

The Linux Kernel: Configuring the Kernel

Part 16

In this article, we will discuss the input/output ports.

First, the "i8042 PC Keyboard controller" driver is needed for PS/2 mice and AT keyboards. Before USB, mice and keyboards used PS/2 ports which are circular ports. The AT keyboard is an 84-key IBM keyboard that uses the AT port. The AT port has five pins while the PS/2 port has six pins.

Input devices that use the COM port (sometime called RS232 serial port) will need this driver (Serial port line discipline). The COM port is a serial port meaning that one bit at a time is transferred.

The TravelMate notebooks need this special driver to use a mouse attached to the QuickPort (ct82c710 Aux port controller).

Parallel port adapters for PS/2 mice, AT keyboards, and XT keyboards use this driver (Parallel port keyboard adapter).

The "PS/2 driver library" is for PS/2 mice and AT keyboards.

"Raw access to serio ports" can be enabled to allow device files to be used as character devices.

Next, there is a driver for the "Altera UP PS/2 controller".

The PS/2 multiplexer also needs a driver (TQC PS/2 multiplexer).

The ARC FPGA platform needs special driver for PS/2 controllers (ARC PS/2 support).

NOTE: I want to make it clear that the PS/2 controllers that are discussed in this article are not Sony's game controllers for their PlayStation. This article is discussing the 6-pin mouse/keyboard ports. The controller is the card that holds the PS/2 ports.

The "Gameport support" driver offers support for the 15-pin gameport. Gameport was the 15-pin port used by many input gaming devices until the invention of the USB port.

The next driver is for gameports on ISA and PnP bus cards (Classic ISA and PnP gameport support). ISA stands for Industry Standard Architecture and was a parallel bus standard before PCI. PnP stands for Plug-and-Play and was a common standard before ISA.

"PDPI Lightning 4 gamecard support" provides a driver for a proprietary gamecard with gameports.

The SoundBlaster Audigy card is a proprietary gameport card (SB Live and Audigy gameport support).

The ForteMedia FM801 PCI audio controller has a gameport on the card (ForteMedia FM801 gameport support). This driver only supports the gameport.

Next, we can move on to "Character devices". Character devices transfer data character by character.

First, TTY can be enabled or disabled (Enable TTY). Removing TTY will save a lot of space, but TTY is needed for terminals and such. Unless you know what you are doing, do not disabled TTY.

NOTE TO MY FANS: If you know of a reason for disabling TTY, could you post the answer below and share with us. Mahalo!

Next, support for "Virtual terminals" can be enabled/disabled. Again, a lot of space can be saved, but virtual terminals are very important.

This next driver supports font mapping and Unicode translation (Enable character translations in console). This can be used to convert ASCII to Unicode.

Virtual terminals can be used as system consoles with this driver (Support for console on virtual terminal). A system console manages the logins and kernel messages/warnings.

Virtual terminals must channel through a console driver to interact with the physical terminal (Support for binding and unbinding console drivers). Before the virtual terminal can do so, the console driver must be loaded. When the virtual terminal is closed, the console terminal must be unloaded.

The next driver provides support for Unix98 PTY (Unix98 PTY support). This is Unix98 pseudo terminal.

FUN FACT: The Linux kernel allows a filesystem to be mount many times in many places at once.

Next, "Support multiple instances of devpts" can be supported. The devpts filesystem is for pseudo-terminal slaves.

Legacy support for PTY can also be enabled (Legacy (BSD) PTY support).

The max amount of legacy PTYs in use can be set (Maximum number of legacy PTY in use).

The next driver can be used to offer support to serial boards that the other drivers fail to support (Non-standard serial port support).

Next, there are some drivers for specific boards and cards.

The GSM MUX protocol is supported with this driver (GSM MUX line discipline support (EXPERIMENTAL)).

The next driver enables the kmem device file (/dev/kmem virtual device support). kmem is usually used for kernel debugging. kmem can be used to read certain kernel variables and states.

The Stallion cards have many serial ports on them (Stallion multiport serial support). This driver specifically supports this card.

Next, we can move on to drivers for serial devices. As stated before, serial devices transfer one bit at a time.

The first driver is for standard serial port support (8250/16550 and compatible serial support).

Plug-and-Play also exists for serial ports with this driver (8250/16550 PNP device support).

The following driver allows the serial ports to be used for connecting a terminal to be used as a console (Console on 8250/16550 and compatible serial port).

Some UART controllers support Direct Memory Access (DMA support for 16550 compatible UART controllers). UART stands for Universal Asynchronous Receiver/Transmitter. UART controllers convert serial to parallel and vice versa.

Next, this driver offers support for standard PCI serial devices (8250/16550 PCI device support).

16-bit PCMCIA serial devices are supported by this driver (8250/16550 PCMCIA device support). Remember, PCMCIA is a PC-card that is usually used in laptops.

The maximum number of supported serial ports can be set (Maximum number of 8250/16550 serial ports) and then the maximum that are registered during boot-up (Number of 8250/16550 serial ports to register at runtime).

For extended serial abilities like HUB6 support, enable this driver (Extended 8250/16550 serial driver options).

A special driver is needed to support more than four legacy serial ports (Support more than 4 legacy serial ports).

Serial interrupts can be shared when this driver is used (Support for sharing serial interrupts).

Serial port IRQs can be autodetected using this driver (Autodetect IRQ on standard ports).

RSA serial ports are also supported by the Linux kernel (Support RSA serial ports). RSA stands for Remote Supervisor Adapter. RSA is an IBM-specific hardware.

Next, there are various vendor/device specific drivers.

This is a TTY driver that uses `printk` to output user messages (TTY driver to output user messages via `printk`). `Printk` (print kernel) is a special piece of software that usually prints the boot-up messages. Any string that is displayed by `printk` is usually put in the `/var/log/messages` file. The shell command "`dmesg`" displays all strings that were used by `printk`.

Next, we can enable/disable support for parallel printers (Parallel printer support).

The next driver allows a printer to be used as a console (Support for console on line printer). This means kernel messages will be literally printed at the printer. Normally when the word "`print`" was used in this article series, it meant putting data on the screen. This time, this literally means putting the data on paper.

The following driver makes the device files at `/dev/parport/` (Support for user-space parallel port device drivers). This allows some processes to access.

Again, the Linux kernel has many features and drivers, so we will discuss more drivers in the next article.

The Linux Kernel: Configuring the Kernel

Part 17

This next article will cover various drivers.

First, the "virtio console" is a virtual console that is used with hypervisors.

The "IPMI top-level message handler" is a message manager for the IPMI system. IPMI stands for Intelligent Platform Management Interface. IPMI is an interface for managing the system via network without using a shell.

"/dev/nvram support" permits the system to read and write memory in the real time clock's memory. Generally, this feature is used for saving data during a power loss.

The next driver supports the Siemens R3964 packet protocol (Siemens R3964 line discipline). This is a device-to-device protocol.

Now, we can move on to PCMCIA character devices. However, most of the drivers here are vendor/device specific.

The RAW driver allows block devices to be bound to the device files /dev/raw/rawN (RAW driver (/dev/raw/rawN)). The advantage to this is efficient zero-copy. However, most software will still prefer to access the storage through /dev/sd** or /dev/hd**.

Next, the maximum number of RAW devices can be supported is set.

The following driver makes the device file /dev/hpet (HPET - High Precision Event Timer).

NOTE: Many of you may be wondering why enabling these device file matter. Well, these device files serve as an interface between the software and hardware.

The HPET timers can be mapped with this driver (Allow mmap of HPET). Mapping is the process of making a list of address in memory of devices and files. The files can then be found faster by getting the address from the memory and then commanding the hard-drive to get the data from the address.

The "Hangcheck timer" is used to detect whether or not the system has locked-up. This timer watches for locked-up processes. As soon as a process freezes, a timer starts. After the timer goes off, if the process has not restarted or closed, then the timer will force the process to close.

Linus Torvalds Quote: Portability is for people who cannot write new programs.

The TPM security chip that uses Trusted Computing Group's specification will need this driver (TPM Hardware Support).

Now, we can move on to I2C devices. I2C stands for Inter-Integrated Circuit and is spoken as "eye two see". However, some people say "eye squared see". I2C is a serial bus standard.

Some old software used I2C adapters as class devices, but software now does not do that (Enable compatibility bits for old user-space). So, this driver will offer backwards compatibility for older software.

