Serie COMO de Linux

por Greg Hankins, greg.hankins@cc.gatech.edu
Traducido por Jesús Jiménez Sánchez, jesus.jimenez@writeme.com v1.9, 2 Enero 1997, traducción del 2 Junio 1997

Este documento describe cómo configurar los dispositivos de comunicación serie en una máquina Linux.

Índice General

1 Introducción

Este es el Serie COMO de Linux. Todo sobre cómo configurar módems y terminales bajo Linux, algunos trucos de serie, y soluciones a problemas. Este COMO está dirigido a Linux funcionando bajo un hardware basado en un x86, aunque debe funcionar en otras arquitecturas.

1.1 Copyright

1.2 Nuevas versiones de este documento

Las nuevas versiones de Serial HOWTO (versión en inglés[1]) se pueden conseguir en

- `ftp://sunsite.unc.edu:/pub/Linux/docs/HOWTO/Serial-HOWTO`

y servidores espejo (*mirror sites*). Existen otros formatos, así como PostScript y DVI en el directorio `other-formats`. El Serial-HOWTO también está disponible en

- `http://sunsite.unc.edu/LDP/HOWTO/Serial-HOWTO.html`

y se enviará a `comp.os.linux.answers` mensualmente.

1.3 Realimentación

Por favor, envíeme cualquier pregunta, comentario, sugerencia, o material adicional. Siempre estoy ansioso de oír lo que opinan sobre el HOWTO. ¡También estoy atento a mejoras!. Dígame exactamente qué es lo que no entiende, o que debe aclararse. Puede contactar conmigo en `greg.hankins@cc.gatech.edu` vía email. Todas las cartas serán contestadas, aunque puede llevarme una semana o así, dependiendo de lo ocupado que esté. Tengo muchas cartas de `root` y sitios desconfigurados. Por favor, envíeme el correo como una persona real, y asegúrese de que su sistema de correo funcione, así podré contestarle. Si no le respondo en dos semanas, por favor, vuelva a enviarme la carta de nuevo.

También puede contactar conmigo en: Greg Hankins
College of Computing
801 Atlantic Drive
Atlanta, GA 30332-0280
a través del correo lento, y en `http://www.cc.gatech.edu/staff/h/Greg.Hankins`

Por favor, incluya el número de versión de el Serial HOWTO cuando escriba, esta es la versión 1.9.

1.4 Renuncia de responsabilidad

Los resultados pueden variar. Las respuestas dadas puede que no funcionen en todos los sistemas y con todas las configuraciones posibles.

2 Hardware serie soportado

Se sabe que Linux funciona con el siguiente hardware serie.

2.1 Placas serie normales de PC

- placas serie estándar de PC (COM1 - COM4), a los cuales los dispositivos serie externos (módems, ratón serie, etc, ...) pueden ser conectados.
- módems internos de PC estándar (COM1 - COM4)
- Quickpath Systems Port-Folio 550e (IRQs permitidas 3, 4, 5, 9, 10, 11, 12, y 15)

Nota: debido a conflictos de dirección, no puede usar COM4 y la placa de vídeo IBM8514 simultáneamente. Esto es debido a un error en la placa IBM8514.

[1] Ver sección 16 para las traducciones al Castellano

2.2 Módems Plug and Play (PnP)

Los módems PnP funcionan con Linux. Sin embargo, recomiendo que no se usen porque son complejos. Existen un par de cosas que puede probar:

- Intente desconectar la característica PnP en la BIOS de su ordenador y módem.
- Arranque Win95, configure el módem, y vea si la configuración permanece cuando arranque Linux. Si no, entonces:
- Arranque Win95, configure el módem, y entonces haga un *arranque en caliente* de Linux, así se preservará la configuración. Puede hacer esto con el cargador `loadlin`.
- Consiga las herramientas de PnP en `http://www.redhat.com/linux-info/pnp`.
- Y, por supuesto, puede adquirir un módem de verdad.

2.3 Placas serie multipuerto no-inteligentes (con UARTs 8250/16450/16550A)

- AST FourPort y clónicas (4 puertos)
- Accent Async-4 (4 puertos)
- Arnet Multiport-8 (8 puertos)
- Bell Technologies HUB6 (6 puertos)
- Boca BB-1004 (4 puertos), BB-1008 (8 puertos), BB-2016 (16 puertos)
- Boca IOAT66 (6 puertos)
- Boca 2by4 (4 puertos serie, 2 puertos paralelos)
- Computone ValuePort V4-ISA (compatible AST FourPort)
- Digi PC/8 (8 puertos)
- GTEK BBS-550 (8 puertos)
- Longshine LCS-8880, Longshine LCS-8880+ (compatible AST FourPort)
- Moxa C104, Moxa C104+ (compatible AST FourPort)
- PC-COMM (4 puertos)
- SIIG I/O Expander 2S IO1812 (4 puertos)
- STB-4COM (4 puertos)
- Twincom ACI/550
- Usenet Serial Board II (4 puertos)

En general, Linux soportará cualquier placa que use una UART 8250, 16450, 16550, 16550A (o compatible), o un módem interno que emule una de las UARTs anteriores.

Nota: el BB-1004 y BB-1008 no soporta líneas DCD y RI, y no se pueden usar para recibir llamadas. Funcionan bien en otras tareas.

2.4 Placas serie multipuerto inteligentes

- Comtrol RocketPort (36MHz ASIC; 4, 8, 16 ó 32 puertos)
contacto: `info@comtrol.com` o `http://www.comtrol.com`
estado del controlador: soportado por Comtrol
localización del controlador: `ftp://tsx-11.mit.edu/pub/linux/packages/comtrol`

- Computone IntelliPort II (16MHz 80186; 4, 8 ó 16 puertos),
IntelliPort II EXpandable (20MHz 80186; 16 - 64 puertos)
contacto: Michael H. Warfield, `mhw@wittsend.atl.ga.us`
estado del controlador: pre-ALPHA

- Cyclades Cyclom-Y (UARTs CD1400 de Cirrus Logic; 8 - 32 puertos),
Cyclom-Z (25MHz MIPS R3000; 8 - 128 puertos)
contacto: `sales@cyclades.com` o `http://www.cyclades.com`
estado del controlador: soportado por Cyclades
localización del controlador: `ftp://ftp.cyclades.com/pub/cyclades` e incluido en el núcleo del Linux desde la versión 1.1.75

- Digi PC/Xi (12.5MHz 80186; 4, 8, ó 16 puertos),
PC/Xe (12.5/16MHz 80186; 2, 4, u 8 puertos),
PC/Xr (16MHz IDT3041; 4 u 8 puertos),
PC/Xem (20MHz IDT3051; 8 - 64 puertos)
contacto: `sales@dgii.com` o `http://www.dgii.com`
estado del controlador: soportado por Digi
localización del controlador: `ftp://ftp.dgii.com/drivers/linux` e incluido en el núcleo de Linux desde la versión 2.0

- Digi COM/Xi (10MHz 80188; 4 u 8 puertos)
contacto: Simon Park, `si@wimpol.demon.co.uk`
estado del controlador: ALPHA
nota: Simon está a menudo desconectado del correo durante meses debido a su trabajo. Mark Hatle, `fray@krypton.mankato.msus.edu` se ha ofrecido voluntario cortésmente para tener el controlador disponible por si usted lo necesita.

- Equinox SuperSerial Technology (30MHz ASIC; 2 - 128 puertos)
contacto: `sales@equinox.com` or `http://www.equinox.com`
estado del controlador: soportado por Equinox
localización del controlador: `ftp://ftp.equinox.com/library/sst`

- GTEK Cyclone (16C654 UARTs; 6, 16 y 32 puertos),
SmartCard (24MHz Dallas DS80C320; 8 puertos),
BlackBoard-8A (16C654 UARTs; 8 puertos),
PCSS (15/24MHz 8032; 8 puertos)
contacto: `spot@gtek.com` o `http://www.gtek.com`
estado del controlador: soportado por GTEK
localización del controlador: `ftp://ftp.gtek.com/pub`

- Hayes ESP (8 - 64 puertos)
contacto: Andrew J. Robinson, `arobinso@nyx.net` o `http://www.nyx.net/~arobinso`
estado del controlador: BETA
localización del controlador: `http://www.nyx.net/~arobinso` e incluido en el núcleo del Linux desde la versión 2.1.15

- Maxpeed SS (Toshiba; 4, 8 y 16 puertos)
contacto: `info@maxpeed.com` o `http://www.maxpeed.com`

estado del controlador: soportado por Maxpeed
localización del controlador: `ftp://maxpeed.com/pub/ss`

