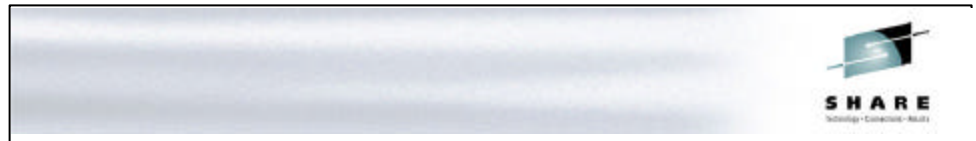


Linux 2.4

Changes and Enhancements

Acknowledgements

- Larry Woodman - Technical Director/Kernel Engineering, Mission Critical Linux
- Joe Pranevich
- Thomas Wolfgang Burger

Background

- Linux 0.95 in 1991
  - First release to public via Internet
- Linux 2.0
  - Starting to get public notice
- Linux 2.2
  - New file systems
  - Redesign of caching
  - Greater scalability
- Linux 2.4

Linux 2.2 was a great improvement over Linux 2.0. It supported many new file systems, it supported a completely rethought caching system for file names, and it was much more scalable than Linux 2.0. Linux 2.4 will build on the great advancements provided under Linux 2.2 to become an even better platform for desktop, server, and embedded tasks. However, it is the intent of the Linux kernel developers to get Linux 2.4 into the hands of the end users more quickly. To meet this goal, Linux 2.4 will understandably not be as different from Linux 2.2 as Linux 2.2 was from Linux 2.0. I think you will agree however that the advancements in Linux 2.4 will be just as noteworthy as previous versions. (Or else I wouldn't need to be writing this!)

What, at the core, is the Linux kernel? Just as the kernel is the heart of the Linux (or GNU/Linux or whatever) Operating System, the kernel itself can be divided into core and non-core parts. Linux is much more than just a collection of assorted device drivers, as any operating system must be. It's what binds these drivers together into a cohesive unit that matters. It's the scheduler, the resource allocator, the virtual filesystem layer, memory management, and so many other unsung features that are the real heroes of the Linux world. These are the portions of the Linux operating system that really define what is Linux because on every platform that Linux has been ported to from i386 (Intel-compatible PC), to ARM (embedded devices), to Sparc64 (high-end servers) this code is the same. In many ways, this "heart" of Linux 2.4 is different than Linux 2.2's and most of the subsystems that I just listed have been changed in one way or another.

Linux 2.2 and earlier Linux's included a base resource management system which was used rather bluntly to allocate and keep track of IO ports and IRQ lines and the other limited niceties of computer architectures. Unfortunately it was deficient in a number of important ways which proved crucial to the needs of a modern desktop operating system. The new system under Linux 2.4 includes a much more generic implementation which allows for nested resource groups, removed the dependencies on pre-defined resource types, and otherwise made it easier to use for a majority of the tasks required by driver developers. Additionally, this has laid the groundwork for ISA PnP support which is discussed more fully later in this article. This quick hack by Linus will probably be one of the most influential changes to go into the 2.4 kernel.

Overview

- Linux kernel architecture features
- Linux kernel hardware support features
- File System Enhancements
- Networking Enhancements
- Device Support Enhancements

Architecture Features

- ELF and POSIX Foundation
  - More dependent on ELF
  - More POSIX compliant:
    - Clocks and Timers support

Linux ELF and POSIX foundation strengthened
Linux kernel 2.4 is more dependent on the ELF (Executable and Linking Format) than Linux 2.2. ELF is an advanced binary format that includes support for multiple code and data sections and increases the support for shared libraries. ELF, most notably used in the Solaris OS, is like the Win32 format, but better designed. Fully exploiting the ELF binary format will allow Linux kernel developers to make some pieces of code more modular and easy to maintain. ELF will allow drivers to be initialized based on how they are linked rather than by having an explicit initialization line in the core code. With the adoption of support for POSIX clocks and timers, Linux 2.4 becomes more POSIX compliant, allowing for non-rtc devices to be used as clocks internally.

Architecture Features

- Memory Usage
  - About the same as 2.2
- Shared Memory
  - More compliant with Industry standards
  - Introduces a special “shared memory” filesystem

Linux 2.4 will use about the same amount of memory as Linux 2.2. Exact numbers will change with each configuration. Some key subsystems, such as the file-caching layer, will use less memory to do the same work, so some memory decreases may be seen.
The way Linux handles shared memory has been changed. Linux 2.4 will be more compliant with industry standards. The changes require a special "shared memory" filesystem to be mounted in order for shared memory segments to work. Distribution vendors will handle this when they are ready for Linux 2.4.

Architecture features

- 2.2 Page Replacement Problems
  - Page eviction
  - Simplistic NRU replacement
  - Clock algorithm can evict accessed pages
  - Sub-optimal reaction to variable load or load spikes after inactivity

•Simple NRU replacement cannot accurately identify the working set versus incidentally accessed pages and can lead to extra page faults. This doesn't hurt noticeably for most workloads, but it makes a big difference in some workloads and can be fixed easily, mostly since the LFU replacement used in older Linux kernels is known to work.
•Due to the simple clock algorithm in shrink_mmap, sometimes clean, accessed pages can get evicted before dirty, old pages. With a relatively small file cache that mostly consists of dirty data, eg unpacking a tarball, it is possible for the dirty pages to evict the (clean) metadata buffers that are needed to write the dirty data to disk. A few other corner cases with amusing variations on this theme are bound to exist.
•The system reacts badly to variable VM load or to load spikes after a period of no VM activity. Since kswapd, the pageout daemon, only scans when the system is low on memory, the system can end up in a state where some pages have referenced bits from the last 5 seconds, while other pages have referenced bits from 20 minutes ago. This means that on a load spike the system has no clue which are the right pages to evict from memory, this can lead to a swapping storm, where the wrong pages are evicted and almost immediately afterwards faulted back in, leading to the pageout of another random page, etc...
•Under very heavy loads, NRU replacement of pages simply doesn't cut it. More careful and better balanced pageout eviction and flushing is called for. With the fragility of the Linux 2.2 pageout framework this goal doesn't really seem achievable.
The facts that shrink_mmap is a simple clock algorithm and relies on other functions to make process-mapped pages freeable makes it fairly unpredictable. Add to that the balancing loop in try_to_free_pages and you get a VM subsystem which is extremely sensitive to minute changes in the code and a fragile beast at its best when it comes to maintenance or (shudder) tweaking.
•Balancing between evicting pages from the file cache, evicting unused process pages and evicting pages fromshm segments. If memory pressure is "just right" shrink_mmap is always successful in freeing cache pages and a process which has been idle for a day is still in memory. This can even happen on a system with a fairly busy filesystem cache, but only with the right phase of moon.

