The Wonderful World of Linux 2.6

Joseph Pranevich - jpranevich <at> kniggit.net

Although it seems like only yesterday that we were booting up our first Linux 2.4 systems, time has ticked by and the kernel development team has just released the 2.6 kernel to the public. This document is intended as a general overview of the features in the new kernel release, with a heavy bias toward i386 Linux. Please also be aware that some of the "new" features discussed here may have been back-ported to Linux 2.4 after first appearing in Linux 2.6, either officially or by a distribution vendor. I have also included information on a handful of cases where a new feature originated during the maintenance cycle of Linux 2.4, and those will be marked as appropriate in the text.

At present, this document has been translated into ten languages. Please see the "Translations" section at the very bottom for more information.

The Story So Far...

The Linux kernel project was started in 1991 by Linus Torvalds as a Minix-like Operating System for his 386. (Linus had originally wanted to name the project Freax, but the now-familiar name is the one that stuck.) The first official release of Linux 1.0 was in March 1994, but it supported only single-processor i386 machines. Just a year later, Linux 1.2 was released (March 1995) and was the first version with support for different hardware platforms (specifically: Alpha, Sparc, and Mips), but still only single-processor models. Linux 2.0 arrived in June of 1996 and also included support for a number of new architectures, but more importantly brought Linux into the world of multi-processor machines (SMP). After 2.0, subsequent major releases have been somewhat slower in coming (Linux 2.2 in January 1999 and 2.4 in January 2001), each revision expanding Linux's support for new hardware and system types as well as boosting scalability. (Linux 2.4 was also notable in being the release that really broke Linux into the desktop space with kernel support for ISA Plug-and-Play, USB, PC Card support, and other additions.) Linux 2.6, released 12/17/03, stands not only to build on these features, but also to be another "major leap" with improved support for both significantly larger systems and significantly smaller ones (PDAs and other devices.)

Core Hardware Support

One of the most important strengths of Linux-powered operating systems is their flexibility and their ability to support a wide range of hardware platforms. While this document is geared specifically to the uses of Linux on PC-derived hardware types, the Linux 2.6 kernel has made some remarkable improvements in this area that deserve to be pointed out.

Scaling Down -- Linux for Embedded Systems

One of the two most fundamental changes to Linux in 2.6 comes through the acceptance and merging of much of the uClinux project into the mainstream kernel. The uClinux project (possibly pronounced "you-see-Linux", but more properly spelled with the Greek character "mu") is the Linux for Microcontrollers project. This variant of Linux has already been a major driver of support for Linux in the embedded market, and its inclusion in the official release should encourage further development in this space. Unlike the "normal" Linux ports that we are generally accustomed to, embedded ports do not have all the features that we associate with the kernel, due to hardware limitations. The primary difference is that these ports feature processors that do not feature an MMU. ("memory management unit" - what makes a protected-mode OS "protected") While these are generally true multitasking Linux systems, they are missing memory protection and other related features. (Without memory protection, it is possible for a wayward process to read the data of, or even crash, other processes on the system.) This may make them unusable for a multi-user system, but an excellent choice for a low-cost PDA or dedicated device. It is difficult to over-emphasize this architecture shift in Linux 2.6; all versions of Linux up to this point were derived (however indirectly) from the limitations inherent with Linus' initial work on his Intel 80386.

There are several new lines of embedded processors supported by Linux 2.6, including Hitachi's H8/300

series, the NEC v850 processor, and Motorola's line of embedded m68k processors. Motorola's offerings are the most familiar to the average Linux user as they are the guts underneath Palm Pilots starting with the first (the Palm 1000), up until the Palm III. Other models go by names such as Dragonball and ColdFire and are included on systems and evaluation boards manufactured by Motorola, Lineo, Arcturus, and others. Sadly, support for other, older m68k processors without MMUs (such as the 68000s used in early Macintoshes) is not yet covered by the new release but it is highly possible that "hobbyist" projects of the future will seek to support Linux on these and other antique system.

Although not a part of the uClinux merge, the new revision of Linux also include support for Axis Communications' ETRAX CRIS ("Code Reduced Instruction Set") processors. (Actual support for this processor arrived as a feature during the 2.4 kernel's maintenance cycle -- well after the 2.4.0 release-- so it deserves a brief mention.) These are embedded processor, but with MMUs, that is primarily used in network hardware. Related support for MMU-less variants of these processors has not yet been accepted into the kernel, but are being worked on by outside projects.

In addition to pure hardware support, there have been a number of other wins through the integration of the embedded work into the mainline kernel. While most of these changes are under the hood, changes such as ability to build a system completely without swap support add to the overall robustness of the OS.

Scaling Up -- NUMA and Bigger Iron

The second of the two most fundamental changes in Linux 2.6 happens to work in the other direction: to make Linux a more acceptable kernel on larger and larger servers. (Some of these larger servers will be i386 based, and some not.) The big change in this respect is Linux's new support for NUMA servers. NUMA (or "Non-Uniform Memory Access") is a step beyond SMP in the multi-processing world and is a major leap forward for efficiency on systems that have many processors. Current multiprocessing systems were designed with many of the same limitations as their uniprocessor counterparts, especially as only a single pool of memory is expected to serve all processors. On a many-processor system, there is a major performance bottleneck due to the extremely high contention rate between the multiple cpus onto the single memory bus. NUMA servers get around that difficulty by introducing the concept that, for a specific processor, some memory is closer than others. One easy way (and not terribly technically incorrect) to imagine this is that you have a system built with separate cards, each containing CPUs, memory, and possibly other components (I/O, etc.) There are many of these cards in a system and while they can all talk to each other, it's pretty clear that the CPUs will have the easiest time talking to the local memory (the memory on the cpu card rather than on a separate card.)You can imagine the new NUMA architecture being somewhat similar to a very tight-knit cluster at the lowest levels of hardware.

