



EL KONSULTORIO DE KLAUS



Klaus Knopper es el creador de Knoppix y co-fundador de la LinuxTag Expo. En la actualidad trabaja como profesor, programador y consultor. Si tiene algún problema de configuración, o simplemente quiere conocer mejor cómo funciona Linux, no dude en escribir sus preguntas a: klaus@linux-magazine.com

Asus con Hardy

? Mi equipo de sobremesa, basado en la placa Asus M2V-MX, se queda en modo suspendido correctamente con Windows Vista, sin embargo no se suspende con Ubuntu Hardy. He cambiado las versiones de S3 y ACPI en la BIOS para ver si funciona. Aunque esto causa algunas diferencias en su comportamiento, el sistema sigue sin funcionar con Ubuntu. ¿Qué puedo hacer?

💡 Echo en falta alguna información en su descripción que ayudaría a resolver el problema.

¿Se refiere a “suspender a RAM” (STR) o a “suspender a disco” (S2D)?

¿El problema es que al suspender simplemente no pasa nada, o es que el equipo no se despierta desde el modo suspendido? Haré un breve resumen de lo que se necesita hacer para utilizar la suspensión a RAM y la suspensión a disco.

Para usar la suspensión a disco necesitamos habilitar esta funcionalidad en el kernel. (Supongo que esto ya está activado en Ubuntu). Véase la Figura 1.

Además de esto, el kernel también necesita saber cómo y dónde debe guardar el contenido de la RAM del equipo mientras está suspendido.

El “dónde” se configura en la línea de comandos de arranque del kernel. La mía es algo así:

```
$ cat /proc/cmdline
... resume=/dev/hda3 ...
```

Con esta opción de arranque, el kernel sabe que la partición, que probablemente contiene un sistema suspendido, es la tercera del primer disco duro IDE en mi configuración. Al arrancar, el kernel intentará restaurar automáticamente una sesión guardada en esta partición cuando está presente esta opción. Para modificar la línea de comandos de arranque del kernel tenemos que cambiar la línea `APPEND` en la configuración de nuestro cargador de arranque (`/etc/lilo.conf` para LILO, al que se llama tras

realizar algún cambio, o `/boot/grub/grub.cfg` o `menu.lst` en el caso de grub). Como último recurso, podemos teclear la opción en el modo interactivo del cargador de arranque.

Respecto a la partición de swap, que también se convierte en una partición de “hibernación”, tenemos que reservar espacio del tamaño aproximado de nuestra RAM.

Por tanto, verifique si su espacio de swap libre es suficientemente grande para albergar el contenido de la memoria de su ordenador (véase el Listado 1).

El ejemplo del Listado 1 muestra que este ordenador tiene aproximadamente 1.2 GB de RAM, y 1.9 GB de swap, por lo que es posible la suspensión a disco.

En su configuración, si el espacio de swap es menor que la RAM más el swap en uso, la suspensión a disco simplemente fallará y tendremos que reparticionar el disco duro para conseguir una partición de swap con la que pueda funcionar.

Tan pronto como el swap y la línea de comandos de arranque estén configurados, podemos iniciar el procedimiento de suspensión indicando al kernel que suspenda y que guarde el sistema en el disco.

Antes de hacer esto, es buena idea realizar un par de comandos `sync` para evi-

Listado 1: Espacio Swap Libre

```
01 $ grep MemTotal /proc/meminfo
02 MemTotal: 1295172 kB
03
04 $ cat /proc/swaps
05 Filename Type Size Used Priority
06 /dev/hda3 partition 1959920 664036 -1
```

tar que información aún no escrita se pierda debido a problemas al despertar. (Enviar uno sólo podría no ser suficiente debido a su naturaleza asíncrona). Para hacer esto debemos tener permisos de root:

```
# sync; sync; sync
# echo shutdown >
  /sys/power/disk
# echo disk > /sys/power/state
```

Ahora el kernel debería conmutar a modo consola de texto y mostrar lo que está haciendo. En caso de fallo, veremos un mensaje de error en la salida de *dmesg*.

Dependiendo de lo bien o mal que funcione ACPI en nuestro equipo, usando

```
# echo platform >
  /sys/power/disk
# echo disk >
  /sys/power/state
```

en lugar de la sugerencia previa podría incluso funcionar mejor, ya que la BIOS ACPI puede saber ahora cómo conmutar dispositivos periféricos a modo suspendido de manera que se despierten correctamente más tarde.

Tras congelar los procesos, el kernel escribirá el estado del sistema completo en la partición de swap y apagará el sistema.

Cuando vuelva a arrancar, buscará una firma de hibernación en la partición de swap y cargará el sistema suspendido desde allí en caso de que sea posible.