Architecture features

- 2.4 Improvements:
  - Finer-grained SMP locking
  - Unification of buffer and page caches
  - Support for larger memory configurations
  - SYSV shared memory code replaced
  - Page aging reintroduced
  - Active & inactive page lists
  - Optimized page flushing
  - Controlled background page aging
  - Aggressive read-ahead

For Linux 2.4 a substantial development effort has gone into things like making the VM subsystem fully fine-grained for SMP systems and supporting machines with more than 1GB of RAM. Changes to the pageout code were done only in the last phase of development and are, because of that, somewhat conservative in nature and only employ known-good methods to deal with the problems that happened in the page replacement of the Linux 2.2 kernel.

•More fine-grained SMP locking. The scalability of the VM subsystem has improved a lot for workloads where multiple CPUs are reading or writing the same file simultaneously; for example web or ftp server workloads. This has no real influence on the page replacement code.

•Unification of the buffer cache and the page cache. While in Linux 2.2 the page cache used the buffer cache to write back its data, needing an extra copy of the data and doubling memory requirements for some write loads, in Linux 2.4 dirty page cache pages are simply added in both the buffer and the page cache. The system does disk IO directly to and from the page cache page. That the buffer cache is still maintained separately for filesystem metadata and the caching of raw block devices. Note that the cache was already unified for reads in Linux 2.2, Linux 2.4 just completes the unification.

•Support for systems with up to 64GB of RAM (on x86). The Linux kernel previously had all physical memory directly mapped in the kernel's virtual address space, which limited the amount of supported memory to slightly under 1GB. For Linux 2.4 the kernel also supports additional memory (so called "high memory" or highmem), which can not be used for kernel data structures but only for page cache and user process memory. To do IO on these pages they are temporarily mapped into kernel virtual memory and the data is copied to or from a bounce buffer in "low memory". At the same time the memory zone for ISA DMA (0 - 16 MB physical address range) has also been split out into a separate page zone. This means larger x86 systems end up with 3 memory zones, which all need their free memory balanced so we can continue allocating kernel data structures and ISA DMA buffers. The memory zones logic is generalized enough to also work for NUMA systems.

•The SYSV shared memory code has been removed and replaced with a simple memory filesystem which uses the page cache for all its functions. It supports both POSIX SHM and SYSV SHM semantics and can also be used as a swappable memory filesystem (tmpfs).

Here is a short overview of the page replacement changes: they'll be described in more detail below.

•Page aging, which was present in the Linux 1.2 and 2.0 kernels and in FreeBSD has been reintroduced into the VM. However, a few small changes have been made to avoid some artifacts of virtual page based aging.

•To avoid the eviction of "wrong" pages due to interactions from page aging and page flushing, the page aging and flushing has been separated. There are active and inactive page lists.

•Page flushing has been optimised to avoid too much interference by writeout IO on the more time-critical disk read IO.

•Controlled background page aging during periods of little or no VM activity in order to keep the system in a state where it can easily deal with load spikes.

•Streaming IO is detected; we do early eviction on the pages that have already been used and reward the IO stream with more agressive readahead.

Architecture features

- SMP locking optimizations
  - Use of global “kernel_lock” was minimized.
  - More subsystem based spinlock are used.
  - More spinlocks embedded in data structures.
  - Semaphores used to serialize address space access.
  - More of a spinlock hierarchy established.
  - Spinlock granularity tradeoffs.

Architecture features

- Increased number CPUs supported
  - Static increase of maximum CPUs to 64.
  - Realistic scalability of up to 8 CPUs.
    - Bus saturation
    - SMP locking
  - Scheduler optimizations speed up selection of threads and context switching.

Linux 2.4 adds support for three new architectures: Intel ia64 (Itanium/Merced), IBM S/390, and Hitachi SuperH. (Intel Itanium/Merced CPU is not yet available.) Changes made to the 2.4 kernel are based on the current specifications and existing support for current 64-bit CPU architectures like the Alpha and Sparc64.
Linux 2.4 hardware support, in general, is very similar to Linux 2.2. All Intel chips 386 to Pentium III are supported, as are the compatible AMD and Cyrix chips. The Crusoe CPU, designed by Torvalds' employer Transmeta, uses a Code-Morphing firmware that emulates the i386 so there is no need for a Crusoe-specific port of Linux.
Support for high-end hardware to increase speeds comes from support of non-Intel varieties of the Memory Type Range Registers (MTRR) for the AMD K7 processors and the Cyrix processors' variation called MCR, which will improve performance on some high bandwidth devices. Where Linux 2.2 included support for the Memory Type Range Registers (MTRR) used on the newest Intel chips to increase performance of some kinds of high-bandwidth devices, Linux 2.4 will take this even further by supporting MTRR (and MCR) variants from alternative vendors AMD and Cyrix.
There will be no support for any CPU previous to the Intel 386. Anyone wanting to install Linux on an old 286 box will have to look for specialized versions like the Embeddable Linux Kernel Subset (ELKS).