To properly support these new NUMA machines, Linux had to adapt in several respects to make the new model efficient. To start with, an internal topology API was created to actually let the kernel internals understand one processor or one memory pool's relations to I/O devices and each other. Derived from that, the Linux process scheduler now is capable of understanding these relationships and will attempt to optimize tasks for best use of local resources. Additionally, many NUMA machines are built in such a way that they have "holes" in the linear memory space "between" nodes. The new kernel is able to deal with those discontiguous cases in a reasonable way. There are many other internal changes which were made to allow Linux to support these new high-end machines, and this is definitely an area of growth for the kernel as a whole. However, this is an area where Linux is very rapidly growing and maturing and much work remains to be done to make the most efficient use of resources possible. Over the course of the next year, we can expect to see many more improvements in Linux's support for these very high-end systems.

Subarchitecture Support

While not quite as core as the two previous changes, the new revision of the kernel also includes a new concept called a "subarchitecture" which further extends Linux's reach into new hardware types. Previously, Linux often had the underlying assumption that processor types and hardware types went hand in hand. That is, that i386-descendant processors are only used on PC/AT-descendant servers. In Linux 2.4, this assumption was broken for i386 with the addition of support for SGI's Visual Workstation, a "legacy-less" platform running with an Intel chip. (And in fact, it was broken long before on many other architectures. For example, m68k has long supported Amigas, Macintoshes, and other platforms.) The big change in Linux 2.6 is that this

feature and concept was standardized so that all architectures handle this in a similar and saner way that allows for more clear separation of the components that need to be separated.

With this standardization comes two new platforms to support for i386. The first is NCR's Voyager architecture. This is a SMP system (developed before the now-standard Intel MP specification) supporting 486-686 processors in up to 32x configurations. The actual number of configurations that were sold with this architecture is relatively small, and not all machines are supported yet. (The oldest ones are unsupported.) The second architecture supported is the more widespread PC-9800 platform developed by NEC into the (almost) dominant PC platform in Japan until relatively recently. The original PC-9800 machines shipped with an 8086 processor and the line eventually evolved and matured (in parallel with the AT-descendants) until they featured Pentium-class processors and SMP support. (Of course, the support for Linux is limited to 386 or better.) Although completely unknown in the US, versions of Microsoft products up until Windows 95 were ported to run on this hardware. The line has been officially discontinued by the manufacturer in favor of more "standard" PCs.

By formalizing Linux's support for these "slightly different" hardware types, this will more easily allow the kernel to be ported to other systems, such as dedicated storage hardware and other components that use industry-dominant processor types. To be absolutely clear though, one should not take this subdivision too far. These subarchitecture have been separated because very low-level components of the system (such as IRQ routing) are slightly or radically different. This is quite different than running Linux on an X-Box, for example, where relatively little other than hardware drivers and some quirks separate the system from being a "generic" i386 system. Support for the X-Box would not be a subarchitecture.

Hyperthreading

Another major hardware advancement supported under Linux 2.6 is hyperthreading. This is the ability, currently only built into modern Pentium 4 processors but applicable elsewhere, allows a single physical processor to masquerade (at the hardware level) as two or more processors. This allows for performance boosts in some circumstances, but also adds scheduling complexity and other issues. Key in the kernel's improved support for this feature is that the scheduler now knows how to recognize and optimize processor loads across both real and virtual processors within a machine. In previous versions of Linux, it was quite possible to overwork a single processor because it was not possible to factor in the workload as a whole. One of the great things to note about this feature is that Linux was ahead of the market curve on supporting this new hardware feature transparently and intelligently. (Windows 2000 servers can see the additional faux-processors, but does not recognize them as virtual. Thus, you also require additional CPU licenses to take advantage of the feature. It was not until the Windows XP release that Microsoft completely supported this feature.)

Linux Internals

Scalability Improvements

In addition to the previously described generic features such as NUMA and hyperthreading, Linux 2.6 also has other changes for Intel servers at the top of the food chain. First and foremost is improved support for other new Intel hardware features including Intel's PAE ("Physical Address Extension") which allows most newer 32-bit x86 systems to access up to 64GB of RAM, but in a paged mode. In addition, IRQ balancing has been significantly improved on multiprocessor systems through major improvements to Linux's APIC support.

In addition to just supporting new hardware features, internal limits have been also increased when possible. For example, the number of unique users and groups on a Linux system has been bumped from 65,000 to over 4 billion. (16-bit to 32-bit), making Linux more practical on large file and authentication servers. Similarly, The number of PIDs (Process IDs) before wraparound has been bumped up from 32,000 to 1 billion, improving application starting performance on very busy or very long-lived systems. Although the maximum number of open files has not been increased, Linux with the 2.6 kernel will no longer require you to set what the limit is in advance; this number will self-scale. And finally, Linux 2.6 will include improved 64-bit support on block devices that support it, even on 32-bit platforms such as i386. This allows for filesystems up to 16TB on common hardware.

Another major scalability improvement in Linux 2.6 is that the kernel itself can now not only support more types of devices, but also support more devices of a single type. Under all iterations of Linux (and indeed, most UNIX-derived operating systems), users and applications running on a system communicate with the attached hardware using numbered device nodes. (The entries in the "/dev" directory.) These device nodes were limited to 255 "major" devices (generally, one type of device gets one or more device nodes) and 255 "minor" numbers (generally, specific devices of that type.) For example, the "/dev/sda2" device (the second partition on the first detected SCSI disk), gets a major number of 8, common for all SCSI devices, and a minor number of 2 to indicate the second partition. Different device types allocate their major and minor numbers differently, so it can't be said with assurance how many devices you can have on a Linux system. Unfortunately, this system breaks down badly on large systems where it would be possible, for example, to have many more than 255 of any specific device in a system. (Think large storage arrays, print farms, etc.) Under Linux 2.6, these limitations have been eased to allow for 4095 major device types and a more than a million subdevices per type. This increase should be more than adequate to support high-end systems for the time being.

Interactivity and Responsiveness

In addition to just scaling up, another priority with the new release has been to make the system more responsive. This is useful not only for the general desktop user (who always likes to see things respond quickly), but also to more time-critical applications where absolute preciseness is required to achieve the desired effect. Despite these changes, Linux 2.6 will not be a "hard" Real Time OS, which has very strict requirements for absolutely ensuring that actions happen predictably, but the overall responsiveness improvements should appeal to all classes of Linux users. (That said, there are external projects which have unofficial patches to provide RTOS functionality. Those projects could conceivably be made official in the next major release.)

