

If you say yes here, you will need to run *make modules_install* to put the modules under */lib/modules* where the module tools can find them.

IOSCHED_NOOP

No-op I/O scheduler

The no-op I/O scheduler is a minimal scheduler that does basic merging and sorting. Its main uses include nondisk-based block devices such as memory devices and specialized software or hardware environments that do their own scheduling and require only minimal assistance from the kernel.

IOSCHED_AS

Anticipatory I/O scheduler

The anticipatory I/O scheduler is the default disk scheduler. It is generally a good choice for most environments, but is quite large and complex compared to the deadline I/O scheduler. It can also be slower in some cases, especially under some database loads.

IOSCHED_DEADLINE

Deadline I/O scheduler

The deadline I/O scheduler is simple and compact. It is often as good as the anticipatory I/O scheduler, and under some database workloads, even better. In the case of a single process performing I/O to a disk at any one time, its behavior is almost identical to the anticipatory I/O scheduler and so is a good choice.

IOSCHED_CFQ

CFQ I/O scheduler

The CFQ I/O scheduler tries to distribute bandwidth equally among all processes in the system. It should provide a fair working environment, suitable for desktop systems.

SMP

Symmetric multiprocessing support

This enables support for systems with more than one CPU. If you have a system with only one CPU, like most personal computers, say no. If you have a system with more than one CPU, say yes.

If you say no here, the kernel will run on single and multiprocessor machines, but will use only one CPU of a multiprocessor machine. If you say yes here, the kernel will run on many, but not all, single-processor machines. On a single-processor machine, the kernel will run faster if you say no here.

Note that if you say yes here and choose architecture 586 or Pentium under Processor family, the kernel will not work on 486 architectures. Similarly, multiprocessor kernels for the PPro architecture may not work on all Pentium-based boards.

Crusoe

Choose this if you have a Transmeta Crusoe series processor.

Efficeon

Choose this if you have a Transmeta Efficeon series processor.

Winchip-C6

Choose this if you have an original IDT Winchip processor.

Winchip-2

Choose this if you have an IDT Winchip 2 processor.

Winchip-2

Choose this if you have an IDT Winchip processor with 3DNow! capabilities.

GeodeGX1

Choose this if you have a Geode GX1 (Cyrix MediaGX) processor.

Geode GX/LX

Choose this if you have an AMD Geode GX or LX processor.

Cyrix III/VIA C3

Choose this if you have a VIA Cyrix III or VIA C3 processor.

VIA C3-2

Choose this if you have a VIA C3-2 "Nehemiah" (model 9 and above) processor.

If you don't know what to do, choose 386.

X86_GENERIC

Generic x86 support

Instead of just including optimizations for the selected x86 variant (e.g., PII, Crusoe, or Athlon), include some more generic optimizations as well. This will make the kernel perform better on x86 CPUs other than the one selected.

This is really intended for distributors who need more generic optimizations.

NR_CPUS

Maximum number of CPUs (2-255)

This allows you to specify the maximum number of CPUs that this kernel will support. The maximum supported value is 255 and the minimum value that makes sense is 2.

This option is purely to save memory; each supported CPU adds approximately 8 KB to the kernel image.

SCHED_SMT

SMT (HyperThreading) scheduler support




SMT scheduler support improves the CPU scheduler's decision-making on Intel Pentium 4 chips with HyperThreading, at a cost of slightly increased overhead in some places.

memory, not all of it can be permanently mapped by the kernel. The physical memory that's not permanently mapped is called *high memory*.

If you are compiling a kernel that will never run on a machine with more than 1 GB total physical RAM, answer `off` here (the default choice, and suitable for most users). This will result in a 3 GB/1 GB split: 3 GB are mapped so that each process sees a 3 GB virtual memory space and the remaining part of the 4 GB virtual memory space is used by the kernel to permanently map as much physical memory as possible.

If the machine has between 1 and 4 GB physical RAM, answer 4GB here.

If more than 4 GB is used, answer 64GB here. This selection turns Intel PAE (Physical Address Extension) mode on. PAE implements three-level paging on IA32 processors. PAE is fully supported by Linux, and PAE mode is implemented on all recent Intel processors (Pentium Pro and better).

 If you say 64GB here, the kernel will not boot on CPUs that don't support PAE!







The actual amount of total physical memory will either be autodetected or can be forced by using a kernel command line option such as `mem=256M`. (See Chapter 9 for details about how to pass options to the kernel at boot time, and what options are available.)