Architecture features

- Kernel multi-threading improvements
  - Multiple threads can access address space data structures simultaneously.
  - Single mem->msem semaphore was replaced with multiple reader/single writer semaphore.
  - Reader lock is now acquired for reading per address space data structures.
  - Exclusive write lock is acquired when altering per address space data structures.

Architecture features

- 32 bit UIDs and GIDs
  - Increase from 16 to 32 bit UIDs allow up to 4.2 billion users.
  - Increase from 16 to 32 bit GIDs allow up to 4.2 billion groups.

Architecture features

- 64 bit virtual address space
  - Architectural limit of the virtual address space was expanded to a full 64 bits.
  - IA64 currently implements 51 bits (16 disjoint 47 bit regions)
  - Alpha currently implements 43 bits (2 disjoint 42 bit regions)
  - S/390 currently implements 42 bits
  - Future Alpha is expanded to 48 bits (2 disjoint 47 bit regions)

Architecture features

- Unified file system cache
  - Single pagecache was unified from previous pagecache read/buffermem write functionality
  - Eliminates copying buffers from buffermem to pagecache on file read operations.
  - Reduces memory consumption by eliminating double buffered copies of file system data.
  - Eliminates overhead of searching two levels of data cache.

Architecture features

- Distributed Interrupts
  - Hardware interrupt service routines can be processed simultaneously on all CPUs.
  - Software interrupts (softIRQs) can be processed simultaneously on all CPUs.
  - SMP spin locks are maintained within device specific data structures.

Architecture features

- Increased number of threads and tasks
  - Default maximum number of tasks/address spaces was increased.
  - Default maximum number of threads per task was increased.
  - Configuration of both maximums was changed to be runtime tunable via /proc file system.
  - Scheduler optimizations minimize overhead of context switching between sibling threads.

One common problem with Linux 2.2 that interfered with high-end (Intel?) machines was its process limitations. Linux 2.2 only allowed you to have 1024 processes or threads running at once. With high-end systems with many thousands of users, this could become a problem very quickly. Linux 2.4 has gotten rid of this relic and implemented a scalable limit which can be configured at run time and is only limited by the amount of memory in the system. On high-end servers with as little as half a gigabyte of RAM installed, it is easily possible to support as many as 16 thousand processes at once. Other users have reported being able to run many more than that on their specific systems. This was one of the major bottlenecks that kept Linux out of the Enterprise markets.

Hardware Support Features

- IA64 Port and Architecture Optimizations
  - Support for IA64 processor features:
    - IA64 specific TLB optimizations.
    - Large rotating register file.
    - IA64 SMP specifics.
    - IA64 IO specifics.
  - 64 bit virtual address space.
    - Itanium is actually 51 bits; sixteen 47 bit regions.
  - NUMA support under development.

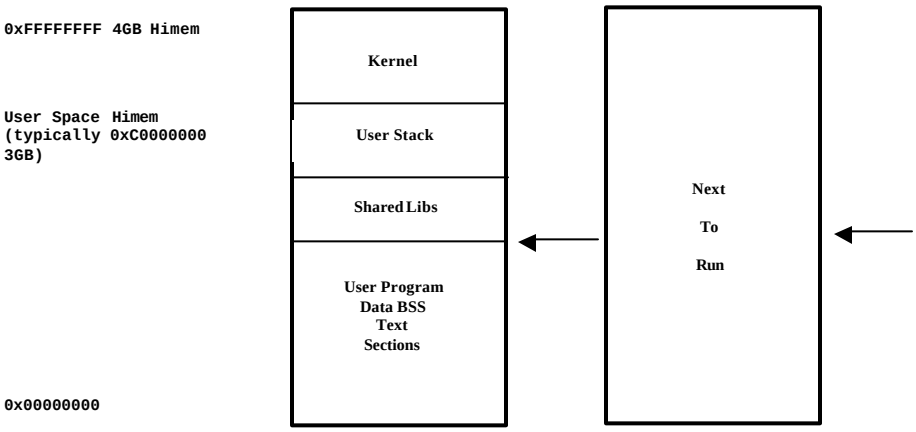
Hardware Support Features

- Alpha Architecture Optimizations
  - 64 bit virtual address space.
    - EV67 is 43 bits; half user, half kernel.
    - EV7 supports 48 bits; half user, half kernel.
  - 2TB(41 bit) physical address limit.
  - Highly accurate SMP compatible processor time optimizations.
  - NUMA support under development.

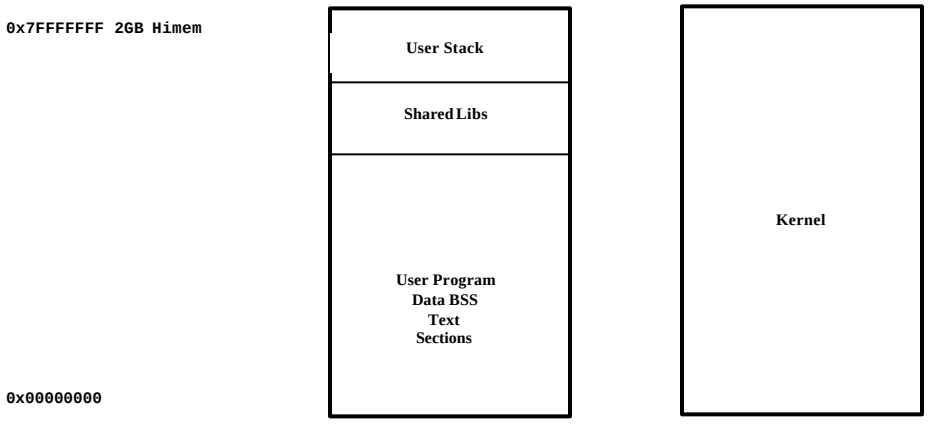
Hardware Support Features

- S/390 Architecture Optimizations
  - 64 bit virtual address space
    - 42 bits used - separate address spaces for users & kernel
  - 16EB physical address limit.
  - Highly accurate SMP compatible processor time optimizations.
  - NUMA support under development.

