

# Volume 1 – Vidéo sous linux

Installation des outils vidéo

V8.0 du 7 janvier 2023

Par Olivier Hoarau ([olivier.hoarau@funix.org](mailto:olivier.hoarau@funix.org))

# Vidéo sous linux

- Volume 1 - Installation des outils vidéo**
- Volume 2 - Tutoriel Kdenlive**
- Volume 3 - Tutoriel Cinelerra**
- Volume 4 - Tutoriel OpenShot Video Editor**
- Volume 5 - Tutoriel Shotcut**
- Volume 6 – Tutorial Flowblade Movie Editor**
- Volume 7 – Tutorial Pitivi Video Editor**

## Table des matières

<b>1</b>	<b>HISTORIQUE DU DOCUMENT.....</b>	<b>5</b>
<b>2</b>	<b>PRÉAMBULE ET LICENCE.....</b>	<b>6</b>
<b>3</b>	<b>PRÉSENTATION ET AVERTISSEMENT.....</b>	<b>6</b>
<b>4</b>	<b>DÉFINITIONS ET AUTRES NOTIONS VIDÉO.....</b>	<b>7</b>
4.1	CONTENEUR.....	7
4.2	CODEC.....	7
<b>5</b>	<b>LES OUTILS DE BASE POUR LA VIDÉO.....</b>	<b>8</b>
5.1	PRÉSENTATION.....	8
5.2	INSTALLATION DE BIBLIOTHÈQUES AUDIO.....	8
5.2.1	<i>Ogg vorbis</i> .....	8
5.2.1.1	Présentation.....	8
5.2.1.2	Installation.....	8
5.2.2	<i>Installation de FLAC</i> .....	9
5.2.3	<i>Installation de LAME</i> .....	10
5.2.4	<i>Installation de a52dec</i> .....	10
5.2.5	<i>Installation de faad</i> .....	10
5.2.6	<i>Installation de faac</i> .....	11
5.2.7	<i>installation de FDK AAC</i> .....	11
5.2.8	<i>Installation de vo-aacenc</i> .....	11
5.2.9	<i>Installation de Rubber band</i> .....	12
5.3	INSTALLATION DE CODECS ET OUTILS VIDÉO.....	12
5.3.1	<i>Installation de xvid</i> .....	12
5.3.2	<i>Installation de GPAC</i> .....	12
5.3.3	<i>Installation de libx264</i> .....	14
5.3.4	<i>Installation de libx265</i> .....	15
5.3.5	<i>Installation de mpeg2dec</i> .....	15
5.4	INSTALLATION DE CODECS ET OUTILS MULTIMÉDIA.....	15
5.4.1	<i>Installation d'OpenCV</i> .....	15
5.4.2	<i>Installation de frei0r</i> .....	19
5.4.3	<i>Installation de vid.stab</i> .....	20
5.4.4	<i>Installation de movit</i> .....	20
5.4.5	<i>Installation de ffmpeg</i> .....	21
<b>6</b>	<b>LES CONTENEURS VIDÉOS.....</b>	<b>24</b>
6.1	PRÉSENTATION.....	24
6.2	LE CONTENEUR AVI.....	24
6.3	OGM TOOLS.....	24
6.3.1	<i>Installation</i> .....	24
6.3.2	<i>Utilisation</i> .....	24
6.4	VORBIS TOOLS.....	25
6.5	MATROSKA.....	26
6.5.1	<i>Installation</i> .....	26
6.5.2	<i>Utilisation</i> .....	27

<b>7</b>	<b>LES LECTEURS VIDÉO SOUS LINUX.....</b>	<b>28</b>
7.1	INSTALLATION DES BIBLIOTHÈQUES DE BASE.....	28
7.1.1	<i>Installation de mad</i> .....	28
7.1.2	<i>Installation de libvcdread</i> .....	28
7.1.3	<i>Installation de libvcdnav</i> .....	29
7.1.4	<i>Installation de libvcdcss</i> .....	29
7.1.5	<i>Installation de libvcdpsi</i> .....	29
7.1.6	<i>Installation de live555</i> .....	29
7.2	LES LECTEURS MULTIFORMATS.....	30
7.2.1	<i>Présentation</i> .....	30
7.2.2	<i>VLC</i> .....	30
7.2.2.1	<i>Installation</i> .....	30
7.2.2.2	<i>Utilisation</i> .....	31
7.2.3	<i>Xine</i> .....	31
7.2.3.1	<i>Installation</i> .....	31
7.2.3.2	<i>Utilisation</i> .....	35
7.2.4	<i>MPlayer</i> .....	35
<b>8</b>	<b>STREAMING VIDÉO.....</b>	<b>38</b>
8.1	PRÉSENTATION.....	38
8.2	DIFFUSER UNE VIDÉO VERS UN POSTE EN PARTICULIER.....	38
8.3	FAIRE DU STREAMING SUR UN RÉSEAU.....	38
8.4	VIDÉO À LA DEMANDE (VoD).....	38
<b>9</b>	<b>LOGICIELS DE MONTAGE VIDÉO.....</b>	<b>39</b>
9.1	PRÉSENTATION.....	39
9.2	OUTILS DE MONTAGE VIDÉO.....	39
9.2.1	<i>KDEnlive</i> .....	39
9.2.1.1	<i>Présentation</i> .....	39
9.2.1.2	<i>Installation</i> .....	39
9.2.1.3	<i>Utilisation</i> .....	44
9.2.2	<i>Pitivi Video Editor</i> .....	44
9.2.2.1	<i>Installation</i> .....	44
9.2.2.2	<i>Utilisation</i> .....	44
9.2.3	<i>Flowblade Movie Editor</i> .....	45
9.2.3.1	<i>Installation</i> .....	45
9.2.3.2	<i>Utilisation</i> .....	46
9.2.4	<i>Shotcut</i> .....	46
9.2.4.1	<i>Installation</i> .....	46
9.2.4.2	<i>Utilisation</i> .....	47
9.2.5	<i>Cinelerra</i> .....	47
9.2.5.1	<i>Installation</i> .....	47
9.2.5.2	<i>Utilisation</i> .....	49
9.2.6	<i>Fork de Cinelerra</i> .....	50
9.2.6.1	<i>Présentation</i> .....	50
9.2.6.2	<i>Installation</i> .....	50
9.2.6.3	<i>Utilisation</i> .....	53
9.2.7	<i>OpenShot Video</i> .....	53
9.2.7.1	<i>Installation</i> .....	53
9.2.7.2	<i>Utilisation</i> .....	55
9.3	OUTILS DIVERS.....	55
9.3.1	<i>Synfig Studio</i> .....	55
9.3.1.1	<i>Installation</i> .....	55
9.3.1.2	<i>Utilisation</i> .....	56
9.3.2	<i>Blender</i> .....	56
<b>10</b>	<b>LES OUTILS DE TRANSCODAGE ET DE RIPPAGE DE DVD.....</b>	<b>57</b>
10.1	PRÉSENTATION.....	57
10.2	TRANSCODAGE AVEC AVIDEMUX.....	58
10.2.1	<i>Présentation</i> .....	58
10.2.2	<i>Installation</i> .....	58
10.2.3	<i>Utilisation</i> .....	58
10.3	COPIER DES DVD.....	60
10.3.1	<i>Mencoder</i> .....	60
10.3.2	<i>HandBrake</i> .....	66
10.3.2.1	<i>Installation</i> .....	66
10.3.2.2	<i>Utilisation</i> .....	67



