LUENTO 23

Käynnistyminen, multimediajärjestelmät

1

Sisältöä

- Käynnistyminen
  - Miten käyttöjärjestelmä ladataan?
  - Miten tietokone pääsee käyntiin sähkön kytkemisen jälkeen
  - (Miten tämä tapahtuu virtuaalikoneessa?)
- Multimediajärjestelmät
  - Miten videodataa esimerkiksi käsitellään?

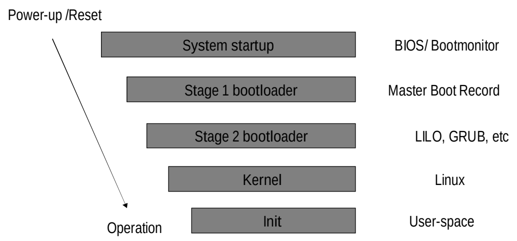
2

Booting

- Key problem: How do you initiate a system using only itself?
- Booting <- Bootstraping
- Bootstraping - why this term?
  - Baron Munchausen pulled himself out of swamp using his boot straps

3

View of the Linux boot process



Source: http://www-128.ibm.com/developerworks/library/l-linuxboot/index.html

4

BIOS tasks

- BIOS – Basic Input / Output System
  - BIOS refers to the firmware code run by a personal computer when first powered on.
  - BIOS can also be said to be a coded program embedded on a chip that recognizes and controls various devices that make up x86 personal computers. (source: wikipedia)
- Execution starts from a fixed location (on x386 that location is 0xFFFF0000)
- Check hardware, locate the boot device
  - Disk, CD, ...
- Load the first 'loader' from Master Boot Record

5

BIOS example sequence

1. Check the CMOS Setup for custom settings
2. Load the interrupt handlers and device drivers
3. Initialize registers and power management
4. Perform the power-on self-test (POST)
5. Display system settings
6. Determine which devices are bootable
7. Initiate the bootstrap sequence

6

BIOS loading the boot loader – pseudocode example

- 0: set the P register to 8
- 1: check paper tape reader ready
- 2: if not ready, jump to 1
- 3: read a byte from paper tape reader to accumulator
- 4: if end of tape, jump to 8
- 5: store accumulator to address in P register
- 6: increment the P register
- 7: jump to 1

7

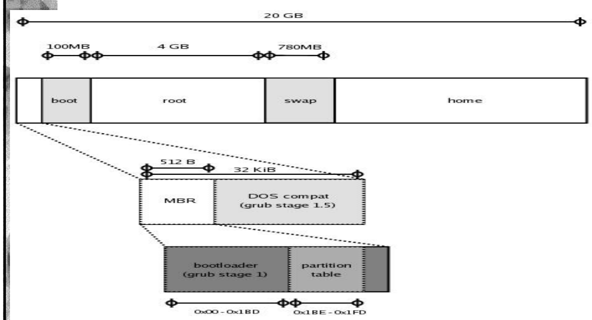
Boot loader

- Linux: Lilo, GRUB
- Windows NT: NTLDR
- Initial location on primary disk, for example: Master Boot Record

- Main task: Load and start operating system
- Dual boot: offer menu and start the chosen one

8

GRUB on disk



http://www.pixelbeat.org/docs/disk/

9

GRUB - GRand Unified Bootloader

- GRUB is independent of any particular operating system and may be thought of as a tiny, function-specific OS.
- It's primary task is to load the actual operating system and pass control to it.
- GRUB is large service:
  - It does not fit to MBR: needs to be handled in two phases
  - Phase 1 (loaded from MBR by BIOS) load phases 1.5 and 2
- A lot of features: file systems, command line interface, chain-loading other boot loaders, ...

10

Loading Linux

- Bootsector loads setup, decompression routines and compressed kernel image.
- The kernel is uncompressed in protected mode.
- Low-level initialisation is performed by asm code arch/i386/kernel/head.S:
  - Initialise segment values and page tables.
  - Enable paging by setting PG bit in %cr0.
  - Copy the first 2k of bootup parameters (kernel commandline).
  - The first CPU calls start_kernel(), all others call initialize_secondary().
- High-level C initialisation

Source: http://www.faqs.org/docs/kernel_2_4/lki-1.html

11

OS Initialising: start_kernel()

- Arch-specific setup (memory layout analysis, copying boot command line again, etc.).
- Initialise traps, irqs, data required for scheduler, time keeping data, softirq subsystem.
- Initialise console.
- Enable interrupts.
- Set a flag to indicate that a schedule should be invoked at "next opportunity"
- Create a kernel thread init()
- Go into the idle loop, this is an idle thread with pid=0.

