda1` es la partición no-Linux destino y `/mnt` puede usarse como un punto de montaje. Primero, genere un subdirectorio en el destino para los archivos de Linux:

```
mount /dev/hda1 /mnt
mkdir /mnt/linux
cp [imagen-del-kernel] /mnt/linux/vmlinuz
cp [imagen-initrd] /mnt/linux/initrd
```

En este ejemplo, digamos que `/dev/sda1` es la partición raíz de Linux deseada, en un disco duro SCSI montado vía un adaptador PCMCIA SCSI. Para instalar `lilo`, genere un archivo `lilo.conf` que contenga:

```
boot=/dev/hda
map=/mnt/linux/map
compact
image=/mnt/linux/vmlinuz
        label=linux
        root=/dev/sda1
        initrd=/mnt/linux/initrd
        read-only
other=/dev/hda1
        table=/dev/hda
        label=windows
```

La línea `boot=` dice que se instale el cargador de inicio en el MBR (*master boot record*) del dispositivo especificado. La línea `root=` identifica el sistema de archivos raíz deseado a usar después de cargar la imagen `initrd`, que puede resultar innecesario si la imagen del kernel ya se encuentra configurada de esta forma. La sección `other=` se usa para describir el otro sistema operativo instalado en `/dev/hda1`.

Para instalar `lilo` en este caso, teclee:

```
lilo -C lilo.conf
```

Nótese que en este caso, el archivo `lilo.conf` usa rutas absolutas que incluyen `/mnt`. Hice esto en el ejemplo porque el sistema de archivos destino puede no soportar la creación de archivos de dispositivos para las opciones `boot=` y `root=`.

6 Problemas con tarjetas no soportadas

6.1 Configuración de tarjetas no reconocidas

Asumiendo que su tarjeta está soportada por algún controlador existente, todo lo que se necesita hacer es añadir una entrada a `/etc/pcmcia/config` para decirle a `cardmgr` cómo identificar la tarjeta, y qué controlador(es) necesitan ser asociados a esta tarjeta. Consulte la página del manual de `pcmcia` para más información acerca del formato del archivo de configuración. Si inserta una tarjeta desconocida, `cardmgr` normalmente almacenará parte de información de la identificación en el registro del sistema, lo cual puede usarse para elaborar la entrada de configuración. Esta información puede mostrarse también con el comando `cardctl ident`.

He aquí un ejemplo de cómo avisa `cardmgr` de una tarjeta no soportada en `/usr/adm/messages`

```
cardmgr[460]: unsupported card in socket 1
cardmgr[460]: product info: "MEGAHERTZ", "XJ2288", "V.34 PCMCIA MODEM"
cardmgr[460]: manfid: 0x0101, 0x1234  function: 2 (serial)
```

La entrada correspondiente en `/etc/pcmcia/config` podría ser:

```
card "Megahertz XJ2288 V.34 Fax Modem"
  version "MEGAHERTZ", "XJ2288", "V.34 PCMCIA MODEM"
  bind "serial_cs"
```

o usar los códigos de ID más compactos del producto:

```
card "Megahertz XJ2288 V.34 Fax Modem"
  manfid 0x0101, 0x1234
  bind "serial_cs"
```

Puede usar `*` para comparar cadenas que no necesiten concordar exactamente, como los números de versión. Cuando haga nuevas entradas en la configuración, hay que ser cuidadosos para copiar las cadenas exactamente, preservando mayúsculas y minúsculas, y espacios en blanco. Asegúrese también de que la entrada en la configuración tiene el místmo número de cadenas que aparecen en el archivo de registro.

