kernal configuration parameter help—Non networking part

CONFIG_EXPERIMENTAL:

Some of the various things that Linux supports (such as network drivers, file systems, network protocols, etc.) can be in a state of development where the functionality, stability, or the level of testing is not yet high enough for general use. This is usually known as the "alpha-test" phase amongst developers. If a feature is currently in alpha-test, then the developers usually discourage uninformed widespread use of this feature by the general public to avoid "Why doesn't this work?" type mail messages. However, active testing and use of these systems is welcomed. Just be aware that it may not meet the normal level of reliability or it may fail to work in some special cases. Detailed bug reports from people familiar with the kernel internals are usually welcomed by the developers (before submitting bug reports, please read the documents README, MAINTAINERS, REPORTING-BUGS, Documentation/BUG-HUNTING, and Documentation/oops-tracing.txt in the kernel source).

This option will also make obsoleted drivers available. These are drivers that have been replaced by something else, and/or are scheduled to be removed in a future kernel release.

Unless you intend to help test and develop a feature or driver that falls into this category, or you have a situation that requires using these features, you should probably say N here, which will cause this configure script to present you with fewer choices. If you say Y here, you will be offered the choice of using features or drivers that are currently considered to be in the alpha-test phase.

In this project: Yes

In "Networking Options", **IPv6** protocol and **IPv6 netfilter** configuration are "EXPERIEMENTAL"

CONFIG_MODULES:

Kernel modules are small pieces of compiled code which can be inserted in or removed from the running kernel, using the programs insmod and rmmod. This is described in the file Documentation/modules.txt, including the fact that you have to say "make modules" in order to compile the modules that you chose during kernel configuration. Modules can be device drivers, file systems, binary executable formats, and so on.

In this project: Yes

My implementation will be a module.

CONFIG_MODVERSIONS:

Usually, modules have to be recompiled whenever you switch to a new kernel. Saying Y here makes it possible, and safe, to use the same modules even after compiling a new kernel; this requires the program modprobe. All the software needed for module support is in the modutils package (check the file

Documentation/Changes for location and latest version). NOTE: if you say Y here but don't have the program genksyms (which is also contained in the above mentioned modutils package), then the building of your kernel will fail. If you are going to use modules that are generated from non-kernel sources, you would benefit from this option. Otherwise it's not that important. So, N ought to be a safe bet.

In this project: Yes

I am not sure. But it doesn't hurt to select it.

CONFIG_KMOD:

Normally when you have selected some drivers and/or file systems to be created as loadable modules, you also have the responsibility to load the corresponding modules (using the programs insmod or modprobe) before you can use them. If you say Y here however, the kernel will be able to load modules for itself: when a part of the kernel needs a module, it runs modprobe with the appropriate arguments, thereby loading the module if it is available. (This is a replacement for kerneld.) Say Y here and read about configuring it in Documentation/kmod.txt.

In this project: Yes

Obvious

CONFIG_M386:

This is the processor type of your CPU. This information is used for optimizing purposes. In order to compile a kernel that can run on all x86 CPU types (albeit not optimally fast), you can specify "386" here.

What to choose?

London.cs.unh.edu is **PII**, Madrid,Dublin,Prague is **PIII**

CONFIG_MICROCODE:

If you say Y here and also to "/dev file system support" in the 'File systems' section, you will be able to update the microcode on Intel processors in the IA32 family, e.g. Pentium Pro, Pentium II, Pentium III, Pentium 4, Xeon etc. You will obviously need the actual microcode binary data itself which is not shipped with the Linux kernel.

For latest news and information on obtaining all the required ingredients for this driver, check:
http://www.urbanmyth.org/microcode/

Default is YES. It doesn't hurt.

CONFIG_X86_MSR:

This device gives privileged processes access to the x86 Model-Specific Registers (MSRs). It is a character device with major 202 and minors 0 to 31 for /dev/cpu/0/msr to /dev/cpu/31/msr. MSR accesses are directed to a specific CPU on multi-processor systems.

Default is YES. It doesn't hurt.

CONFIG_X86_CPUID:

This device gives processes access to the x86 CPUID instruction to be executed on a specific processor. It is a character device with major 203 and minors 0 to 31 for /dev/cpu/0/cpuid to /dev/cpu/31/cpuid.