If unsure, say `off`.

HIGHMEM4G	4GB Select this if you have a 32-bit processor and between 1 and 4 GB of physical RAM.
HIGHMEM64G	64GB Select this if you have a 32-bit processor and more than 4 GB of physical RAM.
FLATMEM_ MANUAL	Flat memory This option allows you to change some of the ways that Linux manages its memory internally. Most users will see only have one option here: <code>FLATMEM</code> . This is normal and a correct option. Some users of more advanced features, such as NUMA and memory hotplug, may have different options here. <code>DISCONTIGMEM</code> is a more mature, better tested system, but is incompatible with memory hotplug and may suffer decreased performance over <code>SPARSEMEM</code> . If

the exact hardware interface is strongly in flux, so no good recommendation can be made.

HOTPLUG_CPU	<p>Support for hot-pluggable CPUs (experimental)</p> <p>Say yes here to experiment with turning CPUs off and on, and to enable suspend on SMP systems. CPUs can be controlled through the <code>/sys/devices/system/cpu</code> interface.</p>
PM	<p>Power management support</p> <p>Power management allows parts of your computer to shut off or be put into a power-conserving sleep mode if they are not being used. There are two competing standards for doing this: APM and ACPI. If you want to use either one, say yes here and then also enable one of those two standards.</p> <p>Power management is most important for battery-powered laptop computers; if you have a laptop, check out the Linux Laptop home page at http://www.linux-on-laptops.com, Tuxmobil-Linux on Mobile Computers at http://www.tuxmobil.org, and the "Battery Powered Linux" mini-HOWTO at http://www.tldp.org/docs.html#howto.</p> <p>Note that, even if you say no here, Linux on the x86 architecture will issue the HLT instruction if nothing is being done, thereby sending the processor to sleep and saving power.</p>
SOFTWARE_SUSPEND	<p>Software suspend</p> <p>Enable machine suspension.</p> <p>When the machine is suspended, an image is saved in your active swap. Upon next boot, pass the <code>resume=/dev/swappartition</code> argument to the kernel to have it detect the saved image, restore memory state from it, and continue to run as before. If you do not want the previous state to be reloaded, use the <code>noresume</code> kernel argument. However, note that your partitions will be <i>fsck'd</i> and you must issue <i>mkswap</i> on your swap partitions again. The procedure does not work with swap files.</p> <p>Right now you may boot without resuming and then resume later, but in the meantime you cannot use those swap partitions/files that were involved in suspending. In this case, also, there is a risk that buffers on disk won't match with saved ones.</p> <p>For more information, see <i>Documentation/power/swsusp.txt</i>.</p>
ACPI	<p>ACPI Support</p> <p>Advanced Configuration and Power Interface (ACPI) support for Linux requires ACPI-compliant hardware and firmware, and assumes the presence of OS-directed configuration and power</p>

CPU_FREQ_GOV_POWERSAVE	<p> powersave CPUFreq policy governor</p> <p>This sets the frequency statically to the lowest available CPU frequency.</p>
CPU_FREQ_GOV_USERSPACE	<p> userspace CPUFreq policy governor</p> <p>Enable this CPUFreq policy governor either when you want to set the CPU frequency manually or when a userspace program should be able to set the CPU dynamically, as on LART (http://www.lart-maker.nl).</p> <p>For details, take a look at <i>Documentation/cpu-freq</i>.</p>
CPU_FREQ_GOV_ONDEMAND	<p> ondemand CPUFreq policy governor</p> <p>This driver adds a dynamic CPUFreq policy governor. The governor polls the CPU and changes its frequency based on CPU utilization. Support for this governor depends on the CPU's ability to do fast frequency switching (i.e., very low latency frequency transitions).</p> <p>For details, take a look at <i>Documentation/cpu-freq</i>.</p>
CPU_FREQ_GOV_CONSERVATIVE	<p> conservative CPUFreq policy governor</p> <p>This driver is similar to the Ondemand governor both in its source code and its purpose. The difference is that the Conservative governor is optimized for a battery-powered system. The frequency is gracefully increased and decreased rather than jumping to 100 percent when speed is required.</p> <p>If you are using a laptop, a PDA, or an AMD64-based computer (due to the unacceptable step-by-step latency issues between the minimum and maximum frequency transitions in the CPU), you will probably want to use this governor. If you have a desktop machine, consider the Ondemand governor instead.</p> <p>For details, take a look at <i>Documentation/cpu-freq</i>.</p>
PCI	<p>PCI support</p> <p>PCI is a bus system used by the processor to talk to internal devices and add-on cards. It is extremely common and found in almost all modern computers.</p> <p>Say yes to this option unless you have a special reason not to.</p>
PCCARD	<p>PCCard (PCMCIA/CardBus) support</p> <p>Say yes here if you want to attach PCMCIA or PC cards to your Linux computer. These are credit-card size devices such as network cards, modems, or hard drives often used with laptop computers.</p>