Tenga en cuenta que puede especificar cualquier controlador para una tarjeta, pero si sólo está dando palos de ciego, no hay mucha razón para esperar que esto resulte productivo. Puede tener suerte y encontrar que su tarjeta está soportada por un controlador existente. Sin embargo, el resultado más probable es que el controlador no funcione, y puede tener efectos laterales desafortunados como el congelamiento de su sistema. A diferencia de la mayoría de los controladores de dispositivos, los cuales comprueban la pressencia de la tarjeta apropiada, el sondeo para un dispositivo PCMCIA se hace con `cardmgr`, y el controlador por sí mismo puede no verificar antes de intentar comunicarse con el dispositivo.

Después de editar `/etc/pcmcia/config`, envíe una señal a `cardmgr` para recargar el archivo con:

```
kill -HUP `cat /var/run/cardmgr.pid`
```

Si configura una entrada para una tarjeta nueva, por favor, envíeme una copia para que pueda incluirla en el archivo de configuración estándar.

6.2 Soporte para una tarjeta ethernet compatible con NE2000

Antes de empezar: este procedimiento sólo funcionará para tarjetas ethernet simples. Las tarjetas multifunción (por ejemplo, las tarjetas «combo» ethernet/módem) tienen una capa extra de complejidad en relación a cómo están integradas las dos funciones, y generalmente no pueden soportarse sin obtener algo de información de la configuración provista por el fabricante de la tarjeta. Usar el procedimiento siguiente con una tarjeta multifunción no resultará productivo en absoluto.

Primero, compruebe si la tarjeta es reconocida por `cardmgr`. Algunas tarjetas que no están listadas en `SUPPORTED.CARDS` son realmente versiones OEM de tarjetas que sí están soportadas. Si encuentra una tarjeta como ésta, hágamelo saber para que pueda añadirla a la lista.

Si su tarjeta no es reconocida, siga las instrucciones en la sección 6.1 (Configuración de tarjetas no reconocidas) para crear una entrada en la configuración para su tarjeta, y relacionar la tarjeta con el controlador `pcnet_cs`. Reinicie `cardmgr` para utilizar el archivo de configuración actualizado.

Si el controlador `pcnet_cs` dice que no puede determinar la dirección ethernet del hardware de la tarjeta, edite su nueva entrada en la configuración para relacionar la tarjeta con el controlador de memoria `memory_cs`.

Reinicie `cardmgr` para utilizar el nuevo archivo de configuración actualizado. Necesitará conocer la dirección ethernet del hardware de la tarjeta. Esta dirección es una serie de seis números hexadecimales de dos dígitos, impresos normalmente en la misma tarjeta. Si no están impresos en la tarjeta, puede usar un controlador de DOS para mostrar la dirección. En cualquier caso, una vez que la sepa, ejecute:

```
dd if=/dev/mem0a count=20 | od -Ax -t x1
```

y busque el volcado de información de su tarjeta. Sólo los bytes pares están definidos, así que ignore los bytes impares del volcado. Anote el desplazamiento hexadecimal del primer byte de la dirección. Ahora, edite `clients/pcnet_cs.c` y busque la estructura `hw_info`. Necesitará crear una nueva entrada para la tarjeta. El primer campo es el desplazamiento de memoria. Los siguientes tres campos son los primeros tres bytes de la dirección de hardware. El campo final contiene algunos indicadores de características especiales de la tarjeta; para empezar, pruebe estableciéndola a `0`.

Después de editar `pcnet_cs.c`, compile e instale el nuevo módulo. Edite nuevamente `/etc/pcmcia/config/`, y cambie la relación de `memory_cs` con `pcnet_cs`. Siga las instrucciones para recargar el archivo de configuración, y habrá terminado. Por favor mándeme copias de sus nuevas entradas de configuración a `hw_info`.

Si no puede encontrar la dirección hardware de su tarjeta en el vaciado hexadecimal, como un último recurso, puede «forzar» la dirección cuando se inicializa el módulo `pcnet_cs`. Edite `/etc/pcmcia/config.opts` y añada una opción `hw_addr`, como esta:

```
module "pcnet_cs" opts "hw_addr=0x00,0x80,0xc8,0x01,0x02,0x03"
```

Por supuesto, sustituya su propia dirección de hardware de la tarjeta en el punto apropiado. Tenga en cuenta que si ha tenido que hacer esto, es muy difícil que su tarjeta sea genuinamente compatible con NE2000. De hecho, no estoy seguro de la existencia de tarjetas que no sean manejadas por alguno de los dos primeros métodos.