A la hora de restaurar, algunas cosas podrían ir mal:

1. Algún hardware puede no reiniciarse adecuadamente al despertar de este modo suspendido, de manera que el sistema se cuelgue. Se sabe que esto ocurre con drivers de sonido no suspendibles. Para mayor seguridad, debemos asegurarnos de que nuestro script de suspensión/reanudación descarga todos los drivers que se sabe que son inestables tras la reanudación. Un ejemplo (antes de suspender):

```
fuser -k /dev/dsp
# Stop sound output
rmmmod
snd_intel8x0
# Unload onboard sound driver
```

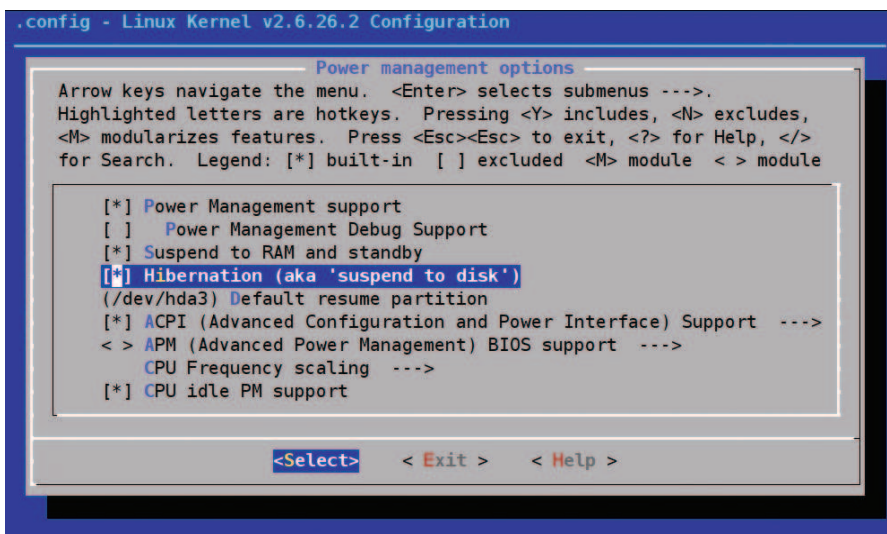


Figura 1: Captura de pantalla de la configuración del kernel relativa a la suspensión a disco.

2. Debido a la manera en la que el kernel haya sido configurado en una distribución específica, la partición de swap podría no estar disponible suficientemente pronto para que sea posible leerla. Por ejemplo, si los controladores del disco duro no están compilados estáticamente dentro del kernel, éste podría no tener oportunidad de inicializar el procedimiento de recuperación posteriormente si el disco RAM inicial que contiene los drivers no lo ha hecho explícitamente.

De nuevo, el log de arranque del kernel mostrado por *dmesg* puede ofrecer pistas acerca de lo que ha ido mal.

Para suspender a RAM (en caso de estar soportado), usaremos *mem* en lugar de *disk* en los anteriores comandos de suspensión.

Aparentemente, para esto no necesitamos una partición de suspensión. El modo *standby* también podría ser una opción, aunque solamente se reduciría el consumo de energía de algunos dispositivos, pero no se suspendería realmente el sistema:

```
# echo mem >
  /sys/power/state
```

o

```
# echo standby >
  /sys/power/state
```

En lugar de usar la más reciente interfaz */sys/power/...*, también podemos probar con el archivo que controla los estados de suspensión de ACPI:

```
# echo 4 >
  /proc/acpi/sleep
# Suspend-to-disk
```

o

```
# echo 3 >
  /proc/acpi/sleep
# Suspend-to-ram
```

Banda Ancha Inalámbrica USB


? Dispongo de tres módems USB de banda ancha inalámbrica Huawei HSDPA CE0682. No consigo que funcionen en Fedora 8. ¿Podría explicarme por qué es tan difícil añadir cualquier módem USB a las plataformas Fedora, Red Hat y Ubuntu?

💡 Me temo que sólo el fabricante original del hardware sabe porqué sólo proporciona drivers para Windows con su hardware. Si el “módem” tiene un modo DHCP en el cual distribuye direcciones automáticamente a un ordenador conectado por Ethernet, podría utilizarlo sin ningún driver especial. En el caso de protocolos USB propietarios que sólo conoce el fabricante del hardware, es una tarea muy tediosa – y en absoluto rentable – desarrollar un driver mediante ingeniería inversa. Por eso, casi nadie lo hará gratuitamente.

La decisión estratégica de una compañía de vender hardware que sólo funciona en un sistema operativo específico no tiene una solución técnica razonable. El mejor consejo que puedo ofrecerle es que devuelva el dispositivo aparentemente defectuoso haciendo uso de su

garantía (¿no es cierto que no funciona?) y reclamar la devolución del dinero.

Puppy en un Pendrive

 Soy estudiante universitario y usuario de Linux amateur. ¿Podría indicarme cómo instalar *mbr.bin* en mi pendrive? Quiero arrancar una versión de Pendrivelinux desde este dispositivo. ¿Es posible instalar Puppy 3.10? No me funciona cuando trato de encontrarlo con la instrucción:


```
locate mbr.bin
```

Igualmente, me gustaría saber si es posible instalar SysLinux con el siguiente comando:

```
syslinux -s /dev/sda1
```

Ya he visitado la página *pendrivelinux.com* y he seguido las instrucciones, pero siempre tengo problemas al arrancar desde mi pendrive.