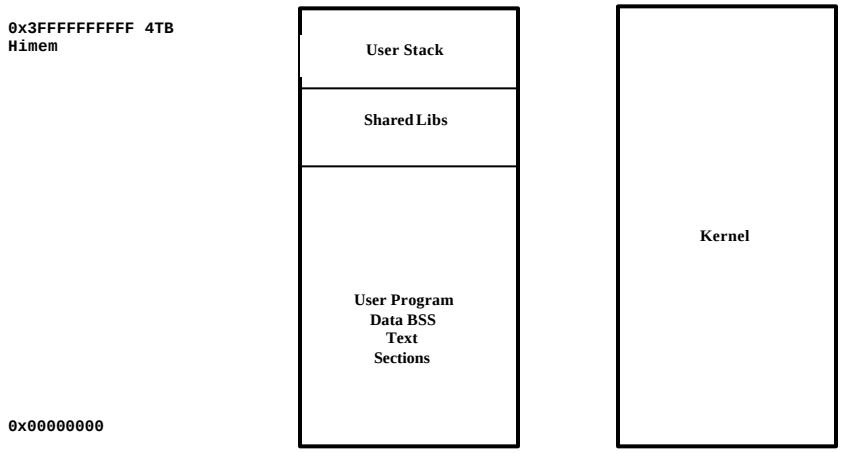
Linux – Intel Address Spaces



Linux – S/390 Address Spaces



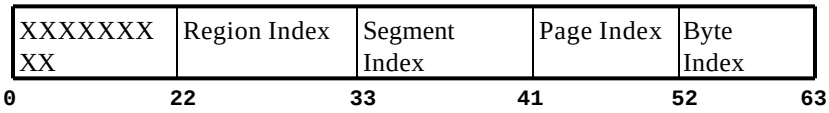
Linux – zSeries Address Spaces



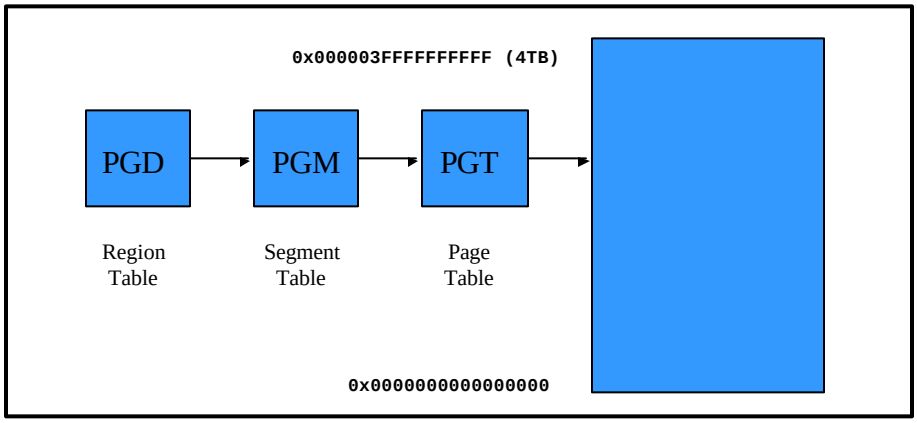
S/390 & z/Architecture Addressing

- A virtual address on S/390 is made up of 3 parts:

- On z/Architecture in Linux we currently make up an address from 4 parts:



zArchitecture Address Spaces



Address Spaces

- Kernel runs in Primary Space mode
- User programs run in Home Space mode
- Copy to/from user just a MVC(L/E) in Access Register mode with AR set for kernel/user address spaces
- Compare this to some of the other elaborate schemes used

Hardware Support Features

- BIGMEM for IA32 (and other 32 bit systems)
  - 1GB physical memory limitation in the Linux kernel.
  - 4GB physical memory limitation for 32 bit systems.
  - 4GB physical memory optimizations in the Linux kernel.
  - 64GB physical memory using PAE on IA32.

Hardware Support Features

- Special instructions for some processors
  - Use of processor specific memory transfer instructions for:
    - Intel Pentium
    - AMD
    - Cyrix
    - WinChip

2.4 Kernel Hardware Support Features

- NUMA infrastructure
  - Machine independent Non-Uniform Memory Architecture (NUMA) infrastructure.
  - Support for:
    - multiple memory domains
    - processor subsets
    - binding of devices and interrupts to processors
  - Machine dependent NUMA portion under development for multiple architectures.

File System Enhancements

- File System size increase
  - File system data offset was increased from 31 bits to 44 bits in the VFS layer.
    - Increases file system size to 16TB.
    - Increases individual file size to 16TB.
    - Still need to consider file system overhead...
  - Several local file systems have been enhanced to take advantage of larger files.