6.3 Tarjetas PCMCIA para unidades de disquete

La interfaz para disquete PCMCIA que se usa en los *Compaq Aero* y otros equipos todavía no está soportada por este paquete. La dificultad para soportar el disquete Aero radica en que el Aero parece usar un controlador PCMCIA personalizado para soportar DMA en el disquete. Sin saber exáctamente cómo se hace esto, no hay forma de implementar soporte bajo Linux.

Si la tarjeta del adaptador de disquete está presente cuando se inicia, la BIOS configurará la tarjeta, y Linux la identificará como una unidad de disquete normal. Cuando se cargan los controladores PCMCIA de Linux, notarán que la tarjeta ya está configurada y conectada al controlador de Linux, y este socket se dejará solo. Así que, la unidad puede usarse si está presente al momento de iniciar, pero la tarjeta no se puede intercambiar en caliente.

6.4 ¿Qué hay de las tarjetas Xircom?

El paquete actual PCMCIA incluye un controlador para las tarjetas ethernet y ethernet/modem de Xircom, gracias al trabajo de Werner Koch. He dispuesto un foro especialmente para la discusión del desarrollo del controlador Xircom, en *http://hyper.stanford.edu/HyperNews/get/pcmcia/xircom.html*.

Durante mucho tiempo, las tarjetas Xircom no fueron soportadas porque Xircom tenía como política de la compañía no divulgar información técnica acerca de sus tarjetas. Sin embargo, han modificado sus reglas, y ahora, distribuyen información de los controladores...

7 Trucos para depurar e información de programación

7.1 Envío de informes de *bugs* que son de utilidad

La mejor forma de informar de *bugs* es usar las listas de mensajes de HyperNews en el servidor web de Linux PCMCIA. De este modo, otras personas podrán ver los problemas actuales (y reparaciones o trabajos relacionados, si están disponibles). He aqui algunas cosas que se deben incluir en los informes de *bugs*:

- El tipo de sistema, y la salida del comando `probe`.
- Qué tarjetas PCMCIA está usando.
- Su versión del kernel de Linux, y la versión del controlador PCMCIA.
- Cualquier cambio que haya hecho a los archivos de inicio en `/etc/pcmcia`, o al script de inicio de PCMCIA.
- Todos los mensajes relacionados con PCMCIA en el registro de su sistema.

Todos los módulos PCMCIA y el demonio `cardmgr` envían mensajes de estado al registro del sistema, que estará normalmente en sitios como `/var/log/messages` o `/usr/adm/messages`. Este archivo debe ser el primer lugar a comprobar cuando se esté rastreando un problema. Cuando envíe una notificación de bug, incluya siempre el contenido de este archivo. Si tiene problemas para encontrar los mensajes de su sistema, revise `/etc/syslog.conf` para ver cuantas clases diferentes de mensajes se manejan.

Antes de enviar una notificación de bug, por favor asegúrese que no esté usando una copia obsoleta del paquete de controladores. Aunque resulte gratificante leer informes sobre un bug que ya he reparado, no supone un uso particularmente constructivo de mi tiempo.

Si no tiene acceso a web, puede enviarme los informes de *bugs* a *dhinds@hyper.stanford.edu*. Sin embargo, prefiero que sean introducidos en mi servidor web, así pueden ser vistos por otros.

7.2 Interpretación de los informes generados por los *traps* del kernel

Si su problema incluye un fallo del kernel, el vaciado del registro del fallo sólo es útil si puede traducir la dirección del error, EIP, o algo semejante. Las versiones recientes de `klogd` intentan traducir las direcciones de fallos basándose en el mapa actual de símbolos del kernel, pero puede que no funcione si el error se produce en un módulo, o si el problema es lo bastante severo como para que que `klogd` no pueda terminar de escribir la información del fallo en el registro del sistema.