Next, the I2C device files can be made (I2C device interface).

I2C can support multiplexing with this driver (I2C bus multiplexing support).

I2C can support GPIO-controlled multiplexing with this driver (GPIO-based I2C multiplexer).

Various tests can be performed on I2C and SMBus with this driver for developers (I2C/SMBus Test Stub).

The I2C system will produce debugging messages with this feature enabled (I2C Core debugging messages).

The next driver produces additional I2C debugging messages (I2C Algorithm debugging messages).

Linus Torvalds Quote: The main reason there are no raw devices [in Linux] is that I personally think that raw devices are a stupid idea.

The following driver will cause the I2C drivers to produce debugging messages (I2C Bus debugging messages).

Next, we have Serial Peripheral Interface support (SPI support). SPI is a synchronous serial protocol used on SPI buses.

After that, there is a driver for High speed synchronous Serial Interface support (HSI support). HSI is a synchronous serial protocol.

PPS can also be supported by the Linux kernel (PPS support).

The "IP-over-InfiniBand" driver allows IP packets to be transported over InfiniBand.

After that, there is a debugging driver for IP-over-InfiniBand (IP-over-InfiniBand debugging).

SCSI's RDMA protocol can also travel over InfiniBand (InfiniBand SCSI RDMA Protocol).

There is also an extension for the iSCSI protocol to transmit over InfiniBand (iSCSI Extensions for RDMA (iSER)).

Sometimes, errors occur in the core system that the whole system must know (EDAC (Error Detection And Correction) reporting). This driver sends the core errors to the system. Generally, such low-level errors are reported in the processor and then seen by this driver to let other system processes know about or handle the error.

This driver provides legacy support for EDAC to use older versions of sysfs (EDAC legacy sysfs).

EDAC can be set to send debugging information to the logging system of Linux (Debugging).

Linus Torvalds Quote: Nobody actually creates perfect code the first time around, except me.

The Machine Check Exceptions (MCEs) are converted to a readable form via this driver (Decode MCEs in human-readable form (only on AMD for now)).

MCEs are hardware errors detected by the CPU. MCEs usually trigger kernel panics.

The decoding process for MCE to a readable form can be injected to test error handling (Simple MCE injection interface over /sysfs).

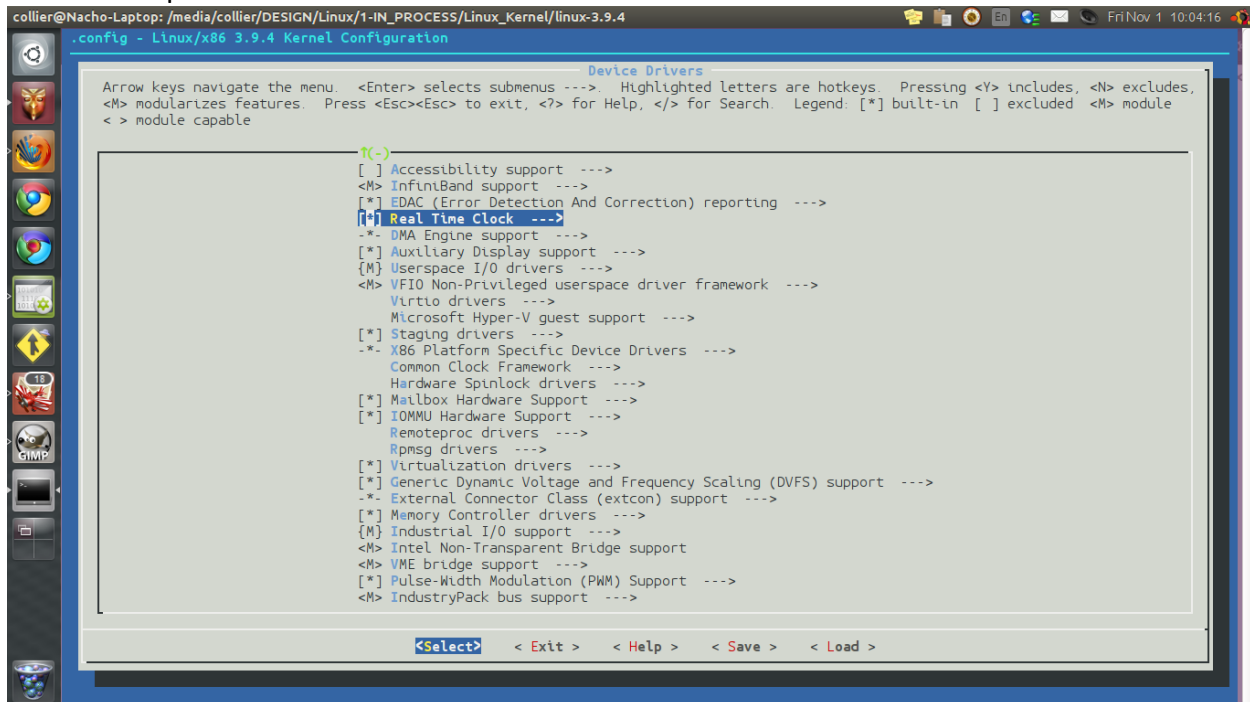
The next driver allows errors to be detected in memory and then corrected (Main Memory EDAC (Error Detection And Correction) reporting).

Next, there are many drivers that detect and correct errors on specific hardware sets.

Linus Torvalds Quote: Theory and practice sometimes clash. And when that happens, theory loses. Every single time.

Now, we can move on to the "Real Time Clock". This is commonly abbreviated "RTC".

The RTC keeps track of time.



The next setting allows us to make the Linux system use the time from the RTC as the time on the "wall clock" (Set system time from RTC on startup and resume). The wall clock is the clock on the desktop or the time seen using the "date" command.

Alternately, the wall clock can get the time from an NTP server and then sync with the RTC (Set the RTC time based on NTP synchronization).

Some systems have more than one RTC, so the user must set which is the default (RTC used to set the system time).

It is best to make the first one (/dev/rtc0) the primary clock.

Debugging abilities can be set for the RTC system (RTC debug support).

The RTC can use various interfaces for giving the operating system the current time. Using sysfs will require this driver (/sys/class/rtc/rtcN (sysfs)) while using proc will require this driver (/proc/driver/rtc (procfs for rtcN)). Special RTC character devices can be made and used (/dev/rtcN (character devices)). The shell command "hwclock" uses /dev/rtc, so the RTC character devices are needed.

The next driver allows interrupts of the RTC to be emulated on the /dev/ interface (RTC UIE emulation on dev interface). This driver reads the clock time and allows the new time to be retrieved from /dev/.

The RTC system can be tested with the test driver (Test driver/device).

Next, we will discuss the Direct Memory Access system.

DMA is the process of hardware accessing the memory independently of the processor. DMA increases system performance because the processor will have less to do if the hardware is performing more tasks for itself. Otherwise, the hardware would be waiting for the processor to complete the task.

The debugging engine is for debugging the DMA system (DMA Engine debugging).

Next, there are many vendor/device specific drivers for DMA support.

Some DMA controllers support big endian reading and writing with this driver (Use big endian I/O register access).

Big endian refers to the arrangement of the binary code. The number system used in English speaking countries places the largest end of the number on the left. For example, in the number 17, the most left numbers place is the tens place which is larger than the ones place. In big endian, each byte is arranged with the largest portion on the left. A byte is eight bits. Example: 10110100. Each place has a value of 128, 64, 32, 16, 8, 4, 2, and 1 respectively. So the byte mentioned converts to the decimal number 180.

The DMA system can use the network to reduce CPU usage (Network: TCP receive copy offload).

The "DMA Test Client" is used for testing the DMA system.

In the next article, we will discuss the display/video drivers.

The Linux Kernel: Configuring the Kernel

Part 18

Aloha! Ready for the next article? In this article, we will discuss the auxiliary-screen. The auxiliary displays are small LCD screens; most are equal to or less than 128x64. Then, we will discuss Userspace IO drivers, some virtual drivers, Hyper-V, staging drivers, IOMMU, and other kernel features.

The first driver to configure for the auxiliary display is the "KS0108 LCD Controller" driver. The KS0108 LCD Controller is a graphics controller made by Samsung.