- Moxa C218 (12MHz 80286; 8 puertos),
  Moxa C320 (40MHz TMS320; 8 - 32 puertos)
  contacto: `info@moxa.com.tw` o `http://www.moxa.com.tw`
  estado del controlador: soportado por Moxa
  localización del controlador: `ftp://ftp.moxa.com.tw/drivers/c218-320/linux`
- SDL RISCom/8 (CD180 de Cirrus Logic; 8 puertos)
  contacto: `sales@sdlcomm.com` or `http://www.sdlcomm.com`
  estado del controlador: soportado por SDL
  localización del controlador: `ftp://ftp.sdlcomm.com/pub/drivers`
- Omega COMM-8 (8 puertos)
  contacto: Vance Petree, `vpetree@infi.net`
- Specialix SIO (20MHz Zilog Z280; 4 - 32 puertos),
  XIO (25MHz Zilog Z280; 4 - 32 puertos)
  contacto: Simon Allen, `simonallen@cix.compulink.co.uk`
  estado del controlador: BETA
  localización del controlador: `ftp://sunsite.unc.edu/pub/Linux/kernel/patches/serial`
- Stallion EasyIO-4 (4 puertos), EasyIO-8 (8 puertos), y
  EasyConnection (8 - 32 puertos) - cada uno con UARTs CD1400 de Cirrus Logic,
  Stallion (8MHz 80186 CPU; 8 o 16 puertos),
  Brumby (10/12 MHz 80186 CPU; 4, 8 o 16 puertos),
  ONboard (16MHz 80186 CPU; 4, 8, 12, 16 o 32 puertos),
  EasyConnection 8/64 (25MHz 80186 CPU; 8 - 64 puertos)
  contacto: `sales@stallion.com` o `http://www.stallion.com`
  estado del controlador: soportado por Stallion
  localización del controlador: `ftp://ftp.stallion.com/drivers/ata5/Linux` e incluido en el núcleo del Linux desde la versión 1.3.27

Una revisión de los productos de Comtrol, Cyclades, Digi y Stallion fue imprimida en Junio de 1995 en la distribución de *Linux Journal*. El artículo está disponible en `http://www.ssc.com/lj/issue14`.

2.5 Hardware serie no soportado

Los módems que afirman ser “Winmodems”, por ejemplo el USR Sporster Winmodem e IBM Aptiva MWAVE, *no* están soportados por Linux. Estos módems usan diseños propietarios que requieren controladores especiales de Windows. Evite este tipo de módems. Además, evite los módems que requieran los controladores Rockwell RPI por las mismas razones.

Las placas serie inteligentes que requieran controladores no disponibles para Linux tampoco funcionarán .

3 ¿Cuáles son los nombres de los puertos serie?

Un puerto de E/S es un modo de conseguir que los datos entren y salgan de un ordenador. Existen muchos tipos de puertos de E/S como los puertos serie, puertos paralelos, controladores de disqueteras, placas ethernet, etc. Trataremos con puertos serie ya que los módems y los terminales son dispositivos serie. Cada puerto serie debe tener una dirección de E/S, y una interrupción (IRQ).

Estos son los cuatro puertos serie correspondientes a COM1 - COM4:

```
/dev/cua0, /dev/ttyS0 (COM1) direccion 0x3f8 IRQ 4
/dev/cua1, /dev/ttyS1 (COM2) direccion 0x2f8 IRQ 3
/dev/cua2, /dev/ttyS2 (COM3) direccion 0x3e8 IRQ 4
/dev/cua3, /dev/ttyS3 (COM4) direccion 0x2e8 IRQ 3
```

Si Linux no detecta ningún puerto serie cuando arranca, entonces asegúrese de que el soporte de comunicaciones serie está compilado y activo en el núcleo.

Los dispositivos `/dev/ttyS`N son para conexiones de entrada y los dispositivos `/dev/cua`N son para conexiones de salida. N es el número del puerto serie.

En este documento me referiré a COM1 como `ttyS0`, COM2 como `ttyS1`, COM3 como `ttyS2`, y COM4 como `ttyS3`. Si me refiero a un dispositivo específico en `/dev`, siempre lo precederé de `/dev` para evitar confusiones. Nótese que por defecto estos dispositivos tienen IRQs que se solapan. No puede usar todos los puertos con la configuración por defecto, debe reasignar IRQs diferentes. Vea la sección 8 para configurar las IRQs.

En algunas instalaciones se crean dos dispositivos extras, `/dev/modem` para el módem y `/dev/mouse` para el ratón. Los dos son enlaces simbólicos al dispositivo `/dev/cua`N apropiado que usted especificó durante la instalación (a menos que tenga un bus mouse, entonces `/dev/mouse` apuntará al dispositivo del bus mouse).

Existen algunas discusiones sobre los méritos de `/dev/mouse` y `/dev/modem`. Yo estoy *totalmente* en contra del uso de estos enlaces. En particular, si está planeando usar su módem para recibir llamadas puede tener problemas debido a que los ficheros de bloqueo puede que no funcionen correctamente si usa `/dev/modem`. Úselos si lo desea, pero *asegúrese de que apuntan al dispositivo correcto*. Sin embargo, si mueve o borra este enlace, algunas aplicaciones (`minicom` por ejemplo) pueden necesitar una reconfiguración.

3.1 Números de dispositivo mayor y menor de los puertos serie en `/dev`

```
/dev/ttyS0 mayor 4, menor 64    /dev/cua0 mayor 5, menor 64
/dev/ttyS1 mayor 4, menor 65    /dev/cua1 mayor 5, menor 65
/dev/ttyS2 mayor 4, menor 66    /dev/cua2 mayor 5, menor 66
/dev/ttyS3 mayor 4, menor 67    /dev/cua3 mayor 5, menor 67
```

Nótese que todas las distribuciones deben traer estos dispositivos correctamente creados. Puede comprobarlo escribiendo:

```
linux% ls -l /dev/cua*
linux% ls -l /dev/ttyS*
```

3.1.1 Creación de dispositivos en `/dev`

Si le falta un dispositivo, tendrá que crearlo con el comando `mknod`. Por ejemplo, suponga que necesita crear un dispositivo para `ttyS0`:

```
linux# mknod -m 666 /dev/cua0 c 5 64
linux# mknod -m 666 /dev/ttyS0 c 4 64
```

Puede usar el script `MAKEDEV`, que está en `/dev`. Esto simplifica la creación de dispositivos. Por ejemplo, si necesita crear el dispositivo para `ttyS0` puede escribir:

```
linux# cd /dev
linux# ./MAKEDEV ttyS0
```

Esto dirige la creación de dispositivos para entrada y salida, y debería establecer los permisos adecuados.

3.2 Notas sobre placas multipuerto no-inteligentes

Los dispositivos que su placa multipuerto usa dependen del tipo de placa que tenga. Estas están listadas con detalle en `rc.serial` que viene con el programa `setserial`. Recomiendo conseguir la última versión de `setserial` si está intentando usar placas multipuerto. Seguramente necesitará crear estos dispositivos. Puede usar el comando `mknod`, o el script `MAKEDEV`. Los dispositivos para placas multipuerto se crean sumando “64 + número de puerto”. Así, si quiere crear un dispositivo para `ttyS17`, debe escribir:

```
linux# mknod -m 666 /dev/cua17 c 5 81
linux# mknod -m 666 /dev/ttyS17 c 4 81
```

Note que “64 + 17 = 81”. Si usa el script `MAKEDEV`, puede escribir:

```
linux# cd /dev
linux# ./MAKEDEV ttyS17
```

Nota: el manual SIIG en el listado de IO1812 para COM5-COM8 es erróneo. Debería ser COM5=0x250, COM6=0x258, COM7=0x260, y COM8=0x268.

Nota: El Registro de Estado de Interrupción de la Digi PC/8 es en 0x140.

Nota: para una AST Fourport, puede necesitar especificar `skip_test` en `rc.serial`.

3.3 Notas sobre placas multipuerto inteligentes

Lea la información que viene con el controlador. Estas placas usan dispositivos especiales, y no los estándar. Esta información varía dependiendo de su hardware.

4 Programas interesantes que debería conocer

4.1 ¿Qué es `getty`?

`getty` es un programa que controla el proceso de login cuando se conecta a una máquina Unix. Puede necesitar `getty` si quiere que sea posible llamar a su máquina Linux con un módem. No necesita `getty` si lo único que quiere es llamar con su modem. Existen tres versiones que son normalmente usadas con Linux: `getty_ps`, `mgetty` y `agetty`. La sintaxis de estos programas difieren, así que asegúrese de comprobar que está usando la sintaxis correcta para el `getty` que esté usando.

4.1.1 Acerca de `getty_ps`

Esta versión de `getty` fue originalmente escrita por Paul Sutcliffe Jr., `paul@devon.lns.pa.us`. Muchas distribuciones vienen con el paquete `getty_ps` instalado. El paquete `getty_ps` contiene dos programas. `getty` se usa para la consola, y dispositivos de terminal - y `uugetty` se usa para los módems. Yo uso esta versión de `getty`, así que me concentraré en el paquete `getty_ps` en este COMO.

4.1.2 Acerca de `mgetty`

`mgetty` es una versión de `getty` escrita por Gert Döring, `gert@greenie.muc.de`, para ser usada con el módem. Además de permitir acceso telefónico, `mgetty` también da soporte de FAX con `sendfax`, que acompaña a `mgetty`. La documentación de `mgetty` es bastante buena, y no necesita ningún suplemento. Por favor, diríjase a ella para las instrucciones de instalación. Puede encontrar la última información sobre `mgetty` en `http://www.leo.org/~doering/mgetty/`.

4.1.3 Acerca de `agetty`

`agetty` es la tercera variación de `getty`. Fue originalmente escrita por W.Z. Venema, `wietse@wzv.win.tue.nl`. Es una simple, y completamente funcional, implementación de `getty` que va mejor con las consolas virtuales o los terminales que con los módems.