# 1 Historique du document

- 07/01/2023 V8.0 - chapitre outils de base pour la vidéo changement de version de flac, faad, rubberband, gpac, libx265, opencv, libx265, frei0r, movit et ffmpeg  
- chapitre conteneur vidéo, changement de version de libmatroska, mkvtoolnix  
- chapitre lecteurs vidéo, changement de version de libdvdread, vlc et Mplayer  
- chapitre montage vidéo, changement de version de mlt, kdenlive, openshot video editor, cinelerra, cinelerra GG, blender et synfig studio  
- chapitre transcodage, changement de version d'avidemux et HandBrake
- 26/12/2021 V7.0 - chapitre outils de base pour la vidéo, passage à libogg 1.3.5, libvorbis 1.3.7, flac 1.3.3, faad 2.10.0 fdk\_aac 2.0.2, xvid 1.3.7, gpac 1.0.1, libx265 3.4, opencv 4.5.4, frei0r last, libx264 last, ffmpeg 4.4.1, rajout de rubberband et suppression de libtheora et swfdec-gtk  
- chapitre conteneur vidéo, passage à vorbis tools 1.4.2, mkvtoolnix 63.0.0, libmatroska 1.6.3  
- chapitre lecteurs vidéos, passage à libdvdread 6.1.2, libdvdnav 6.1.1, vlc 3.0.16 et xine-lib 1.2.11  
- chapitre montage vidéo, passage à mlt 7.4.0, kdenlive 21.12, cinelerra 7.4, openshot video 2.6.1, synfig studio 1.5.1, blender 2.93.5, rajout de pitivi, shotcut et flowblade  
- chapitre transcodage, passage à avidemux 2.7.8 et handrake 1.4.2
- 20/3/2020 V6.3 - chapitre outils de base pour la vidéo, passage à x265 3.3, OpenCV 4.2.0 et ffmpeg 4.2.2  
- chapitre conteneur vidéo, passage à mkvtoolnix 44.0.0  
- chapitre montage vidéo, passage à mlt 6.20.0, kdenlive 19.12.3, cinelerra GG 5.1.20200229, OpenShot Video Editor 2.5.1, Synfig Studio 1.3.13 et blender 2.82a  
- chapitre lecteurs vidéo, passage à xine lib 1.2.10 et xine ui 0.99.12  
- chapitre transcodage, passage à HandBrake 1.3.1
- 1/6/2019 V6.2 - chapitre outils de base pour la vidéo, passage à faad2 2.8.8, x264-snapshot-20190507-2245, OpenCV 4.1.0, et ffmpeg 4.1.3  
- chapitre conteneur vidéo, passage à mkvtoolnix 34.0.0  
- chapitre montage vidéo, passage à mlt 6.16.0, kdenlive 19.04.1, cinelerra GG 5.1.20190430, OpenShot Video Editor 2.4.4 et Synfig Studio 1.3.11  
- chapitre lecteurs vidéo, passage à live 2019.05.29 et MPlayer 1.4.0  
- chapitre transcodage, passage à avidemux 2.7.3 et HandBrake 1.2.2
- 12/2/19 V6.1 - chapitre outils de base pour la vidéo, passage à fdk\_aac 2.0.0, x264-snapshot-20190208-2245, x265 3.0, OpenCV 4.0.1, et ffmpeg 4.1  
- chapitre conteneur vidéo, passage à mkvtoolnix 30.1.0  
- chapitre montage vidéo, passage à mlt 6.12.0 et kdenlive 18.12.2, LiVES 2.10.2, cinelerra 7.1 et cinelerra GG 5.1.20190131  
- chapitre lecteurs vidéos, passage à vlc 3.0.6  
- chapitre transcodage, passage à Handbrake 1.2.0
- 12/10/18 V6.0 - chapitre outils de base pour la vidéo, passage à libvorbis 1.3.6, faac 1\_29\_9\_2, fdk\_aac 0.1.6, x264-snapshot-20181006-2245, x265 2.9, movit 1.6.2 et ffmpeg 4.0.2  
- chapitre conteneur vidéo, passage à libebml 1.3.6, libmatroska 1.4.9 et mkvtoolnix 27.0.0  
- chapitre montage vidéo, passage à mlt 6.10.0 et kdenlive 18.08.1, LiVES 2.10.0, cinelerra GG 5.1.20180930 et OpenShot Video 2.4.3  
- chapitre lecteurs vidéo, passage à libdvdcss 1.4.2, live 2018.09.18 et VLC 3.0.4  
- chapitre transcodage, passage à avidemux 2.7.1 et HandBrake 1.1.2
- 22/10/17 V5.9 - chapitre outils de base pour la vidéo, passage à lame 3.100, faac 1.29.9.2, xvid 1.3.5, x264-snapshot-20171221-2245, x265 2.6, opencv 3.3.1 et ffmpeg 3.4.1  
- chapitre conteneur vidéo, passage à mkvtoolnix 19.0.0  
- chapitre outils multimedia, passage à audacious 3.9  
- chapitre montage vidéo, passage à kdenlive 17.12.0, cinelerra 7 et Openshot Video 2.4.1  
- chapitre mediacenter, passage à kodi krypton 17.6  
- chapitre lecteurs vidéo, passage à VLC 2.2.8  
- chapitre transcodage, présentation d'une fonctionnalité de VLC pour copier des DVD
- 9/10/17 V5.8 - chapitre outils de base, passage à gpac 0.7.1, x264-snapshot-20171007-2245, movit 1.5.3, opencv 3.3.0, ffmpeg 3.3.4  
- chapitre conteneur vidéo, passage à libmatroska 1.4.8 et mkvtoolnix 16.0.0  
- chapitre montage vidéo, passage à kdenlive 17.08.1, Openshot Video 2.4.0, Synfig Studio 1.2.1 et blender 2.79
- 17/07/17 V5.7  
1 - chapitre outils de base pour la vidéo passage à faad 2.8.0, x264-snapshot-20170715-2245, x265 2.5, frei0r 1.6.1, vid.stab 1.1.0, movit 1.5.1, eigen 3.3.4 et libepoxy 1.4.3  
- chapitre conteneur vidéo, passage à mkvtoolnix 13.0.0  
- chapitre lecteur vidéo, passage à live.2017.06.04, passage à vlc 2.2.6 et xine-lib-1.2.8  
- chapitre montage vidéo, passage à kdenlive 17.04.3, LiVES 2.8.7, Openshot Video 2.3.4 et blender 2.78c
- 07/05/17 V5.7 - chapitre outils de base pour la vidéo, passage à gpac 0.7.0, x264-snapshot-20170505-2245, x265 2.4, frei0r 1.6.0 et movit 1.5.0  
- chapitre transcodage vidéo, passage à avidemux 2.6.20 et HandBrake 1.0.7  
- chapitre conteneur vidéo, passage à libmatroska 1.4.7 et mkvtoolnix 11.0.0  
- chapitre lecteur vidéo, passage à libdvpsi 1.3.1 quelques précisions pour l'installation de la version en développement 3.0.0 de vlc  
- chapitre montage vidéo, passage à kdenlive 17.04.0, LiVES 2.8.5, Openshot Video 2.3.1 et blender 2.78c
- 21/02/17 V5.6  
1 - chapitre outils de base passage à flac 1.3.2, présentation du codec audio FDK AAC, passage à x264-snapshot-20170210-2245, x265 2.2, OpenCV 3.2.0, movit 1.4.0 et ffmpeg 3.2.4, quelques modifications dans la compilation pour utiliser l'accélération matérielle et le calcul parallèle à base de GPU nvidia  
- chapitre transcodage passage à avidemux 2.6.18 et HandBrake 1.0.2  
- chapitre passage à mkvtoolnix 9.8.0  
- chapitre montage vidéo passage à mlt 6.4.1, kdenlive 16.12.2, LiVES 2.8.4, cinelerra 6, Openshot Video 2.2, SynfigStudio 1.2.0 et blender 2.78b
- 20/12/16 V5.6 - chapitre outils de base pour la vidéo, passage à x264-snapshot-2016121907-2245, opencv 3.1.0, frei0r-plugin 1.5, movit 1.4.0 et ffmpeg 3.2.2, suppression de libquicktime et mjpegtools obsolètes  
- chapitre transcodage vidéo, passage à avidemux 2.6.15 et HandBrake 0.10.5  
- chapitre conteneur vidéo, passage à libmatroska 1.4.5 et mkvtoolnix 9.6.0  
- chapitre lecteur vidéo, passage à libdvdread 5.0.3, vlc 2.2.4 et MPlayer 1.3.0  
- chapitre montage vidéo, passage à kdenlive 16.12.0, LiVES 2.8.2, OpenShot Video 2.1 et blender 2.78a  
- chapitre outils multimedia pour mediacenter, passage à audacious 3.8,1 et kodi 16.1