Next, the parallel port address for the LCD can be set (Parallel port where the LCD is connected). The first port address is 0x378, the next is 0x278 and the third is 0x3BC. These are not the only choices of addresses. The majority of people will not need to change this. The shell command "cat /proc/ioports" will list the available parallel ports and the addresses.

The kernel developer can set the writing delay of the KS0108 LCD Controller to the parallel port (Delay between each control writing (microseconds)). The default value is almost always correct, so this typically does not need to be changed.

The "CFAG12864B LCD" screen is a 128x64, two-color LCD screen. This screen relies on the KS0108 LCD Controller.

The refresh rate of these LCD screens can be changed (Refresh rate (hertz)). Generally, a higher refresh rate causes more CPU activity. This means slower systems will need a smaller refresh rate.

After the auxiliary displays are configured, the "Userspace I/O drivers" are then set. The userspace system allows the user's applications and processes to access kernel interrupts and memory addresses. With this enabled, some drivers will be placed in the userspace.

The "generic Hilscher CIF Card driver" is a userspace driver for Profibus cards and Hilscher CIF Cards.

The "Userspace I/O platform driver" creates a general system for drivers to be in the userspace.

The next driver is the same as above, but adds IRQ handling (Userspace I/O platform driver with generic IRQ handling).

The following driver is again like the one before, but with dynamic memory abilities added

(Userspace platform driver with generic irq and dynamic memory).

Next, some vendor/device specific drivers are available.

Then, there is a generic PCI/PCIe card driver (Generic driver for PCI 2.3 and PCI Express cards).

The following driver is for "VFIO support for PCI devices". VFIO stands for Virtual Function Input/Output. VFIO allows devices to directly access userspace in a secure fashion.

The "VFIO PCI support for VGA devices" allows VGA to be supported by PCI through VFIO.

Next, are virtio drivers. Virtio is a IO virtualization platform. This virtual software is for operating system virtualization. This is required for running an operating system in a virtual machine on the Linux system.

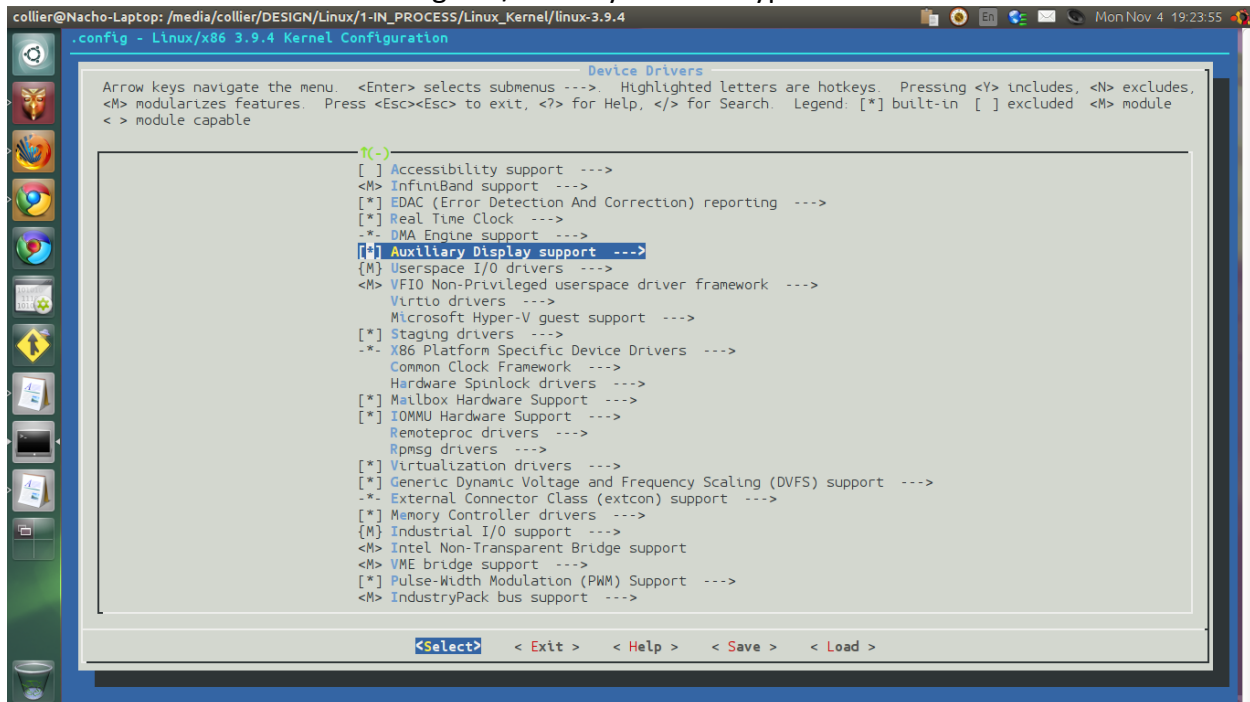
The first virtio driver we can configure is the "PCI driver for virtio devices". This allows virtual access to PCI.

The "Virtio balloon driver" allows the memory owned by a virtual system to be expanded or decreased as needed. Generally, no one wants a virtual system to reserve memory it may never use when the host operating system needs the memory.

The following driver supports memory mapped virtio devices (Platform bus driver for memory mapped virtio devices).

If the Linux kernel being configured is intended to run on a Microsoft Hyper-V system, then enable this driver (Microsoft Hyper-V client drivers).

This would allow Linux to be the guest/client system on Hyper-V.



Next, we have the staging drivers. These are drivers that are under development, may change soon, or are not up to the standard quality for the Linux kernel. The only group of drivers in this category (in this kernel version 3.9.4) are the Android drivers. Yes, Android uses the Linux kernel which would make Android a Linux system. However, this is still debated. If the kernel is intended for Android, then it may be wise to enable all of the drivers.

The "Android Binder IPC Driver" provides support for Binder which is a system that allows processes to communicate with each other on Android systems.

The ashmem driver can be enabled next (Enable the Anonymous Shared Memory Subsystem). Ashmem stands for "Anonymous SHared MEMory" or "Android SHared MEMory". This supports a file-based memory system for userspace.

The "Android log driver" offers the complete Android logging system.

The "Timed output class driver" and "Android timed gpio driver" allow the Android system to manipulate GPIO pins and undo the manipulations after the timeout.

The "Android Low Memory Killer" closes processes when more memory is needed. This feature kills the tasks that are not used or inactive.

The "Android alarm driver" makes the kernel wakeup at set intervals.

After the staging drivers are configured, the next set of drivers are for the X86 platform. These drivers are vendor/device specific for X86 (32-bit) hardware.

The next driver is for "Mailbox Hardware Support". This framework controls mailbox queues and interrupt signals for hardware mailbox systems.

"IOMMU Hardware Support" links the memory to devices that are able to use DMA. IOMMU enhances DMA. The IOMMU maps addresses and blocks faulty devices from accessing the memory. IOMMU also allows hardware to access more memory than it could without IOMMU.

The "AMD IOMMU support" driver offers better IOMMU support for AMD devices.

Debugging abilities exist for the AMD IOMMU support (Export AMD IOMMU statistics to debugfs).

A newer version of the IOMMU driver exists for AMD hardware (AMD IOMMU Version 2 driver).

The Linux kernel also provides an IOMMU driver specifically for Intel devices (Support for Intel IOMMU using DMA Remapping Devices).

Some devices may be able to accept a variety of voltages and clock frequencies. This driver allows the operating system to control the device's voltage output and clock rate (Generic Dynamic Voltage and Frequency Scaling (DVFS) support). With this driver enabled, other kernel features can be enabled for power/performance management as seen below.

"Simple Ondemand" is like above, but specifically changes the clock rate based on the device's activity. Generally, more activity means the device needs a faster clock speed to accommodate for the larger resource demand.

"Performance" allows the system to set the clock speed to the maximum supported amount for best performance. This increases power consumption.

"Powersave" sets the clock rate to the lowest value to save power.

"Userspace" allows the userspace to set the clock speed.

"External Connector Class (extcon) support" provides the userspace with a way to watch external connectors like USB and AC ports. This allows applications to know if a cable was plugged into a port. Users will almost always want this enabled. If anyone has purposely disabled this for a legitimate reason, please share with us why that would be needed.

The "GPIO extcon support" driver is just like the above driver, but is made specifically for GPIO pins.

Next, there is a list of various vendor/device specific controllers for memory (Memory Controller drivers). Memory chip controllers may be separate devices or built inside the memory chips. These controllers manage the incoming and outgoing data flow.

