Linux Kernel

Subsystems of kernel

Hüseyin YAĞCI

Department of Computer Science
University of Yaşar

-2 Presentations, 2015

Outline

Outline

Scheduler
Scheduling

- Every time processes request services from the kernel; rendezvous style for some older operating systems.
- In Unix and Linux systems its vice versa, process executes kernel code for itself.
- This situation may cause disaster for kernel but entering kernel space is strictly controlled with hardware support.

Scheduler

Scheduling

- Pre-emptibility: Fairness in user space but kernel space tasks cannot be pre-emted.
- Interrupts: will temproraily interrupt the running task.
- Scheduling algorithm: Latency / fairness allowing each task to run as soon as possible.
- Locking and Scheduling: Symmetric multiprocessing (SMP) systems. Synchronisation. Spinlock.

Outline

Memory Management

Kernel Memory Allocation

- In the kernel, malloc() is not available...
  Three zones ZONE_DMA (accessible by instruction set architecture (ISA) DMA), ZONE_NORMAL, and ZONE_HIGHMEM (directly accessible by kernel but requires virtual-to physical address translation through the memory management unit (MMU);)

Memory Management
Kernel Memory Allocation

- Memory allocation functions for kernel;
  alloc_bootmem_....(): used at boot time.
  get_fre_pages():get a power-of-two multiple of PAGE_SIZE contiguous physical pages.
  kmalloc(): get any size memory (Max 128KIB).
  kmem_cache_alloc(): et predefined size from a kmem_cache.
  vmalloc(): allocate contiguous virtual memory.
  request_mem_region(): reserves specific physical addresses for device I/O.
  remap_pfn_range():reserves a virtual address range and maps it to a given range of physical pages.

Memory Management
HIGHMEM

- **HIGHMEM**
  In 32 bit x86 architecture traditionally the Linux kernel has split 4GB virtual memory address space into 3GB for user programs and 1GB for the kernel. At first it is not directly addressable. To address it , first kmap() function has to be called to enter memory page into kernel page table. Then the address is valid, until kunmap() is called.

Memory Management
DMA

- **DMA**
  DMA requires some memory space that can be accessed by the hardware, which is not cached. Therefore, drivers of DMA hardware use dma_alloc_chorenet() to allocate DMA-able space.
  Since dma_alloc_coherent() allocates at least a full page, use dma_pool_create() to allocate space for smaller transfers. Then, take some space from the pool with dma_pool_alloc().

Outline

Device driver infrastructure

Initcall Mechanism

- Linux kernel has clever and well-optimised mechanism for calling initialisation code in drivers.
- The kernel makes clever use of macros and GCC attributes to ensure that initialisation functions and pointers to them are stored in unique sections of the ELF. Initialisation code at kernel startup then iterates through these function pointers and executes them in turn. Finally, one all init code has been executed, the entire ELF section is freed for re-use. The best part of this mechanism is that the provided macros completely hide its underlying complexity, thus leaving more time for driver developers.

Outline

Networking
General Information

- **Netfilter** is a framework that provides hook mechanism for those who need to write their own functions for mangling packets within GNU/Linux kernel.
  Netfilter can filter the packet of the kernel network stack. It has 4 tables (filter, nat, mangle, raw) and 5 chains in protocol stack. The different protocol families(IPV4, IPV6, etc) of hooks linked each other by linked list. Every table may have multiple policies stored in an array.

Networking
Linux Networking Systems

- Linux kernel's network sub-systems including the NIC driver's initialization, delivery/receipt of packets, IP packet's processing.
- NIC drivers are different from other driver classes in that they do not rely on /dev or /sys to communicate with user space. Rather, applications interact with a NIC driver via a network interface (for example, eth0 for the first Ethernet interface) that abstracts an underlying protocol stack.

Networking

Linux Networking Systems

- RX Packets: Polling mode while many hardware interrupts are raising in period of time, and then turn back to interrupt mode when not many packets need processing.

- there will be 5 steps to start running when the packets travels to rtl8169_rx_interrupt:
  - netdev_alloc_skb() in rtl8169_alloc_rx_skb(), allocate a receive buffer.
  - skb_reserve(), add a 2-byte padding between the start of the packet buffer and the beginning of the payload for align with IP header which is 16-byte.
  - NIC hardware maps a memory space from DMA to memory. Copy the data in DMA into a preallocated sk_buff when the data arrives.
  - skb_put(), extend the used data area of the buffer.
  - netif_receive_skb(), enqueue the packet for upper protocols/levels to process.

Networking
Linux Networking Systems

- TX packets: Each NIC has its own buffer for packets (ring buffer). Kernel will write packets into the buffer and send TX instructions to control register. The NIC takes packets from the buffer and hits the wire. The linux kernel will copy the packets to kernel space by invoking the memcpy_fromiovec() function which is invoked by packet_snd() function, which invoked is by the packet_sendmsg() function.

Outline

File systems
Introduction

- ”On a UNIX system, everything is a file; if something is not a file, it is a process.”
  This statement is true because there are special files that are more than just files (named pipes and sockets, for instance), but to keep things simple, saying that everything is a file is an acceptable generalization. A Linux system, just like UNIX, makes no difference between a file and a directory, since a directory is just a file containing names of other files. Programs, services, texts, images, and so forth, are all files. Input and output devices, and generally all devices, are considered to be files, according to the system.

File systems

Sorts of files

- Directories: files that are lists of other files.
- Special files: the mechanism used for input and output.
- Links: a system to make a file or directory visible in multiple parts of the system's file tree.
- Sockets: a special file type, similar to TCP/IP sockets, providing inter-process networking protected by the file system's access control.
- Named pipes: Act more or less like sockets and from a way for processes to communicate with each other, without using network socket semantics.

Outline

Power Management

Advanced power management (APM)

- To conserve energy while remaining quickly available, ACPI-compatible PCs may enter system sleep states. The ACPI specification defines 5 of these states, known as S-states.

Power Management

Advanced power management (APM)

- Stopgrant: Power to cpu is maintained, but no instructions are executed. The CPU halts itself and may shut down many of its internel components. Standby in Microsoft Windows.
- Suspend to RAM: All power to cpu is shut off, and the contents of its registers are flushed to RAM.
- Suspend to Disk: CPU power shut off, but RAM is written to disk and shut off as well. Hibernate in Microsoft Windows.
- Soft off: System is shut down, however some power may be supplied to certain devices to generate a wake event to support automatic startup from a LAN or USB device.

Outline

Read-Copy Update (RCU)

What is RCU, Fundamentally?

- RCU is a synchronization mechanism that is used in the Linux kernel to replace some uses of reader-writer locking and reference counting. The concepts behind RCU are quite simple: At its core, RCU is just a way of waiting for things to get done.

References

- http://kernelnewbies.org/Documentation/Subsystems
- http://www.makelinux.net/ldd3/?u=chp-15-sect-1
- http://linuxgazette.net/157/amurray.html
- http://kernelnewbies.org/Networking?action=AttachFile&do=get&target=hacking_the_wholism_of_linux_net.txt
- http://www.tldp.org/LDP/intro-linux/html/sect_03_01.html

Thanks for Listening.