One of the key improvements in Linux 2.6, is that the kernel is finally preemptible. In all previous versions of Linux, the kernel itself cannot be interrupted while it is processing. (On a system with multiple processors, this was true on a per-CPU basis.) Under Linux 2.6, the kernel now can be interrupted mid-task, so that other applications can continue to run even when something low-level and complicated is going on in the background. Of course, there are still times when the kernel cannot be interrupted in its processing. In reality, most users never saw these delays, which are rarely over small fractions of a second. Despite that, many users may notice an improvement in interactive performance with this feature enabled; things like user input will "feel" faster, even when the system is bogged down.

Linux's Input/Output (I/O) subsystems has also undergone major changes to allow them to be more responsive under all sorts of workloads. These changes include a complete rewrite of the I/O scheduler, the code of the kernel that determines what processes get to read from devices and when. The newly rewritten layer is now better capable of ensuring that no processes get stuck waiting in line for too long, while still allowing for the older optimizations which made sure that reading data still happens in the most efficient way for the underlying hardware.

On the application software side, another change that will help make Linux programs more responsive (if they use the feature) is support for new "futexes" (or "Fast User-Space Mutexes") Futexes are a way in which multiple processes or threads can serialize events so that they don't trample on each other (a "race condition"). Unlike the traditional mutex operations that most threading libraries support, this concept is partially kernel based (but only in the contention case) and it also supports setting priorities to allow applications or threads of higher priority access to the contested resource first. By allowing a program to prioritize waiting tasks, applications can be made to be more responsive in timing-critical areas.

In addition to all of the above, there have been a number of other smaller changes which will improve interactivity and performance in many cases. These include more removals of the "Big Kernel Lock" (non-fine-grained locks which were used in the early days' of Linux's support for multiple processors), optimizations of filesystem readahead, writeback, and manipulating small files, and other similar changes.

Other Improvements

Linux, like the Open Source movement in general, has always been a flag-bearer for the benefits of open

standards. Another major change in the 2.6 release, is that the kernel's internal threading infrastructure has been rewritten to allow the Native POSIX Thread Library (NPTL) to run on top of it. This can be a major performance boost for Pentium Pro and better processors in heavily threaded applications, and many of the top players in the "enterprise" space have been clamoring for it. (In fact, RedHat has already backported the support to Linux 2.4 and includes it starting with RedHat 9 and Advanced Server 3.0) This change includes new concepts to the Linux thread space including thread groups, local memory for individual threads, POSIX-style signals, and other changes. One of the major drawbacks is that applications (such as some versions of Sun Java) not written to spec that rely on old Linux-isms will break with the new support enabled. As the benefits overwhelm the cost (and with so many large players in the game), it's clear that most important applications will support the changes before too long after the new kernel is released.

Module Subsystem and the Unified Device Model

Increasingly in modern operating systems, the device handling subsystems have taken on new prominence as they are forced to deal with a myriad of internal and external bus types and more devices by more vendors than you can shake a stick at. It should come as no surprise then, that the upcoming upgrade to the Linux kernel will include improved support both in its module loader, but also in its internal understanding of the hardware itself. These changes range from the purely cosmetic (driver modules now use a ".ko" extension, for "kernel object", instead of just ".o") to a complete overhaul of the unified device model. Throughout all of these changes is an emphasis on stability and better grasp of the limitations of the previous revision.

Strictly in the module (driver) subsystem, there are a handful of major changes to improve stability. The process for unloading modules have been changed somewhat to reduce cases where it is possible for modules to be used while they are still being unloaded, often causing a crash. For systems where this problem cannot be risked, it is now even possible to disable unloading of modules altogether. Additionally, there has been extensive effort to standardize the process by which modules determine and announce what hardware they support. Under previous versions of Linux, the module would "know" what devices it supported, but this information was not generally available outside of the module itself. This change will allow hardware management software, such as RedHat's "kudzu", to make intelligent choices even on hardware that would not otherwise recognize. Of course, in the event that you know better than the current version of the driver what it supports, it is still possible to force a driver to try to work on a specific device.

Outside of just module loading, the device model itself has undergone significant changes in the updated kernel release. Unlike the module loader, which just has to concern itself with detecting the resource requirements of incoming hardware, the device model is a deeper concept which must be completely responsible for all of the hardware in the system. Linux versions 2.2 and earlier had only the barest support for a unified device model, preferring instead to leave almost all knowledge of the hardware solely at the module level. This worked fine, but in order to use all of the features of modern hardware (especially ACPI), a system needs to know more than just what resources a device uses: it needs to know things like what bus it is connected to, what subdevices it has, what its power state is, whether it can be reconfigured to use other resources in the event of contention, and even to know about devices that haven't had modules loaded for them yet. Linux 2.4 expanded on this foundation to become the first edition to unify the interfaces for PCI, PC Card, and ISA Plug-and-Play busses into a single device structure with a common interface. Linux 2.6, through its new kernel object ("kobject") subsystem, takes this support to a new level by not only expanding to know about all devices in a system, but also to provide a centralized interface for the important little details like reference counting, power management, and exports to user-space.

Now that an extensive amount of hardware information is available within the kernel, this has allowed Linux to better support modern laptop and desktop features that require a much more in-depth knowledge of hardware. The most readily apparent application is this is the increasing proliferation of so called "hot plug" devices like PC Cards, USB and Firewire devices, and hot-plug PCI. While it's hard to think back that far now, Linux didn't offer true support for any of these devices until the 2.2 kernel. Given that hot-plugging is the rule these days and not the exception, it is fitting that the new device infrastructure essentially eliminates the differences between a hot-plug and a legacy device. Since the kernel subsystem does not directly differentiate between a device discovered at boot time from one discovered later, much of the infrastructure for dealing with pluggable devices has been simplified. A second up and coming driver of this newly rewritten subsystem is for improved support of modern power management. The new power management standard in recent years, called ACPI for "Advanced Configuration and Power Interface", was first supported in rough form for the previous version of the kernel. Unlike old-fashioned APM ("Advanced Power

Management"), OSes run on systems with this new interface are required to individually tell all compatible devices that they need to change their power states. Without a centralized understanding of hardware, it would be impossible for the kernel to know what devices it needs to coordinate with and in what order. Although these are just two obvious examples, there are clearly other areas (such as hardware auditing and monitoring) that will benefit from a centralized picture of the world.

