

Unterrichten mit Open Source

Nicht nur für Linux!



Dipl.-Ing. Klaus Knopper <knopper@knopper.net>



Das wichtigste zuerst: Standard

Was ist ein Standard-Format?

Formale Definition: „Es gibt eine öffentliche Dokumentation oder Norm, die das Format genau beschreibt.“

→ Offene Standards

Fiktion: „Alle Programme können das Format verarbeiten“

→ Idealismus, oder Festhalten an historischen Formaten

Missverständnis: „Der Hersteller definiert seinen Standard.“ → Interoperabilitäts-Probleme



„Gute“ Formate (1)

- Video:
 - Container: AVI (auch als Dateiendung)
 - Codec: MPEG2, DIVX/XVID, MPEG4, H.xxx
 - Generell: ältere Formate werden von Geräten besser unterstützt, komprimieren aber schlechter oder haben schlechtere Qualität.
 - Audio:
 - WAV (unkompr.), MP3 (patent.), OGG, MP2
- Problem: Qualität vs. Kompatibilität



„Gute“ Formate (2)

- Dokumente (Text, Bilder und Multimedia-Links)
 - LibreOffice/OpenOffice
Open Document Format (OASIS/ODF)
ISO/IEC 26300:2006
 - PDF
 - V1.7 ISO 32000-1:2008
 - PDF/A-1 (Langzeitarchivierung)
ISO 19005-1:2005
 - HTML
 - Aktuell 4.01, 5 aber „Living Standard“
- Problem: Unterschiede im Grad der Unterstützung durch verschiedene Programme



Wie erstelle ich mein Unterrichtsmaterial nachhaltig?

- Information vs. Präsentation: Das Frontend zur Erstellung und Darstellung soll möglichst austauschbar sein, das Dateiformat aber standardisiert.
- Beispiel: TeX/LaTeX (schon etwas älter, aber immer noch gut): Erzeugung von DVI, PDF, HTML aus Quelltexten.
- Beispiel: LibreOffice: Leichte, jedoch eher grafik- und layoutzentrische interaktive Bearbeitung, Ausgabeformat wählbar.
- Probleme beim Wechsel des Frontends: Word-Doc(X) enthält keine verlässliche Layout-Information, PDF ist layoutgetreu, aber nicht editierbar. (Ausnahme: OO-PDF Hybridformat!)



Lösung?

- Das programmspezifische, offene „Meta“-Format zur Archivierung und Weiterverarbeitung (z.B. Hybrid-PDF (LibreOffice), XCF (Gimp), Multimedia-Rohaufnahmen mit archivieren).
- Einzelne Dateien auch nach Einbetten in „Gesamtdokument“ aufheben, da sie möglicherweise dort möglicherweise nicht mehr in der Originalqualität vorliegen.
- Wichtig: Meta-Informationen zu allen Dateien (Datum, Erzeuger/Herkunft, Lizenz) aufheben, ggf. als unsichtbarer Kommentar im archivierten Dokument!
→ Verbreitung unter CC oder anderen Lizenzen möglich, Einschränkungen?



Worst Case Szenarien

- Word-Dokument in LibreOffice öffnen, Kommentare verändern oder einfügen, wieder als Word-Dokument speichern.
Ergebnis: Kommentare sind weg, Dokument sieht anders aus als vorher, obwohl alle bekannten Spezifikationen eingehalten wurden.
→ Probleme mit proprietären Erweiterungen
- Dokumente lassen sich in X Jahren mit der „völlig neuen und verbesserten“ Version eines proprietären Programms nicht mehr lesen.
→ *Neal Stephenson: „Die Diktatur des schönen Scheins“, ISBN-13: 978-344215177*



Skizzen/Bilder

- **xfig** – Zeichenprogramm ursprünglich für LaTeX, kann auch EPS/PDF/SVG Vektorsupport. Absolut primitiv. Reicht oft völlig aus.
- **inkscape** – Profi-Vektorzeichenprogramm mit Gradienten und 3D-Effekten
- **gimp** – Universal-Grafikprogramm, tausende von Funktionen, gut für Bildkonvertierung und Fotobearbeitung
- **argouml, dia, yaoquiang** – Spezielle UML/BPMN-Zeichenprogramme
- **libreoffice Draw** – Ähnlich einfach gehalten wie Xfig, aber mit vielen Erweiterungen wie Beschriftung, klebenden Pfeilen und Diagrammelementen aus EPK und ERM.



OSS-freundliche Peripherie

- Viele Grafiktablets und Zeichengeräte mit USB-Anschluss funktionieren ohne zusätzliche „Treiber“ auf Anhieb, leider verrät der Verkäufer nicht immer, welche kompatiblen Chipsätze vorhanden sind. → „Googlen“ oder genau auf der Packung nachschauen.
- DVB-T, Webcams etc.: dito.
- Moderne Videokameras, die auf Festplatte oder SD-Karte aufzeichnen, sind problemlos als „USB-Festplatte“ auslesbar.
- Generelle Regel: Kontrollieren, ob ein Standard-Protokoll verwendet wird (z.B. Bluetooth-/USB-HID, uvcvideo, MPEG-Video, JPEG)
- Sind unter Windows proprietäre Treiber erforderlich, hat man oft auch dort nicht lange Freude am Gerät.



Video-Konvertierung

- Universaltool **ffmpeg** (alias **avconv**)
- Syntax:
avconv -i Original.xxx [Optionen] Ausgabe.xxx
- Beispiel: Film von DVD in „kompatibles DivX“ wandeln

```
mpplayer -dumpstream -dumpfile film1.vob dvd:///1
```

```
avconv -i film1.vob -f avi  
-c:v libxvid -vtag DX50 -b:v 2500k  
-trellis 2 -cmp 2 -subcmp 2  
-c:a libmp3lame -b:a 192k film1.avi
```

- Ergebnis: AVI-Container mit Videostream im DivX5-Codec und Audiodaten im MP3-Codec
- Nachteil: Bei jeder Konvertierung wird die Kopie (ggf. unmerklich) „schlechter“



Video-Schnitt ohne Videoschnittprogramm

- Möchte man nur den „Container“ wechseln oder das Video „trimmen“, den Codec aber beibehalten, helfen die „copy“-Funktionen → Kein Qualitätsverlust!

```
avconv -i film.mpg -f avi  
-acodec copy -vcodec copy  
-aspect 16:9  
-ss hh:mm:ss[.frm] -t hh:mm:ss[.xxx]  
film-geschnitten.avi
```

- Die interaktive Variante: **avidemux**
- Für Profis, mit Effekten: **openshot, kino, lives**



Lehrvideos auf dem Handy?!

- Problem: Unterschiedliche Auflösungen, unterschiedliche „optimale“ Videoformate.
- MP4+AAC können fast alle, aber ggf. Framerate niedriger setzen!
- Beispiel für ein etwas älteres Modell mit 16:9 Bildformat:

```
avconv -i video.mpg -f mp4 -qscale 4  
-vcodec mpeg4 -r 15 -ab 64k  
-s 256x144 video.mp4
```

- Smartphones haben meist genug CPU-Power, um auch höhere Auflösungen in Echtzeit umzurechnen.
- Youtube konvertiert automatisch.



Ende