In this project: NO

There's only one CPU in London, Dublin, Madrid, Prague. And I will deal with CPU directly

CONFIG_NOHIGHMEM:

Linux can use up to 64 Gigabytes of physical memory on x86 systems. However, the address space of 32-bit x86 processors is only 4 Gigabytes large. That means that, if you have a large amount of physical 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 which will never run on a machine with more than 1 Gigabyte total physical RAM, answer "off" here (default choice and suitable for most users). This will result in a "3GB/1GB" split: 3GB are mapped so that each process sees a 3GB virtual memory space and the remaining part of the 4GB virtual memory space is used by the kernel to permanently map as much physical memory as possible.

In this project: NO

None of the London, Madrid, Dublin, Prague has more than 1 GByte memory

CONFIG_MATH_EMULATION:

Linux can emulate a math coprocessor (used for floating point operations) if you don't have one. 486DX and Pentium processors have a math coprocessor built in, 486SX and 386 do not, unless you added a 487DX or 387, respectively. (The messages during boot time can give you some hints here ["man dmesg"].) Everyone needs either a coprocessor or this emulation.

If you don't have a math coprocessor, you need to say Y here; if you say Y here even though you have a coprocessor, the coprocessor will be used nevertheless. (This behavior can be changed with the kernel command line option "no387", which comes handy if your coprocessor is broken. Try "man bootparam" or see the documentation of your boot loader (lilo or loadlin) about how to pass options to the kernel at boot time.) This means that it is a good idea to say Y here if you intend to use this kernel on different machines.

More information about the internals of the Linux math coprocessor emulation can be found in arch/i386/math-emu/README.

In this project: YES

I am not sure whether my computer has this math coprocessor, but according to the help, it doesn't hurt to have it.

CONFIG_MTRR:

On Intel P6 family processors (Pentium Pro, Pentium II and later) the Memory Type Range Registers (MTRRs) may be used to control processor access to memory ranges. This is most useful if you have a video (VGA) card on a PCI or AGP bus. Enabling write-combining allows bus write transfers to be combined into a larger transfer before bursting over the PCI/AGP bus. This can increase performance of image write operations 2.5 times or more. Saying Y here creates a /proc/mtrr file which may be used to manipulate your processor's MTRRs. Typically the X server should use this.

In this project: NO

I won't deal with the processors. Just to keep my kernel minimum

CONFIG_SMP:

This enables support for systems with more than one CPU. If you have a system with only one CPU, like most personal computers, say N. If you have a system with more than one CPU, say Y.

In this project: NO

None of London, Dublin, Madrid, Prague has more than one CPU.

CONFIG_X86_UP_IOAPIC:

APIC (Advanced Programmable Interrupt Controller) is a scheme for delivering hardware interrupt requests to the CPU. It is commonly used on systems with several CPU's. If you have a single-CPU system which uses APIC, you can say Y here to use it. If you say Y here even though your machine doesn't have APIC, then the kernel will still run with no slowdown at all.

If you have system with several CPU's, you do not need to say Y here: APIC will be used automatically.

In this project: NO

I am not sure. But it doesn't hurt if I don't choose it.

CONFIG_NET:

Unless you really know what you are doing, you should say Y here. The reason is that some programs need kernel networking support even when running on a stand-alone machine that isn't connected to any other computer. If you are upgrading from an older kernel, you should consider updating your networking tools too because changes in the kernel and the tools often go hand in hand. The tools are contained in the package net-tools, the location and version number of which are given in Documentation/Changes.

In this project: YES
OBVIOUS

CONFIG_PCI:

Find out whether you have a PCI motherboard. PCI is the name of a bus system,i.e. the way the CPU talks to the other stuff inside your box. Other bus systems are ISA, EISA, Microchannel (MCA) or VESA. If you have PCI, say Y, otherwise N.

In this project: YES

London, Dublin, Madrid, Prague all has only PCI and IDE. no ISA/EISA

CONFIG_PCI_GOBIOS:

On PCI systems, the BIOS can be used to detect the PCI devices and determine their configuration. However, some old PCI motherboards have BIOS bugs and may crash if this is done. Also, some embedded PCI-based systems don't have any BIOS at all. Linux can also try to detect the PCI hardware directly without using the BIOS.

With this option, you can specify how Linux should detect the PCI devices. If you choose "BIOS", the BIOS will be used, if you choose "Direct", the BIOS won't be used, and if you choose "Any", the kernel will try the direct access method and falls back to the BIOS if that doesn't work. If unsure, go with the default, which is "Any".

In this project: YES

CONFIG_PCI_NAMES:

By default, the kernel contains a database of all known PCI device names to make the information in /proc/pci, /proc/ioports and similar files comprehensible to the user. This database increases size of the kernel image by about 80KB, but it gets freed after the system boots up, so it doesn't take up kernel memory. Anyway, if you are building an installation floppy or kernel for an embedded system where kernel image size really matters, you can disable this feature and you'll get device ID numbers instead of names.