Block filesystems can be used in many ways. The most common way to use a block devices is to mount a filesystem on it. (Internally, the filesystem code is like an overlay on the block device driver.) It should also be mentioned that these filesystems (as well as nearly everything else) will work with all versions of Linux and are not only applicable to i386 Linux.
Linux 2.4 includes all of the new filesystems present in Linux 2.2. These filesystems include FAT (for MSDOS), NTFS (for Windows NT/2000), VFAT and FAT32 (for Windows 9x), HFS (for Macintoshes), and many, many others. All of these filesystems have been rewritten to some extent, sometimes a very large extent, to support the new page caching system and will be more efficient because of it. On the flip side however, binary-only filesystem modules designed for Linux 2.2 will not work with Linux 2.4. (Unlike some software firms, Linux does not generally provide for back-compatibility at the module level. Generally, open source modules can adapt quickly enough and binary module providers are expected to do the same or release the code.)
Some users will however notice major improvements to allow for better compatibility with other systems. OS/2 users will finally be able to both read and write to their disks under Linux. (This change is a long time in coming.) NT users unfortunately don't yet have that luxury unless they wish to use an "experimental" driver which may lead to disk corruption under certain situations.
Linux 2.4 will also include a couple of improvements designed to make it interoperate better with other UNIX-like operating systems. Key to this is Linux 2.4's upcoming support for the IRIX efs filesystem and the IRIX disklabel (partition table) format. Also, support for NextStep has also improved as the UFS driver now supports its CDROMs.
Users who mount Windows shared drives via SMB (Server Message Block protocol) will be pleased that there will no longer be a compile time option for enabling workarounds for (released broken) Win9x systems. Instead, Linux will be able to detect what kind of system it is connecting to and enable bug fixes as needed. This will make Linux a considerably better option for heterogeneous networks. (This is a SMB client only, the popular Samba package can be used if server features or access to printers is desired.)
Of special importance to many Linux users is Linux's ability to mount the shared drives of UNIX operating systems. Linux 2.4 includes for the first time the ability to access NFS shares which conform to version 3 of the NFS protocol. NFS version 3 includes many advantages over previous versions and it has been one of Linux's most often requested features for the enterprise user.
There are still some pieces of support that is currently lacking in Linux 2.4. There is no support for journalizing filesystems, for instance. Due to the relatively low fsck times and the ease of data recovery journalizing filesystems support, this is considered by many to be an entrance requirement to the enterprise. HFS+, the successor to HFS and the filesystem used on some Macintosh disks, is not yet supported. Also not supported is the UDF format, the format commonly used on DVD drives. It is hoped that these and other "missing" features will be completed before 2.4 is ready for release however there will be a code freeze coming soon.

File System Enhancements

- VFS layer redesign to use single cache
  - buffermem and pagecache functionality was unified in 2.4
  - VFS layer was changed to use pagecache for generic file read() and write() operations.
  - Eliminated coping between buffermem and pagecache.
  - Saves memory be eliminating multiple copies of buffered file system data.

The virtual filesystem layer (VFS) has also been heavily modified from earlier Linuxes. Linux 2.2 featured a number of wonderful changes to this layer that allowed for better caching and a much more efficient system overall. However, the system in Linux 2.2 still had a number of important limitations which were resolved in time for Linux 2.4. One major limitation to the way Linux 2.2 handled things was its use of two buffers for caching: one for reading and one for output. As you can imagine, this made things very complicated as the kernel developers had to code with kid gloves to always ensure that these caches were in synch when they had to be. Linux 2.4 brings this wall completely down by removing the multiple cache system and putting all the work into a single page caching layer. This change makes Linux 2.4 more efficient, the code is easier to understand for developers, and the amount of memory needed for the caches have been split roughly in two. During the course of this rewrite, many race conditions (errors caused when multiple processes "race" for access to unprotected variables) were removed and the code streamlined to allow significantly better scaling to higher-end systems and disk writes to happen faster when multiple volumes are involved.

File System Enhancements

- RawIO support to bypass file system cache
  - New RawIO interface was added to file systems.
  - This results in:
    - DMA directly to buffer wired in user address space.
    - Bypassing the pagecache.
    - Eliminates coping between pagecache pages and user buffer pages.
    - More efficient for databases.

File System Enhancements

- Several Journaling File Systems introduced
  - Pending file system updates are continually maintained in a single journal file.
  - The FSCK at reboot time is reduced to replaying the journal.
  - Speeds up reboot FSCK by several orders of magnitude.
  - ext3fs, reiserfs, xfs, jfs

File System Enhancements

- Inclusion of Logical Volume Manager into the Linux kernel
  - Allows file systems to span multiple disks.
  - Dynamic runtime resizing of file systems.
  - More flexible file system device management.
  - Standards compliant.
  - Familiar to users of commercial UNIX.

Networking Enhancements

- Network re-write for optimal performance
  - Redesigned to take advantage of improved multitasking and multithreading.
  - Improvement performance for simultaneous/multiple network interfaces.
  - Distributes networking load much more evenly on SMP systems.
  - Kernel uses wakeup_one to minimize wasted cycles

The Linux model of network sockets is one which is standard across most UNIX variants. Unfortunately however, the standard does have some deficiencies but these deficiencies can be corrected without breaking the standard altogether. Under Linux 2.2 and previous versions, if you have a number of processes all waiting on an event from a network socket (a web server, for instance), they will all be woken up when activity if detected. So, for every web page request received, Linux would wake up a number of processes which would each try and get at the request. As it does not make sense for multiple processes to serve the same request, only one will get to the data; the remainder will notice that it doesn't have anything to process and fall back asleep. Linux is quite efficient at making this all happen as quickly as possible, however it is still very inefficient... but there is a better way. Linux 2.4 includes changes which implement "wake one" under Linux which will allow us to completely remove the "stampede effect". In short, "wake one" does exactly as its name indicates: wakes up only one process in the case of activity. This will allow applications such as Apache to be even more efficient and make Linux an even better choice as a web server. Linux 2.4 also includes a completely rewritten networking layer. In fact, it has been made an unserialized as possible so that it will scale far better than any previous version of Linux. In addition, it contains many optimizations to allow it to work with the particular quirks of the networking stacks in use in many common operating systems, including Windows. It should also be mentioned at this point that Linux is still the only operating system completely compatible with the IPv4 specification (Yes, IPv4) and Linux 2.4 boasts an IPv4 implementation that is much more scalable than its predecessor. Linux 2.4 has a completely rewritten networking layer. It has been made as un-serialized as possible to allow for better scaling. The network subsystem is redesigned for stability on multiprocessor systems and many ***races*** have been eliminated. There are also many updates to the existing network driver set as well as many new devices, including support for ATM network adapters for high-speed networking.