4.2 ¿Qué es `setserial`?

`setserial` es un programa que le permite ver y modificar los distintos atributos de un dispositivo serie, incluyendo las direcciones de los puertos, las interrupciones, y otras opciones de los puertos serie. Fue inicialmente escrito por Rick Sladkey, y fue bastante modificado por Ted T'so, `tytso@mit.edu`, que actualmente lo mantiene. Puede saber que versión tiene ejecutando `setserial` sin argumentos. Recomiendo conseguir la última versión de algún servidor FTP de Linux.

Cuando su sistema Linux arranca, sólo son configurados `ttyS{0-3}`, usando las IRQs 3 y 4 por defecto. Por eso, si tiene cualquier otro puerto serie en otras placas o si `ttyS{0-3}` tiene una IRQ no estándar, *debe* usar este programa para configurar esos puertos serie. Para la lista completa de opciones, consulte la página man.

5 ¿Cómo puedo llamar con mi módem?

5.1 Requerimientos de hardware

Primero, asegúrese de que tiene el cable correcto. El módem requiere un cable directo, sin pins cruzados. Cualquier tienda de informática puede tenerlo. Asegúrese de tener el género correcto. Si está usando el puerto serie DB25, siempre debe ser el DB25 macho. No lo confunda con el puerto paralelo, que es el DB25 hembra. Conecte su módem a uno de los puertos serie. Consulte el manual de su módem para saber cómo hacerlo si necesita ayuda.

5.1.1 Notas sobre módems internos

Para un módem interno, no necesitará un cable. Un módem interno no necesita un puerto serie, tiene uno interno. Todo lo que necesita hacer es configurarlo para usar una interrupción que no esté siendo usada, y configurar la dirección de E/S del puerto. Consulte el manual de su módem si se atasca. Además, vea la sección 8 si necesita ayuda para elegir las interrupciones y las direcciones.

En en algunas placas madres tendrá que desconectar el puerto serie que el módem está suplantando para evitar conflictos. Esto se puede hacer con los puentes o en la configuración de la BIOS, dependiendo de su placa madre. Consulte el manual de su placa madre.

Debido a un error en la placa de vídeo IBM8514, puede tener problemas si quiere poner su módem interno en `ttyS3`. Si Linux no detecta su módem interno en `ttyS3`, puede usar `setserial` y el módem funcionará bien. Los módems internos conectados en `ttyS{0-2}` no deben tener ningún problema para ser detectados. Linux no efectúa ninguna autoconfiguración en `ttyS3` debido a este error en la placa de vídeo.

5.2 Hablando con el módem

Use `kermit` o algún otro programa simple de comunicaciones para probar la configuración, antes de meterse en programas de comunicación más complejos. (Por razones legales, `kermit` no se distribuye en las distribuciones comerciales. Puede encontrar la última versión de `kermit` en `ftp://sunsite.unc.edu/pub/Linux/apps/comm` y servidores espejo). Por ejemplo, digamos que su módem está en `ttyS3`, y su velocidad es 38400 bps. Puede hacer lo siguiente:

```
linux# kermit
C-Kermit 5A(188), 23 Nov 92, POSIX
Type ? or HELP for help
C-Kermit>set line /dev/cua3
C-Kermit>set speed 38400
/dev/cua3, 38400 bps
C-Kermit>c
Connecting to /dev/cua3, speed 38400.
The escape character is Ctrl-\ (ASCII 28, FS)
Type the escape character followed by C to get back,
or followed by ? to see other options.
ATV1
OK
<ctrl>-\-C
(Back at linux)
C-Kermit>quit
linux#
```

Si su módem responde a los comandos `AT`, puede suponer que está funcionando correctamente en lo que respecta a Linux. Intente llamar a otro módem. Si no le gusta `kermit`, pruebe uno de los programas de comunicaciones más avanzados. Mire en la sección 10 sobre programas de comunicaciones si necesita una guía.

Cuando llame con su módem, configure la velocidad a la más alta que éste soporte. Las versiones de Linux con una versión de `libc` superior a 5.x tienen soporte para velocidades de hasta 115200 bps. `libc` normalmente está en `/lib`, así que mire allí para saber que versión tiene. Si Linux no reconoce una velocidad de 57600 ó 115200 bps, entonces debe usar el programa `setserial` para configurar el puerto serie a una velocidad mayor. Vea la sección 9 para saber cómo hacer esto. Después, configure la velocidad a 38400 bps en el programa de comunicaciones.

5.3 Configuración del módem para llamar

Si sólo va a hacer llamadas, puede configurar el módem como quiera. Si tiene la intención de usar el módem para recibir llamadas, *debe* configurar el módem a la misma velocidad a la que quiera que funcione `getty`. Es decir, si quiere usar `getty` a 38400 bps, ponga la velocidad a 38400 bps cuando configure el módem. Esto se hace para prevenir desajustes entre el ordenador y el módem. En general, la configuración de fábrica, que activa la corrección de errores y el control de flujo por hardware, es la mejor configuración para hacer llamadas, consulte el manual de su módem para activar estas opciones.

A mi me gusta ver los códigos resultantes, por eso activo `Q0` - devuelve los códigos resultantes. Para activar esto en mi módem, tengo que preceder el nombre de registro con el comando `AT`. Usando `kermit` u otro programa de comunicaciones, conéctelo al módem y escriba: `ATQ0`.

También me gusta ver lo que escribo, por eso activo `E1` - activa el eco de comandos. Si su módem tiene capacidad de compresión de datos, probablemente quiera activarlo. Consulte el manual de su módem para más ayuda, y una lista de opciones completa. Si su módem puede grabar la configuración activa, asegúrese de guardarla (se suele hacer con `AT&W`, pero varia entre los fabricantes de módems), si no tendrá que configurar los registros cada vez que active el módem, o lo reinicie.

5.4 Control de flujo por hardware

Si su módem soporta el control de flujo por hardware (RTS/CTS), le recomiendo que lo use. Esto es particularmente importante para módems que soporten la compresión de datos. Primero, tendrá que activar el control de flujo RTS/CTS del puerto serie. Es mejor hacerlo en el arranque, en el fichero

/etc/rc.d/rc.local o /etc/rc.d/rc.serial. ¡Asegúrese de que estos ficheros son llamados desde el fichero principal rc.sysinit!. Necesita hacer lo siguiente para cada puerto serie en que quiera activar el control de flujo por hardware:

```
stty crtscts < /dev/cuaN
```

También debe activar el control de flujo RTS/CTS en el módem. Consulte el manual de su módem para saber cómo hacerlo, ya que varía entre fabricantes de módems. Asegúrese de guardar la configuración del módem si éste soporta el almacenamiento del perfil.

6 ¿Cómo puedo llamar y recibir llamadas con mi módem usando getty_ps?

Consiga que su módem llame correctamente. Si no ha leído la sección 5, *¡léala ahora!*. Contiene información de configuración *muy* importante. No necesita leer esta sección si lo único que quiere hacer es llamar con el módem.

6.1 Configuración del módem para llamar y recibir llamadas

Para llamar y recibir llamadas, *tiene* que configurar el módem de un cierto modo (de nuevo, usando los comandos AT del módem):

```
E1      activa el eco de comandos
Q0      devuelve los codigos resultantes
V1      habilita el codigo de resultados largos (texto)
S0=0    nunca contesta (uugetty hace esto con la opcion WAITFOR)
```

Si no configura esto correctamente, la cadena INIT de su fichero de configuración puede fallar, echando a perder el proceso completo. Pero, más sobre ficheros de configuración después ...

```
&C1     DCD se activa solamente despues de conectar
&S0     DSR siempre esta activo
DTR activa/desactiva el reinicio del modem (depende del fabricante - RTFM)
```

Esto afecta a lo que hace su módem cuando inicia o finaliza una llamada.

Si su módem no puede almacenar el perfil, puede usar la cadena INIT de su fichero de configuración. Véalo más adelante. Algunos módems tienen interruptores DIP que afectan a la configuración de los registros. Asegúrese de que estén configurados correctamente.

He iniciado una colección de configuraciones para diferentes tipos de módems. Hasta ahora sólo tengo unos pocos, si desea enviarme una configuración que funcione, por favor, ¡hágalo!. Puede conseguirla en ftp://ftp.cc.gatech.edu/pub/people/gregh/modem-configs.

Nota: para conseguir que mi módem USR Courier V.34 se reinicie correctamente cuando se desactiva el DTR, he tenido que activar &D2 y S13=1 (esto activa el bit 0 del registro S13). Se ha confirmado que esto funciona igualmente en módems USR Sportster V.34.

Nota: algunos módems Supra tratan el DCD de modo diferente a otros módems. Si está usando un Supra, pruebe a poner &C0 y *no* &C1. También debe poner &D2 para manejar el DTR correctamente.

6.2 Instalación de getty_ps

Consiga la última versión en `sunsite.unc.edu:/pub/Linux/system/Serial`. Por defecto, **getty_ps** se configura para usar la norma FSSTND (**F**ile**S**ystem **ST**a**ND**ard) de Linux, lo cual significa que los binarios estarán en `/sbin`, y los ficheros de configuración se llamarán `/etc/conf.{uu}getty.ttyS`*N*. ¡Esto no está claro en la documentación!. También se espera que los ficheros de bloqueo estén en `/var/lock`. Asegúrese de que existe el directorio `/var/lock`.