Si se localiza en el kernel principal, la dirección de fallo puede encontrarse en el archivo `System.map`. El cual puede estar instalado en `/System.map` o en `/boot/System.map`. Si está en un módulo, el comando `nm` proporciona la misma información; sin embargo, la dirección del fallo necesita ajustarse basándose en la dirección de carga del módulo. Digamos que experimenta el siguiente fallo del kernel:

```
Unable to handle kernel NULL pointer dereference
current->tss.cr3 = 014c9000, %cr3 = 014c9000
*pde = 00000000
Oops: 0002
CPU:    0
EIP:    0010:[<c2026081>]
EFLAGS: 00010282
```

La dirección de fallo es `0xc2026081`. Si buscamos en `System.map`, vemos que esto está más allá de los límites del kernel, por ejemplo, es un módulo del kernel. Para determinar qué módulo, revise la salida de `ksyms -m | sort`

```
Address   Symbol                            Defined by
c200d000  (35k)                             [pcmcia_core]
c200d10c  register_ss_entry                 [pcmcia_core]
c200d230  unregister_ss_entry               [pcmcia_core]
          ...
c2026000  (9k)                              [3c574_cs]
c202a000  (4k)                              [serial_cs]
```

Así, `0xc2026081` está en el módulo `3c574_cs` con un desplazamiento de `0x0081` desde el inicio del módulo. Todavía no podemos ver más allá de este desplazamiento en `3c574_cs.o`: cuando el kernel carga un módulo, inserta un encabezado en la dirección de carga del mismo, así el inicio real se desplaza desde la dirección mostrada en `ksyms`. El tamaño del encabezado varía con la versión del kernel: para encontrar el tamaño en su kernel, busque un módulo que exporte símbolos (como `pcmcia_core`), y compare la dirección del símbolo con la salida de `nm` para ese mismo símbolo. En este ejemplo, `register_ss_entry` se carga con un desplazamiento de `0xc200d10c - 0xc200d000 = 0x010c`, mientras que `nm pcmcia_core.o` muestra el desplazamiento como `0x00c0`, así que el tamaño del encabezado es `0x010c - 0x00c0 = 0x004c` bytes.

Regresando a `3c574_cs.o`, nuestro desplazamiento de fallo es `0x0081`, y restando el encabezado `0x004c`, el desplazamiento real del módulo es `0x0035`. Ahora comprobando el resultado de un `nm 3c574_cs.o | sort`, observamos:

```
0000002c d if_names
0000002c t tc574_attach
00000040 d mii_preamble_required
00000041 d dev_info
```

El fallo se localiza en `tc574_attach()`.

En este ejemplo, el fallo no causó un congelamiento total del sistema, así que `ksyms` puede ejecutarse después de haber tenido lugar el fallo. En otros casos, puede que tenga que deducir indirectamente las direcciones de carga del módulo. La misma secuencia de eventos cargará normalmente los módulos en el mismo orden y en las mismas direcciones. Si se produce un fallo cuando se inserta cierta tarjeta, obtenga la salida de `ksyms` antes de insertar la tarjeta, o con una tarjeta diferente insertada. Puede cargar manualmente los módulos controladores de la tarjeta con `insmod` y ejecutar `ksyms` antes de insertarla.