In this project: YES

I am not sure. It doesn’t hurt.

CONFIG_EISA:

The Extended Industry Standard Architecture (EISA) bus was developed as an open alternative to the IBM MicroChannel bus.

The EISA bus provided some of the features of the IBM MicroChannel bus while maintaining backward compatibility with cards made for the older ISA bus. The EISA bus saw limited use between 1988 and 1995 when it was made obsolete by the PCI bus.

In this project: NO

There is no EISA slot on london, dublin, madrid, prague machines

CONFIG_MCA:

MicroChannel Architecture is found in some IBM PS/2 machines and laptops. It is a bus system similar to PCI or ISA. See Documentation/mca.txt (and especially the web page given there) before attempting to build an MCA bus kernel.

In this project: NO
Obvious

CONFIG_HOTPLUG:

Say Y here if you want to plug devices into your computer while the system is running, and be able to use them quickly. In many cases, the devices can likewise be unplugged at any time too.

In this project: NO

I won't use any USB, PCMCIA devices on london/dublin/madrid/prague machines

CONFIG_SYSVIPC:

Inter Process Communication is a suite of library functions and system calls which let processes (running programs) synchronize and exchange information. It is generally considered to be a good thing, and some programs won't run unless you say Y here. In particular, if you want to run the DOS emulator dosemu under Linux (read the DOSEMU-HOWTO, available from http://www.linuxdoc.org/docs.html#howto), you'll need to say Y here.

You can find documentation about IPC with "info ipc" and also in section 6.4 of the Linux Programmer's Guide, available from
http://www.linuxdoc.org/docs.html#guide .

In this project: YES
See the help above

CONFIG_BSD_PROCESS_ACCT:

If you say Y here, a user level program will be able to instruct the kernel (via a special system call) to write process accounting information to a file: whenever a process exits, information about that process will be appended to the file by the kernel. The information includes things such as creation time,owning user,command name, memory usage, controlling terminal etc. (the complete list is in the struct acct in include/linux/acct.h). It is up to the user level program to do useful things with this information. This is generally a good idea, so say Y.

In this project: YES

I am not sure, but it doesn’t hurt.

CONFIG_SYSCTL:

The sysctl interface provides a means of dynamically changing certain kernel parameters and variables on the fly without requiring a recompile of the kernel or reboot of the system. The primary interface consists of a system call, but if you say Y to "/proc file system support", a tree of modifiable sysctl entries will be generated beneath the /proc/sys directory. They are explained in the files in Documentation/sysctl/. Note that enabling this option will enlarge the kernel by at least 8 KB.

In this project: YES

As it is generally a good thing, you should say Y here unless
building a kernel for install/rescue disks or your system is very
limited in memory.

CONFIG_KCORE_ELF:

If you enabled support for /proc file system then the file /proc/kcore will contain the kernel core image. This can be used in gdb:

```
$ cd /usr/src/linux ; gdb vmlinux /proc/kcore
```

You have two choices here: ELF and A.OUT. Selecting ELF will make /proc/kcore appear in ELF core format as defined by the Executable and Linking Format specification. Selecting A.OUT will choose the old "a.out" format which may be necessary for some old versions of binutils or on some architectures.

This is especially useful if you have compiled the kernel with the "-g" option to preserve debugging information. It is mainly used for examining kernel data structures on the live kernel so if you don't understand what this means or are not a kernel hacker, just leave it at its default value ELF.

CONFIG_BINFMT_AOUT:

A.out (Assembler.OUTput) is a set of formats for libraries and executables used in the earliest versions of UNIX. Linux used the a.out formats QMAGIC and ZMAGIC until they were replaced with the ELF format.

As more and more programs are converted to ELF, the use for a.out will gradually diminish. If you disable this option it will reduce your kernel by one page. This is not much and by itself does not warrant removing support. However its removal is a good idea if you wish to ensure that absolutely none of your programs will use this older executable format. If you don't know what to answer at this point then answer Y. If someone told you "You need a kernel with QMAGIC support" then you'll have to say Y here. You may answer M to compile a.out support as a module and later load the module when you want to use a program or library in a.out format. The module will be called binfmt_aout.o. Saying M or N here is dangerous though, because some crucial programs on your system might still be in A.OUT format.