sockets even if your machine is not connected to any network. Unless you are working on an embedded system or something similar, you definitely want to say yes here.

INET

TCP/IP networking

These are the protocols used on the Internet and on most local Ethernets. It is highly recommended that you say yes here, since some programs (e.g., the X Window System) use TCP/IP even if your machine is not connected to any other computer. They use the so-called *loopback device*, which this option sets up. It will enlarge your kernel by about 144 KB.

For an excellent introduction to Linux networking, please read the “Linux Networking” HOWTO, available from <http://www.tldp.org/docs.html#howto>.

IP_ADVANCED_ROUTER

IP: advanced router

If you intend to run your Linux box mostly as a router, i.e., as a computer that forwards and redistributes network packets, say yes here. You will then be presented with several options that allow more precise control about the routing process.

The answer to this question won’t directly affect the kernel: answering no will just cause the configurator to skip all the questions about advanced routing.

Note that your box can act as a router only if you enable IP forwarding in your kernel; you can do that by saying yes to the */proc* filesystem support and Sysctl support options and executing the line:

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

at boot time after the */proc* filesystem has been mounted.

If you turn on IP forwarding, you will also get *rp_filter*, which automatically rejects incoming packets if the routing table entry for their source address doesn’t match the network interface they’re arriving on. This has security advantages because it prevents IP spoofing; however, it can pose problems if you use asymmetric routing (packets from you to a host take a different path from packets that go from that host to you) or if you operate a nonrouting host that has several IP addresses on different interfaces. To turn *rp_filter* off, enter:

```
echo 0 > /proc/sys/net/ipv4/conf/device/rp_filter
```

or:

```
echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
```

NETFILTER

Network packet filtering

Netfilter is a framework for filtering and mangling network packets that pass through your Linux box.

NET_SCHED**QoS and/or fair queueing**

When the kernel has several packets to send out over a network device, it has to decide which ones to send first, which ones to delay, and which ones to drop. This is the job of queueing disciplines. Several different algorithms for how to do this “fairly” have been proposed.

If you say no here, you will get the standard packet scheduler, which is a FIFO (first come, first served) scheduler. If you say yes here, you will be able to choose from among several alternative algorithms that can then be attached to different network devices. This is useful, for example, if some of your network devices are real-time devices that need a certain minimum data flow rate, or if you need to limit the maximum data flow rate for traffic that matches specified criteria.

To administer these schedulers, you’ll need the user-level utilities from the package *iproute2+tc* at <http://linux-net.osdl.org/index.php/Iproute2>.

This Quality of Service (QoS) support will enable you to use Differentiated Services (diffserv) and Resource Reservation Protocol (RSVP) on your Linux router if you also say yes to the corresponding options. Documentation and software is at <http://diffserv.sourceforge.net>.

IRDA**IrDA (infrared) subsystem support**

Say yes here if you want to build support for the IrDA protocols. The Infrared Data Association specifies standards for wireless infrared communication and is supported by most laptops and PDAs.

To use Linux support for the IrDA protocols, you will also need some userspace utilities such as *irattach*. For more information, see the file *Documentation/networking/irda.txt*. You also want to read the IR-HOWTO, available at <http://www.tldp.org/docs.html#howto>.

If you want to exchange bits of data (e.g., vCal, vCard) with a PDA, you will need to install an OBEX application, such as OpenObex from <http://sourceforge.net/projects/openobex>.

IRLAN**IrLAN protocol**

Say yes here if you want to build support for the IrLAN protocol. IrLAN emulates an Ethernet and makes it possible to put up a wireless LAN using infrared beams.

The IrLAN protocol can be used to talk with infrared access points such as the HP NetbeamIR or the ESI JetEye NET. You can also connect to another Linux machine running the IrLAN protocol for ad hoc networking.