Otro problema es que las máquinas a las que tengo acceso sólo tienen 256 MB de RAM, por lo que no puedo arrancar Ubuntu 7.10 ó 8.04. Si he entendido las cosas correctamente, con esas versiones de Ubuntu es posible instalar LILO y Syslinux.

 Para crear un pendrive USB arrancable en FAT16/FAT32 sólo son necesarios unos pocos comandos. En este ejemplo, la unidad USB está presente en */dev/sdc* (véase el Llistado 2).

Esta tabla de partición ya está preparada para el modo “arranque desde disco duro USB” de la BIOS. La primera partición es de tipo FAT32 y está marcada como “arrancable”.

Si sólo falta el flag “arrancable” en la primera partición, se puede añadir con:

```
# sfdisk -A1 /dev/sdc
```

Generalmente, no se necesita un sector de arranque especial como *mbr.bin* al que hace referencia. Pero si su ordenador necesita uno y ha encontrado un sector de arranque personalizado que se recomienda para algún propósito específico (debe tener un tamaño máximo de sólo 488 bytes), podemos instalar este sector de arranque personalizado copiándolo en el archivo de dispositivo:

```
# cat mbr.bin > /dev/sdc
```

Listado 2: Ejemplo de Pendrive

```
01 # fdisk -l /dev/sdc
02 Disk /dev/sdc: 8021 MB, 8021606400 bytes
03 16 heads, 32 sectors/track, 30600 cylinders
04 Units = cylinders of 512 * 512 = 262144 bytes
05 Disk identifier: 0x10d72c95
06
07 Device Boot Start End Blocks Id System
08 /dev/sdc1 * 16 30600 7829568 c W95 FAT32 (LBA)
```

Debe esperar unos instantes hasta que la nueva información se haya copiado en disco o bien forzarlo mediante el comando *sync* o con *eject*.

Tras esto, verificamos si el particionado del pendrive aún es correcto. Para hacer saber al kernel que la información de particionado ha cambiado, puede que tengamos que reconectar la unidad. La manera más fácil de hacer esto es con:

```
# eject /dev/sdc
# eject -t /dev/sdc
```

o bien podemos sacar físicamente la unidad y reinsertarla.

En lugar de instalar un sector de arranque personalizado, recomiendo que lo intente primero con uno estándar:

```
# ms-sys -s /dev/sdc
```

Para instalar el cargador de arranque SysLinux en la primera partición, el comando necesario es:

```
# syslinux /dev/sdc1
```

Nótese que en realidad debe teclear el nombre de su partición, no el del dispositivo entero.

SysLinux es una especie de “programa DOS”, escrito en una ubicación específica y arrancable en la partición. Por tanto, no podemos simplemente “montar y copiar” *dlinux.sys* en la unidad.

Para pendrives USB no debería ser necesaria la opción *-s* (baja) de SysLinux. Sin embargo, si ni siquiera se muestra la línea de comandos de arranque de SysLinux, pero el pendrive parece que es accedido, merece un intento.

SysLinux requiere un archivo de configuración, *syslinux.cfg*, que le indica a SysLinux qué tipo de sistema operativo debe iniciar y, posiblemente, que cargue también un disco RAM.

Lo que sigue es un ejemplo del contenido de *syslinux.cfg*:

```
default linux
timeout 10
prompt 1
label linux
kernel linux
append initrd=minirt.gz
root=/dev/ram0
```

El kernel de Linux (*linux*) y el disco RAM inicial (*minirt.gz*) deben copiarse a la misma partición que contenga el cargador de arranque. El sistema de archivos de Linux puede ubicarse en una partición separada, formateada con un sistema de archivos de Linux como ext2, ext3 o ReiserFS, si es que no se instala como un archivo de imagen loopback.

En caso de instalar Knoppix en un pendrive, simplemente se copiaría el directorio *KNOPPIX* en la partición FAT(32) del mismo, el cual contiene toda la información y los programas:

```
# mkdir -p /media/sdc1
# mount -t vfat -o
shortname=winnt /dev/sdc1
/media/sdc1
# cp -rL syslinux.cfg
linux minirt.gz KNOPPIX
... /media/sdc1/
# umount /media/sdc1
```

Nótese la opción de montaje *shortname=winnt*, que crea nombre de archivo en la partición FAT con deletreo en mayúsculas y minúsculas, justo como en el nombre de archivo original. Muchas distribuciones Live son quisquillosas con los nombres de archivo en mayúsculas y minúsculas.

Una vez que se han copiado todos los archivos necesarios para ejecutar la distribución (y que hayamos creado una partición de Linux que contenga el sistema de archivos de la distribución), ya podremos arrancar desde el pendrive. ■