CONFIG_BINFMT_ELF:

ELF (Executable and Linkable Format) is a format for libraries and executables used across different architectures and operating systems. Saying Y here will enable your kernel to run ELF binaries and enlarge it by about 13 KB. ELF support under Linux has now all but replaced the traditional Linux a.out formats (QMAGIC and ZMAGIC) because it is portable (this does *not* mean that you will be able to run executables from different architectures or operating systems however) and makes building run-time libraries very easy. Many new executables are distributed solely in ELF format. You definitely want to say Y here.

Information about ELF is contained in the ELF HOWTO available from http://www.linuxdoc.org/docs.html#howto .

If you find that after upgrading from Linux kernel 1.2 and saying Y here, you still can't run any ELF binaries (they just crash), then you'll have to install the newest ELF runtime libraries, including ld.so (check the file Documentation/Changes for location and latest version).

If you want to compile this as a module (= code which can be inserted in and removed from the running kernel whenever you want),say M here and read Documentation/modules.txt. The module will be called binfmt_elf.o. Saying M or N here is dangerous because some crucial programs on your system might be in ELF format.

In this project: set to default: BOTH

CONFIG_BINFMT_MISC:

If you say Y here, it will be possible to plug wrapper-driven binary formats into the kernel. You will like this especially when you use programs that need an interpreter to run like Java, Python or Emacs-Lisp. It's also useful if you often run DOS executables under the Linux DOS emulator DOSEMU (read the DOSEMU-HOWTO, available from http://www.linuxdoc.org/docs.html#howto). Once you have registered such a binary class with the kernel, you can start one of those programs simply by typing in its name at a shell prompt; Linux will automatically feed it to the correct interpreter.

You can do other nice things, too. Read the file Documentation/binfmt_misc.txt to learn how to use this feature, and Documentation/java.txt for information about how to include Java support.

You must say Y to "/proc file system support" (CONFIG_PROC_FS) to use this part of the kernel.

You may say M here for module support and later load the module when you have use for it; the module is called binfmt_misc.o. **If you don't know what to answer at this point, say Y.**

CONFIG_PM:

"Power Management" means that parts of your computer are shut off or 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 Y here and then also to the requisite support below.

In this project: NO

For Desktop machines, it doesn't hurt to rule it out

CONFIG_PARPORT:

If you want to use devices connected to your machine's parallel port (the connector at the computer with 25 holes), e.g. printer, ZIP drive, PLIP link (Parallel Line Internet Protocol is mainly used to create a mini network by connecting the parallel ports of two local machines) etc., then you need to say Y here; please read Documentation/parport.txt and drivers/parport/BUGS-parport.

In this project: NO

No printers or other parallel port device will be attached.

CONFIG_PNP:

Plug and Play (PnP) is a standard for peripherals which allows those peripherals to be configured by software, e.g. assign IRQ's 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 user-space utility.

Say Y here if you would like Linux to configure your Plug and Play devices. You should then also say Y to "ISA Plug and Play support", below. Alternatively, you can say N here and configure your PnP devices using the user space utilities contained in the isapnptools package.

This support is also available as a module (= code which can be inserted in and removed from the running kernel whenever you want).If you want to compile it as a module, say M here and read Documentation/modules.txt.

In this project: YES

There might be some NIC installed later

CONFIG_BLK_DEV_FD:

If you want to use the floppy disk drive(s) of your PC under Linux, say Y. Information about this driver, especially important for IBM Thinkpad users, is contained in Documentation/floppy.txt. That file also contains the location of the Floppy driver FAQ as well as location of the fdutils package used to configure additional parameters of the driver at run time.

This driver is also available as a module (= code which can be inserted in and removed from the running kernel whenever you want). The module will be called floppy.o. If you want to compile it as a module, say M here and read Documentation/modules.txt.

In this project: NO
There’s no floppy disk driver in London, Madrid, Prague, Bublin.

CONFIG_BLK_DEV_XD:

Very old 8 bit hard disk controllers used in the IBM XT computer will be supported if you say Y here.

In this project: NO

CONFIG_PARIDE:

There are many external CD-ROM and disk devices that connect through your computer's parallel port. Most of them are actually IDE devices using a parallel port IDE adapter. This option enables the PARIDE subsystem which contains drivers for many of these external drives.

Read Documentation/paride.txt for more information.

In this project: NO

CONFIG_BLK_DEV_NBD:

CONFIG_BLK_DEV_LOOP:

CONFIG_BLK_DEV_RAM:

In this project: USE DEFAULT. It doesn’t hurt.