Important thing to note here that the init() kernel thread calls do_basic_setup() which in turn calls do_initcalls()

More detailed version available from: http://www.faqs.org/docs/kernel_2_4/lki-1.html

12

Process Init

- Init is the father of all processes. Its primary role is to create processes from a script stored in the file /etc/inittab (man init)
- Establishes and operates the entirety of user space.
  - checking and mounting file systems,
  - starting up necessary user services, and
  - switching to a user-based environment when system startup is completed
  - Uses etc/rc directory hierarchy

13

etc/rc.d

- debian: run /etc/init.d/rcS which runs:
- /etc/rcS.d/S* scripts
- /etc/rc.boot/* (depreciated)
- run programs specified in /etc/inittab

- Scripts in /etc/rc*.d/* are symlinks to /etc/init.d
- Scripts prefixed with S will be started when the runlevel is entered, eg /etc/rc5.d/S99xdm
- Scripts prefixed with K will be killed when the runlevel is entered, eg /etc/rc6.d/K20apache
- Executed in numerical order

14

Runlevel	Scripts Directory (Red Hat/Fedora Core)	State
0	/etc/rc.d/rc0.d/	Shutdown/halt system
1	/etc/rc.d/rc1.d/	Single user mode
2	/etc/rc.d/rc2.d/	Multiuser with no network services exported
3	/etc/rc.d/rc3.d/	Default text/console only start. Full multiuser
4	/etc/rc.d/rc4.d/	Reserved for local use. Also X-windows (Slackware/BSD)
5	/etc/rc.d/rc5.d/	XDM X-windows GUI mode (Redhat/System V)
6	/etc/rc.d/rc6.d/	Reboot
s or S		Single user/Maintenance mode (Slackware)
M		Multiuser mode (Slackware)

http://www.yolinux.com/TUTORIALS/LinuxTutorialInitProcess.html

15

Windows NT

- Power-on self test (POST) phase
- Initial startup phase
- Boot loader phase: NTLDR
  - Startup phase: create interrupt descriptor table, page tables and enable paging
- Detect and configure hardware phase
  - osloader.exe: knows file systems,
    - read boot.ini, check if hibernated,
    - Load Ntoskrnl.ece, hal.dll (hardware abstraction)
    - Load boot drivers (hard drive, file system, etc)
- Kernel loading phase
  - Interrupt controller, memory manager, object manager, process manager, System idle process
  - Load and initialize system drivers
- Logon phase
  - Start Session manager subsystem

16

Multimediajärjestelmät

Silbershatz, Galvin & Gagne:
Operating system concepts with Java,

Luku 20

17

Objectives

- To identify the characteristics of multimedia data
- To examine several algorithms used to compress multimedia data
- To explore the operating system requirements of multimedia data, including CPU and disk scheduling and network management

18

What is Multimedia?

- Multimedia data includes
  - audio and video clips (i.e. MP3 and MPEG files)
  - live webcasts
- Multimedia data may be delivered to
  - desktop PC's
  - handheld devices (PDAs, smart phones)

19

Media Delivery

- Multimedia data is stored in the file system like other ordinary data.
- However, multimedia data must be accessed with specific timing requirements.
- For example, video must be displayed at 24-30 frames per second. Multimedia video data must be delivered at a rate which guarantees 24-30 frames/second.
- Continuous-media data is data with specific rate requirements.

20

Streaming

- Streaming is delivering a multimedia file from a server to a client - typically the deliver occurs over a network connection.
- There are two different types of streaming:
  1. Progressive download - the client begins playback of the multimedia file as it is delivered. The file is ultimately stored on the client computer.
  2. Real-time streaming - the multimedia file is delivered to - but not stored on - the client's computer.

21

Real-time Streaming

- There are two types of real-time streaming:

  (1) Live streaming - used to deliver a live event while it is occurring.

  (2) On-demand streaming - used to deliver media streams such as movies, archived lectures, etc. The events are not delivered in real-time.

22

Multimedia Systems Characteristics

- Multimedia files can be quite large.
- Continuous media data may require very high data rates.
- Multimedia applications may be sensitive to timing delays during playback of the media.

23

Compression

- Because of the size and rate requirements of multimedia systems, multimedia files are often compressed into a smaller form.
- MPEG Compression:

  (1) MPEG-1 - 352 X 240 @ 30 frames/second

  (2) MPEG-2 - Used for compressing DVD and high-definition television (HDTV)

  (3) MPEG-4 - Used to transmit audio, video, and graphics. Can be delivered over very slow connections (56 Kbps)

24

Operating Systems Issues

- The operating system must guarantee the specific data rate and timing requirements.
- Such requirements are generally known as Quality-of-Service (QoS) guarantees.
- Guaranteeing QoS has effects in a computer system:
  - (1) CPU processing
  - (2) Scheduling
  - (3) File systems
  - (4) Network protocols

25

Requirement of Multimedia Operating Systems

- There are three levels of QoS

(1) Best-effort service - the system makes a best effort with no QoS guarantees.

(2) Soft QoS - allows different traffic streams to be prioritized, however no QoS guarantees are made.

(3) Hard QoS - the QoS requirements are guaranteed.

