FFmpeg und weitere Tools

Reto Kromer • AV Preservation by reto.ch

Weiterführender Memoriav-Workshop
Qualitätskontrolle
von audiovisuellen Dateien
Lichtspiel, Bern, 26. Januar 2023

QCTools und FFmpeg

- QCTools ist auf FFmpeg aufgebaut
- alle FFmpeg-Befehle, die QCTools benützt, können angezeigt und auch eigenständig benützt werden

FFmpeg

Die FFmpeg-Familie

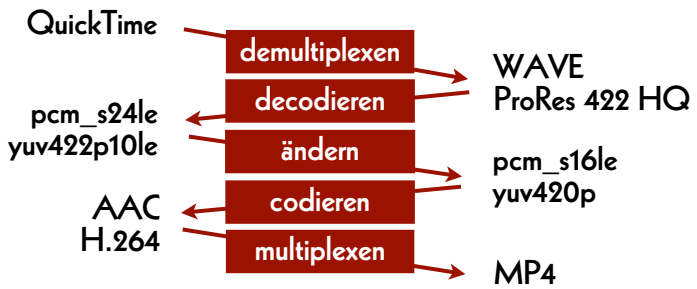
Programme

- ffmpeg
- ffprobe
- ffplay

Bibliotheken

- libavformat
- libavcodec
- libavfilter
- libavutil
- libavdevice
- libswscale
- libswresample
- libpostproc

Beispiel: Bild und Ton



Programmbibliotheken

- demultiplexen: libavformat
- decodieren: libavcodec
- ändern: libavfilter
- codieren: libavcodec
- multiplexen: libavformat

Dateiumwandlungen

ffmpeg (CLI)

→ ffmpeg.org

FFmpeg Cookbook for Archivists

→ avpres.net/FFmpeg/

ffmprovisr

→ amiaopensource.github.io/ffmprovisr/

Metadaten extrahieren

MediaInfo (GUI), **mediainfo** (CLI)

→ mediaarea.net/MediaInfo

ffprobe (CLI)

→ ffmpeg.org

Mediaplayer

VLC (GUI)

➔ www.videolan.org/vlc/

mpv (CLI)

➔ mpv.io

ffplay (CLI)

➔ ffmpeg.org

qcli

qcli recipes

Learn how to use
qcli -h

Create a report from a video file
qcli -i test.mkv

Create a named report from a video file (not working?)
qcli -i test.mkv -o hello.qztools.xml.gz

Specify filters to use on a file
qcli -i test.mkv -f signalstats+cropdetect

Make reports for all files in a folder you go into (and overwrite existing files)

cd folder
for file in *.mkv; do qcli -y -i "$file"; done

Make reports for all files in the folder you are in (and overwrite existing files)

find ./ -name "*.mov" -o -name "*.mkv"
-exec qcli -y -i "{}" \;

Bonus!

Add time in front of your commands!

Check if report has been uploaded to SignalServer (as configured with QCTools)
qcli -c file.mkv.qctools.xml.gzip

~*The Pie Guy*~

MediaInfo

Input Formats

- files locally or over a network
- FTP/FTPS/SFTP
- HTTP/HTTPS
- direct memory access

Export Formats

- flat text, XML, JSON
- EBUCore 1.5 and 1.6
- PBCore 1.2, 2.0 and 2.1
- MPEG-7
- FIMS 1.1 and 1.2

mediainfo-Befehle

```
mediainfo file.ext
mediainfo -f file.ext
mediainfo --Details=1 file.ext
mediainfo --Output=JSON file.ext
```

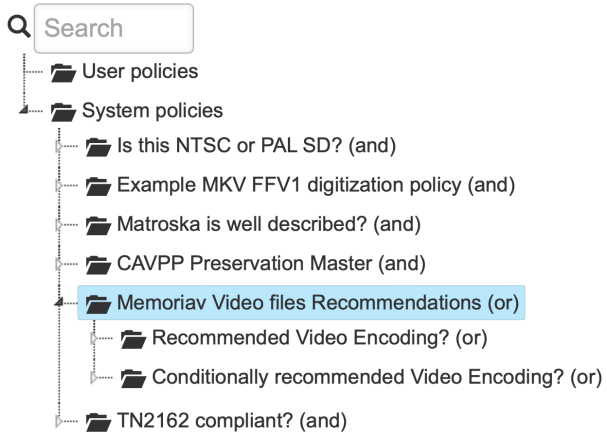

MediaConch

Die MediaConch-Familie

- GUI
- CLI
- MediaConchOnline (Web User Interface)