HIDP

Module Human Interface Device Protocol

To use the Linux Bluetooth subsystem, you will need several user-space utilities, such as *hciconfig* and *hcid*. These utilities and updates to Bluetooth kernel modules are provided in the BlueZ packages at <http://www.bluez.org>.

IEEE80211

Generic IEEE 802.11 networking stack

This option enables the hardware-independent IEEE 802.11 networking stack.

MTD

Memory Technology Device (MTD) support

Memory Technology Devices are flash, RAM, and similar chips, often used for solid-state filesystems on embedded devices. This option provides the generic support for MTD drivers to register themselves with the kernel and for potential users of MTD devices to enumerate the devices present and obtain a handle on them. It also allows you to select individual drivers for particular hardware and users of MTD devices.

PARPORT

Parallel port support

If you want to use devices connected to your machine's parallel port (the connector at the computer with 25 holes), e.g., a printer, ZIP drive, or Parallel Line Internet Protocol (PLIP) link, you need to say yes here.

Please read *Documentation/parport.txt* and *drivers/parport/BUGS-parport* for more information. For extensive information about drivers for many devices attaching to the parallel port, see <http://www.torque.net/linux-pp.html>.

It is possible to share a single parallel port among several devices, and it is safe to compile all the corresponding drivers into the kernel. If you have more than one parallel port and want to specify which port and IRQ will be used by this driver at module load time, take a look at *Documentation/parport.txt*.

PNP

Plug and Play support

Plug and Play (PnP) is a standard for peripherals that allows them to be configured by software—for example, to assign IRQs or other parameters. No jumpers on the cards are needed; instead, the values are provided to the cards from the BIOS, from the operating system, or using a userspace utility.

Say yes here if you would like Linux to configure your PnP devices. You should then also say yes to all of the protocols needed. Alternatively, you can say no here and configure your PnP devices using userspace utilities such as the *isapnptools* package.

SMART IDE (self-monitoring, -analysis, and -reporting technology) was designed in order to prevent data corruption and disk crashes by detecting pre-hardware failure conditions (heat, access time, and the like). Disks built after June 1995 may follow this standard. The kernel itself doesn't manage this; however, there are quite a number of user programs, such as *smart*, that can query the status of SMART parameters from disk drives.

For further information, please read *Documentation/ide.txt*.

BLK_DEV_IDE

Enhanced IDE/MFM/RLL disk/CD-ROM/tape/floppy support

If you say yes here, you will use the full-featured IDE driver to control up to 10 ATA/IDE interfaces, each one able to serve a “master” and a “slave” device, for a total of up to 20 ATA/IDE disk/CD-ROM/tape/floppy drives.

Useful information about large (540 MB) IDE disks, multiple interfaces, what to do if ATA/IDE devices are not automatically detected, sound card ATA/IDE ports, module support, and other topics is contained in *Documentation/ide.txt*. For detailed information about hard drives, consult the Disk-HOWTO and the Multi-Disk-HOWTO, available at <http://www.tldp.org/docs.html#howto>.

To fine-tune ATA/IDE drive/interface parameters for improved performance, look for the *hdparm* package at <ftp://ibiblio.org/pub/Linux/system/hardware>.

Do not compile this driver as a module if your root filesystem (the one containing the directory /) is located on an IDE device.

If you have one or more IDE drives, enable this option. If your system has no IDE drives or if memory requirements are really tight, you could say no here, and select the old hard disk driver option instead to save about 13 KB of memory in the kernel.

BLK_DEV_IDEDISK

Include IDE/ATA-2 disk support

This includes enhanced support for MFM/RLL/IDE hard disks. If you have a MFM/RLL/IDE disk and there is no special reason to use the old hard disk driver instead, say yes. If you have an SCSI-only system, you can say no here.

Do not compile this driver as a module if your root filesystem (the one containing the directory /) is located on the IDE disk.

BLK_DEV_IDECD

Include IDE/ATAPI CD-ROM support

If you have a CD-ROM drive using the ATAPI protocol, say yes here. ATAPI is a newer protocol used by IDE CD-ROM and tape drives, similar to the SCSI protocol. Most new CD-ROM drives use ATAPI, including the NEC-260, Mitsumi FX400, Sony 55E, and just about all non-SCSI double (2×) or better speed drives.

Do not compile this driver as a module if your root filesystem (the one containing the directory /) is located on a SCSI disk. In this case, do not compile the driver for your SCSI host adapter as a module either.

