



Klaus Knopper is the creator of Knoppix and co-founder of the LinuxTag expo. He currently works as a teacher, programmer, and consultant. If you have a configuration problem, or if you just want to learn more about how Linux works, send your questions to:

klaus@linux-magazine.com

ASK KLAUS!



suspend mode? I'll give you a short overview on what needs to be done to use suspend to RAM and suspend to disk here.

To use suspend to disk, you need to enable this feature in the kernel. (I assume that this is already done in Ubuntu.) See Figure 1.

For suspend to disk, the kernel also needs to know how and where to store the content of your notebook's RAM while it is in power-off mode.

The "where" part is configured at the kernel boot command line. Mine looks like this:

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

With this preset boot option, the kernel knows that the partition, which possibly contains a suspended system, is the third partition on the first IDE hard disk in my setup. On boot, the kernel will automatically try to restore a saved session from this partition when this option is present. To modify the boot kernel command line, you have to change the *APPEND* line in your bootloader's setup (*/etc/lilo.conf* for LILO, with LILO being called after a change, or */boot/grub/grub.cfg* or *menu.lst* for the grub bootloader). As a last resort, you can type the option in the bootloader's interactive mode.

On the swap partition, which will also become the "hibernate" partition, you need to reserve space about the size of your RAM.

So, please check to see whether your free swap space

is large enough to hold the content of your computer's memory (see Listing 1).

The example in Listing 1 shows that this computer has about 1.2GB RAM, and 1.9GB of swap, so suspend to disk is possible.

In your setup, if the available swap space is lower than RAM plus swap in use, then suspend to disk will just fail and you would have to repartition the hard drive for a larger swap partition to make it work.

As soon as swap and the boot command line are set up, you can start the suspend procedure by telling the kernel to freeze then store the running system to disk.

Before this, sending a couple of *sync* commands is a good idea, just in case that unwritten data gets lost because of a failed wake up later. (Sending only one might not be sufficient because of its asynchronous nature.) You must be root for this procedure:

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

Now the kernel should switch to text console mode and display what it's doing. In case of failure, you will see an error message in the output of *dmesg*.

Asus with Hardy

? My Asus M2V-MX-based desktop suspends well in Windows Vista, but it does not suspend in Ubuntu Hardy. I have changed S3 and ACPI versions in the BIOS to see if that works. Although it causes some differences in behavior, the system still does not work in Ubuntu. What should I do?

💡 Some information is missing in your description that could help solve the problem.

Do you mean "suspend to RAM" (STR) or "suspend to disk" (S2D)?

Is the problem that suspend just does not do anything, or is the problem that the notebook does not wake up from

Listing 1: Free swap space

```
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
```

Depending on how well ACPI works on your notebook, using

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

instead of the previous suggestion might work even better because the ACPI BIOS might know how to switch peripheral devices to suspend mode in order to wake up correctly later.

After freezing processes, the kernel will write the entire system state to the swap partition and power off the system.

On next boot, the kernel will search for a hibernate signature on the swap partition and load the suspended system from there if at all possible.

Some things that could go wrong when resuming:

1. Some hardware does not reinitialize properly when being awoken from suspend, so your system freezes. This result is known to happen with non-suspendable sound drivers. To be on the safe side, make sure that in your own suspend/resume script, all drivers known to be unstable after wake up are unloaded. Example (before suspend):

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

2. Because of the way the kernel has been configured in a specific distribution, the swap partition might not be accessible early enough to be able to

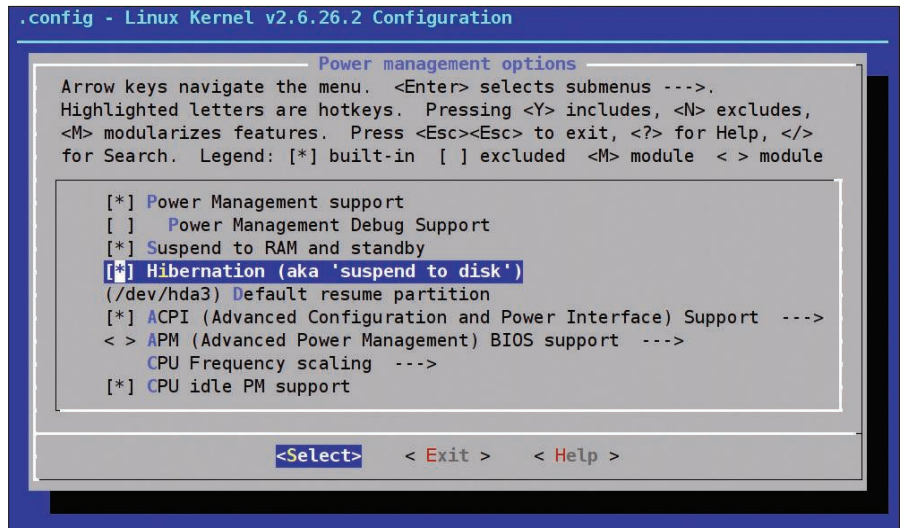


Figure 1: Screen shot from the kernel config regarding suspend to disk.

read from it. For example, if hard disk controller drivers are not statically compiled into the kernel, the kernel might not have a chance to initialize the resume procedure later if the initial ramdisk containing the drivers does not explicitly do this.