CONFIG_SCSI:

If you want to use a SCSI hard disk, SCSI tape drive, SCSI CDROM or any other SCSI device under Linux, say Y and make sure that you know the name of your SCSI host adapter (the card inside your computer that "speaks" the SCSI protocol, also called SCSI controller),because you will be asked for it.

You also need to say Y here if you want support for the parallel port version of the 100 MB IOMEGA ZIP drive.

In this project: NO
There is no SCSI devices.

CONFIG_BLK_DEV_SD:

CONFIG_SD_EXTRA_DEVS:

CONFIG_I2O:

The Intelligent Input/Output (I2O) architecture allows hardwaredrivers to be split into two parts: an operating system specific module called the OSM and an hardware specific module called the HDM. The OSM can talk to a whole range of HDM's, and ideally the HDM's 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), thus allowing high speeds since the CPU does not have to deal with I/O.

If you say Y here, you will get a choice of interface adapter drivers and OSM's with the following questions.

This support is also available as a module (= code which can be inserted in and removed from the running kernel whenever you want). If you want to compile it as a module, say M here and read Documentation/modules.txt. You will get modules called i2o_core.o and i2o_config.o.

In this project: NO
Unsure

CONFIG_INPUT:

Say Y here if you want to enable any of the following options for USB Human Interface Device (HID) support.

CONFIG_INPUT_KEYBDEV:

CONFIG_INPUT_MOUSEDEV:

CONFIG_INPUT_MOUSEDEV_SCREEN_X:

CONFIG_INPUT_MOUSEDEV_SCREEN_Y:

In this project: NO
There won’t be any USB input devices

CONFIG_VT:

If you say Y here, you will 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. This is rather useful, for example one virtual terminal can collect system messages and warnings, another one can be used for a text-mode user session, and a third could run an X session, all in parallel. Switching between virtual terminals is done with certain key combinations, usually Alt-<function key>. The setterm command ("man setterm") can be used to change the properties (such as colors or beeping) of a virtual terminal. The man page console_codes(4) ("man console_codes") contains the special character sequences that can be used to change those properties directly. The fonts used on virtual terminals can be changed withthe setfont ("man setfont") command and the key bindings are defined with the loadkeys ("man loadkeys") command.

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 N here in order to save some memory; the only way to log into such a system is then via a serial or network connection.

If unsure, say Y, or else you won't be able to do much with your new shiny Linux system :-)

CONFIG_VT_CONSOLE:

In this project: USE default set value

CONFIG_SERIAL:

This selects whether you want to include the driver for the standard serial ports. The standard answer is Y. People who might say N here are those that are setting up dedicated Ethernet WWW/FTP servers, or users that have one of the various bus mice instead of a serial mouse and don't intend to use their machine's standard serial port for anything. (Note that the Cyclades and Stallion multi serial port drivers do not need this driver built in for them to work.)

If you want to compile this driver as a module, say M here and read Documentation/modules.txt. The module will be called serial.o. [WARNING: Do not compile this driver as a module if you are using non-standard serial ports, since the configuration information will be lost when the driver is unloaded. This limitation may be lifted in the future.]

BTW1: If you have a mouseman serial mouse which is not recognized by the X window system, try running gpm first.

BTW2: If you intend to use a software modem (also called Winmodem) under Linux, forget it. These modems are crippled and require proprietary drivers which are only available under Windows.

Most people will say Y or M here, so that they can use serial mice, modems and similar devices connecting to the standard serial ports.

CONFIG_SERIAL_CONSOLE:

If you say Y here, it will be possible to use a serial port as the system console (the system console is the device which receives all kernel messages and warnings and which allows logins in single user mode). This could be useful if some terminal or printer is connected to that serial port.

Even if you say Y here, the currently visible virtual console (/dev/tty0) will still be used as the system console by default, but you can alter that using a kernel command line option such as "console=ttyS1". (Try "man bootparam" or see the documentation of your boot loader (lilo or loadlin) about how to pass options to the kernel at boot time.)

If you don't have a VGA card installed and you say Y here, the kernel will automatically use the first serial line, /dev/ttyS0, as system console.

If unsure, say N.

CONFIG_SERIAL_EXTENDED:

If you wish to use any non-standard features of the standard "dumb" driver, say Y here. This includes HUB6 support, shared serial interrupts, special multiport support, support for more than the four COM 1/2/3/4 boards, etc.

Note that the answer to this question won't directly affect the kernel: saying N will just cause this configure script to skip all the questions about serial driver options. If unsure, say N.