08/02/16 V5.5 - chapitre outils de base pour la vidéo, passage à xvid 1.3.4, x264 20160205-2245, x265 1.9, opencv 3.0.0 et ffmpeg 2.8.6  
- chapitre transcodage vidéo, passage à HandBrake 0.10.3  
- chapitre conteneur vidéo, passage à Matroska 1.4.4 mkvtoolnix 8.8.0  
- chapitre lecteur vidéo, passage à libdvdcss 1.4.0, libdvdpsi 1.3.0 et MPlayer 1.2.1  
- chapitre montage vidéo, passage à mlt 0.9.8, kdenlive 15.12.1, LiVES 2.6.0, cinelerra 4.6.1, synfigstudio 1.0.2 et blender 2.76b  
- suppression du chapitre sur la gestion des caméscopes minDV, plus personne ne n'utilise, c'est devenu obsolète et ce n'est plus mis à jour  
....  
16/12/02 V1.0 Création du document.

## 2 Préambule et licence

Ce document présente les outils de visualisation de la télé, d'acquisition, de lecture et de montage vidéo sous Linux.

La dernière version de ce document est téléchargeable à l'URL <https://www.funix.org>.

Ce document est sous licence Creative Commons Attribution-ShareAlike 4.0 Unported, le détail de la licence se trouve sur le site <http://creativecommons.org/licenses/by-sa/4.0/legalcode>. Pour résumer, vous êtes libres

- de reproduire, distribuer et communiquer cette création au public
- de modifier cette création

suivant les conditions suivantes:

- **Paternité** — Vous devez citer le nom de l'auteur original de la manière indiquée par l'auteur de l'oeuvre ou le titulaire des droits qui vous confère cette autorisation (mais pas d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent votre utilisation de l'oeuvre).
- **Partage des Conditions Initiales à l'Identique** — Si vous transformez ou modifiez cette oeuvre pour en créer une nouvelle, vous devez la distribuer selon les termes du même contrat ou avec une licence similaire ou compatible.

Par ailleurs ce document ne peut pas être utilisé dans un but commercial sans le consentement de son auteur. Ce document vous est fourni "dans l'état" sans aucune garantie de toute sorte, l'auteur ne saurait être tenu responsable des quelconques misères qui pourraient vous arriver lors des manipulations décrites dans ce document.

## 3 Présentation et avertissement

Ce document n'a pas pour objet d'être un cours magistral et pointu sur la vidéo et encore moins un mode d'emploi pour le montage. Le but de ce document est seulement de vous présenter l'installation d'un ensemble d'outils vous permettant de travailler la vidéo sous Linux. Une présentation très succincte de chaque outil est également proposée.

Ce document se veut le plus didactique possible pour qu'il puisse être utile à un maximum de personnes, novices comme gourous ! Cependant il ne prétend être un livre de recette qu'on suit à la lettre. Dans la plupart des cas ce sera le cas, mais il m'a parfois été nécessaire de réaliser certaines opérations, parfois peu orthodoxes (modification du code), pour pouvoir installer tel ou tel logiciel. En fonction de votre configuration logicielle, vous ne rencontrerez peut être pas de difficultés ou, au contraire, en rencontrerez là où je n'en ai pas eu.

J'ai essayé de détailler au possible les opérations réalisées pour arriver à compiler un logiciel, cela vous sera utile si vous rencontrez les mêmes difficultés ou si ce n'est pas le cas pour comprendre la méthodologie à adopter pour résoudre un problème de compilation.

Les machines de test utilisées pour écrire ce document tourne avec une distribution Mageia. Ce document est donc orienté Mageia, mais il s'applique aussi à d'autres distributions étant donné que je présente l'installation de logiciels en utilisant la compilation des sources et que l'arborescence et les fichiers système varient peu d'une distribution à une autre.

Si vous éprouvez des difficultés de compréhension, rencontrez des problèmes de compilation inattendus, ou d'une manière générale si vous avez une remarque qui pourrait contribuer à améliorer ce document, n'hésitez pas à en faire part à l'auteur ([olivier.hoarau@funix.org](mailto:olivier.hoarau@funix.org)).

Avant de rentrer dans le vif du sujet, voici une présentation du plan suivi par ce document :

- Définitions et autres notions vidéo : quelques pré requis nécessaires pour pouvoir aller plus loin.
- Les outils de base en vidéo : présentation et installation des outils de bas niveau gérant la vidéo.
- Les conteneurs vidéo.
- Les lecteurs vidéo : présentation des logiciels de lecture de fichiers vidéo (DVD compris).
- Montage vidéo : présentation de logiciel de haut niveau réalisant le montage de vidéo et l'export dans une grande variété de format.
- Conversion de format vidéo (transcodage): ce chapitre traite des logiciels permettant de convertir un format vidéo en un autre.

## 4 Définitions et autres notions vidéo

### 4.1 Conteneur

Les principaux types de formats de fichier vidéo sont mp4, avi, ogm, et matroska. Ces fichiers ne sont en fait que des enveloppes (ou conteneurs) pour du flux compressé vidéo et audio. Il existe plusieurs moyens de compresser ces données, on appellera codec le logiciel de compression et de décompression chargé de lire ces données. Un fichier avi peut donc très bien contenir un flux vidéo avec un codec xvid ou bien encore mpeg2.

