FeM C3 Streaming and Recording Setup Documentation

** **!!!! Under construction !!!!** **

Questions go to cccongress@fem.tu-ilmenau.de

Encoding Pipeline

Recording

- Recording using BlackMagic Decklink SDI cards, a capture tool and a ffmpeg chain
- Recording is done in fixed length segments for continous capturing (and not missing any lecture starts) without reaching file system limits

Marking

The main goal of the marking process is to avoid the massive I/O that was necessary in earlier encoding setups. So the core concept is to use a custom FUSE filesystem that provides virtual files as concatenation of input files without copying large files and storing redundant data. There is one FUSE instance per lecture, not e.g. one single instance for a whole congress. The mountpoints of the FUSE instances are shared over network via glusterFS.

In 2012 there was a remake of the recording pipeline, especially in this process to support multiple audio tracks for 29C3 and maybe higher resolutions in the future. The pipeline used before 29C3 will be referenced as “v1”, the current version as “v2”.

Marking v1 workflow (< 29C3)

- recording format: rawDV 720×576 (if source is 16:9 then it is recorded anamorph) with 1x stereo PCM 16bit 48ks/s
- files containing 4500 frames each, split by simple perl script
- custom FUSE file system: **fuse-vdv** providing a file `uncut.dv`
- fuse-vdv provides a generated **kdenlive project file**
  - this file is opened by an operator on her workstation directly via the glusterFS network share.
  - in the provided kdenlive project the virtually concatenated recording is trimmed and the project is saved
  - on save, fuse-vdv parses the written project file and extracts the trim marks
- using the trim marks and the knowledge of constant frame sizes in raw DV fuse-dvd provides three additional virtual files
  - a file `cut.dv` which contains just the frames between the trim marks
  - a file `cut-complete.dv` which contains the frames between the trim marks and a

    - prepended intro as well as an appended outro file
    - a file `cut.wav` which provides the demuxed audio data of `cut.dv`
- subsequent processes (e.g. encoding the final files) just use `cut-complete.dv` or `cut.wav` (audio-only formats) as input file

Marking v2 workflow (>= 29C3)

(only the differences between v1 and v2 are shown here)

- recording format: MPEG-TS container, which can contain a vast number of video and audio tracks and different codecs for them
- files are split by the ffmpeg segmenter every 3 minutes
- custom FUSE file system: **fuse-ts** providing a file `uncut.ts`
- fuse-ts also provides a kdenlive project file
    - operator trims like in workflow v1
    - after saving the kdenlive project file, fuse-ts **does not** provide trimmed versions of the virtual files, since it doesn't know anything about MPEG-TS or the contained tracks or their codecs
    - it just extracts the trim marks from the project file, converts them from frame numbers to fractioned seconds and stores them for late use
- subsequent processes (e.g. encoding the final files) use the untrimmed `uncut.ts` as input file and must do the trimming (via ffmpeg's `-s` and `-t` options) and prepending intro etc. (via ffmpeg's `concat`-video filter) by themselves

Ticket Tracker

- Ticket tracker tracking the state of recording tickets (for each Fahrplan entry) and encoding tickets (for each format)
- PostgreSQL, PHP
- Code: http://git.fem.tu-ilmenau.de/?p=cccongress.git;a=summary

Encoding Cloud

- LiveCD image to boot multiple encoder instances
- Recording segments available via central storage box
- Encoding ticket processing with tracker connection (XML-RPC)
- Code: http://subversion.fem.tu-ilmenau.de/repository/cccongress/trunk/tools/job-control/

Streaming

Encoding

- each room one RTMP encoder
    - sdi card, capture tool, ffmpeg pipeline
    - multiple data sinks using enhanced tee → **supertee**
    - Code: http://hg.fem.tu-ilmenau.de/supertee/,

http://subversion.fem.tu-ilmenau.de/repository/cccongress/trunk/tools/sdi-script/
- each room one WMV encoder
  - sdi card, windows media encoder 9
- each room one slide only encoder
  - sdi card, capture tool providing v4l2 device (video only), motion
  - Code: http://subversion.fem.tu-ilmenau.de/repository/cccongress/trunk/tools/bmd2v4l2/, http://www.lavrsen.dk/twiki/bin/view/Motion/WebHome
  - http://subversion.fem.tu-ilmenau.de/repository/cccongress/trunk/tools/slides-only-http-streaming/saal_encoder/slides , siehe REAME

Relay

- RTMP and HTTP Relay using Nginx
  - new RTMP relay module for nginx
  - Code: https://github.com/arut/nginx-rtmp-module
- WMV relay using Windows Server 200X with Media Services
- CDN Controll software and config files
  - http://subversion.fem.tu-ilmenau.de/websvn/wsvn/cccongress/trunk/tools/fem-cdn-mm/

From:
https://wiki.fem.tu-ilmenau.de/ - **FeM-Wiki**

Permanent link:
https://wiki.fem.tu-ilmenau.de/public/projekte/c3/start?rev=1357229301

Last update: **2013/01/03 17:08**