The "Industrial I/O support" driver provides a standard interface for sensors despite the bus type they are on (that is, PCIe, spi, GPIO, etc.). IIO is a common abbreviation for Industrial Input/Output.

The Linux kernel offers support for a large variety of accelerometers, amplifiers, analog to digital converters, inertial measurement units, light sensors, magnetometer sensors, and many other sensors and converters.

The "Intel Non-Transparent Bridge support" driver supports PCIe hardware bridges which connect to systems. All writes to mapped memory will be mirrored on both systems.

"VME bridge support" is the same as above except the bridge uses VME which is a different bus standard.

"Pulse-Width Modulation (PWM) Support" controls the back-light and fan speed by regulating the average power received by such devices.

"IndustryPack bus support" offers drivers for the IndustryPack bus standards.

In the next article, we will configure the firmware drivers.

The Linux Kernel: Configuring the Kernel

Part 19

Aloha and welcome to the next article of the Linux kernel series! We are getting closer to the end of the configuration process. In this article, we will discuss firmware drivers and then the filesystem drivers.

The first driver in this category is for finding the boot-disk (BIOS Enhanced Disk Drive calls determine boot disk). Sometimes, Linux does not know which drive is the bootable drive. This driver allows the kernel to ask the BIOS. Linux then stores the information on sysfs. Linux needs to know this for setting up bootloaders.

Even if BIOS EDD services are compiled in the kernel, this option can set such services to be inactive by default (Sets default behavior for EDD detection to off). EDD stands for Enhanced Disk Drive.

When using kexec to load a different kernel, performance can be increased by having the firmware provide a memory map (Add firmware-provided memory map to sysfs).

The "Dell Systems Management Base Driver" gives the Linux kernel better control of the Dell hardware via the sysfs interface.

The hardware's information can be accessed by the software via `/sys/class/dmi/id/` with this driver enabled (Export DMI identification via sysfs to userspace). DMI stands for Desktop Management Interface. The DMI manages the components of the hardware and can access the hardware's data. The structure of the data in the BIOS and hardware is regulated by the System Management BIOS (SMBIOS) specification.

The raw data tables from the DMI can be accessed with this driver (DMI table support in sysfs).

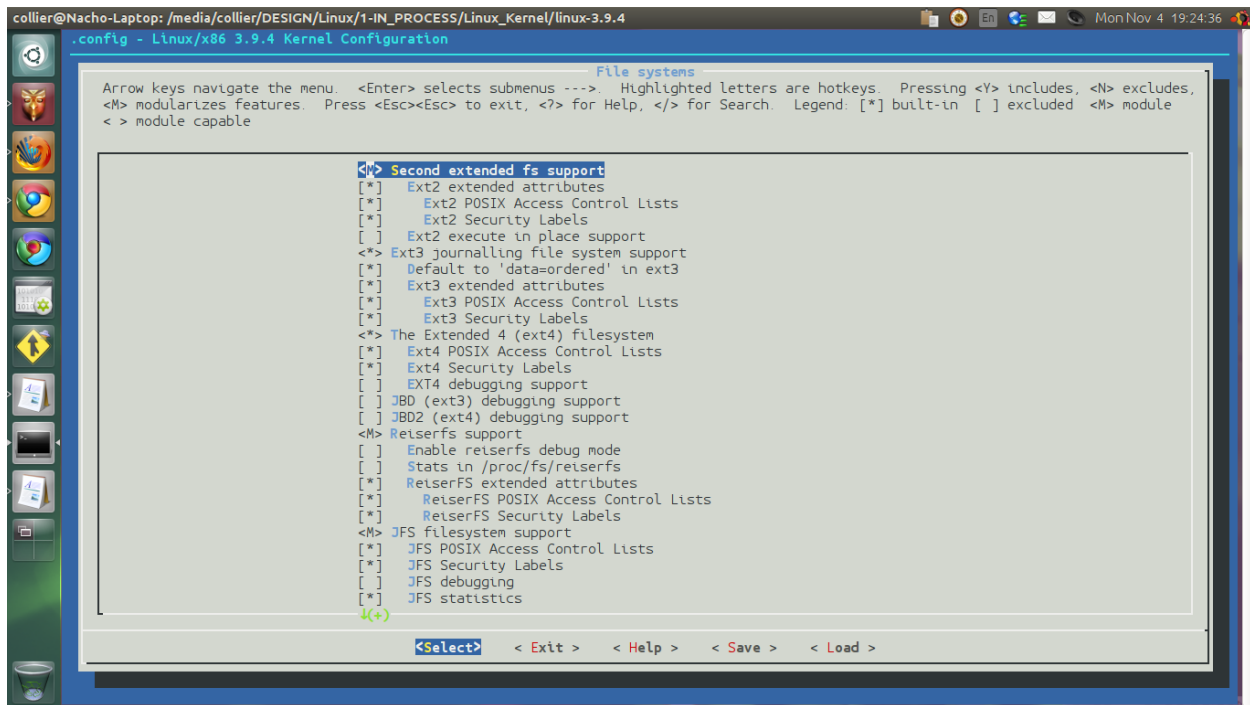
To boot from an iSCSI driver, enable this driver (iSCSI Boot Firmware Table Attributes).

The last firmware driver is a set of "Google Firmware Drivers". These are drivers for Google-specific hardware. Do not enable this driver unless you work for Google and need to use Linux on such hardware or if you are making a Linux kernel for a computer you stole from Google.

Next, we can configure the file system support of the kernel.

The "Second extended fs support" driver provides the EXT2 filesystem.

<http://www.linux.org/threads/ext-file-system.4365/>



The "Ext2 extended attributes" offers the ability to use extra metadata not natively supported by the filesystem.

The "Ext2 POSIX Access Control Lists" driver adds additional permission schemes not native to EXT2.

The "Ext2 Security Labels" enhances the security provided by SELinux.

Enabling "Ext2 execute in place support" allows executables to be executed in the current location without being executed using the paged cache.

The EXT3 filesystem is offered by this driver (Ext3 journalling file system support).

<http://www.linux.org/threads/ext-file-system.4365/>

The "Default to 'data=ordered' in ext3" driver sets the data ordering mode to "Ordered". This deals with the way the journalling and writing work. Data ordering is explained in this article -

<http://www.linux.org/threads/journal-file-system.4136/>

The "Ext3 extended attributes" offers the ability to use extra metadata not natively supported by the filesystem. Again, the following EXT3 drivers/features are the same as for EXT2 - "Ext3 POSIX Access Control Lists" and "Ext3 Security Labels". Also, the same is true for the following EXT4 drivers/features - "Ext4 POSIX Access Control Lists", "Ext4 Security Labels", and "EXT4 debugging support".

Journal Block Device debugging is supported by EXT3 (JBD debugging support) and EXT4 (JBD2 debugging support).

The next driver offers the Reiser filesystem (Reiserfs support).

<http://www.linux.org/threads/reiser-file-system-reiser3-and-reiser4.4403/>

Debugging exists for the Reiser filesystem (Enable reiserfs debug mode).

The kernel can store ReiserFS statistics in /proc/fs/reiserfs (Stats in /proc/fs/reiserfs).

The following Reiser drivers/features are the same as the ones for EXT2/3/4 - "ReiserFS extended attributes", "ReiserFS POSIX Access Control Lists", and "ReiserFS Security Labels".

JFS is also supported by the Linux kernel and includes various features - "JFS filesystem support", "JFS POSIX Access Control Lists", "JFS Security Labels", "JFS debugging", and "JFS statistics". <http://www.linux.org/threads/journald-file-system-jfs.4404/>

Again, XFS is supported with drivers/features that can be enabled - "XFS filesystem support", "XFS Quota support", "XFS POSIX ACL support", "XFS Realtime subvolume support", and "XFS Debugging support". <http://www.linux.org/threads/xfs-file-system.4364/>

The Global FileSystem 2 is supported by the Linux kernel (GFS2 file system support). This filesystem is used to share storage in a cluster.

The "GFS2 DLM locking" driver offers a distributed lock manager (DLM) for GFS2.

The Oracle Cluster FileSystem 2 is supported by the Linux kernel (OCFS2 file system support). This filesystem is used to share storage in a cluster.

The "O2CB Kernelspace Clustering" driver offers various services for the OCFS2 filesystem.