Para profundizar, consulte `man insmod`, `man ksyms`, y `man klogd`. En el árbol de los fuentes del kernel, `Documentation/oops-tracing.txt` también es relevante. He aquí unas cuantas pistas para depurar el kernel:

- Dependiendo del error, puede ser útil traducir direcciones en el *Trazado de llamadas*, usando el mismo procedimiento para la dirección de error principal.
- Para diagnosticar un congelamiento silencioso, pruebe provocar el problema con X desactivado, porque los mensajes del kernel se envían a la consola en texto, y no serán visibles bajo X.
- Si mata a `klogd` muchos de los mensajes del kernel harán eco directamente a la consola de texto, el cual puede ser útil si el problema impide a `klogd` escribir en el registro del sistema.
- Para hacer que todos los mensajes del kernel se envíen a la consola, para kernels `2.1.x`, si existe `/proc/sys/kernel/printk`, hacer:

```
echo 8 > /proc/sys/kernel/printk
```

- La combinación de teclas <RightAlt><ScrLk> imprime un vaciado del registro en la consola de texto. Esto puede funcionar en caso de que el sistema esté o no completamente sin responder, y la dirección EIP puede interpretarse como fallo del kernel.
- Para los kernels `2.1.x` configurados con `CONFIG_MAGIC_SYSRQ` activado, se pueden activar varias funciones de emergencia por medio de las combinaciones especiales de las teclas <Alt><SysRq>, que están documentadas en `Documentation/sysrq.txt` dentro del árbol de los fuentes del kernel.

7.3 Primeros auxilios al depurar a bajo nivel

Los módulos PCMCIA contienen bastante código de depuración compilado de forma condicional. La mayor parte de este código está bajo el control de las definiciones del preprocesador de `PCMCIA_DEBUG`. Si no está definido, el código de depuración no se compilará. Si se establece a `0`, se compilará pero no estará activo. Los números mayores especifican el incremento del nivel de detalle del registro. Cada módulo compilado con `PCMCIA_DEBUG` definido tendrá un parámetro entero, `pc_debug`, que controla el nivel de detalle de su salida. Esto puede ajustarse cuando se carga el módulo, así la salida puede controlarse en base a cada módulo sin necesidad de recompilar.

Su configuración por omisión para `syslogd` puede descartar los mensajes de depuración del kernel. Para asegurarse de que se están registrando, edite `/etc/syslog.conf` y compruebe que los mensajes `kern.debug` se registren en algún lugar. Consulte `man syslog.conf` para más detalles.

Hay algunas herramientas de depuración en el subdirectorio `debug_tools` dentro de la distribución de PCMCIA. Las utilidades `dump_tcic` y `dump_i365` generan volcados completos de los controladores PCMCIA, y decodifican mucha de la información del registro. Son útiles si tiene acceso a una hoja con los datos del chip controlador correspondiente. El comando `dump_cis` (`dump_tuples` en las distribuciones `pre-3.0.2`) lista el contenido de la CIS (*Card Information Structure*) (Estructura de Información de Tarjeta), y decodifica algunos bits importantes. `dump_cisreg` muestra los registros de configuración local de una tarjeta.

El controlador de tarjetas de memoria `memory_cs` a veces también es útil para depurar problemas con PC Cards de 16 bits. Puede utilizarse con cualquier tarjeta, y no interfiere con otros controladores. Puede usarse para acceder directamente a los atributos de memoria o memoria común de cualquier tarjeta. De igual modo, con las tarjetas *CardBus*, el controlador `memory_cb` puede utilizarse con cualquier tarjeta de 32 bits, para dar acceso directo a los espacios de direcciones de esa tarjeta. Revise las páginas del manual para más información.

7.4 /proc/bus/pccard

A partir de los kernels `2.1.103`, el paquete PCMCIA crea un árbol de información de estado bajo `/proc/bus/pccard`. La entrada `memory` muestra las posiciones de memoria para dispositivos PC Card en un formato similar a `/proc/ioports`. Cada socket tiene también su propio subdirectorio de entradas de estado. La entrada `info` identifica el controlador del host y describe sus características. La entrada `exca` es un volcado del registro `ExCA` compatible con *Intel i82365sl* que se configura para ese socket. Para los puentes *CardBus*, la entrada `pci` es el volcado del espacio de la configuración PCI del puente, y la entrada `cardbus` es el vaciado de los registros de configuración de *CardBus*.