26

Parameters Defining QoS

- Throughput - the total amount of work completed during a specific time interval. (*läpimeno*)
- Delay - the elapsed time from when a request is first submitted to when the desired result is produced. (*viive*)
- Jitter - the delays that occur during playback of a stream. (*huojunta, viipeen vaihtelu*)
- Reliability - how errors are handled during transmission and processing of continuous media. (*luotettavuus*)

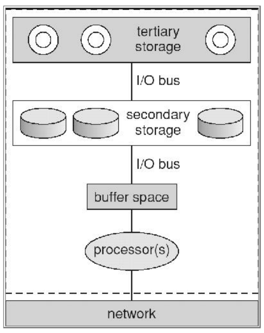
27

Further QoS Issues

- QoS may be negotiated between the client and server.
- Operating systems often use an admission control algorithm that admits a request for a service only if the server has sufficient resources to satisfy the request.

28

Figure 20.1 Resources on a file server



29

CPU Scheduling

- Multimedia systems require hard realtime scheduling to ensure critical tasks will be serviced within timing deadlines.
- Most hard realtime CPU scheduling algorithms assign realtime processes static priorities that do not change over time.

30

Disk Scheduling

- Disk scheduling algorithms must be optimized to meet the timing deadlines and rate requirements of continuous media.
- Earliest-Deadline-First (EDF) Scheduling
  - The EDF scheduler uses a queue to order requests according to the time it must be completed (its deadline.).
- SCAN-EDF Scheduling
  - SCAN-EDF scheduling is similar to EDF except that requests with the same deadline are ordered according to a SCAN policy.

31

Network Management

- Three general methods for delivering content from a server to a client across a network:

(1) Unicasting - the server delivers the content to a single client.

(2) Broadcasting - the server delivers the content to all clients, regardless whether they want the content or not.

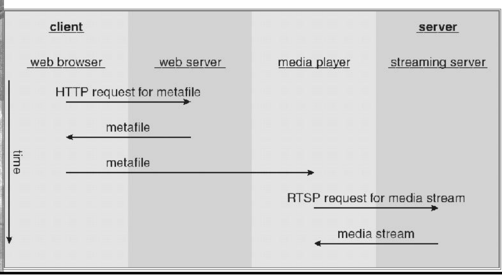
(3) Multicasting - the server delivers the content to a group of receivers who indicate they wish to receive the content.

32

RealTime Streaming Protocol (RTSP)

- Standard HTTP is stateless whereby the server does not maintain the status of its connection with the client.

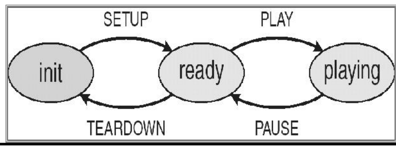
[SGG07] Fig 20.3



33

RTSP States

- SETUP - the server allocates resources for a client session.
- PLAY - the server delivers a stream to a client session.
- PAUSE - the server suspends delivery of a stream.
- TEARDOWN - the server breaks down the connection and releases the resources allocated for the session.



[SGG07] Fig 20.4

34

CineBlitz Multimedia Server

- CineBlitz supports both realtime and non-realtime clients.
- CineBlitz provides hard QoS guarantees to realtime clients using an admission control algorithm.
- The disk scheduler orders requests using C-SCAN order.

35

CineBlitz Admission Controller

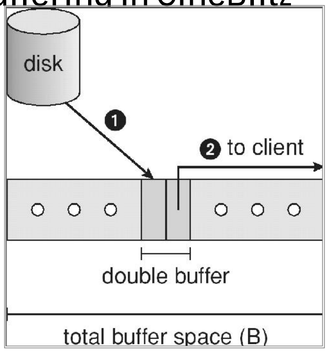
- Controller monitors resourses: disk bandwidth, latency, available buffer space
- Total buffer space required for *N* clients where each client has rate requirement of r_i

$$\sum_{i=1}^{N} 2 \times T \times r_i \le B.$$

T duration of the period
r_i rate

36

Figure 20.05
Double buffering in CineBlitz



37

CineBlitz Admission Controller (cont)

- Admission controller accepts new requests only, if the new one does not increase the estimated service cycle duration above a set upper limit (T).
- If t_{seek} and t_{rot} are the worst-case seek and rotational delay times, the maximum latency for servicing N requests is

$$2 \times t_{seek} + \sum_{i=1}^{N} \left(\lceil \frac{T \times r_i}{b} \rceil + 1 \right) \times t_{rot}.$$

38

CineBlitz Admission Controller (cont)

- The CineBlitz admission controller only admits a new client if there is at least *2* x *T* x r_i bits of free buffer space and the following equation is satisfied

$$2 \times t_{seek} + \sum_{i=1}^{N} \left(\lceil \frac{T \times r_i}{b} \rceil + 1 \right) \times t_{rot} + \sum_{i=1}^{N} \frac{T \times r_i}{r_{disk}} \leq T.$$

Total latency

Service time

39