Chaque conteneur possède en entête les informations qui décrivent les flux qu'il contient, par opposition aux fichiers .mpeg qui ne peuvent contenir que du mpeg. Le conteneur .avi est encore un des plus utilisés, il est cependant limité, on ne peut pas mettre du flux ogg, ou bien encore plusieurs pistes sons et chapitrage comme pour les DVD. C'est pourquoi des nouveaux conteneurs ont fait leur apparition pour offrir de nouvelles possibilités, je pense notamment au conteneur ogm (<http://www.bunkus.org/videotools/ogmtools/index.html>), matroska (<http://www.matroska.org>) et bien sûr [mp4](#).

### 4.2 Codec

Un certain nombre de codecs sont directement inspirés par un organisme de standardisation appelé MPEG pour Motion Picture Expert Group. Il a mis au point la norme MPEG1 pour la réalisation des disques laser vidéo. Le MPEG2 a été mis au point pour la télévision numérique. Le MPEG3 est mort né, car ses fonctionnalités ont été reprises pour une grande part avec le développement de MPEG2. Le MPEG4 permet la diffusion de la vidéo avec un support de faible débit.

Les codes les plus anciens et les plus universels sont:

- VCD compatible norme MPEG1, résolution 352\*288, 25img/s PAL/SECAM, débit 1123kb/s, qualité légèrement inférieur à une K7 VHS
- XVCD compatible norme MPEG1 , résolution 352\*576 ou 352\*288, 25img/s PAL/SECAM, débit jusqu'à 2500kb/s
- SVCD compatible norme MPEG2, résolution 480\*576, 25img/s PAL/SECAM, débit 2520kb/s, qualité SVHS
- DVD compatible norme MPEG2, résolution 720\*576, 25img/s PAL/SECAM, débit de 28400Kb/s qualité DV supérieure

Le codec VCD est le plus universel, il peut être lu par tous les logiciels de lecture vidéo sur PC (et autres) et par tous les lecteurs DVD de salon. Le codec SVCD peut être par certains logiciels de lecture et quelques lecteurs DVD de salon. Le codec DVD est évidemment lu par les lecteurs DVD de salon.

Les codecs vidéo les plus récents et aujourd'hui les plus utilisés sont:

- Le x264 qui repose sur la norme [MPEG4 AVC/H264](#) est actuellement le codec le plus en vogue et le plus populaire, compact tout en préservant une très bonne qualité d'image, pour donner une idée pour un fichier de qualité DVD la taille est deux fois plus petite.

- Le x265 qui repose sur la norme [MPEG4 HEVC/H265](#) qui devrait remplacer à terme le x264, il améliore sensiblement le codage/compression par rapport à AVC, en d'autres termes, les fichiers obtenus sont beaucoup plus compacts pour une qualité identique.
- [Theora](#) est un codec vidéo totalement libre et dénué de brevets, il n'est pas totalement compatible à la norme MPEG4 ce qui fait que son utilisation n'entraîne pas le paiement d'une redevance au consortium MPEG.
- DivX est un codec tombé aujourd'hui en désuétude, c'est une implémentation d'une version non définitive des normes MPEG4 H263 et H264. Ce n'est pas d'un codec libre (au sens OpenSource), il est développé par DivXNetwork. Il est apparu il y a quelques années sous le terme DivX;-) dans sa version 3.11, en creusant un peu on s'est rendu compte que c'était une version piratée d'un codec Microsoft (MS-MPEG4-v3, qui porte mal son nom puisqu'il n'est pas compatible à la norme MPEG4). Du coup DivXNetwork a été contraint à développer son propre codec, compatible à la norme MPEG4, et en plus a décidé de le rendre OpenSource, de là est né OpenDivX. Son existence a été plus qu'éphémère, il a très vite laissé la place au codec DivX 4.0 qui lui n'avait rien d'OpenSource. Cependant les sources d'OpenDivX ont été la base du développement d'un codec compatible MPEG4, totalement OpenSource, en l'occurrence XviD (l'opposé de DivX...). En résumé Xvid est un codec compatible MPEG4 sous licence GNU GPL (OpenSource) et DivX (4.0 et 5.0) est un codec qui n'est pas totalement compatible MPEG4 et qui n'est pas libre.

Les codes audio les plus courant sont:

- Le codec audio [MP3](#) est encore très utilisé, c'est la spécification audio des normes MPEG1 et MPEG2, les puristes trouveront qu'il dégrade la qualité audio, il existe maintenant d'autres codes permettant de mieux préserver la qualité audio.
- Le codec audio [AAC](#) fait partie de ces nouveaux codecs, on dit qu'un fichier codé en 128kb/s en AAC est équivalent à un fichier MP3 codé à 256kb/s.
- Le codec audio [Vorbis](#) qui contrairement aux deux précédents codecs est complètement libre et dénué et de brevets, il est plus performant que le MP3 mais reste largement moins populaire à tort.

## 5 Les outils de base pour la vidéo

### 5.1 Présentation

Les outils présentés dans ce paragraphe sont nécessaires pour le fonctionnement de la plupart des interfaces graphiques traitant de la vidéo, que ce soit des outils pour l'acquisition, le montage, la conversion et tout simplement la lecture vidéo.

Notez bien que j'ai pris le parti de présenter l'installation de ces outils à partir des sources pour pouvoir m'adapter à toutes les distributions linux (et accessoirement pour avoir un code optimisé et pouvoir choisir éventuellement mes options de compilation). La plupart des distributions proposent déjà ces outils en packages précompilés, libre à vous d'utiliser votre "package manager" préféré (urpmi, apt-get, ...) pour les installer. Cette remarque est valable pour tous les autres paragraphes.

### 5.2 Installation de bibliothèques audio

#### 5.2.1 Ogg vorbis

##### 5.2.1.1 Présentation

**Ogg Vorbis** est un nouveau format de compression audio. Il est comparable à d'autres formats équivalents comme le MP3, VQF ou AAC à la différence qu'il est complètement libre. Il est généralement considéré comme plus performant que MP3. Ogg est en fait le nom d'un conteneur qui peut contenir aussi bien de l'audio que de la vidéo, alors que vorbis est l'algorithme de compression utilisé pour l'audio inclus dans un conteneur ogg. Notez bien que le conteneur ogg peut contenir de l'audio qui utilise un moyen de compression différent du vorbis (FLAC par exemple).

Le site officiel de **Ogg Vorbis** est <http://www.vorbis.com/> on y récupérera les bibliothèques **libogg** et **libvorbis** ainsi que la bibliothèque audio **libao**.