The final, but possibly the most obvious, ramification of the new centralized infrastructure is the creation of a new "system" filesystem (to join 'proc' for processes, 'devfs' for devices, and 'devpts' for UNIX98 pseudo-terminals) called 'sysfs'. This filesystem (intended to be mounted on '/sys') is a visible representation of the device tree as the kernel sees it (with some exceptions). This representation generally includes a number of known attributes of the detected devices, including the name of the device, its IRQ and DMA resources, power status, and that sort of thing. However, one aspect of this change that may be confusing on the short term is that many of the device-specific uses of the "/proc/sys" directory may be moved into this new filesystem. This change has not (yet) been applied consistently, so there may continue to be an adjustment period.

System Hardware Support

As Linux has moved forward over the years and into the mainstream, each new iteration of the kernel appeared to be leaps and bounds better than the previous in terms of what types of devices it could support--both in terms of emerging technologies (USB in 2.4) and older "legacy" technologies (MCA in 2.2). As we arrive at the 2.6 however, the number of major devices that Linux does not support is relatively small. There are few, if any, major branches of the PC hardware universe yet to conquer. It is for that reason that most (but certainly not all) of improvements in i386 hardware support have been to add robustness rather than new features.

Internal Devices

Arguably as important as the processor type, the underling bus(es) in a system are the glue that holds things together. The PC world has been blessed with no shortness of these bus technologies, from the oldest ISA (found in the original IBM PC) to modern external serial and wireless busses. Linux has always been quick to adapt to a new bus and device type as they have become popular with consumer devices, but significantly less quick adapting to technologies that get relatively little use.

Improvements in Linux's support for internal devices are really spread across the board. One specific example where Linux is playing "catch up" is support for the old ISA ("Industry Standard Architecture") Plug-and-Play extensions. Linux didn't offer any built-in support for PnP at all until the 2.4 release. This support has been rounded out with the upcoming kernel to include full PnP BIOS support, a device name database, and other compatibility changes. The sum of all of those modifications, is that now Linux is now a "true" Plug-and-Play OS and may be set as such in a compatible machine's BIOS. Other legacy busses such as MCA ("Microchannel Architecture") and EISA ("Extended ISA") have also been wrapped into the new device model and feature device naming databases. On a more modern front Linux 2.6 brings to the table a number of incremental improvements to its PCI ("Peripheral Component Interconnect") subsystem including improved Hot-Plug PCI and power management, support for multiple AGPs ("accelerated graphics ports" -- a separate high-speed extension to the PCI bus), and other changes. And finally, in addition to all of the "real" busses, Linux 2.6 has internally added the concept of a "legacy" bus that is specific to each architecture and contains all of the assumed devices that you would expect to find. On a PC, for example, this may include on-board serial, parallel, and PS/2 ports-- devices that exist but are not enumerated by any real busses on the system. This support may require more complicated work (such as querying firmware) on some platforms, but in general this is just a wrapper to ensure that all devices are handled in a standard way in the new driver paradigm.

External Devices

While it is true that the older-style internal device busses have not seen many new features during the most recent development cycle, the same cannot be said for hot new external hardware. Possibly the most important change in this space is the new support for USB 2.0 devices. These devices, commonly referred to

as "high speed" USB devices, support device bandwidth of up to 480 megabits per second, compared to 12 mbit/sec of current USB. A related new standard, USB On-the-Go (or USB OTG), a point-to-point variant on the USB protocol for connecting devices directly together (for example, to connect a digital camera to a printer without having a PC in the middle) is not currently supported in Linux 2.6. (Patches for this feature are available, but not yet rolled into the official release.) In addition to device support, much of the way USB devices have been internally enumerated has been revised so that it is now possible to have many more devices of the same type all accessible from within Linux. In addition to the large changes, there has been an emphasis placed in this development cycle on simplification, stability, and compatibility that should improve the support of USB devices for all Linux users.

On the complete opposite end of the field, Linux 2.6 for the first time includes support that allows a Linux-powered machine to be a USB device, rather than a USB host. This would allow, for example, your Linux-powered PDA to be plugged into your PC and to have both ends of the line speaking the proper protocol. Much of this support is new, but this is an essential direction for Linux to move into for embedded devices.

Wireless Devices

Wireless technology has really taken off within the public in the past several years. It often seems as if cords (except power... maybe?) will be a thing of the past within a handful of years. Wireless devices encompass both networking devices (the most common currently) and also more generic devices such as PDAs, etc.

In the wireless networking space, devices can generally be divided into long range (for example, AX.25 over amateur radio devices) and short range (usually 802.11, but some older protocols exist.) Support for both of these has been a hallmark of Linux since the early days (v1.2) and both of these subsystems have been updated during development of 2.6. The largest change here is that major components of the short range subsystems for the various supported cards and protocols has been merged into a single "wireless" subsystem and API. This merge resolves a number of minor incompatibilities in the way different devices have been handled and strengthens Linux's support for the subsystem by making a central set of userspace tools that will work with all supported devices. In addition to just standardization, Linux 2.6 introduces a number of overall improvements including better capability to notify in the event of a state change (such as a device that has a "roaming" state) and a change to TCP to better handle periodic delay spikes which occur with wireless devices. Due to the immediate desire to better support wireless devices in the current Linux 2.4 kernel, many of these changes have already been back-ported and are available for use.

In the generic wireless devices space, there have been similar major advancements. IrDA (the infrared protocol named for the Infrared Data Associates group) has received some advancements since the last major release such as power management and integration into the new kernel driver model. The real advancements however have been made in providing Linux support for Bluetooth devices. Bluetooth is a new wireless protocol that is designed to be short range and low on power consumption, but does not have the line of sight limitations that IrDA has. Bluetooth as a protocol is designed to go "anywhere" and has been implemented in devices like PDAs, cell phones, printers, and more bizarre things such as automotive equipment. The protocol itself is made up of two different data link types: SCO, or "Synchronous Connection Oriented", for lossy audio applications; and L2CAP, or "Logical Link Control and Adaptation Protocol", for a more robust connection supporting retransmits, etc. The L2CAP protocol further supports various sub-protocols (including RFCOMM for point-to-point networking and BNEP for Ethernet-like networking.) Linux's support for the things that Bluetooth can do continues to grow and we can expect this to mature significantly once more devices are in the hands of the consumers. It should also be mentioned that initial support for Bluetooth has been integrated into later editions of the 2.4 kernel.