CHR_DEV_ST	<p>SCSI tape support</p> <p>If you want to use a SCSI tape drive under Linux, say <i>yes</i> and read the SCSI-HOWTO, available from http://www.tldp.org/docs.html#howto, and <i>Documentation/scsi/st.txt</i> in the kernel source. This is <i>not</i> for SCSI CD-ROMs.</p>
BLK_DEV_SR	<p>SCSI CD-ROM support</p> <p>If you want to use a SCSI or FireWire CD-ROM under Linux, say <i>yes</i> and read the SCSI-HOWTO and the CDRM-HOWTO at http://www.tldp.org/docs.html#howto for more directions. Also make sure to enable the ISO 9660 CD-ROM filesystem support option.</p>
CHR_DEV_SG	<p>SCSI generic support</p> <p>If you want to use SCSI scanners, synthesizers, or CD writers, or just about anything having “SCSI” in its name other than hard disks, CD-ROMs, or tapes, say <i>yes</i> here. These won’t be supported by the kernel directly, so you need some additional software that knows how to talk to these devices using the SCSI protocol.</p> <p>For scanners, look at SANE http://www.sane-project.org. For CD writer software look at Cdrtools, http://cdrecord.berlios.de/old/private/cdrecord.html, and for burning a “disk at once,” check out CDRDAO, http://cdrdao.sourceforge.net. Cdparanoia is a high-quality digital reader of audio CDs (http://www.xiph.org/paranoia). For other devices, it’s possible that you’ll have to write the driver software yourself. Please read the file <i>Documentation/scsi/scsi-generic.txt</i> for more information.</p>
CHR_DEV_SCH	<p>SCSI media changer support</p> <p>This is a driver for SCSI media changers. The most common such devices are tape libraries and MOD/CD-ROM jukeboxes. This option is for real jukeboxes; you don’t need it for tiny six-slot CD-ROM changers. Media changers are listed as “Type: Medium Changer” in <i>/proc/scsi/scsi</i>. Check <i>Documentation/scsi/scsi-changer.txt</i> for details.</p>
SCSI_MULTI_LUN	<p>Probe all LUNs on each SCSI device</p> <p>If you have a SCSI device, such as a CD jukebox, that supports more than one LUN (Logical Unit Number), and only one LUN is</p>

I2O**I2O support**

The Intelligent Input/Output (I2O) architecture allows hardware drivers to be split into two parts: an operating-system-specific module called the OSM and a hardware-specific module called the HDM. The OSM can talk to a whole range of HDMs, and ideally the HDMs are not OS-dependent. This allows for the same HDM driver to be used under different operating systems if the relevant OSM is in place. In order for this to work, you need to have an I2O interface adapter card in your computer. This card contains a special I/O processor (IOP), allowing high speeds because the CPU does not have to deal with I/O.

If you say yes here, you will get a choice of interface adapter drivers and OSMs and will have to enable the correct ones.

NETDEVICES**Network device support**

You can say no here if you do not intend to connect your Linux box to any other computer.

You'll have to say yes if your computer contains a network card that you want to use under Linux. If you are going to run SLIP or PPP over a telephone line or null modem cable you also need to say yes here. Connecting two machines with parallel ports using PLIP needs this, as well as AX.25/KISS, for sending Internet traffic over amateur radio links.

See also the *Linux Network Administrator's Guide* by Tony Bautts et al. (O'Reilly), available at <http://www.tldp.org/guides.html>.

NET_ETHERNET**Ethernet (10 or 100 Mbit)**

Ethernet (also called IEEE 802.3 or ISO 8802-2) is the most common type of Local Area Network (LAN) in universities and companies.

Common varieties of Ethernet are 10-base2 or Thinnet (10 Mbps over coaxial cable, linking computers in a chain), 10-baseT or twisted pair (10 Mbps over twisted pair cable, linking computers to central hubs), 10-baseF (10 Mbps over optical fiber links, using hubs), 100-baseTX (100 Mbps over two twisted pair cables, using hubs), 100-baseT4 (100 Mbps over four standard voice-grade twisted pair cables, using hubs), 100-baseFX (100 Mbps over optical fiber links), and gigabit Ethernet (1 Gbps over optical fiber or short copper links). The 100-base varieties are also known as Fast Ethernet.