Si no quiere usar la norma FSSTND, los binarios pueden ir en `/etc`, los ficheros de configuración pueden ir en `/etc/default/{uu}getty.ttyS`*N*, y los ficheros de bloqueo pueden ir en `/usr/spool/uucp`. Recomiendo hacer las cosas de esto modo si está usando UUCP, porque UUCP puede tener problemas si mueve los ficheros de bloqueo a donde no los busca.

`getty_ps` también puede usar `syslogd` para guardar los mensajes. Vea las páginas man de `syslogd(1)` y `syslog.conf(5)` para configurar `syslogd`, si no lo tiene funcionando todavía. Los mensajes se guardan con prioridad LOG_AUTH, los errores usan LOG_ERR, y la depuración usa LOG_DEBUG. Si no quiere usar `syslogd` puede editar `tune.h` en los ficheros fuente de `getty_ps` para usar otro fichero de almacenamiento en su lugar, llamado `/var/adm/getty.log` por defecto.

Decida si quiere usar la norma FSSTND y la capacidad syslog. También puede elegir una combinación de ambos. Edite los ficheros `Makefile`, `tune.h` y `config.h` para reflejar sus decisiones. Después compile e instale siguiendo las instrucciones incluidas en el paquete.

A partir de este punto, todas las referencias a `getty_ps` se referirán a `getty_ps`. Las referencias a `uugetty` se referirán a `uugetty` que viene con el paquete `getty_ps`. Estas instrucciones no funcionarán en `mgetty` o `agetty`.

6.3 Configurando uugetty

Para recibir llamadas y llamar con su módem, vamos a usar `uugetty`. `uugetty` hace importantes comprobaciones de los ficheros de bloqueo. Actualice `/etc/gettydefs` para incluir las entradas de los módems si todavía no están allí (nótese que cada entrada apunta a otra, esto no es para velocidades fijas - las líneas en blanco son necesarias entre cada entrada):

```
# Entradas del modem
38400# B38400 CS8 # B38400 SANE -ISTRIP HUPCL #@S @L @B login: #19200

19200# B19200 CS8 # B19200 SANE -ISTRIP HUPCL #@S @L @B login: #9600

9600# B9600 CS8 # B9600 SANE -ISTRIP HUPCL #@S @L @B login: #2400

2400# B2400 CS8 # B2400 SANE -ISTRIP HUPCL #@S @L @B login: #1200

1200# B1200 CS8 # B1200 SANE -ISTRIP HUPCL #@S @L @B login: #300

300# B300 CS8 # B300 SANE -ISTRIP HUPCL #@S @L @B login: #38400
```

Si tiene un módem de 9600 bps o más rápido con compresión de datos, puede bloquear la velocidad del puerto serie y permitir que el módem controle las transiciones a otras velocidades. Entonces, en vez de la serie descendente de líneas listadas antes, `/etc/gettydefs` sólo necesita contener una línea para el módem:

```
# velocidad fijada a 38400
F38400# B38400 CS8 # B38400 SANE -ISTRIP HUPCL #@S @L @B login: #F38400

# velocidad fijada a 19200
F19200# B19200 CS8 # B19200 SANE -ISTRIP HUPCL #@S @L @B login: #F19200
```

```
# velocidad fijada a 9600
F9600# B9600 CS8 # B9600 SANE -ISTRIP HUPCL #@S @L @B login: #F9600
```

Si tiene configurado su modem con control de flujo por hardware RTS/CTS, puede añadir `CRTSCTS` a las entradas:

```
# velocidad fijada a 38400 con control de flujo por hardware
F38400# B38400 CS8 CRTSCTS # B38400 SANE -ISTRIP HUPCL CRTSCTS #@S @L @B login: #F38400

# velocidad fijada a 19200 con control de flujo por hardware
F19200# B19200 CS8 CRTSCTS # B19200 SANE -ISTRIP HUPCL CRTSCTS #@S @L @B login: #F19200

F9600# B9600 CS8 CRTSCTS # B9600 SANE -ISTRIP HUPCL CRTSCTS #@S @L @B login: #F9600
```

Si quiere, puede hacer que `uugetty` escriba cosas interesantes en el mensaje de login. En los ejemplos, tenemos el nombre del sistema, la línea serie, y la velocidad actual en bps. Puede añadir otras cosas:

```
@B    Velocidad actual en bps (calculada en el momento que se lee @B).
@D    Fecha actual, en MM/DD/AA.
@L    Linea serie a la cual est\'{a} conectado getty.
@S    Nombre del sistema.
@T    Hora actual, en HH:MM:SS (24 horas).
@U    Numero de usuarios actualmente conectados. Esto es, el numero
      de entradas en el fichero /etc/utmp que tienen un campo ut_name
      no nulo.
@V    Valor de VERSION, dado en el fichero de opciones por defecto.
Para que se vea el car\'{a}cter '@', use '\@' o '@@'.
```

Después, asegúrese de que tiene un dispositivo de entrada y otro de salida para el puerto serie en que esté el módem. Si tiene el módem en `ttyS3`, necesitará los dispositivos `/dev/cua3` y `/dev/ttyS3`. Si no tiene los dispositivos adecuados, vea la sección 3.1.1 para saber como crearlos.

6.4 uugety a medida

Existen un montón de parámetros que puede configurar para cada puerto. Están implementados en distintos ficheros de configuración para cada puerto. El fichero `/etc/conf.uugetty` será usado por *cada* ejecución de `uugetty`, y `/etc/conf.uugetty.ttyS`*N* será usado solamente para ese puerto. Se pueden encontrar ejemplos de ficheros de configuración por defecto en los ficheros fuentes de `getty`, que viene con la mayoría de las distribuciones de Linux. Debido a lo que ocupan, no lo vamos a listar aquí. Nótese que si está usando una versión antigua de `getty` (anterior a la 2.0.7e), o no está usando FSSTND, entonces el fichero por defecto debe ser `/etc/default/uugetty.ttyS`*N*. Mi fichero `/etc/conf.uugetty.ttyS3` es como sigue:

```
# ejemplo de fichero de configuracion de uugetty para un modem compatible
# Hayes para permitir las conexiones a traves de m\'{o}dem
#
# fichero de bloqueo alternativo a comprobar ... si este fichero de bloqueo
# existe, entonces uugetty se ejecuta de nuevo y el modem se reinicia
ALTLOCK=cua3
ALTLINE=cua3
# l\'{\i}nea a inicializar
INITLINE=cua3
# tiempo para desconectar si est\'{a} inactivo ...
```

```
TIMEOUT=60
# cadena de inicializacion del modem ...
# formato: <esperado> <enviado> ... (secuencia de chat)
INIT="" AT\r OK\r\n
WAITFOR=RING
CONNECT="" ATA\r CONNECT\s\A
# esta linea configura el tiempo de espera antes de enviar el mensaje de login
DELAY=1
#DEBUG=010
```

Añada la siguiente línea a `/etc/inittab`, así `uugetty` se ejecutará en el puerto serie (sustituyéndolo por la información correcta para su entorno - localización del fichero de configuración, puerto, velocidad, y tipo de terminal por defecto):

```
S3:456:respawn:/sbin/uugetty -d /etc/default/uugetty.ttyS3 ttyS3 F38400 vt100
```

Vuelva a ejecutar `init`:

```
linux# init q
```

En el parámetro de velocidad de `/etc/inittab`, debe poner la velocidad en bps más alta que soporte su módem. Ya que `uugetty` no soporta velocidades de 57600 ó 115200 bps, debe usar el programa `setserial` para configurar el puerto serie a una velocidad superior. Vea la sección 9 para saber cómo hacer esto. Entonces, ponga 38400 bps en `/etc/inittab`.

Ahora Linux vigilará el puerto serie por posibles conexiones. Llame desde otro sitio y conéctese a su sistema Linux.

`uugetty` tiene muchas opciones, vea la página man de `getty(1m)` para una descripción completa. Además de otras cosas hay una característica de planificación y otra de rellamada.

7 ¿Cómo puedo tener un terminal conectado a mi PC?

Las instrucciones de esta sección sirven para la conexión de terminales, así como otros ordenadores a un puerto serie de su sistema Linux.

7.1 Hardware requerido

Asegúrese de tener el cable correcto. Un cable de módem nulo comprado en cualquier tienda de informática puede valer. ¡Pero debe ser un cable de *modem nulo*!. Muchas tiendas de informática venden este tipo de cable como cable serie de impresora. Asegúrese de estar usando el puerto serie, el DB25 o el DB9 macho, y no el puerto paralelo (DB25 hembra o Centronics).