Block Device Support

Storage Busses

Dedicated storage busses, such as IDE/ATA ("Integrated Drive Electronics/Advanced Technology Attachment") and SCSI ("Small Computer System Interface"), have also received a major update during the 2.6 cycle. The most major changes are centered around the IDE subsystem which has been rewritten (and rewritten again) during the development of the new kernel, resolving many scalability problems and other

limitations. For example, IDE CD/RW drives can now be written to directly through the real IDE disk driver, a much cleaner implementation than before. (Previously, it was required to also use a special SCSI-emulating driver which was confusing and often difficult.) In addition, new support has been added for high-speed Serial ATA (S-ATA) devices, which have transfer rates exceeding 150 MB/sec. On the SCSI side, there have also been many small improvements scattered around the system both for wider support and scalability. One specific improvement for older systems is that Linux now supports SCSI-2 multipath devices that have more than 2 LUNs on a device. (SCSI-2 is the previous version of the SCSI device standard, circa 1994.) Another important change is that Linux can now fall back to test media changing like Microsoft Windows does, to be more compatible with devices that do not completely follow the specification. As these technologies have stabilized over time, so too has Linux's support for them.

Although not a storage bus in itself, Linux now includes support for accessing a newer machine's EDD ("Enhanced Disk Device") BIOS directly to see how the server views its own disk devices. The EDD BIOS includes information on all of the storage busses which are attached to the system that the BIOS knows about (including both IDE and SCSI.) In addition to just getting configuration and other information out of the attached devices, this provides several other advantages. For example, this new interface allows Linux to know what disk device the system was booted from, which is useful on newer systems where it is often not obvious. This allows intelligent installation programs to consider that information when trying to determine where to put the Linux boot loader, for example.

In addition to all of these changes, it should be stressed again that all of the bus device types (hardware, wireless, and storage) have been integrated into Linux's new device model subsystem. In some cases, these changes are purely cosmetic. In other cases, there are more significant changes involved (in some cases for example, even logic for how devices are detected needed to be modified.)

Filesystems

The most obvious use of a block device on a Linux (or any other) system is by creating a filesystem on it, and Linux's support for filesystems have been vastly improved since Linux 2.4 in a number of respects. Key among these changes include support for extended attributes and POSIX-style access controls.

When dealing strictly with conventional Linux filesystems, the extended filesystems (either "ext2" or "ext3") are the systems most associated with a core Linux system. (ReiserFS is the third most common option.) As these are the filesystems that users care about the most, they have also been the most improved during the development of Linux 2.6. Principal among these changes is support for "extended attributes", or metadata that can be embedded inside the filesystem itself for a specific file. Some of these extended attributes will be used by the system and readable and writable by root only. Many other operating systems, such as Windows and the MacOS, already make heavy use of these kinds of attributes. Unfortunately, the UNIX legacy of operating systems have not generally included good support for these attributes and many user-space utilities (such as 'tar') will need to be updated before they will save and restore this additional information. The first real use of the new extended attribute subsystem is to implement POSIX access control lists, a superset of standard UNIX permissions that allows for more fine-grained control. In addition to these changes for ext3, there are several other smaller changes: the journal commit time for the filesystem can now be tuned to be more suited for laptop users (which might have to spin up the drive if it were in a power save mode.), default mount options can now also be stored within the filesystem itself (so that you don't need to pass them at mount time), and you can now mark a directory as "indexed" to speed up searches of files in the directory.

In addition to the classic Linux filesystems, the new kernel offers full support for the new (on Linux) XFS filesystem. This filesystem is derived from and is block-level compatible with the XFS filesystem used by default on Irix systems. Like the extended filesystems and Reiser, it can be used as a root-disk filesystem and even supports the newer features such as extended attributes and ACLs. Many distributions are beginning to offer support for this filesystem on their Linux 2.4-based distributions, but it remains to be seen yet what place this filesystem will have in the already crowded pantheon of UNIX-style filesystems under Linux.

Outside of those, Linux has also made a number of improvements both inside and outside the filesystem layer to improve compatibility with the dominant PC operating systems. To begin with, Linux 2.6 now supports Windows' Logical Disk Manager (aka "Dynamic Disks"). This is the new partition table scheme that Windows 2000 and later have adopted to allow for easier resizing and creation of multiple partitions. (Of course, it is not likely that Linux systems will be using this scheme for new installations anytime soon.)

Linux 2.6 also features improved (and rewritten) support for the NTFS filesystem and it is now possible to mount a NTFS volume read/write. (Writing support is still experimental and is gradually being improved; it may or may not be enabled for the final kernel release.) And finally, Linux's support for FAT12 (the DOS filesystem used on really old systems and floppy disks) has been improved to work around bugs present in some MP3 players which use that format. Although not as dominant in the marketplace, Linux has also improved compatibility with OS/2 by adding extended attribute support into the HPFS filesystem. Like previous releases, the new additions to Linux 2.6 demonstrate the importance of playing well with others and reinforces Linux's position as a "Swiss Army Knife" operating system.

In addition to these changes, there have been a large number of more scattered changes in Linux's filesystem support. Quota support has been rewritten to allow for the larger number of users supported on a system. Individual directories can now be marked as synchronous so that all changes (additional files, etc.) will be atomic. (This is most useful for mail systems and directory-based databases, in addition to slightly better recovery in the event of a disk failure.) Transparent compression (a Linux-only extension) has been added to the ISO9660 filesystem (the filesystem used on CD-ROMs.) And finally, a new memory-based filesystem ("hugetlbfs") has been created exclusively to better support shared memory databases.