##### 5.2.1.2 Installation

On décompresse la première archive en tapant



**tar xvfz libao-1.2.0.tar.gz**

Cela donne le répertoire **libao-1.2.0**. On veillera à ce que les packages **lib64x11-devel**, **lib64alsa2-devel** et **pulseaudio-devel** soient installés, on tape maintenant successivement

```
./configure  
make
```

Puis en tant que root

```
make install
```

Dans le fichier **/etc/ld.so.conf** on rajoutera les lignes suivantes (si ce n'est déjà fait)

```
/usr/local/lib  
/usr/local/lib/ao/plugins-4
```

On tape alors

```
ldconfig
```

Passons à **libogg**, on décompresse l'archive en tapant

```
tar xvfz libogg-1.3.5.tar.gz
```

Cela donne le répertoire **libogg-1.3.5** dans lequel on tape successivement

```
./configure  
make
```

Puis en tant que root

```
make install
```

Passons à **libvorbis**, on décompresse l'archive en tapant

```
tar xvf libvorbis-1.3.7.tar.xz
```

Cela donne le répertoire **libvorbis-1.3.7** dans lequel on tape successivement

```
./configure  
make
```

puis en tant que root

```
make install  
ldconfig
```

## 5.2.2 Installation de FLAC

**FLAC** (Free Lossless Audio Codec) est un algorithme de compression qui est similaire au MP3 à la différence que la compression s'effectue sans perte de qualité. Son site officiel est <https://xiph.org/flac/> on y récupérera l'archive qu'on décompresse en tapant:

```
tar xvf flac-1.4.2.tar.xz
```

Cela donne le répertoire **flac-1.4.2**. Dans le répertoire de **flac** on tape

```
./configure  
make
```

puis en tant que root

**make install**

Vérifiez que la ligne `/usr/local/lib` soit bien dans le fichier `/etc/ld.so.conf` puis tapez

**ldconfig**

### 5.2.3 Installation de LAME

LAME qui signifie Ain't an MP3 Encoder se trouve à l'URL <http://www.mp3dev.org> et plus précisément ici <http://sourceforge.net/projects/lame/>. Comme son nom l'indique c'est un encodeur MP3, il est utilisé notamment par les logiciels pour construire des avi (avec une piste audio en mp3).

On décompresse le tarball en tapant :

**tar xvfz lame-3.100.tar.gz**

Cela donne le répertoire **lame-3.100** dans lequel on tape successivement

**./configure**

**make**

Puis en tant que root

**make install**

### 5.2.4 Installation de a52dec

La bibliothèque **a52** fournit des fonctions pour décoder du flux A/52. Kezako flux A/52 ? C'est le flux audio utilisé par le DVD, il est connu aussi sous le terme AC-3, vous trouverez davantage d'informations dans ce document [http://www.atsc.org/standards/a\\_52a.pdf](http://www.atsc.org/standards/a_52a.pdf). Cette bibliothèque est nécessaire pour pouvoir lire les DVD. L'URL officiel est <http://liba52.sourceforge.net/> on y récupère l'archive qu'on décompresse en tapant

**tar xvfz a52dec-0.7.4.tar.gz**

Cela donne le répertoire **a52dec-0.7.4** dans lequel on tape

**./configure --with-pic --enable-shared --disable-static**

Les options **--with-pic --enable-shared --disable-static** sont utiles en mode 64bits pour la compilation de **ffmpeg** et **transcode** et ce n'est pas suffisant, il faut éditer tous les fichiers **Makefile** et à la place de

**CC = gcc**

on rajoute

**CC = gcc -fPIC**

on tape **make** puis en tant que root

**make install**

**ldconfig**

### 5.2.5 Installation de faad

**Faad** est un codec audio compatible AAC (Advanced Audio Coding). L'AAC est un format de compression audio comparable à MP3, qui a été développé dans le cadre du standard MPEG2 puis MPEG4, il se pose comme le successeur du MP3 qui a été développé auparavant pour le MPEG2. Les oreilles fines pourront noter qu'un fichier AAC codé à 96kbps a une qualité sonore meilleure qu'un fichier mp3 codé à 128 kbps. L'URL officiel est <https://github.com/knik0/faad2/releases> on y récupérera l'archive qu'on décompresse en tapant

**tar xvfz faad2-2.10.1.tar.gz**

Cela donne le répertoire **faad2-2.10.1** dans lequel on tape

```
./bootstrap
./configure --with-drm
make
```

et enfin en tant que root

```
make install
ldconfig
```

## 5.2.6 Installation de faac

La bibliothèque **FAAC** fournit des encodeurs MPEG-4 and MPEG-2 AAC. Le site officiel est <https://github.com/knik0/faac/releases> on y récupère l'archive qu'on décompresse en tapant

```
tar xvfz faac-1_30.tar.gz
```

cela donne le répertoire **faac-1\_30** dans lequel on tape

```
./bootstrap
./configure
make
```

et enfin en tant que root

```
make install
ldconfig
```

## 5.2.7 installation de FDK AAC

**FDK AAC** est une bibliothèque qui supporte différents profils du codec audio AAC, il est considéré comme un des meilleurs encodeurs AAC. Le site officiel est <https://github.com/mstorsjo/fdk-aac> où on récupérera la dernière version qu'on décompressera en tapant

```
tar xvfz fdk-aac-2.0.2.tar.gz
```

cela donne le répertoire **fdk-aac-2.0.2** dans lequel on tape

```
./autogen.sh
./configure
make
```

puis en tant que root

```
make install
```

## 5.2.8 Installation de vo-aacenc

Cette bibliothèque intègre une implémentation du codec audio AAC (Advanced Audio Coding). Le site officiel est <http://sourceforge.net/projects/opencore-amr/files/vo-aacenc/> on y récupère l'archive qu'on décompresse en tapant

```
tar xvfz vo-aacenc-0.1.3.tar.gz
```

Cela donne le répertoire **vo-aacenc-0.1.3** dans lequel on tape

```
./configure
make
```

puis en tant que root

```
make install
```

## 5.2.9 Installation de Rubber band

La bibliothèque **Rubber band** est nécessaire pour certains effets audio qu'on utilise notamment l'effet pitch compensation de [kdenlive](http://kdenlive.org). Le site officiel est <https://breakfastquay.com/rubberband/> on installera d'abord le package **lib64vamp-plugin-sdk-devel** et on décompresse l'archive en tapant :

```
tar xvfj rubberband-3.1.2.tar.bz2
```

cela va donner le répertoire **rubberband-3.1.2** dans lequel on va taper :

```
meson build && ninja -C build
```

et en tant que root

```
ninja -C build/ install
```

```
ldconfig
```

## 5.3 Installation de codecs et outils vidéo

### 5.3.1 Installation de xvid

**xvid** est un codec compatible MPEG4 comme DivX à la différence qu'il est complètement libre. Le site officiel est [www.xvid.org](http://www.xvid.org), on récupère la dernière archive stable qu'on décompresse en tapant

```
tar xvfz xvidcore-1.3.7.tar.gz
```

Cela donne le répertoire **xvidcore**. On veillera à ce que le package **nasm** soit installé. Au niveau de **xvidcore/build/generic** on tape

```
./configure
```

Puis

```
make
```

Puis en tant que root

```
make install
```

j'ai eu cette erreur là

```
In: impossible de créer le lien symbolique « /usr/local/lib/libxvidcore.so.4 »: Le fichier existe
Makefile:144: recipe for target 'install' failed
```

j'ai donc dû supprimer les liens existants

```
unlink /usr/local/lib/libxvidcore.so.4
```

```
unlink /usr/local/lib/libxvidcore.so
```

et relancer **make install**, il faut taper ensuite **ldconfig**

### 5.3.2 Installation de GPAC

**GPAC** est une implémentation de la norme MPEG-4 utilisée par d'autres outils de plus haut niveau, le site officiel est <https://gpac.wp.imt.fr/> on y récupère l'archive qu'on décompresse en tapant

```
tar xvfz gpac-2.2.0.tar.gz
```

cela donne le répertoire **gpac-2.2.0** dans lequel on tape

**./configure --enable-pic**

cela donne

**\*\* System Configuration**

**Install prefix:** /usr/local  
**Source path:** /usr/local/linux/multimedia/gpac-2.2.0  
**C compiler:** gcc  
**C++ compiler:** g++  
**make:** make  
**CPU:** x86\_64  
**Big Endian:** no