Para un conector DB25, necesita como mínimo:

```
DB25 macho en el PC                     DB25 en el Terminal
TxD   Transmision de datos  2 --> 3     RxD   Recepcion de datos
RxD   Recepcion de datos    3 <-- 2     TxD   Transmision de datos
SG    Tierra                7 --- 7     SG    Tierra
```

Si quiere tener señales de sincronización del hardware, debe tener un cable completo de módem nulo:

```
DB25 macho del PC                      DB25 del terminal
TxD  Transmision de datos   2 --> 3    RxD  Recepcion de datos
RxD  Recepcion de datos     3 <-- 2    TxD  Transmision de datos
RTS  Peticion de envio      4 --> 5    CTS  Listo para envio
CTS  Listo para envio       5 <-- 4    RTS  Peticion de envio
DSR  Modem listo            6
                            |
DCD  Deteccion de portadora 8 <-- 20   DTR  Terminal listo
SG   Tierra                 7 --- 7    SG   Tierra
                                  6    DSR  Modem listo
                                  |
DTR  Terminal listo        20 --> 8    DCD  Deteccion de portadora
```

Si tiene un conector DB9 en el puerto serie, intente lo siguiente:

```
DB9 del PC                             DB25 del terminal
RxD  Recepcion de datos     2 <-- 2    TxD  Transmision de datos
TxD  Transmision de datos   3 --> 3    RxD  Recepcion de datos
SG   Tierra                 5 --- 7    SG   Tierra
```

Alternativamente, un cable completo DB9-DB25 de módem nulo:

```
DB9 del PC                             DB25 del terminal
RxD  Recepcion de datos     2 <-- 2    TxD  Transmision de datos
TxD  Transmision de datos   3 --> 3    RxD  Recepcion de datos
                                  6    DSR  Modem listo
                                  |
DTR  Terminal listo         4 --> 8    DCD  Deteccion de portadora
GND  Tierra                 5 --- 7    GND  Tierra
DCD  Deteccion de portadora 1
                            |
DSR  Modem listo            6 <-- 20   DTR  Terminal listo
RTS  Peticion de envio      7 --> 5    CTS  Listo para envio
CTS  Listo para envio       8 <-- 4    RTS  Peticion de envio
(RI  Indicador de llamada   9, no se necesita)
```

(Efectivamente, ¡los pins 2 y 3 *realmente* tienen significados opuestos en los conectores DB9 a los que tienen en los conectores DB25!)

Si no está usando un cable completo de módem nulo, deberá hacer el siguiente truco: en la parte del conector que va al ordenador, conecte juntos RTS y CTS, y además conecte juntos DSR, DCD y DTR. De este modo, cuando el ordenador busque una cierta señal de sincronismo, la cogerá (de él mismo).

Ahora que tiene el tipo de cable adecuado, conecte el terminal al ordenador. Si puede, dígale al terminal que ignore las señales de control de módem. Pruebe a configurar el terminal a 9600 bps, 8 bits de datos, 1 bit de parada y sin bit de paridad.

7.2 Configurando `getty`

Añada las siguientes entradas en `/etc/gettydefs` para usar `getty` en su terminal, si todavía no están allí:

```
# Entrada para un terminal no-inteligente a 38400 bps
DT38400# B38400 CS8 CLOCAL # B38400 SANE -ISTRIP CLOCAL #@S @L login: #DT38400
```

```
# Entrada para un terminal no-inteligente a 19200 bps
DT19200# B19200 CS8 CLOCAL # B19200 SANE -ISTRIP CLOCAL #@S @L login: #DT19200

# Entrada para un terminal no-inteligente a 9600 bps
DT9600# B9600 CS8 CLOCAL # B9600 SANE -ISTRIP CLOCAL #@S @L login: #DT9600
```

Si lo desea, puede hacer que `getty` imprima cosas interesantes en el mensaje de login. En los ejemplos, imprime el nombre del sistema y la línea serie. Puede añadir más cosas:

```
@B    Velocidad actual en bps (calculada en el momento que se lee @B).
@D    Fecha actual, en MM/DD/AA.
@L    Linea serie a la cual esta conectado getty.
@S    Nombre del sistema.
@T    Hora actual, en HH:MM:SS (24 horas).
@U    Numero de usuarios actualmente conectados. Esto es, el numero
      de entradas en el fichero /etc/utmp que tienen un campo ut_name
      no nulo.
@V    Valor de VERSION, dado en el fichero de opciones por defecto.
Para que se vea el caracter '@', use '\@' o '@@'.
```

Asegúrese de que no existe un fichero de configuración de `getty` o `uugetty` para el puerto serie al que está conectado el terminal (`/etc/default/`{uu}`getty.ttyS`*N* o `/etc/conf.`{uu}`getty.ttyS`*N*), ya que probablemente esto interferirá si ejecuta `getty` en un terminal. Borre el fichero si existe.

Edite el fichero `/etc/inittab` para ejecutar `getty` en un puerto serie (substituya la información por la correcta para su entorno - puerto, velocidad, y tipo de terminal por defecto):

```
S1:456:respawn:/sbin/getty ttyS1 DT9600 vt100
```

Vuelva a ejecutar `init`:

```
linux# init q
```

Ahora, debería estar viendo una petición de login en el terminal. Debe pulsar return para conseguir la atención del terminal.

7.3 Notas sobre cómo configurar un PC como terminal

Mucha gente configura otros PCs como terminales conectados a un sistema Linux. Por ejemplo, un viejo 8088 u 80286 es perfecto para este propósito. Todo lo que necesita es un disco de arranque de DOS que contenga una versión de DOS apropiada para un terminal-PC, y un programa de comunicaciones para ejecutar en el terminal-PC.

`kermit` funciona muy bien para esto. Puede encontrar versiones precompiladas de `kermit` para casi todos los SO que existen en `ftp://watsun.cc.columbia.edu/pub/ftp/kermit`. Otros programas de comunicaciones populares de DOS, como `Telix` y `procomm` funcionan igual de bien. Asegúrese de introducir la información del puerto serie en la configuración de comunicaciones del terminal-PC.

8 ¿Puedo usar más de dos dispositivos serie?

No necesita leer esta sección, a menos que quiera usar tres o más dispositivos serie... (suponiendo que tiene una placa multipuerto).

Suponiendo que tiene otro puerto serie sin usar, sí, puede. El número de puertos serie que puede usar está limitado por el número de interrupciones (IRQs) y direcciones de puertos de E/S que tengamos disponibles. Esto no es una limitación de Linux, es una limitación del bus del PC. Cada dispositivo serie debe tener asignado una interrupción propia y una dirección. Un dispositivo serie puede ser un puerto serie, un módem interno, o una placa serie multipuerto.

Las placas serie multipuerto están especialmente diseñadas para tener múltiples puertos serie que comparten la misma IRQ. Linux obtiene los datos de ellos usando una dirección de E/S para cada puerto de la placa.

8.1 Eligiendo las interrupciones de los dispositivos serie

El PC normalmente viene con `ttyS0` y `ttyS2` en la IRQ 4, y `ttyS1` y `ttyS4` en la IRQ 3. Puede saber que IRQs están usándose viendo `/proc/interrupts`. Para usar más de dos dispositivos serie, debe reasignar las interrupciones. Una buena elección es reasignar una interrupción del puerto paralelo. El PC normalmente viene con IRQ 5 e IRQ 7 configuradas como interrupciones para los puertos paralelos, pero poca gente usa dos puertos paralelos. Puede reasignar una de las interrupciones a un dispositivo serie, y aún usar felizmente un puerto paralelo. Necesitará el programa `setserial` para hacer esto. Además, tendrá que jugar con los puentes de su placa, vea los documentos de su placa. Ponga los puentes para la IRQ que quiera para cada puerto.

Necesitará configurar algunas cosas, ya que hay una, y solamente una interrupción para cada dispositivo serie. Así es como tengo mi configuración en `/etc/rc.d/rc.local` - debe hacerlo donde quiera que tenga su fichero de inicialización:

```
/sbin/setserial /dev/cua0 irq 3      # mi raton serie
/sbin/setserial /dev/cua1 irq 4      # mi terminal Wyse no-inteligente
/sbin/setserial /dev/cua2 irq 5      # mi modem Zoom
/sbin/setserial /dev/cua3 irq 9      # mi modem USR
```

Asignación estándar de IRQ:

```
IRQ  0    Reloj en el canal 0
IRQ  1    Teclado
IRQ  2    Cascada para el controlador 2
IRQ  3    Puerto serie 2
IRQ  4    Puerto serie 1
IRQ  5    Puerto paralelo 2
IRQ  6    Disquetera
IRQ  7    Puerto paralelo 1
IRQ  8    Reloj de tiempo real (24 horas)
IRQ  9    Redireccionada a IRQ 2
IRQ 10    no asignada
IRQ 11    no asignada
IRQ 12    no asignada
IRQ 13    Coprocesador matematico
IRQ 14    Controlador de disco duro 1
IRQ 15    Controlador de disco duro 2
```

No existen realmente un Modo Correcto de hacer las cosas cuando se eligen las interrupciones. Sólo asegúrese de que no las está usando la placa madre, o cualquier otra placa. 2, 3, 4, 5, ó 7 son buenas elecciones. “no asignada” significa que actualmente nada estándar usa estas IRQs. Nótese también que la IRQ 2 es la misma que la IRQ 9. Puede llamarlo 2 ó 9, el controlador serie lo entenderá. Si tiene una placa serie con un conector de 16 bits, también puede usar las IRQ 10, 11, 12 ó 15.

¡Sólo asegúrese de no usar las IRQs 0, 1, 6, 8, 13 ó 14!. Estas son usadas por la placa madre. La hará muy desgraciada si coge sus IRQs. Cuando lo haga, compruebe otra vez `/proc/interrupts` y asegúrese de que no haya conflictos.

