DeMuxing

Getting shit done

3. Mai 2010

faber@faberman.de

What do we want?

Getting Audio/Video out of almost any source

Convert media data for processing

Prepare for the show

How do we get there?

Fundamentals of video/audio formats

Explaining Containers and Streams

Methods for multiplexing data

Simple hands-on examples on real life data

More examples

Golden Rule No. 1

If you can play it, you can convert it.

Golden Rule No. 2

There’s more than one way to do it.

Audio file properties

- Base unit: 1 Sample
- Samplerate: 48kHz, 44.1kHz, ...
- Resolution: 24Bit, 16Bit, float, ...
- Number of channels: 1, 2, 6, ...
- Encoding: PCM, mp2, mp3, ...

Video

- Base unit: 1 Frame
- Framerate: 25fps, 24fps, 29.97fps, ...
- Resolution: 720x576 (SD), 1920x1080 (HD), ...
- Encoding: RGB, YUV, MPEG2, ...

What do you want to do?

Streaming

Constraints: Constant Bitrate, no seeking possible

Storage

The only constraints are the time you want to invest in tweaking parameters

Containers

- MOV, AVI, MKV, MP4, OGG, ...
- Carries metadata to describe the data inside
- May contain hints for synchronisation when streaming

Multiplexing

Strategy 1

Store media data a separate blocks

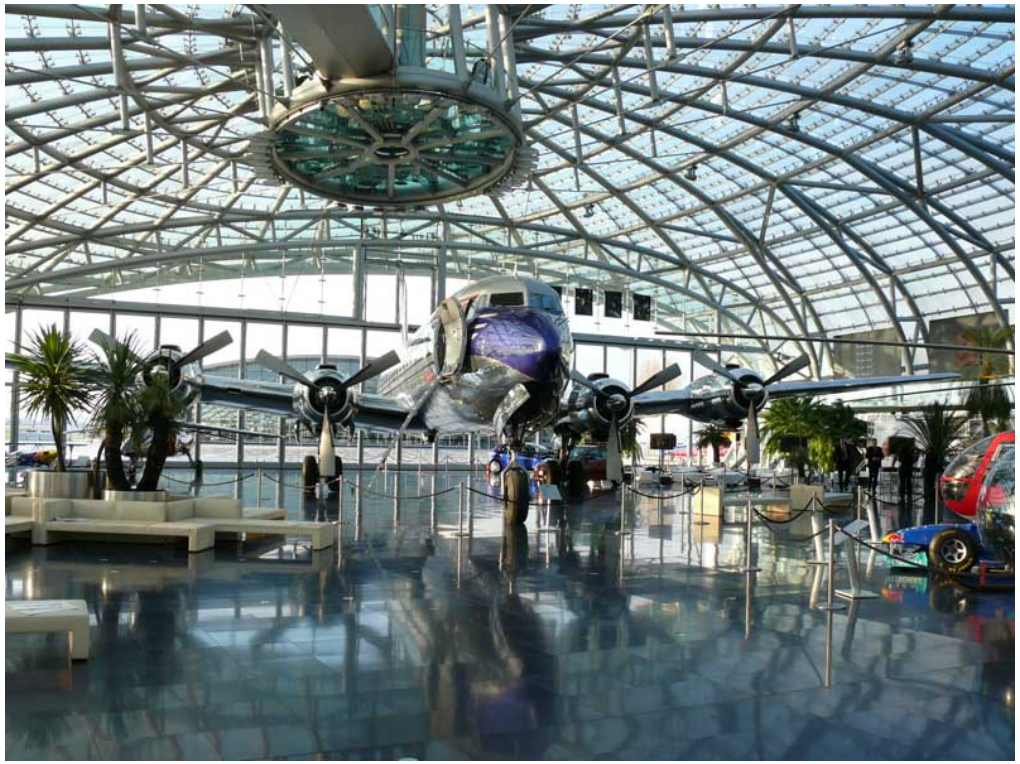
Strategy 2

Interleave small chunks of data

How to play on the venue

- Get specifications and manuals of playback gear
- Optimize all static content for target system
- Prepare a backup system
- MAKE A DRESS REHEARSAL

Fuckups



Shell

- Don’t be fooled: Extremely powerful tools
- The world’s best H.264 encoder is an open source shell tool
- You can automate and parallelize the work easily

Extracting audio from a container

To extract the first audio stream of a video file (any channel number), simply do a

```
ffmpeg -i movie.avi movie.wav
```

If the original audio is not PCM encoded, it will be transcoded on the fly. You can immediately process the result.

Changing Samplerate

libsamplerate brings a tool called `libsndfile-resample` with it. It uses Sinc interpolation, which is as good as you can get (for general purposes).

Converting a 48 kHz file to 44.1 kHz:

```
sndfile-resample -to 44100 -c 0 k_48000.wav k_44100.wav
```

Merging tracks

For simple jobs (result not bigger than 2GB) you can use the WAV format:

```
sox -M in_1.wav in_2.wav in_3.wav in_4.wav out.wav
```

This will combine whatever you throw at it to a multitrack file.

Changing Resolution

If the target system features a much small resolution than your material, e.g. a 720p beamer for 1080p material, it makes sense to downscale the video to reduce resource stress.

```
ffmpeg -i show.mov -s 1280x720 -sameq -acodec copy show_720.av
```

This example takes a QuickTime(tm) input file, scales the video to 1280x720 pixels and leaves the audio untouched. The output container is a AVI.

mplayer

You don't have to multiplex audio and video streams to play them both back. Simply use mplayer like this:

```
mplayer movie.avi -audiofile sound.wav -fs
```

This also starts mplayer in Fullscreen mode (`-fs`).

Your tasks

What do YOU want to do?