The "OCFS2 Userspace Clustering" driver allows the cluster stack to execute in userspace.

The "OCFS2 statistics" driver allows the user to get statistics concerning the filesystem.

Like with most of the Linux kernel, the OCFS2 offers logging (OCFS2 logging support). This may be used to watch for errors or for debugging purposes.

The "OCFS2 expensive checks" driver offers storage consistency checks at the cost of performance. Some Linux users recommend only enabling this feature for debugging purposes.

The kernel also contains the new B-Tree FileSystem; this driver offers the disk formatter (Btrfs filesystem Unstable disk format). BTRFS is still in development and is planned to one day become as popular or more popular than EXT4. <http://www.linux.org/threads/b-tree-file->

[system-btrfs.4430/](http://www.linux.org/threads/system-btrfs.4430/)

The "Btrfs POSIX Access Control Lists" driver adds additional permission schemes not native to BTRFS.

Next, there is a BTRFS check tool (Btrfs with integrity check tool compiled in (DANGEROUS)). Since, BTRFS is a newly developing filesystem, most of the software associated with it are unstable.

The NIL-Filesystem is also supported by Linux (NILFS2 file system support).

<http://www.linux.org/threads/new-implementation-of-a-log-structured-file-system-nilfs.4547/>

To support the flock() system call used by some filesystems, enable this driver (Enable POSIX file locking API). Disabling this driver will reduce the kernel size by about eleven kilobytes. The driver provides file-locking. File-locking is the process of allowing one process to read a file at a time. This is commonly used with network filesystems like NFS.

The "Dnotify support" driver is a legacy filesystem notification system that informs the userspace of events on the file system. One use of this and the successor notifications software is to monitor the filesystem for applications. Certain applications tell this daemon what events to watch. Otherwise, each userspace application would need to complete this task themselves.

Remember, Dnotify is a legacy system, so what is the new notification system? It is Inotify which is provided by this driver (Inotify support for userspace).

An alternative notification system is fanotify (Filesystem wide access notification). Fanotify is the same as Inotify, but fanotify relays more information to the userspace than Inotify.

Fanotify can check permissions with this driver enabled (fanotify permissions checking).

For systems that need to divide the storage space by user will want "Quota support".

<http://www.linux.org/threads/atomic-disk-quotas.4277/>

The following driver allows disk quota warnings and messages to be reported through netlink (Report quota messages through netlink interface). Netlink is a socket interface on the userspace that communicates with the kernel.

Quota messages can also be sent to a console (Print quota warnings to console (OBSOLETE)).

This driver allows the quota system to perform extra sanity checks (Additional quota sanity checks). In computer technology, a sanity check is the process of checking for errors that may be due to poor programming. The files and output are inspected to ensure the data is what it should be and not structured in some odd fashion.

Some old system use the old quota system but want to retain the old quota system when upgrading to a newer kernel. This is easily solved by enabling this driver (Old quota format support). Many readers may be wondering why someone would want to keep the old quota system instead of upgrading. Well, imagine being the manager of the IT department of a very large corporation that has many servers running very important tasks. Would you want to create and configure a new (and possibly large) quota system when you can continue using the one that works well? Generally, with computers, follow the principle - If it is not broken or will not cause security issues, do not fix it.

The newer quota system supports 32-bit UIDs and GIDs with this driver (Quota format vfstv0 and vfstv1 support).

To automatically mount remote storage units, enable this driver (Kernel automounter version 4 support).

FUSE filesystems are supported by this driver (FUSE (Filesystem in Userspace) support). Filesystem in Userspace (FUSE) allows any user to create their own filesystem and utilize it in userspace.

A special extension for FUSE can be used to utilize character devices in userspace (Character device in Userspace support).

In the next article, we will discuss caches, optical disc filesystems, FAT32 on Linux, and other interesting filesystem topics.

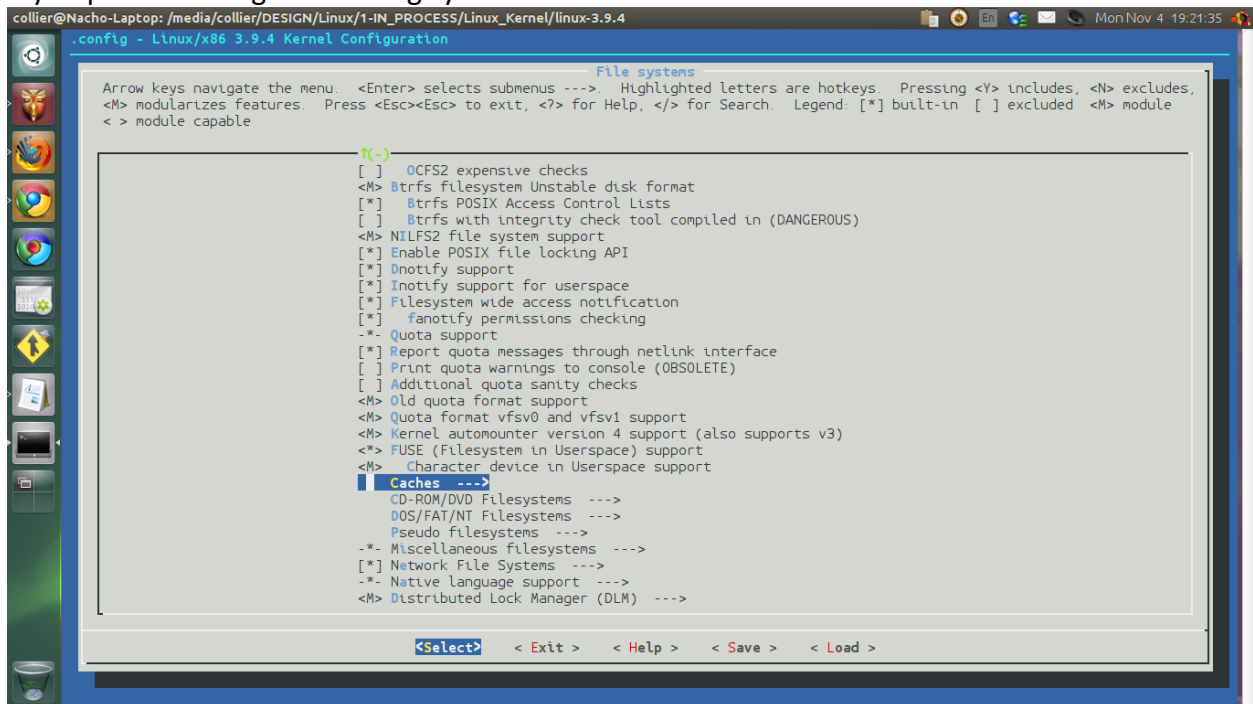
The Linux Kernel: Configuring the Kernel

Part 20

Aloha and welcome to the next Linux kernel article! In this article, we will continue configuring filesystem support.

First, we can enable "General filesystem local caching manager" which allows the kernel to store filesystem cache. This can enhance performance at the cost of storage space.

The caching system can be monitored with statistical information used for debugging purposes (Gather statistical information on local caching). Generally, this feature should only be enabled if you plan to debug the caching system.



This next feature is a lot like the above, but this feature stores latency information (Gather latency information on local caching). Again, this is a debugging feature.

The "Debug FS-Cache" driver offers many other debugging abilities for the cache system.

The next cache debugging tool keeps a global list (any process can access the list) of filesystem cache objects (Maintain global object list for debugging purposes).

To enhance the speed of network filesystems, enable this next driver (Filesystem caching on files). This feature allows a whole local filesystem to be used as cache for remote filesystem and

storage units. The Linux kernel will manage this partition.

Two different debugging drivers exist for this local cache system for remote filesystems (Debug CacheFiles) and (Gather latency information on CacheFiles).

The most common optical disc filesystem is ISO-9660 which is ISO standard 9660, hence the name (ISO 9660 CDROM file system support). This driver is needed to read/write the major of optical discs.

When reading an optical disc with files using long Unicode filenames or writing such files, this driver is required (Microsoft Joliet CDROM extensions). This is an extension to the ISO-9660 filesystem.

The "Transparent decompression extension" allows data to be written to a disc in a compressed form and read off the disc and decompressed transparently. This will allow more data to be placed on the disc.