Input / Output Support

On the more "external" side of any computer system is the input and output devices, the bits that never quite seem as important as they are. These include the obvious things like mice and keyboards, sound and video cards, and less obvious things like joysticks and accessibility devices. Many of Linux's end-user subsystems have been expanded during the 2.6 development cycle, but support for most of the common devices were already pretty mature. Largely, Linux 2.6's improved support for these devices are derived directly from the more general improvments with external bus support, such as the ability to use Bluetooth wireless keyboards and similar. There are however a number of areas where Linux has made larger improvements.

Human Interface Devices

One major internal change in Linux 2.6 is the reworking of much of the human interface layer. The human interface layer is the center of the user experience of a Linux system, including the video output, mice, and keyboards. In the new version of the kernel, this layer has been reworked and modularized to a much greater extent than ever before. It is now possible to create a completely "headless" Linux system without any included support for a display or anything. The primary benefit of this modularity may be for embedded developers making devices that can only be administrated over the network or serial, but end-users benefit as many of the underlying assumptions about devices and architectures has been modularized out. For example, it was previously always assumed that if you had a PC that you would need support for a standard AT (i8042) keyboard controller; the new version of Linux removes this requirement so that unnecessary code can be kept out of legacy-less systems.

Support to Linux's handling of monitor output has also received a number of changes, although most of these are useful only in configurations that make use of the kernel's internal framebuffer console subsystem. (Most Intel Linux boxes are not configured this way, but that is not the case for many other architectures.) In my personal opinion, the best feature is that the boot logo (a cute penguin, if you've never seen it) now supports resolutions up to 24bpp. That aside, other new features for the console include resizing and rotating (for PDAs and similar) and expanded acceleration support for more hardware. And finally, Linux has now included kernel support for querying VESA ("Video Electronics Standard Association") monitors for capability information, although XFree86 and most distributions installation systems already have covered this detail in user-space.

In addition to the big changes, Linux 2.6 also includes a number of smaller changes for human interaction. Touch screens, for example, are now supported. The mouse and keyboard drivers have also been updated and standardized to only export a single device node ("/dev/input/mouse0", for example) regardless of the underlying hardware or protocol. Bizarre mice (with multiple scroll wheels, for example) are now also supported. PC keyboard key mappings have also been updated to follow the Windows "standard" for extended keys. Joystick support has also been improved thanks not only to the addition of many new drivers (including the X Box gamepad), but also to include newer features such as force-feedback. And finally (but not least important), the new release also includes support for the Tieman Voyager braille TTY device to

allow blind users better access to Linux. (This feature is important enough that it has been back-ported to Linux 2.4 already.)

As a side note, Linux has also changed the "system request" interface to better support systems without a local keyboard. The system request ("sysrq") interface is a method for systems administrators at the local console to get debugging information, force a system reboot, remount filesystems read-only, and do other wizardly things. Since Linux 2.6 now supports a completely headless system, it is now also possible to trigger these events using the /proc filesystem. (Of course, if your system hangs and you need to force it to do things, this may not be of much help to you.)

Audio & Multimedia

One of the most anticipated new features of Linux 2.6 for desktop users is the inclusion of ALSA (the "Advanced Linux Sound Architecture") in lieu of the older sound system. The older system, known as OSS for "Open Sound System", has served Linux since the early days but had many architectural limitations. The first major improvement with the new system is that it has been designed from the start to be completely thread and SMP-safe, fixing problems with many of the old drivers where they would not work properly outside the expected "desktop-means-single-cpu paradigm." More importantly, the drivers have been designed to be modular from the start (users of older versions of Linux will remember that modularity was retro-fitted onto the sound system around Linux 2.2), and that this allows for improved support for systems with multiple sound cards, including multiple types of sound cards. Regardless of how pretty the internals are, the system would not be an improvement for users if it did not have neat new whiz-bang features, and the new sound system has many of those. Key among them are support for newer hardware (including USB audio and MIDI devices), full-duplex playback and recording, hardware and non-interleaved mixing, support for "merging" sound devices, and other things. Whether you are an audiophile or just someone that likes to play MP3s, Linux's improved sound support should be a welcome step forward.

Beyond simple audio these days, what users want is support for the really fancy hardware like webcams, radio and TV adapters, and digital video recorders. In all three cases, Linux's support has been improved with the 2.6 release. While Linux has supported (to a greater or lesser extent) radio cards (often through userspace) for many iterations, support for television tuners and video cameras was only added within the last one or two major revisions. That subsystem, known as Video4Linux (V4L), has received a major upgrade during the work on the new edition of the kernel including both an API cleanup and support for more functionality on the cards. The new API is not compatible with the previous one and applications supporting it will need to upgrade with the kernel. And on a completely new track, Linux 2.6 includes the first built-in support for Digital Video Broadcasting (DVB) hardware. This type of hardware, common in set-top boxes, can be used to make a Linux server into a Tivo-like device, with the appropriate software.

Software Improvements

Networking

Leading-edge networking infrastructure has always been one of Linux's prime assets. Linux as an OS already supports most of the world's dominant network protocols including TCP/IP (v4 and v6), AppleTalk, IPX, and others. (The only unsupported one that comes to mind is IBM/Microsoft's obsolete and tangled NetBEUI protocol.) Like many of the changes in the other subsystems, most networking hardware changes with Linux 2.6 are under the hood and not immediately obvious. This includes low-level changes to take advantage of the device model and updates to many of the device drivers. For example, Linux now includes a separate MII (Media Independent Interface, or IEEE 802.3u) subsystem which is used by a number of the network device drivers. This new subsystem replaces many instances where each driver was handling that device's MII support in slightly different ways and with duplicated code and effort. Other changes include major ISDN updates and other things.

On the software side, one of the most major changes is Linux's new support for the IPsec protocols. IPsec, or IP Security, is a collection of protocols for IPv4 ("normal" IP) and IPv6 that allow for cryptographic security at the network protocol level. And since the security is at the protocol level, applications do not have to be explicitly aware of it. This is similar to SSL and other tunneling/security protocols, but at a much lower level. Currently supported in-kernel encryption includes various flavors of SHA ("secure hash algorithm"), DES

("data encryption standard"), and others.