CONFIG_SERIAL_MANY_PORTS:

Say Y here if you have dumb serial boards other than the four standard COM 1/2/3/4 ports. This may happen if you have an AST FourPort, Accent Async, Boca (read the Boca mini-HOWTO, available from http://www.linuxdoc.org/docs.html#howto), or other custom serial port hardware which acts similar to standard serial port hardware. If you only use the standard COM 1/2/3/4 ports, you can say N here to save some memory. You can also say Y if you have an "intelligent" multiport card such as Cyclades, Digiboards, etc.

CONFIG_SERIAL_SHARE_IRQ:

Some serial boards have hardware support which allows multiple dumb serial ports on the same board to share a single IRQ. To enable support for this in the serial driver, say Y here.

CONFIG_SERIAL_NONSTANDARD:

Say Y here if you have any non-standard serial boards – boards which aren't supported using the standard "dumb" serial driver.This includes intelligent serial boards such as Cyclades, Digiboards, etc. These are usually used for systems that need many serial ports because they serve many terminals or dial-in connections.

Note that the answer to this question won't directly affect the kernel: saying N will just cause this configure script to skip all the questions about non-standard serial boards.

In this project: NO

CONFIG_UNIX98_PTYS:

A pseudo terminal (PTY) is a software device consisting of two halves: a master and a slave. The slave device behaves identical to a physical terminal; the master device is used by a process to read data from and write data to the slave, thereby emulating a terminal. Typical programs for the master side are telnet servers and xterms.

Linux has traditionally used the BSD-like names /dev/ptyxx for masters and /dev/ttyxx for slaves of pseudo terminals. This scheme has a number of problems. The GNU C library glibc 2.1 and later, however, supports the Unix98 naming standard: in order to acquire a pseudo terminal, a process opens /dev/ptmx; the number of the pseudo terminal is then made available to the process and the pseudo terminal slave can be accessed as /dev/pts/<number>. What was traditionally /dev/ttyp2 will then be /dev/pts/2, for example.

The entries in /dev/pts/ are created on the fly by a virtual file system; therefore, if you say Y here you should say Y to "/dev/pts file system for Unix98 PTYs" as well.

If you want to say Y here, you need to have the C library glibc 2.1 or later (equal to libc-6.1, check with "ls -l /lib/libc.so.*").Read the instructions in Documentation/Changes pertaining to pseudo terminals. It's safe to say N.

CONFIG_UNIX98_PTY_COUNT:

The maximum number of Unix98 PTYs that can be used at any one time. The default is 256, and should be enough for desktop systems. Server machines which support incoming telnet/rlogin/ssh connections and/or serve several X terminals may want to increase this: every incoming connection and every xterm uses up one PTY.

When not in use, each additional set of 256 PTYs occupy approximately 8 KB of kernel memory on 32-bit architectures.

In this project: The above options use DEFAULT value

CONFIG_I2C:

I2C (pronounce: I-square-C) is a slow serial bus protocol used in many micro controller applications and developed by Philips. 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 called "summary" there.

Both I2C and SMBus are supported here. You will need this for hardware sensors support, and also for Video for Linux support.Specifically, if you want to use a BT848 based frame grabber/overlay boards under Linux, say Y here and also to "I2C bit-banging interfaces", below.

If you want I2C support, you should say Y here and also to the specific driver for your bus adapter(s) below. If you say Y to "/proc file system" below, you will then get a /proc interface which is documented in Documentation/i2c/proc-interface.

This I2C support is also available as a module. If you want to compile it as a module, say M here and read Documentation/modules.txt. The module will be called i2c-core.o.

CONFIG_BUSMOUSE:

Say Y here if your machine has a bus mouse as opposed to a serial mouse. Most people have a regular serial MouseSystem or Microsoft mouse (made by Logitech) that plugs into a COM port (rectangular with 9 or 25 pins). These people say N here. If you have something else, read the Busmouse-HOWTO, available from http://www.linuxdoc.org/docs.html#howto , and say Y here.