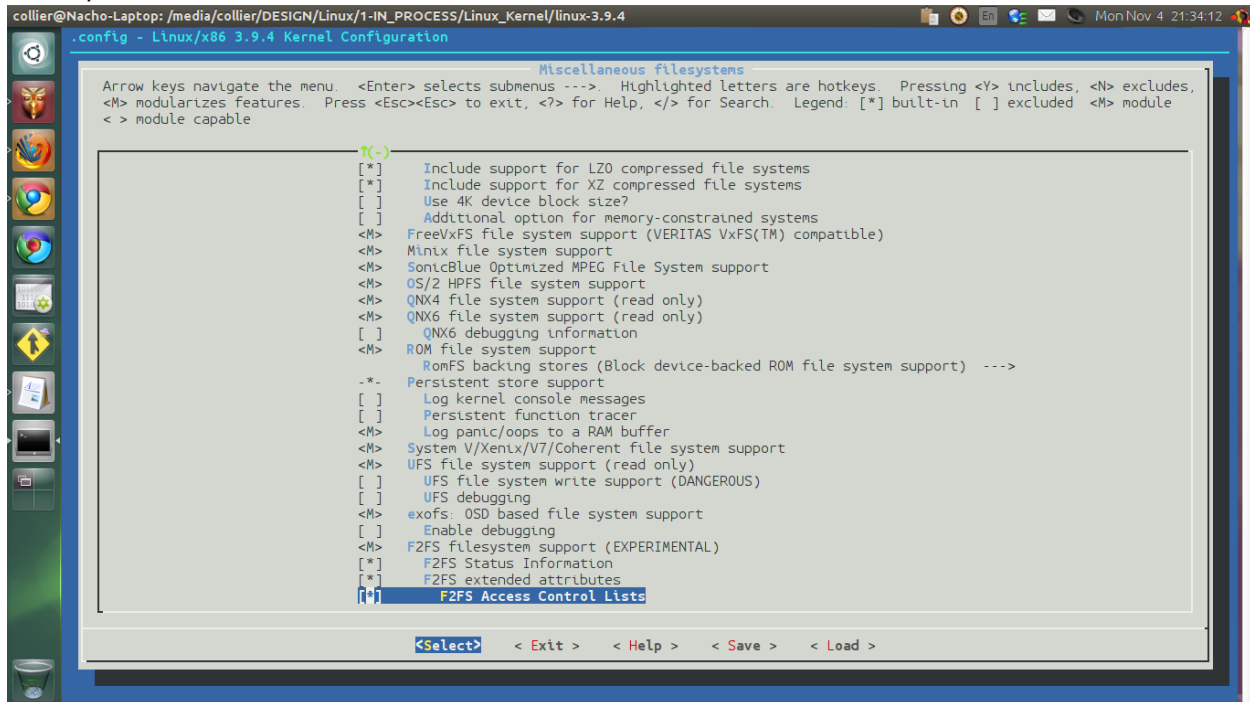
"UDF file system support" allows the kernel to read/write rewritable-optical-discs that are using the UDF filesystem. UDF is designed to manage incremental writes. UDF allows the rewritable optical disc to be used more like flash drives. The system can write and update the optical disc's data more quickly than regular writing on ISO-9660 filesystems. However, this is not faster than using flash drives.

As many of you know, Windows is a very popular system, so many storage units are using the FAT filesystem or NTFS. Thankfully, Linux supports such filesystems. The "MSDOS fs support" driver is a general driver for MS-DOS filesystems. This will increase the kernel size significantly, but since the FAT filesystems are very common, this size increase is usually worth the cost.

<http://www.linux.org/threads/file-allocation-table-fat.4472/>

To support the FAT filesystems, enable this driver (VFAT (Windows-95) fs support).

At the time this article was written, this driver does not support FAT64 (commonly called exFAT).



The size of the codepage can be set here (Default codepage for FAT).

After that, the default character set is configured for the FAT filesystems (Default iocharset for FAT).

The NTFS file system is offered with this driver (NTFS file system support). The driver provides read-only abilities. To write to NTFS, enable this driver (NTFS write support).

The Linux kernel offers debugging tools for the NTFS filesystem (NTFS debugging support).

To have a proc folder in the root, this feature must be enabled (/proc file system support). Some other similar drivers that rely on this one include (/proc/kcore support), (/proc/vmcore support), and (Sysctl support (/proc/sys)). The proc system (short for "process") uses the proc-filesystem sometimes called procfs. This filesystem is in the hardware's memory and is created when Linux boots up. So, when viewing files in proc, the user is browsing the memory as if it were like other storage units. Proc acts as an interface between userspace and the kernelspace. Proc is in the kernelspace.

The "Enable /proc page monitoring" driver offers some proc files that monitor the memory utilization of processes.

The "sysfs file system support" driver creates the /sys/ folder. The sysfs filesystem is in memory and provides an interface to the kernel objects.

The tmp directory is needed by many applications and Linux itself, so it is strongly recommended that this driver be enabled (Tmpfs virtual memory file system support (former shm fs)). The tmp filesystem maybe stored on the hard-drive or in memory and is used only to store temporary files.

The "Tmpfs POSIX Access Control Lists" driver offers extra permission features for the files in the tmpfs virtual filesystem.

The "Tmpfs extended attributes" driver provides more attributes to tmpfs files than what they would normally have without the driver.

The "HugeTLB file system support" driver provides the hugetlbfs filesystem, which is ramfs based. This virtual filesystem contains HugeTLB pages.

The configfs filesystem is a kernel object manager in the form of a filesystem (Userspace-driven configuration filesystem). It is highly recommended that this driver be enabled. ConfigFS is a lot like sysfs. However, ConfigFS is used to create and delete kernel object while sysfs is used to view and modify kernel objects.

Next, we can move back to "real" filesystems. That is, the filesystems users themselves use to store their personal files. Next, the kernel can be given the ability to read ADFS filesystems (ADFS file system support).

The ability to write to ADFS filesystems is provided by a separate and unstable driver (ADFS write support (DANGEROUS)). ADFS stands for Advanced Disc Filing System.

Linux also supports the Amiga Fast FileSystem (Amiga FFS file system support).

<http://www.linux.org/threads/amiga-fast-file-system-affs.4429/>

The "eCrypt filesystem layer support" driver offers a POSIX-compliant cryptographic filesystem layer. This eCrypt can be placed on every and any filesystem no matter what partition table the filesystem resides on. <http://www.linux.org/threads/intro-to-encryption.4376/>

The eCrypt layer can have a device file if this driver is enabled (Enable notifications for userspace key wrap/unwrap). The device path is /dev/ecryptfs.

Linux also supports HFS and HFS+ (Apple Macintosh file system support) and (Apple Extended HFS file system support). <http://www.linux.org/threads/hierarchical-file-system-hfs.4480/> and <http://www.linux.org/threads/hierarchical-file-system-plus-hfs.4493/>

The BeFS filesystem can be used by Linux as a read-only filesystem (BeOS file system (BeFS) support (read only)). Generally, it is easier to program the reading abilities for a filesystem than the writing features.

Special debugging features exist for BeFS (Debug BeFS).

EFS is another filesystem that Linux can only read, not write (EFS file system support (read only)). <http://www.linux.org/threads/extent-file-system-efs.4447/>

Some flash drives may use the JFFS2 filesystem (Journalling Flash File System v2 (JFFS2) support). Next, the debugging level can be set (JFFS2 debugging verbosity).
<http://www.linux.org/threads/journaling-flash-file-system-version-2-jffs2.4495/>

To use JFFS2 on NAND and NOR flash drives, this driver is needed (JFFS2 write-buffering support).

This next driver offers better error protection (Verify JFFS2 write-buffer reads).

JFFS filesystems can be mounted faster with "JFFS2 summary support" enabled. This driver stores information about the filesystem.

Like the other extended/extra attributes drivers for some filesystems, JFFS2 has such a driver (JFFS2 XATTR support).

The JFFS2 filesystem supports various transparent compression systems. This allows files to be smaller on JFFS2 filesystems and be read without the user needing to perform any special actions. (Advanced compression options for JFFS2), (JFFS2 ZLIB compression support), (JFFS2 LZO compression support), (JFFS2 RTIME compression support), and (JFFS2 RUBIN compression support). The default compression format can be defined in the following option (JFFS2 default compression mode).

A successor for JFFS2 exists and is supported by the kernel (UBIFS file system support). The Unsorted Block Image File System (UBIFS) also competes with LogFS.