7.5 Programación de controladores de servicios PCMCIA para nuevas tarjetas

El *Linux PCMCIA Programmer's Guide* constituye la mejor documentación acerca de la interfaz de los controladores. La última versión estará siempre disponible en `hyper.stanford.edu` en `/pub/pcmcia/doc`, o vía WWW en *http://hyper.stanford.edu/HyperNews/get/pcmcia/home.html*.

Con los dispositivos relativamente similares a los dispositivos ISA normales, probablemente pueda Vd. usar parcialmente controladores Linux existentes. En algunos casos, el tropiezo más grande será modificar un controlador existente que pueda manejar la inserción y extracción de dispositivos después del momento de iniciar. De los controladores actuales, el controlador de tarjeta de memoria es el único controlador autónomo, que no depende de otras partes del kernel de Linux para hacer la mayor parte del trabajo sucio.

En muchos casos, el mayor impedimento para soportar un nuevo tipo de tarjeta es el obtener información técnica por parte del fabricante. Puede ser difícil el encontrar a quién preguntar, o a quien explicar que información se necesita. Sin embargo, con pocas excepciones, es muy difícil, si no imposible, el implementar un controlador para una tarjeta sin información técnica por parte del fabricante.

He escrito un controlador modelo con muchos comentarios que explican bastante cómo el controlador se comunica con los Servicios de Tarjetas; lo encontrará en la distribución fuente de PCMCIA en `clients/dummy_cs.c`.

7.6 Sugerencias para los autores de controladores PCMCIA

He decidido que no es realmente factible para mi el distribuir todos los controladores de PCMCIA como parte del paquete PCMCIA. Cada controlador nuevo hace que el paquete principal sea incrementalmente más dificil de mantener, e incluir un controlador inevitablemente transfiere algo del trabajo de mantenimiento del autor del controlador hacia mí. En lugar de ello, decidiré caso por caso si se incluyen o no los controladores que sean contribuciones, basándome en la demanda de los usuarios y también en la facilidad de mantenerlos. Para los controladores que no se incluyen en el paquete principal, sugiero que los autores de los controladores adopten el esquema siguiente para empaquetar sus controladores de cara a su distribución.

Los archivos controladores deben acomodarse en el mismo esquema del directorio que utiliza la distribución fuente de PCMCIA, así el controlador puede ser desempaquetado en la parte más alta del árbol de los fuentes de PCMCIA. Debe incluir los archivos fuentes (en `./modules/`), una página del manual (en `./man/`), y los archivos de configuración (en `./etc/`). El directorio más alto debe incluir también un archivo `README`.

El directorio de más alto nivel debe incluir un *makefile*, configurado para que `make -f ... all` y `make -f ... install` compilen el controlador e instalen los archivos apropiados. Si este archivo tiene una extensión `.mk`, será invocado automáticamente por el `Makefile` de más alto nivel para los destinos `all` e `install`. He aquí un ejemplo de cómo debe elaborarse un `Makefile`:

```
# Un simple Makefile para un controlador de contribución
FILES = sample_cs.mk README.sample_cs \
        modules/sample_cs.c modules/sample_cs.h \
        etc/sample etc/sample.opts man/sample_cs.4
all:
        $(MAKE) -C modules MODULES=sample_cs.o
install:
        $(MAKE) -C modules install-modules MODULES=sample_cs.o
        $(MAKE) -C etc install-clients CLIENTS=sample
        $(MAKE) -C man install-man4 MAN4=sample_cs.4
dist:
        tar czvf sample_cs.tar.gz $(FILES)
```

Este `Makefile` usa los destinos de instalación que se definen en la versión `2.9.10` y versiones posteriores del paquete PCMCIA. Este makefile también incluye un destino `dist` para conveniencia del autor del controlador. Probablemente desee añadir un número de versión al final del nombre del paquete (por ejemplo, `sample_cs-1.5.tar.gz`). Una distribución completa puede ser similar a:

```
sample_cs.mk
README.sample_cs
modules/sample_cs.c
modules/sample_cs.h
etc/sample
etc/sample.opts
man/sample_cs.4
```

De esta forma, cuando un controlador de contribución se desempaquete, se convierte en parte esencial del árbol de los fuentes de PCMCIA. Puede hacer uso de los archivos de encabezados de PCMCIA, así como también de la maquinaria para comprobar la configuración del sistema del usuario, y chequeo automático de dependencias, tal y como un controlador «normal».