Again, the boot/kernel log shown by *dmesg* might give a hint about what went wrong.

To suspend to RAM (if supported), use *mem* instead of *disk* in the suspend commands above.

Apparently, you don't need a suspend partition for this; *standby* could also be an option, which will only reduce power consumption of some devices but not really suspend the system:

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

or

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

Instead of using the newer */sys/power/...* interface, you can also try the ACPI sleep states control file:

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

or

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

USB Wireless Broadband



I have a three USB wireless broadband Huawei HSDPA CE0682 modem. I cannot get it to work on the Fedora 8. Can you explain why it is difficult to add any USB modem to the Fedora, Red Hat, and Ubuntu platforms?



I'm afraid that only the original hardware manufacturer knows why they only provide Windows drivers for their hardware. If the "modem" has a DHCP mode in which it distributes addresses to an Ethernet-networked computer automatically, you should be able to use it without any special drivers. For closed, proprietary USB protocols that only the hardware manufacturer knows about, it is a very tiresome – and in no way cost-effective – task to develop a driver via reverse engineering. Therefore, almost nobody will do this for free.

A company's strategic decision to sell hardware that works only for a specific operating system does not have a reasonable technical solution. The best advice I could give is to return the apparently defective device as a warranty case (it does not work, does it?) and claim your money back.

Puppy on Pen Drive



I am a university student and an amateur Linux user. Can you tell me how to install *mbr.bin* on my pen drive? I want to boot a version of Pendrivelinux from this device. Is it possible to install from Puppy 3.10? It never



works when I try to find it with the instruction:

```
locate mbr.bin
```

Also, I would like to know whether it is possible to install SysLinux with the following command:

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

I have already visited *pendrivelinux.com* and followed the instructions, but I always have problems booting from my pen drive.

Another problem is that the machines I have access to only have 256MB of RAM, so I can't boot into Ubuntu 7.10 or 8.04. If I understand things correctly, with these versions of Ubuntu, it is possible to install LILO and SysLinux.



Making a FAT16/FAT32-formatted USB pen bootable only takes a few commands. In this example, the pen drive is present as */dev/sdc* (Listing 2).

This partition table is already perfect for "boot from USB hard disk" BIOS mode. The first partition is of FAT(32) type and marked as "bootable."

If just the "bootable" flag for the first partition is missing, it can be added with:

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

Usually, you do not need a special boot record like the *mbr.bin* you asked about. But if your computer needs one and you have found a customized boot record file recommended for a specific purpose (it should be a maximum of only 488 bytes), you can install this custom boot record by just copying it over to the device file:

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

You must wait for a while until new data has



Listing 2: Pen drive example							
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)

been flushed to disk or enforce it by calling *sync* or the *eject* commands below.

After this, check to see whether the partitioning of the pen drive is still OK. To let the kernel know that partitioning data has been changed, you might have to reconnect the drive. The easiest way to do so is with

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

or you can physically remove and re-insert the drive.

Instead of installing a custom master boot record, I recommend that you first try a standard one:

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

For installing the SysLinux bootloader on the first partition, the command is:

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

Note that you really have to type the partition name here, not the entire device name.

SysLinux is a kind of "DOS program," just written to a specific, bootable location on the partition; therefore, you cannot just "mount and copy" *ldlinux.sys* to the drive.

For USB pen drives, the *-s[low]* option of SysLinux should not be necessary;

however, if not even the SysLinux boot command line is displayed, but the pen drive seems to be accessed, it is worth a try.

SysLinux requires a configuration file, *syslinux.cfg*, which tells SysLinux what kind of operating system to start and, possibly, also to load a ramdisk.

The following is an example of *syslinux.cfg* contents:

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

Linux kernel (*linux*) and initial ramdisk (*minirt.gz*) must be copied to the same partition containing the bootloader. The Linux filesystem itself can be located on a separate partition, formatted with a Linux filesystem such as ext2, ext3, or ReiserFS, if it is not installed as a loop-back image file.

In the case of installing Knoppix on a pen drive, you would just copy the *KNOPPIX* directory to the FAT(32) partition of the pen drive, which contains all the data and programs:

```
# 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
```

Please note the *shortname=winnt* mount option, which will create file names on the FAT partition with lower- and uppercase spelling, just as in the original file name. Most Live distributions are picky about upper- and lower-case in file names.

After all files needed to run the distribution have been copied (and you have created a Linux partition containing the distribution's filesystem), you should be able to boot from the pen drive. ■

Send your Linux questions to klaus@linux-magazine.com.