**\*\* GPAC 2.2 rev Core Configuration \*\***

**Static Modules:** no  
**debug version:** no  
**GProf enabled:** no  
**Memory tracking enabled:** no  
**Sanitizer enabled:** no  
**Fixed-Point Version:** no  
**IPV6 Support:** yes  
**QuickJS Support:** yes (with qjs-libc)

**\*\* Detected libraries \*\***

**zlib:** system (pkgconfig)  
**OpenGL support:** yes  
**OpenSSL support:** system (pkgconfig)  
**nghttp2:** system (pkgconfig)  
**OSS Audio:** yes  
**ALSA Audio:** yes  
**Jack Audio:** yes  
**Pulse Audio:** yes  
**DirectFB:** no  
**X11:** yes (path: /usr/X11R6)  
**X11 Shared Memory support:** yes  
**X11 XVideo support:** yes  
**SDL:** yes  
**DVB for Linux:** yes  
**FreeType:** system (pkgconfig)  
**JPEG:** system (pkgconfig)  
**OpenJPEG:** system (pkgconfig)  
**PNG:** system (pkgconfig)  
**MAD:** system  
**FAAD:** system (pkgconfig)  
**XVID:** system  
**FFMPEG:** system (pkgconfig) - VVC supported  
**LZMA:** yes  
**Xiph OGG:** system (pkgconfig)  
**Platinum UPnP:** no  
**Xiph Vorbis:** system (pkgconfig)  
**Xiph Theora:** system (pkgconfig)  
**A52 (AC3):** system  
**OpenSVCD decoder:** no  
**OpenHEVC decoder:** no  
**Freenect:** no

**Creating config.mak**

**Check config.log for detection failures**

**Done - type 'make help' for make info, 'make' to build**

on tape ensuite

**make**

puis **make install**

j'ai copié ensuite les fichiers suivants

```
cp -Rf include/gpac/ /usr/local/include
cp bin/gcc/libgpac_static.a /usr/local/lib
```

dans le fichier `/usr/local/include/gpac/configuration.h` j'ai mis en commentaire la ligne 207 comme ceci

```
///error "Unknown target platform used with static configuration file"
```

### 5.3.3 Installation de libx264

**libx264** est une bibliothèque permettant d'encoder du flux vidéo dans le format [H.264/MPEG-4 AVC](http://www.videolan.org/developers/x264.html) on récupère l'archive par ici <http://www.videolan.org/developers/x264.html> on la décompresse en tapant

```
tar xvfj x264-master.tar.bz2
```

cela donne le répertoire **x264-master** dans lequel on tape

```
./configure --enable-pic --enable-shared
```

il faudra sans doute installer préalablement **yasm**. A noter que les options **--enable-pic --enable-shared** sont utiles pour les configurations 64bits pour compiler **ffmpeg**.

on revient à **libx264** et on peut taper maintenant la commande **configure** avec les options décrites plus haut et voilà le résultat

```
platform:  X86_64
byte order: little-endian
system:    LINUX
cli:       yes
libx264:   internal
shared:    yes
static:    no
bashcompletion: yes
asm:       yes
interlaced: yes
avs:       yes
lavf:      yes
ffms:      no
mp4:       gpac
gpl:       yes
thread:    posix
opencl:    yes
filters:   resize crop select_every
lto:       no
debug:     no
gprof:     no
strip:     no
PIC:       yes
bit depth: all
chroma format: all
```

puis

```
make
```

puis en tant que root

```
make install
```

**ldconfig**

### 5.3.4 Installation de libx265

[HEVC](#) ou High Efficiency Video Coding qu'on appelle également H.265 est le codec qui est censé remplacer à terme [H.264](#). Il est donné pour donner la même qualité mais avec un fichier deux fois plus petit, ou si vous préférez une qualité supérieure avec un fichier de taille identique. Le site officiel est [x265.org](#). Il faut s'enregistrer pour télécharger, ça me plaît pas trop donc j'ai récupéré la dernière archive par [https://bitbucket.org/multicoreware/x265\\_git/downloads/](https://bitbucket.org/multicoreware/x265_git/downloads/)

On y récupérera l'archive qu'on décompressera en tapant:

```
tar xvfz x265_3.5.tar.gz
```

cela donne le répertoire **x265\_3.5** dans lequel on tape

```
cd build  
cmake ../source  
make
```

puis en tant que root

```
make install
```

### 5.3.5 Installation de mpeg2dec

**mpeg2dec** est un décodeur vidéo compatible MPEG2, son site officiel est <http://libmpeg2.sf.net> dans lequel on récupère l'archive qu'on décompresse en tapant

```
tar xvfz mpeg2dec-0.4.1.tar.gz
```

Cela donne le répertoire **mpeg2dec-0.4.1** on installe ensuite le package **lib64xv-devel** puis on tape successivement

```
./configure --with-pic
```

l'option **with-pic** est nécessaire pour la compilation pour les plateformes 64 bits, mais ce n'est pas suffisant il faudra éditer tous les **Makefile** et au lieu de

```
CC = gcc
```

mettre

```
CC = gcc -fPIC
```

on tape **make** puis en tant que root

```
make install  
ldconfig
```

## 5.4 Installation de codecs et outils multimédia

### 5.4.1 Installation d'OpenCV

**OpenCV** qui veut dire Open Source Computer Vision library est une bibliothèque qui fournit 500 algorithmes, documentation et divers code d'affichage, elle sert notamment pour le tracking d'éléments dans une vidéo. On installera préalablement les packages **python3-numpy-devel**, **lib64blas-static-devel**, **lib64gphoto-devel**, **lib64vtk-devel**, **lib64ogre-devel**, **lib64openjpeg-devel**, **tesseract-devel** pour la reconnaissance de caractères. Pour les possesseurs GPU NVIDIA on installera les packages **nvdiacuda-toolkit-devel** et **nvdiacuda-toolkit-samples** pour le calcul en parallèle des GPU nvidia, pour les GPU intel on installera en revanche **lib64va-devel**.