Elsewhere on the protocol side, Linux has improved its support for multicast networking. Multicast networks are networks where a single sent packet is intended to be received by multiple computers. (Compare to traditional point-to-point networks where you are only speaking to one at a time.) Primarily, this functionality is used by messaging systems (such as Tibco) and audio/video conferencing software. Linux 2.6 improves on this by now supporting several new SSM (Source Specific Multicast) protocols, including MLDv2 (Multicast Listener Discovery) and IGMPv3 (Internet Group Messaging Protocol.) These are standard protocols that are supported by most high-end networking hardware vendors, such as Cisco.

Linux 2.6 also has broken out a separate LLC stack. LLC, or Logical Link Control protocol (IEEE 802.2), is a low-level protocol that is used beneath several common higher-level network protocols such as Microsoft's NetBeui, IPX, and AppleTalk. As part of this change-over, the IPX, AppleTalk, and Token Ring drivers have been rewritten to take advantage of the new common subsystem. In addition, an outside source has put together a working NetBEUI stack and it remains to be seen whether it will ever be integrated into the stock kernel.

In addition to these changes, there have been a number of smaller changes. IPv6 has received some major changes and it can now also run on Token Ring networks. Linux's NAT/masquerading support has been extended to better handle protocols that require multiple connections (H.323, PPTP, etc.) On the Linux-as-a-router front, support for configuring VLANs on Linux has been made no longer "experimental".

Network Filesystems

Overlaid on top of Linux's robust support for network protocols is Linux's equally robust support for network filesystems. Mounting (and sometimes exporting) a network filesystem is one of the very few high-level network operations that the kernel cares about directly. (The most obvious other, the "network block device", did not receive many changes for 2.6 and is generally used in specialized applications where you end up doing something filesystem-like with it anyway.) All other network operations are content to be relegated to user-space and outside the domain of the kernel developers.

In the Linux and UNIX-clone world, the most common of the network filesystems is the aptly named Network File System, or NFS. NFS is a complicated file sharing protocol that has deep roots in UNIX (and especially Sun Solaris' excellent implementation). The primary transport protocol can utilize either TCP or UDP, but several additional sub-protocols are also required, each of which also run on top of the separate RPC ("remote procedure call") protocol. These include the separate "mount" protocol for authentication and NLM ("network lock manager") for file locking. (The common implementation is also tied closely to other common RPC-based protocols, including NIS-- "network information service"-- for authentication. NIS and its progeny are not commonly used for authentication on Linux machines due to fundamental insecurities.) It is perhaps because of this complexity that NFS has not been widely adapted as an "Internet" protocol.

In Linux 2.6, this core Linux filesystem received many updated and improvements. The largest of these improvements is that Linux now experimentally supports the new and not widely adopted NFSv4 protocol version for both its client and server implementations. (Previous versions of Linux included support for only the v2 and v3 versions of the protocol.) The new version supports stronger and more secure authentication (with cryptography), more intelligent locking, support for pseudo-filesystems, and other changes. Not all of the new NFSv4 features have been implemented in Linux yet, but the support is relatively stable and could be used for some production applications. In addition, Linux's NFS server implementation has been improved to be more scalable (up to 64 times as many concurrent users and a larger request queues), to be more complete (by supporting serving over TCP, in addition to UDP), to be more robust (individual filesystems drivers can adapt the way files on those systems are exported to suit their particularities), and more easily maintainable (management though a new 'nfsd' filesystem, instead of system calls.) There have also been may other under the hood changes, including separating lockd and nfsd, and support for zero-copy networking on supported interfaces. NFS has also been made somewhat easier to secure by allowing the kernel lockd port numbers to be assigned by the administrator. The NFS client side has also benefited from a number of improvements to the implementation of the underlying RPC protocol including a caching infrastructure, connection control over UDP, and other improvements for TCP. Linux's support for using NFS-shared volumes as the root filesystem (for disk-less systems) has also been improved as the kernel now supports NFS over TCP for that purpose.

In addition to improving support for the UNIX-style network filesystems, Linux 2.6 also delivers many improvements to Windows-style network filesystems. The standard shared filesystem for Windows servers (as well as OS/2 and other operating systems) has been the SMB ("server message block") protocol and the Linux kernel has had excellent client support of the SMB protocol for many revisions. Windows 2000 however standardized on an upgraded superset of the SMB protocol, known as CIFS ("common internet filesystem.") The intention of this major update was to streamline and refine certain aspects of SMB which had at that point become a complete mess. (The protocol itself was loosely defined and often extended to the point that there were cases even where the Win95/98/ME version was incompatible with the WinNT/Win2k version.) CIFS delivered on that intention and added UNICODE support, improved file locking, hard linking, eliminated the last vestiges of NetBIOS dependencies, and added a few other features for Windows users. Since Linux users do not like to be kept in the dark for long, Linux 2.6 now includes completely rewritten support for mounting CIFS filesystems natively. Linux 2.6 also now includes support for the SMB-UNIX extensions to the SMB and CIFS protocols which allows Linux to access non-Windows file types (such as device nodes and symbolic links) on SMB servers which support it (such as Samba.) Although not as commonly seen today, Linux has not completely forgotten about the Novell NetWare users. Linux 2.6 now allows Linux clients to mount up to the maximum of 256 shares on a single NetWare volume using its built in NCP ("NetWare Core Protocol") filesystem driver.

Linux 2.6 also includes improved support for the relatively new domain of distributed network filesystems, systems where files on a single logical volume can be scattered across multiple nodes. In addition to the CODA filesystem introduced in Linux 2.4, Linux now includes some support for two other distributed filesystems: AFS and InterMezzo. AFS, the Andrew filesystem (so named because it was originally developed at CMU), is presently very limited and restricted to read-only operations. (A more feature complete version of AFS is available outside the kernel-proper.) The second newly supported filesystem, InterMezzo (also developed at CMU), is also newly supported under Linux 2.6 and it allows for more advanced features such as disconnect operation (so you work on locally cached files) and is suitable for high-availability applications where you need to guarantee that storage is never unavailable (or faked, when down). It also has applications for keeping data in sync between multiple computers, such as a laptop or PDA and a desktop computer. Many of the projects providing support for these new types of filesystems are initially developed on Linux, putting Linux well ahead of the curve in support for these new features.