The Linux kernel also supports LogFS (LogFS file system).

ROM-based embedded systems need support for CramFS (Compressed ROM file system support (cramfs)).

Alternately, embedded systems could use SquashFS which is a read-only compression filesystem (SquashFS 4.0 - Squashed file system support). The Linux kernel also offers extended attributes for SquashFS (Squashfs XATTR support).

There are three different compression formats supported by SquashFS - (Include support for ZLIB compressed file systems), (Include support for LZO compressed file systems), and (Include support for XZ compressed file systems). The block size for SquashFS can be set to four

kilobytes (Use 4K device block size?). Also, the cache size can be set (Additional option for memory-constrained systems).

The Linux kernel supports FreeVxFS (FreeVxFS file system support (VERITAS VxFS(TM) compatible)), Minix (Minix file system support), MPEG filesystem (SonicBlue Optimized MPEG File System support), HPFS (OS/2 HPFS file system support), QNX4 (QNX4 file system support (read only)), QNX6 (QNX6 file system support (read only)), and the ROM filesystem (ROM file system support). <http://www.linux.org/threads/qnx-file-systems.4577/> and <http://www.linux.org/threads/minix-mini-unix-file-system.4545/>

"RomFS backing stores (Block device-backed ROM file system support)" offers a list of various ROMfs extra features and abilities.

The "Persistent store support" driver provides support for the pstore filesystem which allows access to platform level persistent storage.

The pstore filesystem can store kernel logs/messages (Log kernel console messages).

When a kernel panic takes place (equivalent to the "Blue-Screen-of-Death" on Windows), the "Log panic/oops to a RAM buffer" driver will store a log in the RAM.

This next single driver offers support for the Xenix, Coherent, Version 7, and System V filesystems (System V/Xenix/V7/Coherent file system support).

The Linux kernel also supports UFS (UFS file system support (read only)), (UFS file system write support (DANGEROUS)), and (UFS debugging).

exofs is also supported by the kernel (exofs: OSD based file system support).

The Flash-Friendly FileSystem is a special filesystem for flash drives (F2FS filesystem support (EXPERIMENTAL)), (F2FS Status Information), (F2FS extended attributes), and (F2FS Access Control Lists). <http://www.linux.org/threads/flash-friendly-file-system-f2fs.4477/>

In the next article, we will configure network filesystems.

The Linux Kernel: Configuring the Kernel

Part 21

Aloha! In this next article, we will configure network filesystem support for the Linux kernel. A network filesystem is a remote filesystem that computers access via the network.

First, the "NFS client support" driver allows the Linux system to use the NFS network filesystem. There are also three other drivers for different versions of NFS - (NFS client support for NFS version 2), (NFS client support for NFS version 3), (NFS client support for NFS version 4), and (NFS client support for NFSv4.1). If you have a network that possess NFS, either figure out what version of NFS you are using, or enable all of the NFS drivers.

Swap space is not required to be on a local storage unit. This driver allows Linux to use NFS support to use remote swap spaces (Provide swap over NFS support).

The NFS system can be sped up by using a cache system (Provide NFS client caching support). This is local cache.

Enable this driver to allow DNS to use host-names for NFS servers (Use the legacy NFS DNS resolver).

"NFS server support" gives the server providing NFS the features it needs to fulfill such a task. Some other NFS drivers include (NFS server support for NFS version 3) and (NFS server support for NFS version 4).

The "NFS server manual fault injection" driver is a debugging tool that allows developers to make the NFS server think an error occurred with NFS. Specifically, this is used to test how the server handles NFS errors.

The "Secure RPC: Kerberos V mechanism" is needed to make the RPC calls secure. NFS cannot be added to the kernel without this feature for security reasons.

There is a special debugging tool for RPC (RPC: Enable dprintk debugging).

The Linux kernel supports the Ceph filesystem (Ceph distributed file system).

CIFS is a virtual filesystem used by Samba and Windows servers (CIFS support (advanced network filesystem, SMBFS successor)). CIFS stands for Common Internet FileSystem.

There are two features that can be used to debug or monitor the CIFS driver (CIFS statistics) and

(Extended statistics).

A special driver is needed to support servers with LANMAN security (Support legacy servers which use weaker LANMAN security). LANMAN or LM hash is a special password hashing function that has some weaknesses.

CIFS requires that Kerberos tickets be used before mounting to secure servers (Kerberos/SPNEGO advanced session setup). This driver offers the ability for CIFS to use userspace tools which is needed to provide the tickets.

Like other filesystems, CIFS can have extended abilities (CIFS extended attributes) and (CIFS POSIX Extensions).

This driver gets the Access Control List (ACL) from the CIFS server (Provide CIFS ACL support).

CIFS has two other debugging tools (Enable CIFS debugging routines) and (Enable additional CIFS debugging routines).

CIFS can have "DFS feature support" which allows shares to be accessed even when they are moved. DFS stands for Distributed FileSystem.

SMB2 is an improved alternative to CIFS (SMB2 network file system support). SMB2 stands for Server Message Block version 2.

Clients can store CIFS cache with this driver enabled (Provide CIFS client caching support).

Novell NetWare clients need this driver to access NetWare volumes (NCP file system support (to mount NetWare volumes)). NCP stands for NetWare Core Protocol. NCP is a protocol that allows clients to communicate with the servers hosting NetWare volumes.

NetWare servers can use NFS namespaces if this driver is enabled (Use NFS namespace if available).

NetWare servers can use the OS/2 long namespaces if this driver is enabled (Use LONG (OS/2) namespace if available).

If this driver is enabled, then filenames made by DOS or on storage units owned by a DOS system will be converted to lowercase (Lowercase DOS filenames).

Many filesystems depend on native language support (Use Native Language Support). Specifically, Native Language Support (NLS) allows the different character-sets to be used in filenames.

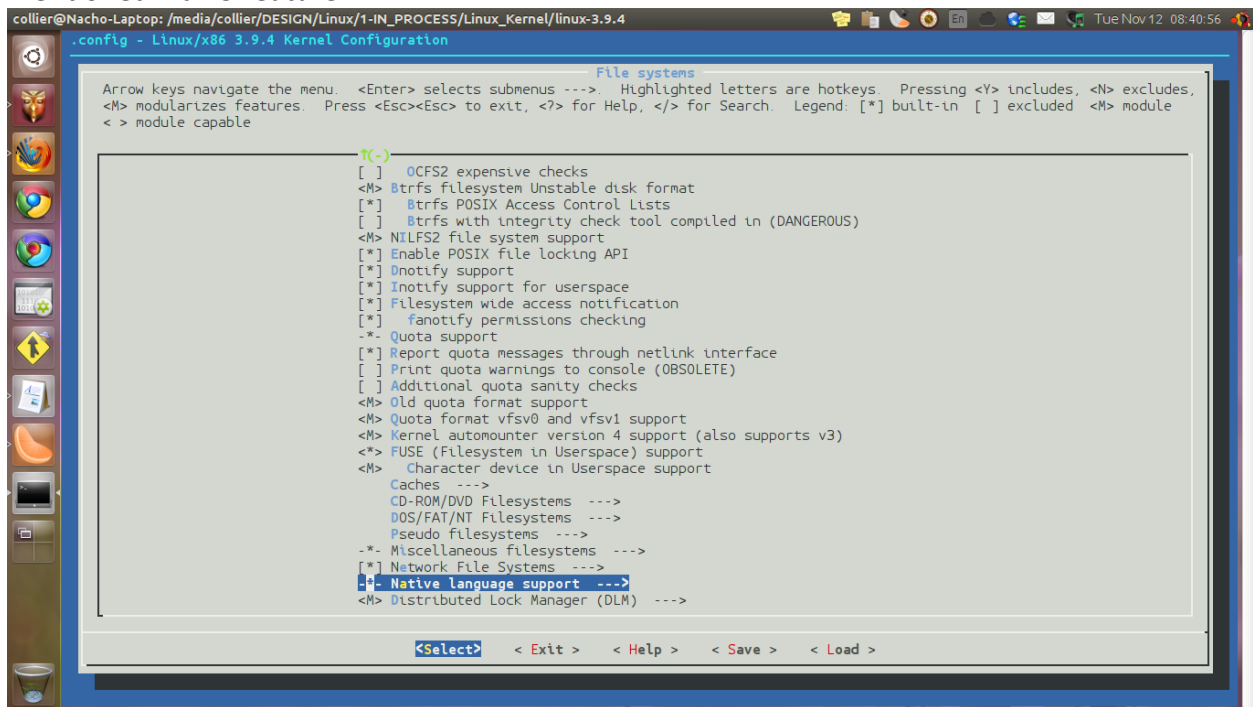
NCP filesystems can support the execute flag and symbolic links with this driver enabled (Enable

symbolic links and execute flags).