If your Linux machine will be connected to an Ethernet and you have an Ethernet network interface card (NIC) installed in your computer, say yes here and read the Ethernet-HOWTO, available from <http://www.tldp.org/docs.html#howto>. You will then also have to say yes to the driver for your particular NIC.

ISDN

ISDN support

ISDN (Integrated Services Digital Networks, called RNIS in France) is a special type of fully digital telephone service; it's mostly used to connect to your Internet service provider (with SLIP or PPP). The main advantage of ISDN is that the speed is higher than ordinary modem/telephone connections and that you can have voice conversations while downloading stuff. It works only if your computer is equipped with an ISDN card and both you and your service provider purchased an ISDN line from the phone company. For details, read <http://www.alumni.caltech.edu/~dank/isdn>.

Select this option if you want your kernel to support ISDN.

PHONE

Linux telephony support

Say yes here if you have a telephony card, which, for example, allows you to use a regular phone for voice over IP applications.



This option has nothing to do with modems. You do not need to say yes here in order to be able to use a modem under Linux.

INPUT

Generic input layer (needed for keyboard, mouse, ...)

Say yes here if you have any input device (mouse, keyboard, tablet, joystick, steering wheel, etc.) connected to your system and want it to be available to applications. This includes a standard PS/2 keyboard and mouse.

Say no here if you have a headless system (no monitor or keyboard). More information is available in *Documentation/input/input.txt*.

VT

Virtual terminal

Say yes here to get support for terminal devices with display and keyboard devices. These are called “virtual” because you can run several virtual terminals (also called virtual consoles) on one physical terminal.

You need at least one virtual terminal device in order to make use of your keyboard and monitor. Therefore, only people configuring an embedded system would want to say no here in order to save some memory. The only way to log into such a system is then via a serial or network connection.

Virtual terminals are useful because, for example, one virtual terminal can display system messages and warnings, another one can be used for a text-mode user session, and a third could run an

If you need more texture memory than you can get with the AGP GART (theoretically up to 256 MB, but in practice usually 64 or 128 MB due to kernel allocation issues), you could use PCI accesses and have up to a couple of gigabytes of texture space.

Note that this is the only way to have X and GLX use write-combining with MTRR support on the AGP bus. Without this option, OpenGL direct rendering will be a lot slower, but still faster than PIO.

You should say yes here if you want to use GLX or DRI.

DRM

Direct Rendering Manager (XFree86 4.1.0 and higher DRI support)

Kernel-level support for the Direct Rendering Infrastructure (DRI) was introduced in XFree86 4.0. If you say yes here, you need to select the module that's right for your graphics card from the list. These modules provide support for synchronization, security, and DMA transfers. Please see <http://dri.sourceforge.net> for details. You should also select and configure AGP (`/dev/agpgart`) support.

I2C

I2C support

I2C (pronounced "I-square-C") is a slow serial bus protocol developed by Philips and used in many micro controller applications. SMBus, or System Management Bus, is a subset of the I2C protocol. More information is contained in the directory *Documentation/i2c*, especially in the file there called *summary*.

Both I2C and SMBus are supported by this option. You will need it for hardware sensors support and Video For Linux support.

If you want I2C support, in addition to saying yes here, you must also select the specific drivers for your bus adapters.

SPI

SPI support

The Serial Peripheral Interface (SPI) is a low-level synchronous protocol. Chips that support SPI can have data transfer rates up to several tens of Mbps. Chips are addressed with a controller and a chipselect. Most SPI slaves don't support dynamic device discovery; some are even write-only or read-only.

SPI is widely used by microcontrollers to talk with sensors, EEPROM and flash memory, codecs and various other controller chips, analog-to-digital and digital-to-analog converters, and more. MMC and SD cards can be accessed using SPI protocol, and for DataFlash cards used in MMC sockets, SPI must always be used.

SPI is one of a family of similar protocols using a four-wire interface (select, clock, data in, and data out), including Microwire (half duplex), SSP, SSI, and PSP. This driver framework should work with most such devices and controllers.

for the x86 architecture, you can say yes if you want to use the frame buffer, but it is not essential.

Please note that running graphical applications that directly touch the hardware (e.g., an accelerated X server) and that are not attuned to the frame buffer device may cause unexpected results.

VGA_CONSOLE

VGA text console

Saying yes here will allow you to use Linux in text mode through a display that complies with the generic VGA standard. Virtually everyone wants that.