Policy editor

Policy list:



hexdump / xxd

Hexdump-Befehle

```
hexdump -C -n 1024 file.ext
xxd -b -c 2 file.ext
```

```
00000000 10010010
11100000 10011010
11000000 10100110
00010000 10010010
00100000 10011010
11010000 10100111
00010000 10010010
00100000 10011010
10000000 10100110
01100000 10010001
00110000 10011001
00010000 10100110
00010000 10010101
00110000 10011100
11110000 10100110
10000000 10010111
00110000 10100011
01100000 10100101
10000000 10010111
10000000 10100010
11010000 10100010
01110000 10010101
10010000 10011010
```

16-bit TIFF?

```
ffmpeg -i file.tif
  -c:v rawvideo
  -f rawvideo -
  | xxd -b -c 2
```

21

fq

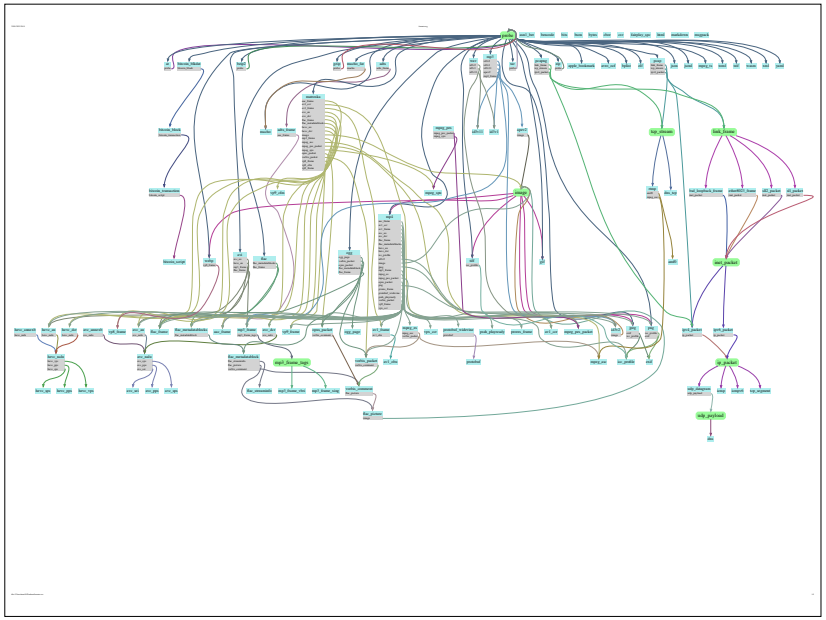
22

Supported formats

aac_frame, adts, adts_frame, amf0, apev2, apple_bookmark, ar, asn1_ber, av1_ccr, av1_frame, av1_obu, avc_annexb, avc_au, avc_dcr, avc_nalu, avc_pps, avc_sei, avc_sps, avi, avro_ocf, bencode, bitcoin_blkdat, bitcoin_block, bitcoin_script, bitcoin_transaction, bits, bplist, bsd_loopback_frame, bson, bytes, bzip2, cbor, csv, dns, dns_tcp, elf, ether8023_frame, exif, fairplay_spc, flac, flac_frame, flac_metadatablock, flac_metadatablocks, flac_picture, flac_streaminfo, gif, gzip, hevc_annexb, hevc_au, hevc_dcr, hevc_nalu, hevc_pps, hevc_sps, hevc_vps, html, icc_profile, icmp, icmpv6, id3v1, id3v11, id3v2, ipv4_packet, ipv6_packet, jpeg, json, jsonl, macho, macho_fat, markdown, matroska, mp3, mp3_frame, mp3_frame_vbri, mp3_frame_xing, mp4, mpeg_asc, mpeg_es, mpeg_pes, mpeg_pes_packet, mpeg_spu, mpeg_ts, msgpack, ogg, ogg_page, opus_packet, pcap, pcapng, png, prores_frame, protobuf, protobuf_widevine, pssh_playready, rtmp, sll2_packet, sll_packet, tar, tcp_segment, tiff, toml, tzif, udp_datagram, vorbis_comment, vorbis_packet, vp8_frame, vp9_cfm, vp9_frame, vpx_ccr, wasm, wav, webp, xml, yaml, zip

It can also work with some common text formats like URL:s, hex, base64, PEM etc and for some serialization formats like XML, YAML etc it can transform both from and to jq values.

23



24

fq-Befehle

```
fq -h
fq . file.ext
fq d file.ext
fq 'd({display_bytes: 200})' file.ext
fq -o bits_format=md5 tovalue file.ext
fq -d mp4 file.mp4
```


AV Preservation by
reto.ch

Sandrainstrasse 3
3007 Bern
Switzerland

Web: reto.ch
Email: info@reto.ch