8.2 Configurando las direcciones de los dispositivos serie

El siguiente paso es configurar la dirección del puerto. Vea el manual de su placa para configurar los puentes. Como las interrupciones, puede haber sólo un dispositivo serie en cada dirección. Los puertos suelen venir configurados de la siguiente manera:

```
ttyS0 direccion 0x3f8
ttyS1 direccion 0x2f8
ttyS2 direccion 0x3e8
ttyS3 direccion 0x2e8
```

Elija la dirección que desee para cada dispositivo serie que tenga y configure los puentes en concordancia. Tengo mi módem en `ttyS3`, el ratón en `ttyS0`, y el terminal en `ttyS2`.

Cuando arranque de nuevo, Linux debería ver los puertos serie en las direcciones en las que lo ha definido. La IRQ que Linux ve puede que no corresponda a la IRQ que ha configurado con los puentes. No se preocupe por esto. Linux no hace ninguna detección de IRQ mientras arranca, porque la detección de IRQ no es fiable y puede ser falsa. Use `setserial` para decirle a Linux que IRQ está usando el puerto. Puede ver `/proc/ioports` para saber que direcciones del puerto de E/S está usando Linux después de arrancar.

9 ¿Cómo puedo configurar el puerto serie para velocidades altas? ¿qué velocidad debo usar con mi módem?

Esta sección le ayudará a calcular la velocidad a usar cuando use su módem con un programa de comunicaciones, o con un programa `getty`.

- Si tiene algo más lento que un módem de 9600 bps (V.32), configure la velocidad a la más alta que su módem soporte. Por ejemplo 300 bps (V.21/Bell 103), 1200 bps (V.22/Bell 212A), ó 2400 bps (V.22bis).
- Si tiene un módem de 9600 bps (V.32), con compresión de datos V.42bis, ponga la velocidad a 38400 bps . La compresión V.42bis tiene una velocidad *teórica* de 4:1, es decir “4 * 9600 = 38400”.
- Si tiene un módem de 14400 bps (V.32bis), con compresión de datos V.42bis, use `setserial` con el parámetro `spd_hi` para configurar el puerto serie a 57600 bps (4 * 14400 = 57600).

  Use el parámetro `spd_hi` si tiene un módem a 28800 ó 33600 bps (V.FC o V.34) (4 * 28800 = 115200).

  Después, use la velocidad de 38400 bps en su programa de comunicaciones, o en `/etc/inittab`. Esta es ahora la velocidad más alta que ha configurado. Asegúrese de tener UARTs 16550A.

  Si su versión de libc es por lo menos 5.x, debe haber velocidades llamadas 57600 y 115200. `libc` está en `/lib`, así que mire allí para saber que versión tiene. Puede usar esto directamente (sin usar `setserial`), si sus aplicaciones han sido compiladas para usarlas. Existen muchas distribuciones por ahí, así que lo mejor que puede hacer es probar las velocidades más altas si tiene una versión reciente de una distribución Linux.

Compruebe primero la configuración de `setserial` en la línea de comandos, y después cuando esté funcionando, póngalo en `/etc/rc.d/rc.serial` o `/etc/rc.d/rc.local`, yo he puesto `ttyS3` a 115200 bps escribiendo:

```
/sbin/setserial /dev/cua3 spd_vhi
```

Asegúrese de estar usando un path válido para `setserial`, y un nombre válido para el dispositivo. Puede comprobar la configuración de un puerto serie ejecutando:

```
setserial -a /dev/cuaN
```

10 Programas de comunicaciones y utilidades

Una vez que todo esté funcionando, puede querer echar un vistazo a programas más avanzados. Todos están asequibles en los servidores habituales de FTP, si no vienen ya con su distribución.

- `ecu` - un programa de comunicaciones
- `minicom` - programa de comunicaciones parecido a `telix`
- `procomm` - programa de comunicaciones con zmodem parecido a `procomm`
- `seyon` - programa de comunicaciones basado en X
- `xc` - paquete de comunicaciones xcomm

Estos programas ofrecen más herramientas que `kermit` solo, incluyendo agenda telefónica, auto-marcado y cosas por el estilo.

- Otros programas muy útiles son `term` y `SLirP`. Ofrecen funcionalidad TCP/IP usando una cuenta de la shell.
- `screen` es otro programa multi-sesión. Se comporta igual que las consolas virtuales.
- `callback` es un programa que hace que su módem le llame al lugar de donde usted le ha llamado.
- `getty+fax` controla las cosas del FAX, y provee un `getty` alternativo.
- `ZyXEL` es un programa de control para los módems ZyXEL U-1496. Controla las llamadas entrantes, llamadas de respuesta de seguridad, enviar FAX, y funciones de buzón de voz.
- El software de SLIP y PPP se puede encontrar en `ftp://sunsite.unc.edu/pub/Linux/system/Network/serial`.
- Se puede encontrar otras cosas en `ftp://sunsite.unc.edu/pub/Linux/system/Serial` y `ftp://sunsite.unc.edu/pub/Linux/apps/comm` o uno de los muchos servidores espejo. Estos son los directorios donde se están manteniendo los programas serie.

11 Trucos serie y misceláneas

Aquí vienen algunos trucos serie que puede encontrar útiles ...

11.1 `kermit` y `zmodem`

Para usar zmodem con `kermit`, añada lo siguiente a `.kermc`:

```
define rz !rz < /dev/cuaN > /dev/cuaN
define sz !sz \%0 > /dev/cuaN < /dev/cuaN
```

Asegúrese de poner el puerto correcto en el que está el módem. Después, para usarlo, solamente escriba `rz` o `sz` <`nombre_de_fichero`> en la línea de comandos de `kermit`.

11.2 Activando los tipos de terminal automágicamente

Para activar automágicamente un tipo de terminal cuando se conecte, añádalo al fichero `/etc/inittab`. Si tengo un terminal vt100 en `ttyS1`, debo añadir "vt100" al comando `getty`:

```
S1:456:respawn:/sbin/getty ttyS1 DT9600 vt100
```

También puede conseguir `tset` en `ftp://sunsite.unc.edu:/pub/Linux/system/Terminal-management` o en un servidor espejo. Vea los documentos que vienen con `tset` para aprender como usarlo. `tset` puede establecer las características de un terminal cuando usted se conecte, y no depende de ninguna opción por defecto.

11.3 `ls` en color en conexiones serie

Si `ls` está causando problemas en la emulación del terminal con la opción de color, desconéctela. `ls --color`, y `ls --colour` usan las características de color. Algunas instalaciones tienen `ls` preparado para usar el color por defecto. Busque en `/etc/profile` y `/etc/csh.cshrc` alias de `ls`. También puede ponerle el alias `ls --no-color` a `ls`, si no quiere cambiar la configuración por defecto del sistema.

11.4 Imprimiendo en una impresora conectada a un terminal

Hay un programa llamado `vtprint` que hace esto, escrito por Garrett D'Amore, `garrett@sdsu.edu`. Está disponible en `ftp://ftp.sdsu.edu/pub/vtprint`, y en `http://www.sdsu.edu/~garrett`. Lo siguiente ha sido extraído del fichero `README` que viene con el programa:

> vtprint es un programa que permite a los usuarios imprimir desde un anfitrión Unix remoto a una impresora conectada a un terminal local o emulador, lo cual es perfecto para imprimir ficheros en casa, etc. (Sólo con ficheros de texto, creo.)

11.5 ¿Puede linux configurar los dispositivos serie automágicamente?

Sí. Para conseguir que Linux detecte y configure los dispositivos serie automáticamente al arrancar, añada la línea:

```
/sbin/setserial /dev/cuaN auto_irq skip_test autoconfig
```

al fichero `/etc/rc.d/rc.local` o `/etc/rc.d/rc.serial`. Haga esto para cada puerto serie que quiera que se autoconfigure. Asegúrese de poner un nombre de dispositivo que realmente exista en su máquina.

11.5.1 Notas sobre placas multipuerto

Para direcciones de placas, e IRQs, mire en `rc.serial` que viene con el programa `setserial`. Trae muchos detalles sobre placas multipuerto, incluyendo las direcciones de E/S y los nombres de los dispositivos.

12 Un paso más ...

Esta sección no necesita ser leída, pero puede ayudarle a comprender algo mejor el Unix, y el mundo de las telecomunicaciones.

12.1 ¿Qué son los ficheros de bloqueo?

Un fichero de bloqueo es simplemente un fichero que dice que dispositivo se está usando. Están en `/usr/spool/uucp`, o `/var/lock`. Los ficheros de bloqueo en Linux se llaman `LCK..`*nombre*, donde *nombre* puede ser un nombre de dispositivo, o un nombre de un sitio UUCP. Ciertos procesos crean estos bloqueos para así poder tener acceso exclusivo a un dispositivo, por ejemplo, si llama con su módem, aparecerá un bloqueo para que otros procesos sepan que alguien está usando actualmente el módem. Los bloqueos principalmente contienen el PID del proceso que tiene bloqueado el dispositivo. Muchos programas comprueban si un dispositivo está bloqueado, e intentan determinar si ese bloqueo es todavía válido buscando en la tabla de procesos el que tiene bloqueado el dispositivo. Si el bloqueo es válido, el programa se cerrará (o debería cerrarse). Si no, algunos programas borran el bloqueo viejo, y usan el dispositivo, creando su propio bloqueo. Otros programas solamente se cierran y dicen que el dispositivo está usándose.