The Linux kernel offers support for the Coda filesystem (Coda file system support (advanced network fs)). Coda is one of many network filesystems.

The Linux kernel can support the Andrew Filesystem (Andrew File System support (AFS)). However, the Linux kernel can only read such filesystems in an insecure manner. This driver is intended to allow Linux systems to access AFS. If your network only contains Linux systems, then select a different network filesystem that the kernel can fully support.

The Linux kernel has an experimental driver for accessing Plan 9 resources via the 9P2000 protocol (Plan 9 Resource Sharing Support (9P2000)). The kernel also has cache support (Enable 9P client caching support) and control lists (9P POSIX Access Control Lists) for the previously mentioned Plan 9 feature.



After the network filesystems have been configured, the next part of the kernel to setup is the "Native Language Support". This whole menu contains the drivers for most or all of the character-sets and encodings. Enabling these encodings allows these character sets to be used by the system and applications. UTF-8 is the most commonly used encoding, but it is not the only one. Most applications and driver need UTF-8, so this encoding is already set to be added to the kernel.

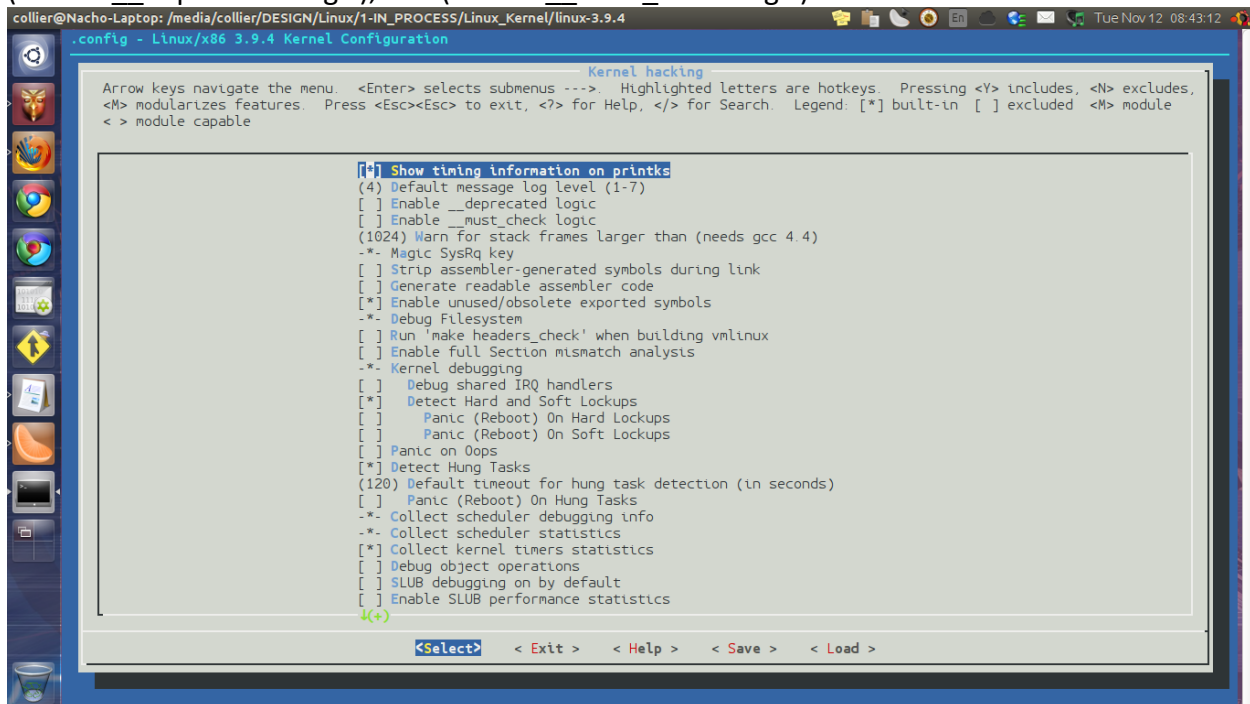
After that menu, the "Distributed Lock Manager (DLM)" can be configured. A DLM is used to keep shared resources in sync and performing well. This driver manages the userspace and kernelspace applications that access or manipulate shared resources (like network filesystems).

Clusters strongly depend on this driver.

Now that we have finally finished configuring the filesystems and related features, we can now move on to the "Kernel hacking" menu seen on the main (first) screen on the kernel configuration tool (when using the command "make menuconfig"). I am using the ncurses interface (seen in the screenshot) which is initiated with the "make menuconfig" command, so other interfaces may be a little different. Many of the features and settings in the kernel hacking menu contain various settings concerning the kernel itself. Some of these features are debugging tools and some control the kernel's behavior.

This first setting adds the printk time stamps to the syslog system call output (Show timing information on printks).

The next three features control various debugging features (Default message log level (1-7)), (Enable __deprecated logic), and (Enable __must_check logic).



The next feature is a debugging feature that is active at compiling time ((1024) Warn for stack frames larger than (needs gcc 4.4)). If stack frames are larger than the specified amount, then the compiler will warn the user.

The "Magic SysRq key" driver will enable support for the Magic SysRq key. This allows users to send the kernel special commands when Alt+PrintScreen is pressed. This works in most cases regardless of the kernel's state. However, exceptions exist. It is highly recommended that the Magic SysRq Key be enabled.

During compilation, the assembler's symbolic links will be removed during a link to reduce the output of `get_wchan()` (Strip assembler-generated symbols during link).

This next feature is for debugging purposes (Generate readable assembler code). If enabled, some kernel optimizations will be disabled so some of the assembly code will be human-readable. This will harm the kernel's speed. Only enable this if you have a specific reason for doing so.

This setting enables/disables commonly unneeded or obsolete symbols (Enable unused/obsolete exported symbols). However, some modules may need such symbols. Enabling this will increase the kernel's size. It is very unlikely that a Linux user will need such symbols. In general, disable this unless you know for a fact the user needs a symbol for an important module.

Sanity checks will be performed on user kernel headers if this setting is enabled (Run 'make headers_check' when building vmlinux).

During compilation, this feature will check for invalid references (Enable full Section mismatch analysis).

The kernel can be configured to detect soft and hard lockups (Detect Hard and Soft Lockups). When the system is frozen for more than twenty second and other tasks cannot execute, this is called a soft-lockup. If the CPU is in a loop that lasts for more than ten seconds and interrupts fail to get execution time, then this is called a hard-lockup.

The next to features set the kernel to reboot on hard and soft lockups respectively, (Panic (Reboot) On Hard Lockups) and (Panic (Reboot) On Soft Lockups).

When the kernel experiences major problems, it can be set to start a kernel panic (Panic on Oops). It is highly recommended that this setting be enabled. This will help to prevent the kernel from causing system damage and data loss.

The kernel can be set to detect hung tasks (Detect Hung Tasks). This is when a process or application locks-up or is frozen. Specifically, the application becomes uninterruptable. The following setting allows the user to define how much time must pass before a process is deemed "hung" (Default timeout for hung task detection (in seconds)).

The kernel can be set to restart when a process hangs (Panic (Reboot) On Hung Tasks). Generally, users will not want to enabled this. Would you like your computer to restart every time an application becomes frozen?

The "Kernel memory leak detector" finds and logs memory leaks.

The kernel uses frame pointers to help report errors more efficiently and include more

information (Compile the kernel with frame pointers). I will skip a lot of the debugging tools because they are self-explanatory.

As many Linux users know, when the system boots up, the boot messages appear too quickly to be read. This feature sets the delay time which will give users more time to read the messages (Delay each boot printk message by N milliseconds).

This is a special developmental feature for testing backtrace code (Self test for the backtrace code). Backtrace code is a self-test.

Block device number can be extended (Force extended block device numbers and spread them). However, this may cause booting issues, so use with caution.

We will continue to configure the kernel itself in the next article.