Kernel Networking Enhancements

- Firewall and IP functions placed in kernel
- Network subsystem split:
  - Packet filtering layer
  - Network Address Translation layer
- PPP code rewritten and modularized
- ISDN updated to support many new cards
- PLIP improved
- DECnet & ARCNet protocols supported
- Autodetection of Windows shares based on SMB
- Completely compatible to the letter of IPv4 spec

The Linux 2.4 rewrite includes placing firewall and Internet protocol functions into the kernel. The network subsystem has been split into two pieces: a packet filtering layer and a network address translation (NAT) layer. Each is more generic than its previous version and allows sophisticated routing through any Linux box. The rewrite includes a new user-space tool to manage the available functionality. To make the upgrade easier by providing backward compatibility, modules that will allow use of the Linux 2.0 ipfwadm or Linux 2.2 ipchains are included.

Much of the PPP (point-to-point protocol) code has undergone major rewriting and modularization. The kernel now combines the PPP layers from the ISDN layer and the serial device PPP layer (like those used for dial-up connections with modems). ISDN has also been updated to support many new cards. The PLIP (PPP over parallel ports) layer has been improved as well, and now uses the new parallel port abstraction layer and PPP over Ethernet (PPPoE) support, a protocol used by some DSL providers.

Enterprise level changes to networking introduce features that will better enable Linux to be integrated into existing network infrastructures. Linux 2.4 adds DECnet support for interoperating with specialized Digital/Compaq systems and ARCNet protocols, as well as hardware that allows for better interoperation with specialized systems (including older Digital/Compaq ones).

Linux 2.4 will be able to auto-detect Windows shares based on Server Message Block (SMB) protocol. This makes Linux a better candidate for use in mixed (heterogeneous) networks. The Linux 2.4 kernel removes the compile-time requirement of selecting support for mounting drives from Windows 9x or NT. 2.4 will be able to auto-detect the remote system type and enable bug fixes (some versions of Windows have SMB bugs) as needed. (This is for an SMB client only. The Samba package can be used if there are server needs.)

The IPv4 implementation is more scalable, and the use of colon-mode for IP "aliasing" has been removed. Linux is still the only operating system completely compatible with the letter of the IPv4 specification.

Networking Enhancements

- iptables/netfilter replacement for ipchains
  - Linux 2.2 replaced ipfwadm with ipchains.
  - Linux 2.4 replaced ipchains with iptables, also known as netfilter.
    - Includes capabilities to construct more sophisticated firewalls.
    - Can be used to implement NAT for supporting masqueraded private networks
    - Compatible with ipfwadm and ipchains command syntax.

Networking Enhancements

- Kernel based HTTP daemon
  - khttpd is a kernel daemon module which serves static web pages.
  - Can cooperate with Apache and other web servers to serve dynamic web pages.
  - Will result in significant web benchmarking improvement (SpecWeb, etc).

Newly integrated into Linux 2.4 is a kernel Web server, khttpd. This kernel-space http ***daemon*** won't have to exist in user-space. It sends data to kernel-space to be taken to the network connection. This results in faster response times. The khttpd can, however, only run static pages. Apache or another httpd will have to be used to run CGI programs, because khttpd is not designed as a replacement for Web servers. If it receives a request for CGI, khttpd will pass the request to user space where a Web server can process it.

Networking Enhancements

- Fully compatible NFSv3 implementation
  - Fully compatible with version 3 of NFS distributed by Sun Microsystems.
  - Eases the burden of Linux sysadmins who maintain heterogeneous environments.
  - Also compatible with:
    - DECnet
    - ARCnet

Device Support

- 2.4 Supports:
  - Up to 10 IDE controllers
  - Up to 16 ethernet cards
  - Multiple AIPCs
  - SCSI TCQ (tagged command queuing)
  - RAID devices
  - ATM

From most user's points of view, there are three different fundamental types of devices under Linux: block devices, character devices, and network devices. We will discuss each of these in turn.
Block devices are hardware whose data can be best expressed in an array of bytes that can be accessed individually. (This is simplified a bit.) To use a more computer savvy term, b lock devices are devices that support random access; allowing a user to seek to a specific place anywhere on the device to read from or write to (this is also simplified a bit). Common examples of block devices are harddisks, floppy drives, (anything that you can imagine as a "drive", mostly.), ramdisks, etc. If a device has special features (for example, can be ejected), it will support these extras through ioctls (I/O controls) which any program can use. Linux 2.2 already supports the most common types of storage media for enterprise and desktop use including RAID controllers, IDE and SCSI disks, and many others. Linux 2.4 will build on this in a number of important ways.
IDE is the most common type of disks used in PCs today. Each IDE controller actually supports two separate disks (harddrives, cdromdrives, etc.) which appear under Linux as separate block devices. Linux 2.4 has improved on Linux 2.2's support of IDE by more than doubling the number of IDE controllers allowed in a system to 10. (Previously, 4 was the maximum allowed.) This boosts Linux to a theoretical limit of 20 IDE devices. There have also been some changes to allow for better support for DVDs and CD-ROM changers. While it may not be ready for Linux 2.4, there is ongoing work to allow Linux to fully support rewritable CDs and DVDs in a transparent fashion, for the time being however these should be considered read-only under normal circumstances but a previously formatted disk image can be copied out to the disk directly. And finally, Linux 2.3 has access the UDMA features of many new hardware chipsets and can work better around the bugs present in some pieces of hardware.
The SCSI subsystem has advanced in Linux 2.4, the most obvious example being in the number of new SCSI controllers supported. The long awaited SCSI rewrite has not happened for Linux 2.4 although a major cleanup effort is underway.
One idea adopted from the commercial UNIX world into Linux is the concept of a "raw" I/O device. A raw device is one whose accesses are not handled through the caching layer and whose actions are immediately and always synchronous with the "hard" data on the disk or elsewhere. This idea has many enterprise uses as it allows Linux to better maintain data integrity in the case of a system failure for ultra-important data. Also, this capability has been exploited by database applications which feel that they can do a better caching job than the native filesystem. What kept this idea from being adopted before was that commercial UNIXes did not provide a scalable process to allocate and access these devices, rather they required that a "raw" device node be allocated for each and every block device on the system. After much thought and many rejected ideas, this functionality was finally allowed in by creating a pool of "raw" device nodes which then can be associated with any arbitrary block device. Thus, we need only have nodes allocated for the number of raw devices that we will be using at any one time.