12.2 “baudios” Vs. “bps”

“baudios” y “bps” son, quizás, unos de los términos peor usados en el campo de los ordenadores y telecomunicaciones. Mucha gente usa estos términos indiferentemente, ¡cuando de hecho no son lo mismo!.

baudios

La velocidad en baudios es una medida de cuantas veces por segundo cambia una señal, por ejemplo una señal enviada por un módem (**mo**ulador-**dem**odulador). Es decir, una velocidad de 1200 baudios implica que una señal cambia cada 833 microsegundos. Las velocidades de los módems normales son 50, 75, 110, 300, 600, 1200, y 2400. La mayoría de los módems rápidos van a 2400 baudios. Debido a limitaciones del ancho de banda en las líneas telefónicas de voz, las velocidades superiores a 2400 baudios son difíciles de alcanzar, y sólo funcionan en líneas telefónicas de alta calidad. Se pueden codificar varios bits por baudio, para conseguir velocidades que excedan la velocidad del baudio. Se llama “baudio” en honor a Emile Baudot, inventor de la impresora asíncrona del telégrafo (teletipo).

bps

La velocidad en bps es una medida de cuantos bits por segundo son transmitidos. Las velocidades en bps de los módem normales son 50, 75, 110, 300, 1200, 2400, 9600, ... 115200. Usando módems con compresión V.42bis (compresión máxima 4:1), son posibles velocidades *teóricas* de hasta 115200 bps. Esto es lo que la mayoría de la gente quiere decir cuando mal usan la palabra “baudios”.

Así, si módems de alta velocidad están funcionando a 2400 baudios, ¿cómo pueden enviar 14400 bps?. Los módems alcanzan una velocidad en bps mayor que la velocidad en baudios codificando varios bits en cada cambio de señal, o cambio de fase. De este modo, cuando 2 o más bits son codificados por baudio, la velocidad en bps supera a la velocidad en baudios. Si su módem se conecta a 14400 bps, estará enviando 6 bits en cada cambio de fase, a 2400 baudios.

¿Cómo empezó esta confusión?. Bien, volviendo a cuando los módems lentos eran módems de alta velocidad, la velocidad en bps realmente era igual a la velocidad en baudios. Un bit podía ser codificado por cambio de fase. La gente podía usa bps y baudios intercambiándolos, porque eran el mismo número. Por ejemplo, un módem de 300 bps también tenía una velocidad de 300 baudios. Todo esto cambió cuando llegaron los módems más rápidos, y la velocidad en bits superó a la velocidad en baudios.

12.3 ¿Qué son las UARTs? ¿Cómo afectan al rendimiento?

Las UARTs (**U**niversal **A**syncronous **R**eceiver **T**ransmitter) son circuitos integrados de la placa serie del PC. Su propósito es convertir los datos a bits, enviarlos a la línea serie, y después reconstruir los datos en el otro terminal. Las UARTs tratan los datos en bloques del tamaño de un byte, que además es, convenientemente, el tamaño de los caracteres ASCII.

Digamos que tiene un terminal conectado a su PC. Cuando escriba un carácter, el terminal da el carácter a su transmisor (también una UART). El transmisor envía ese byte a la línea serie, un bit cada vez, a una determinada velocidad. En el otro PC, la UART que recibe, coge todos los bits, reconstruye el byte y lo coloca en el búfer.

Existen dos tipos diferentes de UARTs. Probablemente ha oído algo sobre las UARTs no inteligentes - la 8250 y 16450, y las UARTs FIFO - la 16550A. Para entender las diferencias, examinemos primero qué ocurre cuando una UART ha enviado o recibido un byte.

La UART no puede hacer nada con los datos por sí misma, sólo los envía y los recibe. La CPU recibe una interrupción del dispositivo serie cada vez que se recibe o envía un byte. Entonces la CPU mueve el byte recibido desde el búfer de la UART a algún lugar de la memoria, o le da a la UART otro byte a enviar. Las UARTs 8250 y 16450 sólo tienen un búfer de 1 byte. Esto significa, que cada vez que se envía o recibe 1 byte, se interrumpe a la CPU. A velocidades bajas, está bien. Pero, a velocidades de transferencia alta, la CPU está demasiado ocupada tratando con la UART, y no le deja tiempo de atender otras tareas. En algunos casos, la CPU no puede atender la interrupción a tiempo, y se sobreescribe el byte, debido a que llegan demasiado rápido.

Ahí es cuando son útiles las UARTs 16550A. Estos integrados tienen un búfer FIFO de 16 bytes. Esto significa que pueden recibir o transmitir hasta 16 bytes antes de tener que interrumpir a la CPU. La CPU no sólo puede esperar, sino que puede entonces transferir los 16 bytes de una sola vez. Aunque es raro que el umbral de interrupción esté a 16, esto es una ventaja significante frente a las otras UARTs, que sólo tienen un búfer de 1 byte. La CPU recibe menos interrupciones, y está libre para hacer otras cosas. No se pierden datos, y todo el mundo es feliz. (También existe una UART 16550, pero es tratada como un 16450 ya que no funciona bien.)

En general, las UARTs 8250 y 16450 deben valer para velocidades de hasta 38400 bps. A velocidades mayores que 38400 bps, puede empezar a tener pérdida de datos. Otros sistemas operativos de PC (definición usada aquí vagamente) como DOS no son multitarea, por eso deben ser capaces de habérselas mejor con 8250 ó 16450. Por eso es por lo que algunas personas no sufren pérdidas de datos, hasta que cambian a Linux.

Las placas sin UART y las multipuerto inteligentes usan integrados DSP para hacer las tareas de búfer y control adicional, esto libera mucho más la CPU. Por ejemplo, las placas Cyclades Cyclom y Stallion EasyIO usan una UART Cirrus Logic CD1400, y muchas placas usan CPUs 80186 o incluso CPUS RISC especiales, para controlar la E/S serie.

Tenga en mente que este tipo de UART no inteligentes no son malas, simplemente no son buenas para velocidades altas. No debe tener problemas conectando un terminal, o un ratón a estas UARTs. Pero, para un módem de alta velocidad, la 16550A es imprescindible.

Puede comprar placas serie con la UART 16550A por un poco más de dinero, solamente pregunte al vendedor que tipo de UART tiene la placa. O si quiere actualizar su placa, simplemente compre el chip 16550A y reemplácelo por su UART 16450. Son compatibles patilla a patilla. Algunas placas traen la UART en zócalo para este propósito, si no, puede soldarlo. Nótese que probablemente se ahorrará un montón de problemas consiguiendo una placa nueva, si tiene el dinero, cuestan menos de 5000 ptas.

12.4 ¿Cuál es la diferencia real entre los dispositivos `/dev/cua`*N* Y `/dev/ttyS`*N*?

La única diferencia es el modo en que se abren los dispositivos. Los dispositivos para recibir llamadas `/dev/ttyS`*N* se abren en modo bloqueante, hasta que se haya detectado la portadora (CD) (esto es, alguien conecta). Así, cuando alguien quiere usar el dispositivo `/dev/cua`*N*, no hay conflictos con un programa que esté vigilando el dispositivo `/dev/ttyS`*N* (a menos que alguien esté conectado, por supuesto).

Las múltiples entradas `/dev`, permiten operaciones sobre el mismo dispositivo físico con diferentes características de operación. También permite a los programas `getty` estándar coexistir con cualquier otro programa serie, sin que `getty` tenga que ser reajustado con bloqueos de ningún tipo. Es especialmente útil ya que los ficheros de bloqueo de los núcleos Unix estándar, y los bloqueos de UUCP son de aviso y no de obligatoriedad.

13 Eliminando problemas

13.1 Me dice: "line *NNN* of inittab invalid"

Asegúrese de estar usando la sintaxis correcta para su versión de `init`. Las diferentes versiones de `init` que hay por ahí usan sintaxis diferentes en el fichero `/etc/inittab`. Asegúrese de estar usando la sintaxis correcta para su versión de `getty`.

13.2 Cuando intento llamar, dice: "/dev/cua*N*: Device or resource busy" (dispositivo o recurso ocupado)

Este problema puede surgir cuando DCD o DTR no están activados correctamente. DCD sólo debe activarse cuando haya una conexión en curso (ej: alguien ha llamado a este sistema), no cuando `getty` esté vigilando el puerto. Compruebe el módem para asegurarse de que está configurado para activar DCD sólo cuando haya una conexión. DTR debe estar activo siempre que alguien esté usando, o vigilando la línea, como `getty`, `kermit`, o algún otro programa de comunicaciones.

Otra causa común de los errores de "device busy" (dispositivo ocupado), es que haya configurado el puerto serie con una interrupción que ya esté siendo usada. Cuando cada dispositivo se inicializa, le pide permiso a Linux para usar las interrupciones hardware. Linux sigue la pista de a quién se le ha asignado cada interrupción, y si la interrupción ya está siendo usada será imposible que el dispositivo se inicialice correctamente. El dispositivo realmente no tiene muchas formas de avisarle de que esto está ocurriendo, excepto que cuando intente usarlo, dará un mensaje de error "device busy". Compruebe las interrupciones de todas las placas (serie, ethernet, SCSI, etc). Busque conflictos de IRQ.