The program `SVGATextMode` can be used to utilize SVGA video cards to their full potential in text mode. Download it from <ftp://ibiblio.org/pub/Linux/utils/console>.

LOGO

Bootup logo

This option enables the pretty penguin logo at boot time. It will show up on the frame buffer while the kernel is booting. The number of penguins shows the number of processors that the kernel has found.

SOUND

Sound card support

If you have a sound card in your computer—i.e., if it can create more than an isolated beep—say yes. Be sure to have all the information about your sound card and its configuration (I/O port, interrupt and DMA channel), because you will be asked for it.

Read the Sound-HOWTO, available from <http://www.tldp.org/docs.html#howto>. General information about the modular sound system is contained in the file *Documentation/sound/oss/Introduction*. The file *Documentation/sound/oss/README.OSS* contains some slightly outdated but still useful information as well. Newer sound driver documentation can be found in files in the *Documentation/sound/alsa* directory.

If you have a PnP sound card and you want to configure it at boot time using the ISA PnP tools (read <http://www.roestock.demon.co.uk/isapnptools>), you need to compile sound card support as a module and load that module after the PnP configuration is finished. To do this properly, read *Documentation/sound/oss/README.modules*.

I'm told that even without a sound card, you can make your computer create more than an occasional beep by programming the PC speaker. Kernel patches and supporting utilities to do that are in the *pcsp* package, available at <ftp://ftp.infradead.org/pub/pcsp>.

EHCI controllers are packaged with “companion” host controllers (OHCI or UHCI) to handle USB 1.1 devices connected to root hub ports. Ports will connect to EHCI if the device is high-speed; otherwise, they connect to a companion controller. If you configure EHCI, you should probably configure the OHCI (for NEC and some other vendors) USB HCD or UHCI (for VIA motherboards) HCD, too.

You may want to read *Documentation/usb/ehci.txt* for more information on this driver.

USB_OHCI_HCD OHCI HCD support

The Open Host Controller Interface (OHCI) is a standard for accessing USB 1.1 host controller hardware. It does more in hardware than Intel’s UHCI specification. If your USB host controller follows the OHCI spec, say yes. On most non-x86 systems, and on x86 hardware that’s not using a USB controller from Intel or VIA, this is appropriate. If your host controller doesn’t use PCI, this is probably appropriate. For a PCI-based system where you’re not sure, the *lspci -v* command will list the right prog-if for your USB controller(s): EHCI, OHCI, or UHCI.

USB_UHCI_HCD UHCI HCD (most Intel and VIA) support

The Universal Host Controller Interface is a standard created by Intel for accessing the USB hardware in the PC (which is also called the USB host controller). If your USB host controller conforms to this standard, you may want to say yes. All recent boards with Intel PCI chipsets (such as Intel 430TX, 440FX, 440LX, 440BX, i810, i820) conform to this standard. All VIA PCI chipsets (like VIA VP2, VP3, MVP3, Apollo Pro, Apollo Pro II, or Apollo Pro 133) also use the standard.

USB_STORAGE USB mass storage support

Say yes here if you want to connect USB mass storage devices to your computer’s USB port. This is the driver you need for USB floppy drives, USB hard disks, USB tape drives, USB CD-ROMs, USB flash devices, and memory sticks, along with similar devices. This driver may also be used for some cameras and card readers.

This option enables the SCSI option, but you probably also need SCSI device support: SCSI disk support for most USB storage devices to work properly.

USB_SERIAL USB serial converter support

Say yes here if you have a USB device that provides normal serial ports, or acts like a serial device, and you want to connect it to your USB bus.

EXT2_FS

Second extended filesystem support

ext2 is a standard Linux filesystem for hard disks. Most systems use the upgrade, *ext3*, instead.

Note that the filesystem of your root partition (the one containing the directory */*) cannot be compiled as a module without using a special boot process, so building it as a module could be dangerous.

EXT3_FS

Third extended filesystem support

This is the journaling version (called *ext3*) of the second extended filesystem, the de facto standard Linux filesystem for hard disks.

The journaling code included in this driver means you do not have to run *fsck* (filesystem checker) on your filesystems after a crash. The journal keeps track of any changes that were being made at the time the system crashed, and can ensure that your filesystem is consistent without the need for a lengthy check.

Other than adding the journal to the filesystem, the on-disk format of *ext3* is identical to *ext2*. It is possible to freely switch between using the *ext3* driver and the *ext2* driver, as long as the filesystem has been cleanly unmounted, or *fsck* is run on the filesystem before the switch.