Aceptaré controladores preparados de acuerdo a esta especificación y los colocaré en el directorio `/etc/pcmcia/contrib` en mi servidor FTP, `hyper.stanford.edu`. El archivo `README` en este directorio describirá cómo desempaquetar un controlador de contribución.

La interface de controlador no ha cambiado mucho a pesar del tiempo, y ha preservado casi siempre su compatibilidad con las versiones anteriores. Un controlador normalmente no necesitará actualizarse para revisiones menores en el paquete principal. Trataré de notificar a los autores de los controladores «externos» de los cambios que se requiera realizar a sus controladores.

7.7 Sugerencias para encargados de las distribuciones de Linux

Si su distribución tiene herramientas para configuración del sistema que quiera que sean compatibles PCMCIA, por favor, use los archivos `*.opts` en `/etc/pcmcia` para su «integración». Dichos archivos no serán modificados si un usuario compila e instala una nueva versión del paquete PCMCIA. Si modifica los scripts principales de configuración, una instalación fresca sobreescribirá silenciosamente sus scripts personalizados y romperá la conexión con sus herramientas de configuración. Contacte conmigo si no está seguro de cómo escribir un script de opciones apropiado, o si necesita características adicionales.

Resulta muy útil para los usuarios (y para mi) que documente cómo deriva su distribución del paquete PCMCIA que se describe en este documento. En particular, por favor documente los cambios al script de inicio y a los scripts de configuración. Si me manda la información apropiada, la incluiré en la sección 2.5 (Notas acerca de distribuciones de Linux específicas).

Cuando construya una distribución PCMCIA, considere el incluir los controladores aportados, que no son parte del paquete PCMCIA principal. Por razones de mantenimiento, estoy tratando de limitar el tamaño del paquete principal, añadiendo solamente controladores nuevos si considero que son de interés general. Los demás controladores se distribuirán por separado, como se describe en la sección anterior. La división entre controladores generales y separados es algo arbitraria y en parte histórica, y no debería implicar diferencia alguna en cuanto a calidad.

8 Anexo: El INSFLUG

El *INSFLUG* forma parte del grupo internacional *Linux Documentation Project*, encargándose de las traducciones al castellano de los Howtos, así como de la producción de documentos originales en aquellos casos en los que no existe análogo en inglés, centrándose, preferentemente, en documentos breves, como los *COMOs* y *PUFs* (**P**reguntas de **U**so **F**recuente, las *FAQs*. :)), etc.

Diríjase a la sede del Insflug para más información al respecto.

En élla encontrará siempre las **últimas** versiones de las traducciones «oficiales»: *www.insflug.org*. Asegúrese de comprobar cuál es la última versión disponible en el Insflug antes de bajar un documento de un servidor réplica.

Además, cuenta con un sistema interactivo de gestión de fe de erratas y sugerencias en línea, motor de búsqueda específico, y más servicios en los que estamos trabajando incesantemente.

Se proporciona también una lista de los servidores réplica (*mirror*) del Insflug más cercanos a Vd., e información relativa a otros recursos en castellano.

En *http://www.insflug.org/insflug/creditos.php3* cuenta con una detallada relación de las personas que hacen posible tanto esto como las traducciones.

¡Diríjase a *http://www.insflug.org/colaboracion/index.php3* si desea unirse a nosotros!.

«Cartel» Insflug, *cartel@insflug.org*.