Device Support

- Buses
  - Integrated into the new resource management subsystem
- Plug-N-Play
  - ISA & S/390 device configuration and detection
- USB
- I2O supported (PCI extension)
- PCMCIA support integrated

Buses - ISA, PCI, USB, MCA, etc.

Processors however are just a small part of the guts of a computer. Equally important to its operation is its bus architecture, the component of the system that is responsible for (or irresponsible towards, as the case may be) internal and external devices. Linux 2.4 has not yet touched much on the internal workings of many of the supported busses, including (E)ISA, VLB, PCI, and MCA except to work them into the new resource management subsystem and fix bugs. The biggest news in this area is that ISA PnP, the somewhat misguided attempt to support device configuration and detection on the ISA bus, is finally supported at the kernel level! In the future, this will allow PnP devices to "just work" and not need any supplementary configuration utilities to function properly.

There is more exciting news from this front however. Universal Serial Bus, a new external bus type just now coming into prominence for devices such as keyboards, mice, sound systems, and scanners is now supported in the Linux kernel. At the time of this writing, the support is not 100% and many individual and common USB devices are not supported or not completely supported. I would be confident however that the number of devices which are supported will only rise over time, just as we observed a similar rise in the number of framebuffer devices that are now supported.

In addition to USB, I2O device (Intelligent Input/Output) support, an extension of PCI, has been added in Linux 2.4. In theory, this will allow for more operating system independent devices and drivers to exist. Many I2O devices are already functioning and more will be added before Linux 2.4.

PCMCIA support, the semi-external bus common in laptop computers, is now supported from within the standard kernel distribution. No longer will PCMCIA users need to download and install separate packages to get their systems to work properly.

Device Support

- Framebuffers
  - New drivers and improvements to old
  - Support of many more “standard” VGA cards

Another, more complicated variety of block device is the frame-buffer. A frame-buffer is simply a section of memory that represents (or is) video memory to such an extent that writing to this memory affects the colors of the pixels on a screen. This is more complicated than some other block devices because it supports ioctls to change the palette and other functions associated to video. (Which it might be possible to "format" this device and mount a filesystem from it, I wouldn't recommend you try.)
Linux 2.4 includes a number of new drivers and improvements to old drivers. Especially important here is Linux's support for many more "standard" VGA cards and configurations, at least in some mode. (Probably less than optimally.) Please remember that this feature can be bypassed and (on i386) is only necessary for people with certain systems which cannot be supported in any other way. At this time, the XFree project provides many more drivers to many more video cards than the kernel can support so it is not necessary to use this feature to get X Windows support. (SVGAlib and other libraries allow you to do direct video manipulation on supported hardware, however the use of these libraries must be done carefully as there are some security concerns.)

Device Support

- Keyboards, Mice, Consoles, and Ports
  - USB support of keyboards and mice
  - Ability to redirect console output to parallel port
  - Serial support has same limitations as 2.2
  - Parallel port support has been overhauled
    - New generic driver
    - DMA support
  - IRDA support
  - Little work done on “WinModems”

The biggest news on this front is that Linux 2.4 will support for the first time keyboards and mice attached to the Universal Serial Bus. When plugged in, these input device will behave just as if they were "normal" keyboards and mice. Additionally, Linux will now work on more systems, including broken (or specially embedded) ones where the keyboard is not pre-initialized by the BIOS. Also, better support is provided for machines without keyboards in some cases.

As much as it may not appear so, all versions of Linux output to the screen in character mode. (Linux supports a built-in extended vt100 interface to handle cursor positioning. This is done using a very small text-mode only frame-buffer device.) In the case of a frame-buffer, Linux 2.2 and later support overlaying the framebuffer driver with a terminal driver allowing identical (sometimes even better) features as (than) the built-in text mode.

Linux 2.4 does not include many major changes to this subsystem however it does for the first time support redirecting the console to the parallel port for, for example, a printer. (Earlier versions of Linux already supported redirecting messages to serial ports.) This functionality will be of primary interest to some developers and server applications which want to maintain a hard-copy of kernel and debug messages that Linux uses.

Serial support for Linux 2.4 has not changed much and many of the same limitations from 2.2 still apply. (In particular, setting module options is generally done with an external utility rather than the standard parameters passed to modules.) Later versions of Linux 2.2 and all versions of Linux 2.4 will allow one to share IRQs on PCI serial boards; previously this was only allowed on ISA cards and on-board serial ports. Some other pieces of multiport hardware will be better supported under Linux 2.2. More updates and new drivers are flowing in regularly.

In contrast, the parallel port subsystem has undergone some major overhauls since 2.2. There is now a generic parallel port driver for abstracted communication with "unknown" types of parallel devices. This could be used, for example, by programs that want to poll the parallel port for Plug-and-Play information as we described earlier. It is these changes that allow us the side-effect of being able to use the parallel port as a console. Also, Linux 2.4 supports using all the different modes of modern parallel ports, including writing to the parallel ports using DMA, if supported in the hardware. This will speed up access to printers and other parallel devices.