Le site officiel est <https://opencv.org> on y récupère l'archive officiel, qu'on désarchive en tapant

## unzip opencv-4.6.0.zip

cela donne le répertoire **opencv-4.6.0** on récupérera également des extensions qu'on trouvera par là [https://github.com/opencv/opencv\\_contrib](https://github.com/opencv/opencv_contrib), on les décompresse en tapant

```
tar xvfz opencv_contrib-4.6.0.tar.gz
```

cela donne le répertoire **opencv\_contrib-4.6.0**. On revient dans le répertoire **opencv-4.6.0** dans ce répertoire on tape

```
mkdir build
cd build
```

```
cmake -DWITH_CUDA=OFF -DCMAKE_BUILD_TYPE=Release -DOPENCV_EXTRA_MODULES_PATH=
usr/local/linux/multimedia/opencv_contrib-4.6.0/modules/ -DWITH_OPENGL=ON -
DOPENCV_ENABLE_NONFREE=ON -DTesseract_INCLUDE_DIR=/usr/include/tesseract -
DCMAKE_CXX_FLAGS="-lcbblas -std=c++11" -DENABLE_CXX11=ON -
DOPENCV_GENERATE_PKGCONFIG=YES -DWITH_QT=ON -DOGRE_DIR=/usr/lib64/OGRE/cmake ..
```

A noter que l'option **-DOPENCV\_GENERATE\_PKGCONFIG=YES** **-DCMAKE\_CXX\_FLAGS="-std=c++11"** **-DENABLE\_CXX11=1** servent pour compiler **ffmpeg**, on désactivera momentanément les fonctions CUDA, une partie du code n'est plus maintenue (voir par là <https://docs.nvidia.com/cuda/video-decoder/index.html>) et ça plante à la compilation. On pourra rajouter les options **-DWITH\_VA=ON** **-DWITH\_VA\_INTEL=OFF** pour une GPU intel. Pour le chemin des contribs **opencv**, il faudra l'adapter bien sûr à votre configuration

Voilà le résultat

```
--          General          configuration          for          OpenCV          4.6.0
=====
-- Version control:          unknown
--
-- Extra modules:
--                               Location                               (extra):
/usr/local/linux/multimedia/opencv_contrib-4.6.0/modules
-- Version control (extra):  unknown
--
-- Platform:
-- Timestamp:                2022-12-26T09:44:45Z
-- Host:                      Linux 6.1.1-desktop-2.mga9 x86_64
-- CMake:                     3.25.1
-- CMake generator:          Unix Makefiles
-- CMake build tool:         /bin/gmake
-- Configuration:           Release
--
-- CPU/HW features:
-- Baseline:                  SSE SSE2 SSE3
-- requested:                 SSE3
-- Dispatched code generation: SSE4_1 SSE4_2 FP16 AVX AVX2 AVX512_SKX
-- requested:                 SSE4_1 SSE4_2 AVX FP16 AVX2 AVX512_SKX
-- SSE4_1 (18 files):         + SSSE3 SSE4_1
-- SSE4_2 (2 files):         + SSSE3 SSE4_1 POPCNT SSE4_2
-- FP16 (1 files):           + SSSE3 SSE4_1 POPCNT SSE4_2 FP16 AVX
-- AVX (5 files):             + SSSE3 SSE4_1 POPCNT SSE4_2 AVX
-- AVX2 (33 files):          + SSSE3 SSE4_1 POPCNT SSE4_2 FP16 FMA3 AVX AVX2
-- AVX512_SKX (8 files):     + SSSE3 SSE4_1 POPCNT SSE4_2 FP16 FMA3 AVX
AVX2 AVX_512F AVX512_COMMON AVX512_SKX
--
-- C/C++:
-- Built as dynamic libs?:    YES
-- C++ standard:              11
-- C++ Compiler:              /bin/c++ (ver 12.2.1)
```



```

-- C++ flags (Release):      -lcbblas -std=c++11 -fsigned-char -W -Wall -Wreturn-
type -Wnon-virtual-dtor -Waddress -Wsequence-point -Wformat -Wformat-security -
Wmissing-declarations -Wmissing-prototypes -Wstrict-prototypes -Wundef -Winit-self
-Wpointer-arith -Wshadow -Wsign-promo -Wuninitialized -Wsuggest-override -Wno-
delete-non-virtual-dtor -Wno-unnamed-type-template-args -Wno-comment -Wimplicit-
fallthrough=3 -Wno-strict-overflow -fdiagnostics-show-option -Wno-long-long -
pthread -fomit-frame-pointer -ffunction-sections -fdata-sections -msse -msse2 -
msse3 -fvisibility=hidden -fvisibility-inlines-hidden -O3 -DNDEBUG -DNDEBUG
-- C++ flags (Debug):      -lcbblas -std=c++11 -fsigned-char -W -Wall -Wreturn-
type -Wnon-virtual-dtor -Waddress -Wsequence-point -Wformat -Wformat-security -
Wmissing-declarations -Wmissing-prototypes -Wstrict-prototypes -Wundef -Winit-self
-Wpointer-arith -Wshadow -Wsign-promo -Wuninitialized -Wsuggest-override -Wno-
delete-non-virtual-dtor -Wno-unnamed-type-template-args -Wno-comment -Wimplicit-
fallthrough=3 -Wno-strict-overflow -fdiagnostics-show-option -Wno-long-long -
pthread -fomit-frame-pointer -ffunction-sections -fdata-sections -msse -msse2 -
msse3 -fvisibility=hidden -fvisibility-inlines-hidden -g -O0 -DDEBUG -D_DEBUG
-- C Compiler:              /bin/cc
-- C flags (Release):      -fsigned-char -W -Wall -Wreturn-type -Wnon-virtual-dtor
-Waddress -Wsequence-point -Wformat -Wformat-security -Wmissing-declarations -
Wmissing-prototypes -Wstrict-prototypes -Wundef -Winit-self -Wpointer-arith -
Wshadow -Wsign-promo -Wuninitialized -Wsuggest-override -Wno-delete-non-virtual-
dtor -Wno-unnamed-type-template-args -Wno-comment -Wimplicit-fallthrough=3 -
Wno-strict-overflow -fdiagnostics-show-option -Wno-long-long -pthread -fomit-frame-
pointer -ffunction-sections -fdata-sections -msse -msse2 -msse3 -fvisibility=hidden -
fvisibility-inlines-hidden -O3 -DNDEBUG -DNDEBUG
-- C flags (Debug):      -fsigned-char -W -Wall -Wreturn-type -Wnon-virtual-dtor
-Waddress -Wsequence-point -Wformat -Wformat-security -Wmissing-declarations -
Wmissing-prototypes -Wstrict-prototypes -Wundef -Winit-self -Wpointer-arith -
Wshadow -Wsign-promo -Wuninitialized -Wsuggest-override -Wno-delete-non-virtual-
dtor -Wno-unnamed-type-template-args -Wno-comment -Wimplicit-fallthrough=3 -
Wno-strict-overflow -fdiagnostics-show-option -Wno-long-long -pthread -fomit-frame-
pointer -ffunction-sections -fdata-sections -msse -msse2 -msse3 -fvisibility=hidden -
fvisibility-inlines-hidden -g -O0 -DDEBUG -D_DEBUG
-- Linker flags (Release): -Wl,--exclude-libs,libippicv.a -Wl,--exclude-
libs,libippiw.a -Wl,--gc-sections -Wl,--as-needed -Wl,--no-undefined
-- Linker flags (Debug):  -Wl,--exclude-libs,libippicv.a -Wl,--exclude-
libs,libippiw.a -Wl,--gc-sections -Wl,--as-needed -Wl,--no-undefined
-- ccache:                 NO
-- Precompiled headers:    NO
-- Extra dependencies:     dl m pthread rt
-- 3rdparty dependencies:
--
-- OpenCV modules:
-- To be built:            alphamat aruco barcode bgsegm bioinspired calib3d ccalib
core cvv datasets dnn dnn_objdetect dnn_superres dpm face features2d flann freetype
fuzzy gapi hdf hfs highgui img_hash imgcodecs imgproc intensity_transform java
line_descriptor mcc ml_objdetect optflow phase_unwrapping photo plot quality rapid
reg rgbd saliency shape stereo stitching structured_light superres surface_matching
text tracking ts video videoio videostab wechat_qrcode xfeatures2d ximgproc
xobjdetect xphoto
-- Disabled:              world
-- Disabled by dependency: -
-- Unavailable:          cudaarithm cudabgsegm cudacodec cudafeatures2d
cudafilters cudaimgproc cudalegacy cudaobjdetect cudaoptflow cudastereo
cudawarping cudev julia matlab ovis python2 python3 sfm viz
-- Applications:         tests perf_tests apps
-- Documentation:       NO
-- Non-free algorithms:  YES
--
-- GUI:                  QT5
-- QT:                   YES (ver 5.15.7 )