13.3 Me dice: "Id S*N* respawning too fast: disabled for 5 minutes" (Id S*N* reiniciado demasiado rápido: desconectado durante 5 minutos) Asegúrese de que el módem está configurado correctamente. Mire los registros `E` y `Q`. Esto puede ocurrir cuando el módem está negociando con `getty`. Asegúrese de estar llamando correctamente a `getty` desde `/etc/inittab`. Si usa una sintaxis o nombre de dispositivo incorrectos puede causar graves problemas. Esto también puede ocurrir cuando esté fallando la inicialización de `uugetty`.

13.4 Los dispositivos serie son lentos, o los dispositivo serie sólo pueden enviar en una dirección

Probablemente tenga un conflicto de IRQ. Asegúrese de que no se están compartiendo IRQs. Compruebe todas las placas (serie, ethernet, SCSI, etc). Asegúrese de que los puentes, y los parámetros de `setserial` son los correctos en todos los dispositivos serie. Revise también `/proc/ioports` y `/proc/interrupts` por si hubiera conflictos.

13.5 Mi módem se bloquea después de colgar, o `uugetty` no se reinicia.

Esto puede ocurrir cuando no se reinicia el módem al desactivar el DTR. He visto que los LEDs RD y SD de mi módem se vuelven locos cuando esto ocurre. Debe tener el módem reiniciado. Muchos módems compatible Hayes hacen esto con `&D3`, pero en mi USR Courier, he tenido que poner `&D2` y `S13=1`. Mire en el manual de su módem.

13.6 Tengo un terminal conectado al PC, pero después de escribir un nombre de registro, sólo se bloquea

- Si está usando `getty`: Probablemente no tendrá puesto `CLOCAL` en ninguna línea de `/etc/gettydefs` para el terminal, y probablemente no está usando un cable completo de módem nulo. Necesita `CLOCAL`, el cual le dice a Linux que ignore las señales del control del módem. Debería parecerse a esto:

  ```
  # 38400 bps, entrada para un Terminal no inteligente
  DT38400# B38400 CS8 CLOCAL # B38400 SANE -ISTRIP CLOCAL #@S @L login: #DT38400

  # 19200 bps, entrada para un Terminal no inteligente
  DT19200# B19200 CS8 CLOCAL # B19200 SANE -ISTRIP CLOCAL #@S @L login: #DT19200

  # 9600 bps, entrada para un Terminal no inteligente
  DT9600# B9600 CS8 CLOCAL # B9600 SANE -ISTRIP CLOCAL #@S @L login: #DT9600
  ```

  Después, elimine el proceso `getty` con el comando `kill` y se generará un proceso nuevo con la nueva entrada.

- Si está usando `agetty`: Añada el parámetro `-L` a la línea de `agetty` en `/etc/inittab`. Esto hará que ignore las señales de control del módem. Después ejecute de nuevo `init` escribiendo `init q`. La línea debería ser como esta:

  ```
  s1:345:respawn:/sbin/agetty -L 9600 ttyS1 vt100
  ```

13.7 A velocidades altas, mi módem pierde datos

Si está intentando usar el módem a más de 38400 bps, y no tiene una UART 16550A, debe conseguirla. Vea la sección 12.3 para saber más sobre las UARTs.

13.8 Al arrancar, linux no informa a los dispositivos serie de cómo los he configurado

Esto es verdad. Linux no realiza ninguna detección de IRQ al arrancar, sólo hace la detección de los dispositivos serie. Así que no haga caso de lo que diga sobre las IRQs, ya que asume que son las IRQs estándar. Esto se hace porque la detección de IRQs no es fiable, y puede ser falsa.

Así, aún cuando tengo mi `ttyS2` en la IRQ5, me sale

```
Jan 23 22:25:28 misfits vmunix: tty02 at 0x03e8 (irq = 4) is a 16550A
```

cuando Linux arranca.

Tiene que usar `setserial` para decirle a Linux la IRQ que está usando. Después de que Linux arranque, puede mirar en el fichero `/proc/interrupts` para ver que IRQs se han configurado realmente.

13.9 rz y/o sz no funcionan cuando llamo a mi máquina linux con un módem

Si Linux busca `/dev/modem` cuando intenta enviar un fichero, mire en `/etc/profile`, y `/etc/csh.cshrc`. Algunas distribuciones definen ahí muchos alias, sobre todo Slackware. Estos alias echan a perder los programas zmodem. Elimínelos o corríjalos.

13.10 En mi pantalla están saliendo caracteres muy graciosos

Esto ocurre en las consolas virtuales cuando envía datos binarios a la pantalla, o a veces en conexiones serie. La forma de arreglar esto es escribiendo `echo ^v^[c`. Para los que son incapaces de identificar los caracteres de control, es:

```
linux% echo <ctrl>v<esc>c
```

13.11 `getty` o `uugetty` no funciona todavia

Existe la opción `DEBUG` que viene con `getty_ps`. Edite el fichero de configuración `/etc/conf.{uu}getty.ttyS`*N* y añada `DEBUG=`*NNN*. Donde *NNN* es una de las combinaciones numéricas siguiente, dependiendo de lo que quiera depurar:

```
D_OPT   001          activacion de las opciones
D_DEF   002          procesamiento del fichero de opciones por defecto
D_UTMP  004          procesamiento de utmp/wtmp
D_INIT  010          inicializacion de la linea (INIT)
D_GTAB  020          procesamiento del fichero gettytab
D_RUN   040          otros diagnosticos de ejecucion
D_RB    100          depuracion de rellamada
D_LOCK  200          procesamiento de bloqueo de uugetty
D_SCH   400          procesamiento de tareas
D_ALL   777          todo lo anterior
```

Poniendo `DEBUG=010` es una buena forma de empezar.

Si está ejecutando `syslogd`, la información de depuración aparecerá en los ficheros log. Si no está usando `syslogd` la información aparecerá en `/tmp/getty:tyyS`*N* si depura `getty` y `/tmp/uugetty:ttyS`*N* si usa `uugetty`, y en `/var/adm/getty.log`. Mire la información de depuración y vea que está ocurriendo. Probablemente necesitará ajustar algunos parámetros del fichero de configuración, y reconfigurar el módem.

También lo puede intentar con `mgetty`. Algunas personas tienen mejores resultados con él.

14 Otras fuentes de información

- páginas man de: `agetty(8)`, `getty(1m)`, `gettydefs(5)`, `init(1)`, `login(1)`, `mgetty(8)`, `setserial(8)`
- El manual del módem
- NET-2 HOWTO: todo sobre redes, incluido SLIP, CSLIP, y PPP
- PPP HOWTO: ayuda para PPP
- Printing HOWTO: para configurar una impresora serie
- Term HOWTO: todo lo que quería saber sobre el programa `term`
- UPS HOWTO: configuración de una UPS conectada al puerto serie
- UUCP HOWTO: información sobre como configurar UUCP

  Grupos de noticias de Usenet:

```
comp.os.linux.answers
FAQs, How-To's. READMEs, etc. sobre Linux.
comp.os.linux.hardware
Compatibilidad de hardware con el sistema operativo Linux.
comp.os.linux.networking
Redes y comunicaciones bajo Linux.
comp.os.linux.setup
Instalaci\'{o}n de Linux y administracion de sistemas.
```

- La lista de correo serie en Linux. Para apuntarse, mande un mail a `majordomo@vger.rutgers.edu`, con "`subscribe linux-serial`" en el cuerpo del mensaje, obtendrá un mensaje de ayuda. El servidor también ofrece otras listas de Linux. Envíe el comando "`lists`" para obtener una lista de las listas de correo.
- Un *libro blanco* para discutir sobre comunicaciones serie y placas serie multipuerto está a su disposición en Cyclades en `http://www.cyclades.com`.
- Modem FAQs:
  Navas 28800 Modem FAQ

15 Contribuciones

Es imposible escribir este HOWTO solo. Aunque una gran parte de este HOWTO lo he escrito yo, he tenido que reescribir muchas contribuciones para mantener la continuidad del estilo y flujo de escritura. Gracias a todos los que han contribuido o comentado algo, la lista de personas es demasiado larga para ponerla aquí (más de cien). En especial quiero darle las gracias a Ted T'so por responder a preguntas sobre los controladores serie, a Kris Gleason que suele mantener `getty_ps`, y a Gert Döring que mantiene `mgetty`.

16 Anexo: El INSFLUG

El *INSFLUG* forma parte del grupo internacional *Linux Documentation Project*, encargándose de las traducciones al castellano de los Howtos (Comos), así como la producción de documentos originales en aquellos casos en los que no existe análogo en inglés.

En el **INSFLUG** se orienta preferentemente a la traducción de 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 la sede del INSFLUG encontrará siempre las **últimas** versiones de las traducciones: `www.insflug.org`. Asegúrese de comprobar cuál es la última versión disponible en el Insflug antes de bajar un documento de un servidor réplica.

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

Francisco José Montilla, `pacopepe@insflug.org`.