In this project: NO
No bus mouse. Only PS/2 mouse

CONFIG_ATIXL_BUSMOUSE:

CONFIG_LOGIBUSMOUSE:

CONFIG_MS_BUSMOUSE:

CONFIG_MOUSE:
In this project: For the above options: NO

CONFIG_PSMOUSE:

In this project: YES

CONFIG_82C710_MOUSE:

CONFIG_PC110_PAD:

In this project: For the above options :NO

CONFIG_VIDEO_DEV:

CONFIG_QUOTA:
In this project: For the above options :NO

CONFIG_AUTOFS_FS:

CONFIG_AUTOFS4_FS:

CONFIG_REISERFS_FS:

CONFIG_REISERFS_CHECK:

CONFIG_ADFS_FS:

CONFIG_AFFS_FS:

CONFIG_HFS_FS:

CONFIG_BFS_FS:

CONFIG_FAT_FS:

CONFIG_MSDOS_FS:

CONFIG_UMSDOS_FS:

CONFIG_VFAT_FS:

CONFIG_EFS_FS:

CONFIG_JFFS_FS:

CONFIG_TMPFS:

CONFIG_RAMFS:

CONFIG_ISO9660_FS:

CONFIG_JOLIET:

CONFIG_MINIX_FS:

CONFIG_VXFS_FS:

CONFIG_NTFS_FS:

CONFIG_NTFS_RW:

CONFIG_HPFS_FS:

CONFIG_PROC_FS:

```
This is a virtual file system providing information about the status
of the system. "Virtual" means that it doesn't take up any space on
your hard disk: the files are created on the fly by the kernel when
you try to access them. Also, you cannot read the files with older
version of the program less: you need to use more or cat.

It's totally cool; for example, "cat /proc/interrupts" gives
information about what the different IRQs are used for at the moment
(there is a small number of Interrupt ReQuest lines in your computer
that are used by the attached devices to gain the CPU's attention --
often a source of trouble if two devices are mistakenly configured
to use the same IRQ). The program procinfo to display some
information about your system gathered from the /proc file system.

Before you can use the /proc file system, it has to be mounted,
meaning it has to be given a location in the directory hierarchy.
That location should be /proc. A command such as "mount -t proc proc
/proc" or the equivalent line in /etc/fstab does the job.

The /proc file system is explained in the file
Documentation/filesystems/proc.txt and on the proc(5) manpage ("man
5 proc").

This option will enlarge your kernel by about 67 KB. Several
programs depend on this, so everyone should say Y here.
```

CONFIG_DEVFS_FS:

CONFIG_DEVFS_FS:

CONFIG_DEVFS_MOUNT:

CONFIG_DEVPTS_FS:

CONFIG_DEVPTS_FS:

CONFIG_QNX4FS_FS:

CONFIG_ROMFS_FS:

CONFIG_EXT2_FS:

This is the de facto standard Linux file system (method to organize files on a storage device) for hard disks.

You want to say Y here, unless you intend to use Linux exclusively from inside a DOS partition using the UMSDOS file system. The advantage of the latter is that you can get away without repartitioning your hard drive (which often implies backing everything up and restoring afterwards); the disadvantage is that Linux becomes susceptible to DOS viruses and that UMSDOS is somewhat slower than ext2fs. Even if you want to run Linux in this fashion, it might be a good idea to have ext2fs around: it enables you to read more floppy disks and facilitates the transition to a *real* Linux partition later. Another (rare) case which doesn't require ext2fs is a diskless Linux box which mounts all files over the network using NFS (in this case it's sufficient to say Y to "NFS file system support" below). Saying Y here will enlarge your kernel by about 44 KB.

The Ext2fs-Undeletion mini-HOWTO, available from http://www.linuxdoc.org/docs.html#howto , gives information about how to retrieve deleted files on ext2fs file systems.

To change the behavior of ext2 file systems, you can use the tune2fs utility ("man tune2fs"). To modify attributes of files and directories on ext2 file systems, use chattr ("man chattr").

Ext2fs partitions can be read from within DOS using the ext2tool command line tool package (available via FTP (user: anonymous) from ftp://metalab.unc.edu/pub/Linux/system/filesystems/ext2) and from within Windows NT using the ext2nt command line tool package from ftp://metalab.unc.edu/pub/Linux/utils/dos . Explore2fs is a graphical explorer for ext2fs partitions which runs on Windows 95 and Windows NT and includes experimental write support; it is available from
http://jnewbigin-pc.it.swin.edu.au/Linux/Explore2fs.htm .

If you want to compile this file system as a module (= code which can be inserted in and removed from the running kernel whenever you want), say M here and read Documentation/modules.txt. The module will be called ext2.o. Be aware however that the file system of your root partition (the one containing the directory /) cannot be compiled as a module, and so this could be dangerous. Most everyone wants to say Y here.