Infra-red support has progressed since Linux 2.2 and there have been many changes in this area, including better network support. In a separate department, there has been some work since 2.2 on supporting so-called "WinModems" (or "soft modems" or "Linmodems"). These are modems which exist largely in software and whose drivers are often only provided by the manufacturer for Windows. While no code has been submitted to Linus for the support of these beasts, it is possible that we may see some support for them before 3.0. One major obstacle here is that each and every WinModem is different; it is unlikely that a driver for one would be applicable to another and the sheer number of different types of WinModems would make this difficult or impossible to ever get a decent selection of hardware supported.

There are some other places where some people feel that Linux 2.4 could improve, of course. With the addition of USB we have the chance to have multiple keyboards and mice attached to the same bus. Linux 2.4 however does not have internal multi-heading of these devices; you cannot assign one keyboard and one mouse to one terminal and another set to a different terminal.

Device Support

- Accessibility
  - Support for speech synthesizer card
- Multimedia
  - No ground breaking changes
  - Updates and new drivers for variety of cards
    - Including full duplex support
  - Ease of configuration enhancements

Accessibility

Linux is not commonly regarded as a "user friendly" operating system. Therefore, one would be surprised to learn that Linux 2.4 (and some later versions of Linux 2.2) includes support for its first speech synthesizer card. These cards will allow Linux users to hear all Linux output, including messages early in the boot process. Very few operating systems can boast such complete support for these devices at the kernel level. (Other patches and utilities are still required to get the full use out if these cards, however their presence in the kernel is a giant leap in the right direction for Linux.)

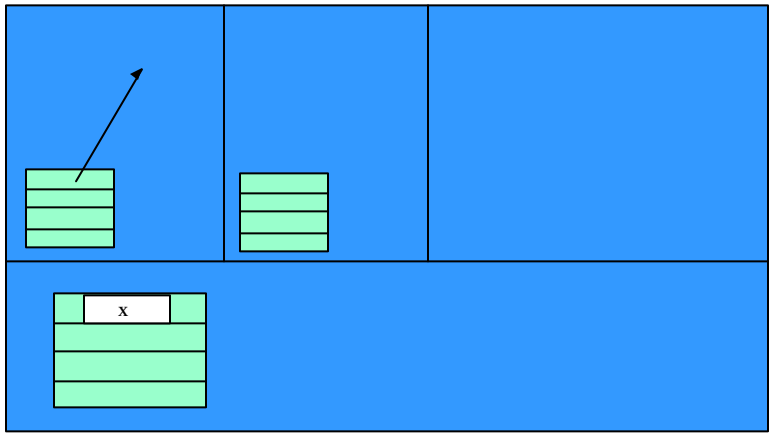
Multimedia: Sound, TV, Radio, etc.

On the complicated side of the character device list, we have some of the "fluffier" devices to be supported by Linux. Linux, in its emerging role as a desktop platform, tries very hard to support these devices, including sound cards, TV and radio tuners, and other sound and video output devices. To be honest, Linux 2.4 does not include as many ground-breaking changes as Linux 2.2 did in this respect. Linux 2.4 does however include updates and new drivers for a variety of sound and video cards, including full duplex support. Linux 2.4 and some later versions of Linux 2.2 also include code which will allow some sound devices to more easily allocate memory in required ranges; this should make the configuration and use of some cards much easier.

Device Support

- S/390 Devices
  - 3270 as console and terminal
  - Enhanced CIP (CLAW) driver(s)
  - Tape support (3480; 3490; 3590)
  - Hipersockets (classic & OSA)
  - z/OS formatted disk (VTOC & DCBs)
    - Linux disks now recognized by z/OS
    - Hitachi provided file system driver for PDS/PDSE (r/o)
  - PAGEX/PFAULT support (VM only)
  - Kernel in NSS (VM only)

PAGEX/PFAULT Support



PAGEX/PFAULT Support

- Eliminate overhead of double paging
- Page fault by Linux virtual machine usually puts it in wait state until VM gets page
- PAGEX/PFAULT handshaking allows VM to inform Linux of page request and have it dispatch another process
- When page operation is complete VM signals Linux again so it can mark task as dispatchable

PAGEX/PFAULT Support

- PAGEX
  - PROG 14 interrupt
  - 32-bit only
- PFAULT
  - External interrupt (x'2603')
  - 32 & 64-bit systems
  - z/VM 4.2 required

Kernel in NSS

- Requires gcc 3.1
  - Proper split of r/o data
- Multiple Linux guests can share r/o data and code
- More significant as you add more guests

SCSI Support

- zfcp device driver now available for 2.4
- Redpaper available: “Getting Started with zSeries Fiber Channel Protocol”
- Initial offering:
  - No IPL
  - No zSeries multipathing
  - Only switched fabric supported

Automatic Shutdown

- Service Signal (ext. x2401) to inform O/S that shutdown is imminent
- O/S “registers” via SERVC with appropriate parameter list
- Signal received – retrieve event data telling how long before shutdown
- Initiate shutdown (e.g. exploit “CTRL-ALT-DEL” logic)
- z/VM 4.3 required for z/VM-based systems

“Jiffies” Patch

- Not part of official 2.4 tree
- Provided by SuSE, Redhat, or by applying patches yourself
- Eliminates 100Hz timer pop
  - Determine when to wake
- Reduces overhead
- Allows virtual machine to drop from run list and pages to be eligible for LRU

What's still needed?

- Greater scalability above 8 processors
  - S/390 already showing scaling to 16
- NUMA
- Improved fiber-channel handling (requires an inappropriate amount of hand waving to work)
- >1TB per file system limit
- Poor I/O throughput on x86 class machines with very large amounts of memory
- Basic fail-over is there but not advanced clustering
- Logical volume manager needs more work