Miscellaneous Features

Security

Another of the big changes in Linux 2.6 that does not receive enough attention is the wealth of new security-related changes. Most fundamentally, the entirety of kernel-based security (powers of the super user under a UNIX-like operating system) has been modularized out to be one out of a potential number of alternate security modules. (At present however, the only offered security model is the default one and an example how to make your own.) As part of this change, all parts of the kernel have now been updated to use "capabilities" as the basis of fine-grained user access, rather than the old "superuser" system. Nearly all Linux systems will continue to have a "root" account which has complete access, but this allows for a Linux-like system to be created which does not have this underlying assumption. Another security-related change is that binary modules (for example, drivers shipped by a hardware manufacturer) can no longer "overload" system calls with their own and can no longer see and modify the system call table. This significantly restricts the amount of access that non-open source modules can do in the kernel and possibly closes some legal loopholes around the GPL. The final change that is somewhat security-related is that Linux with the new kernel is now able to use hardware random number generators (such as those present in some new processors), rather than relying on a (admittedly quite good) entropy pool based on random hardware fluctuations.

Virtualizing Linux

One of the most interesting new features in Linux 2.6 is its inclusion of a "user-mode" architecture. This is essentially a port (like to a different hardware family) of Linux to itself, allowing for a completely virtualized Linux-on-Linux environment to be run. The new instance of Linux runs as if it was a normal application. "Inside" the application, you can configure fake network interfaces, filesystems, and other devices through special drivers which communicate up to the host copy of Linux in a secure way. This has proved quite useful, both for development purposes (profiling, etc.) as well as for security analysis and honeypots. While

most users will never need this kind of support, it is an incredibly "cool" feature to have running on your box. (Impress your friends!)

Laptops

In addition to all of the other general purpose support described above (improved APM and ACPI, wireless support improvements, etc.) Linux also includes two other hard-to-classify features that will best assist laptop users. The first is that the new edition of the kernel now supports software-suspend-to-disk functionality for the Linux user on the go. This feature still has some bugs to iron out, but is looking solid for many configurations. The new version also supports the ability of modern mobile processors to change speed (and, in effect, power requirements) based on whether your system is plugged in or not.

Configuration Management

Linux 2.6 includes another feature which might seem minor to some, but will both greatly assist developers' abilities to debug kernel problems of end-users as well as make it easier for individual administators to know configuration details about multiple systems. In short, the kernel now supports adding full configuration information into the kernel file itself. This information would include details such as what configuration options were selected, what compiler was used, and other details which would help someone reproduce a similar kernel if the need arose. This information would also be exposed to users via the /proc interface.

Legacy Support

Although Linux 2.6 is a major upgrade, the difference to user-mode applications will be nearly non-existent. The one major exception to this rule appears to be threading: some applications may do things that worked under 2.4 or 2.2 but are no longer allowed. Those applications should be the exception to the rule however. Of course, low-level applications such as module utilities will definitely not work. Additionally, some of the files and formats in the /proc and /dev directories have changed and any applications that have dependencies on this may not function correctly. (This is especially true as more things shift over to the new "/sys" virtual filesystem. In the "/dev" case, backwards-compatible device names can easily be configured.)

In addition to those standard disclaimers, there are a number of other smaller changes which may affect some environments. First, very old swap files (from Linux 2.0 or earlier) will need to be reformatted before they can be used with 2.6. (Since swap files do not contain any permanent data, that should not be a problem for any user.) The kHTTPd daemon which allowed the kernel to serve web pages directly has also been removed as most of the performance bottlenecks that prevented Apache, Zeus, et. al. from reaching kernel speeds have been resolved. Autodetection of DOS/Windows "disk managers" such as OnTrack and EzDrive for large harddisk support with older BIOSes has been removed. And finally, support for using a special kernel-included boot sector for booting off of a floppy disk has also been removed; you now need to use SysLinux instead.

Stuff At The Bottom

This document was assembled primarily from long hours looking at BitKeeper changelogs, looking at and playing with the source, reading mailing list posts, and lots and lots of Google and Lycos searches for documentation about this and that. As such, there are likely places where something could have been missed or misunderstood, and places where something could have been backed out after the fact. (I have been especially careful of the two versions of IDE support that were worked on during this time period, but there are other examples.) As a bit of my research was done by looking at the web pages of various kernel projects, I have had to be careful that the independent projects weren't farther ahead with features than were accepted into the mainline Linux code. If you see any errors in this document or want to email me to ask me how my day is going, you can do so at jpranevich <at> kniggit.net.

The newest version of this document can always be found at http://kniggit.net/wwol26.html.

Translations

Not an English speaker? This document (or an older revision) has been translated into a handful of other languages.

Bulgarian - http://kniggit.net/wwol26bg.html (Ivan Dimov)
Chinese - http://www-900.ibm.com/developerWorks/cn/linux/kernel/l-kernel26/index.shtml (Stone Wang, et. al.)
Czech - http://www.linuxzone.cz/index.phtml?ids=10&idc=782 (David Haring)
French - http://dsoulayrol.free.fr/articles/wonderful_2.6.html (David Soulayrol)
Hungarian - http://free.srv.hu/b/e/behun/pn/modules.php?op=modload&name=News&file=index&catid=&topic=12 (Ervin Novak) *(Not yet completed.)*
Italian - http://www.opensp.org/tutorial/vedi.php?appartenenza=42&pagine=1 (Giulio Ciuffi Vampa)
Portuguese (BR) - http://geocities.yahoo.com.br/cesarakg/wwol26-ptBR.html (Cesar A. K. Grossmann)
Russian - http://www.opennet.ru/base/sys/linux26_intro.txt.html (Sergey Prokopenko)
Spanish - http://www.escomposlinux.org/wwol26/wwol26.html (Alex Fernández)

An abridged version also appeared in German in the 09/2003 issue of LanLine magazine. I believe that an unabridged edition may be floating around also, but I am uncertain of the link. If you know of any additional translations to add to this list, please let me know.

This document is Copyright 2003 by Joseph Pranevich. Redistribution online without modification is permitted. Offline distribution (in whole or in part) is also encouraged, but please email me first to discuss the details. Translations are also encouraged; please email me though so that I can help coordinate.