CONFIG_SYSV_FS:

CONFIG_UDF_FS:

CONFIG_UDF_RW:

CONFIG_UFS_FS:

CONFIG_UFS_FS_WRITE:

CONFIG_CODA_FS:

CONFIG_NFS_FS:

CONFIG_NFS_V3:

CONFIG_ROOT_NFS:

CONFIG_NFSD:

CONFIG_NFSD_V3:

CONFIG_SMB_FS:

CONFIG_SMB_NLS_DEFAULT:

ONFIG_SMB_NLS_REMOTE:

CONFIG_NCP_FS:

CONFIG_NCPFS_PACKET_SIGNING:

CONFIG_NCPFS_IOCTL_LOCKING:

CONFIG_NCPFS_STRONG:

CONFIG_NCPFS_NFS_NS:

CONFIG_NCPFS_OS2_NS:

CONFIG_NCPFS_SMALLDOS:

CONFIG_NCPFS_NLS:

CONFIG_NCPFS_EXTRAS:

In this project:
For the above file system options. Only select FAT VFAT for possible Dos partition access. Select Ext2 which is linux de facto file system. Select ISO9600 for possible CD-ROM access

CONFIG_VGA_CONSOLE:

Saying Y 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://metalab.unc.edu/pub/Linux/utils/console .

Say Y.

CONFIG_VIDEO_SELECT:

This enables support for text mode selection on kernel startup. If you want to take advantage of some high-resolution text mode your card's BIOS offers, but the traditional Linux utilities like SVGATextMode don't, you can say Y here and set the mode using the "vga=" option from your boot loader (lilo or loadlin) or set "vga=ask" which brings up a video mode menu on kernel startup. (Try "man bootparam" or see the documentation of your boot loader about how to pass options to the kernel.)

Read the file Documentation/svga.txt for more information about the Video mode selection support. If unsure, say N.

In this project: NO
NOT SURE

CONFIG_MDA_CONSOLE:

Say Y here if you have an old MDA or monochrome Hercules graphics adapter in your system acting as a second head (= video card). You will then be able to use two monitors with your Linux system. Do not say Y here if your MDA card is the primary card in your system; the normal VGA driver will handle it.

This driver is also available as a module (= code which can be inserted and removed from the running kernel whenever you want). The module will be called mdacon.o. If you want to compile it as a module, say M here and read Documentation/modules.txt.

In this project: NO unsure

CONFIG_SOUND:

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

You want to read the Sound-HOWTO, available from
http://www.linuxdoc.org/docs.html#howto . General information

about the modular sound system is contained in the files Documentation/sound/Introduction. The file Documentation/sound/README.OSS contains some slightly outdated but still useful information as well.

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/), then you need to compile the sound card support as a module (= code which can be inserted in and removed from the running kernel whenever you want) and load that module after the PnP configuration is finished. To do this, say M here and read Documentation/modules.txt as well as Documentation/sound/README.modules; the module will be called soundcore.o.

I'm told that even without a sound card, you can make your computer say 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/ .

In this project: NO
Won’t use Sound card, although we have sound card on board.

CONFIG_SOUND_EMU10K1:

Say Y or M if you have a PCI sound card using the EMU10K1 chipset, such as the Creative SBLive!, SB PCI512 or Emu-APS.

For more information about the degree of support for the different card models please check:

    http://opensource.creative.com

It is now possible to load dsp microcode patches into the EMU10K1 chip. These patches are used to implement real time sound processing effects which include for example: signal routing, bass/treble control, AC3 passthrough, ...Userspace tools to create new patches and load/unload them can be found
at the above link. You need to get the source snapshot and then type:

```
      % make tools
      % make install-tools
```

in the top directory.

CONFIG_SOUND_FUSION:

This module drives the Crystal SoundFusion devices (CS4280/46xx series)
when wired as native sound drivers with AC97 codecs. If this driver
does not work try the CS4232 driver.

CONFIG_USB:
CONFIG_USB_DEBUG:
CONFIG_USB_DEVICEFS:
CONFIG_USB_BANDWIDTH:
CONFIG_USB_UHCI:
CONFIG_USB_OHCI:
CONFIG_USB_AUDIO:
CONFIG_USB_BLUETOOTH:
CONFIG_USB_STORAGE:
CONFIG_USB_ACM:
CONFIG_USB_PRINTER:
CONFIG_USB_HID:
CONFIG_USB_KBD:
CONFIG_USB_MOUSE:
CONFIG_USB_WACOM:
CONFIG_USB_DC2XX:
CONFIG_USB_SCANNER:
CONFIG_USB_IBMCAM:
CONFIG_USB_OV511:
CONFIG_USB_USS720:
CONFIG_USB_SERIAL:
In this project: For the above USB options, Say No because we won’t use any USB devices in this project. Just to keep kernel minimum