To add a journal on an existing *ext2* filesystem or change the behavior of *ext3* filesystems, you can use the *tune2fs* utility. To modify attributes of files and directories on *ext3* filesystems, use *chattr*. You need *e2fsprogs* version 1.20 or later in order to create *ext3* journals (available at <http://sourceforge.net/projects/e2fsprogs>).

REISERFS_FS

ReiserFS support

This is a journaled filesystem that stores not just filenames but the files themselves in a balanced tree. Balanced trees can be more efficient than traditional filesystem architectural foundations.

In general, ReiserFS is as fast as *ext2*, but is more efficient with large directories and small files.

JFS_FS

JFS filesystem support

This is a port of IBM's Journaled Filesystem (JFS). More information is available in the file *Documentation/filesystems/jfs.txt*.

XFS_FS

XFS filesystem support

XFS is a high-performance journaling filesystem that originated on the SGI IRIX platform. It is completely multithreaded; supports large files and large filesystems, extended attributes, and variable

To use the automounter, you need the userspace tools from the `autofs` package; you can find the location in *Documentation/Changes*. You also want to answer yes to the NFS filesystem support option.

If you want to use the newer version of the automounter with more features, say no here and say yes to the Kernel automounter v4 support option.

If you are not a part of a fairly large, distributed network, you probably do not need an automounter, and can say no here.

FUSE_FS

Filesystem in userspace support

With FUSE it is possible to implement a fully functional filesystem in a userspace program.

There's also companion library named *libfuse*. This library, along with utilities, is available from the FUSE homepage: <http://fuse.sourceforge.net>.

See *Documentation/filesystems/fuse.txt* for more information. See *Documentation/Changes* for library/utility version you need.

If you want to develop a userspace filesystem, or if you want to use a filesystem based on FUSE, answer yes here.

SMB_FS

SMB filesystem support (to mount Windows shares etc.)

SMB (Server Message Block) is the protocol Windows for Workgroups (WfW), Windows 95/98, Windows NT and later variants, and OS/2 LAN Manager use to share files and printers over local networks. Saying yes here allows you to mount their filesystems (often called "shares" in this context) and access them just like any other Unix directory. Currently, this works only if the Windows machines use TCP/IP as the underlying transport protocol, not NetBEUI. For details, read *Documentation/filesystems/smbfs.txt* and the SMB-HOWTO, available from <http://www.tldp.org/docs.html#howto>.

If you just want your box to act as an SMB server and make files and printing services available to Windows clients (which need to have a TCP/IP stack), you don't need to say yes here; you can use the Samba set of daemons and programs (available from <ftp://ftp.samba.org/pub/samba>).

CIFS

CIFS support (advanced network filesystem for Samba, Window, and other CIFS compliant servers)

This is the client VFS module for the Common Internet File System (CIFS) protocol, which is the successor to the Server Message Block (SMB) protocol, the native file-sharing mechanism for most early PC operating systems. The CIFS protocol is fully supported by file servers such as Windows 2000 (including Windows 2003, NT 4,

immediately, or dump some status information). This is accomplished by pressing various keys while holding down the SysRq (Alt+PrintScreen) key. It also works on a serial console (on PC hardware at least), if you send a BREAK and then within 5 seconds a command keypress. The keys are documented in *Documentation/sysrq.txt*. Don't say yes unless you really know what this hack does.

DEBUG_KERNEL	<p>Kernel debugging</p> <p>Say yes here if you are developing drivers or trying to debug and identify kernel problems.</p> <p>On its own, this option does not do anything except allow you to chance to select other options.</p>
DEBUG_FS	<p>Debug filesystem</p> <p><i>debugfs</i> is a virtual filesystem where kernel developers put debugging files. Enable this option to be able to read and write to these files.</p>
SECURITY	<p>Enable different security models</p> <p>This allows you to configure different security modules into your kernel.</p> <p>If this option is not selected, the default Linux security model will be used.</p>
SECURITY_ SELINUX	<p>NSA SELinux support</p> <p>This selects NSA Security-Enhanced Linux (SELinux). You will also need a policy configuration and a labeled filesystem. You can obtain the policy compiler (<i>checkpolicy</i>), the utility for labeling filesystems (<i>setfiles</i>), and an example policy configuration from http://www.nsa.gov/selinux.</p>