```



```
-- Java tests:          YES
--
-- Install to:         /usr/local
-----
--
-- Configuring done
-- Generating done
-- Build files have been written to: /usr/local/linux/multimedia/opencv-4.6.0/build
```

Puis sous le répertoire **opencv-4.6.0/build** on tape

```
make
```

je suis tombé sur cette erreur

```
/usr/local/linux/multimedia/opencv-4.6.0/modules/videoio/src/ffmpeg_codecs.hpp:81:26:      erreur:
« CODEC_ID_H264 » n'a pas été déclaré dans cette portée; voulez-vous employer « AV_CODEC_ID_H264 » ?
```

pour la résoudre j'ai édité le fichier **opencv-4.6.0/modules/videoio/src/ffmpeg\_codecs.hpp** et mis en commentaire toute cette partie

```
/*#if (LIBAVCODEC_VERSION_INT <= AV_VERSION_INT(54, 51, 100))
(...)
#endif*/
```

ensuite pour cette erreur là

```
dynafu.cpp:
(.text._ZN2cv6dynafu10DynaFuImplINS_3MatEEC2ERKNS_5kinfu6ParamsE[_ZN2cv6dynafu10DynaFuImplI
NS_3MatEEC5ERKNS_5kinfu6ParamsE]+0x39c) : référence indéfinie vers « glGenRenderbuffersEXT »
/bin/ld : dynafu.cpp:
(.text._ZN2cv6dynafu10DynaFuImplINS_3MatEEC2ERKNS_5kinfu6ParamsE[_ZN2cv6dynafu10DynaFuImplI
NS_3MatEEC5ERKNS_5kinfu6ParamsE]+0x3a9) : référence indéfinie vers « glBindRenderbufferEXT »
```

visiblement c'est un conflit de bibliothèque **OpenGL** et le fait de mettre la variable **OpenGL\_GL\_PREFERENCE** à **LEGACY** pour forcer l'utilisation de **/lib64/libGL.so** ne sert à rien. J'ai édité le fichier **opencv-4.6.0/build/CMakeVars.txt** et il faut changer toutes les occurrences de **/lib64/libOpenGL.so;/lib64/libGLX.so;/lib64/libGLU.so** par **/lib64/libGL.so**

De même dans le fichier **opencv-4.6.0/build/modules/rgbd/CMakeFiles/opencv\_rgbd.dir/build.make** on remplacera les lignes

```
lib/libopencv_rgbd.so.4.6.0: /lib64/libOpenGL.so
lib/libopencv_rgbd.so.4.6.0: /lib64/libGLX.so
lib/libopencv_rgbd.so.4.6.0: /lib64/libGLU.so
```

par

```
lib/libopencv_rgbd.so.4.6.0: /lib64/libGL.so
```

et dans le fichier **opencv-4.6.0/build/modules/rgbd/CMakeFiles/opencv\_rgbd.dir/link.txt** on remplace

```
/lib64/libOpenGL.so /lib64/libGLX.so /lib64/libGLU.so par /lib64/libGL.so
```

on retape **make** puis en tant que root **make install**

Alors en théorie **opencv** a besoin de **ffmpeg** pour compiler, je vous conseille donc d'installer d'abord **opencv** sans **ffmpeg**, de compiler **ffmpeg** de recompiler **opencv** avec **ffmpeg** et de recommencer ensuite la compilation de **frei0r**, c'est tordu mais c'est le seul moyen. Il faudra pas hésiter à supprimer le répertoire **build** pour supprimer le cache.

## 5.4.2 Installation de frei0r

Maintenant on va récupérer la bibliothèque **frei0r** qui fournit une API vidéo qui est utile pour les effets vidéo pour les logiciels de montage vidéo, le site officiel est <https://frei0r.dyne.org/>. On installera préalablement le package **lib64gavl-**

**devel.** On décompresse la dernière version en tapant

```
tar xvfz frei0r-2.2.0.tar.gz
```

cela donne le répertoire **frei0r-plugins-1.8.0** dans lequel on tape successivement

```
mkdir build  
cd build  
cmake ..
```

cela donne

```
-- Found Cairo  
-- Checking for module 'gavl'  
-- Found gavl, version 1.4.0  
-- Configuring done  
-- Generating done  
-- Build files have been written to: /usr/local/linux/multimedia/frei0r-2.2.0/build
```

on tape ensuite

```
make
```

Puis en tant que root

```
make install
```

on rajoutera dans le fichier **/etc/ld.so.conf** la ligne

```
/usr/local/lib64/frei0r-1
```

et on tape **ldconfig** attention j'ai dû créer le lien suivant pour que **mlt** retrouve les plugins **frei0r**

```
ln -s /usr/local/lib64/frei0r-1 /usr/local/lib/frei0r-1
```

### 5.4.3 Installation de vid.stab

**vid.stab** est un plugin servant à stabiliser les vidéos, il sert notamment à [kdenlive](http://kdenlive.org), le site officiel est <http://public.hronopik.de/vid.stab/> on y récupérera l'archive qu'on décompressera en tapant:

```
tar xvfz vid.stab-1.1.0.tar.gz
```

cela donne le répertoire **vid.stab-1.1.0** dans lequel on tape

```
cmake . -DCMAKE_BUILD_TYPE=Release  
make
```

puis en tant que root

```
make install
```

### 5.4.4 Installation de movit

**movit** est une bibliothèque d'effets vidéos qui va utiliser les ressources des processeurs graphiques (GPU) généralement présent sur la carte graphique, en complément des CPU pour accélérer les applications grâce au calcul parallèle. On récupère les sources sur le site officiel <https://movit.sesse.net/> qu'on décompresse en tapant

```
tar xvfz movit-1.6.3.tar.gz
```

cela donne le répertoire **movit-1.6.3**. Pour les joueurs on pourra récupérer la version en développement en tapant

**git clone <http://git.sesse.net/movit>**

tout d'abord il faudra installer **eigen** qui est une bibliothèque mathématique pour l'algèbre linéaire. On installe d'abord les packages **fftw3-devel**, **lib64epoxy-devel**, **lib64freeglut-devel** et **lib64sdl2.0-devel**. Le site officiel est par ici [http://eigen.tuxfamily.org/index.php?title=Main\\_Page](http://eigen.tuxfamily.org/index.php?title=Main_Page) on y récupère **eigen 3.4.0** qu'on décompresse en tapant

**tar xvfj eigen-3.4.0.tar.bz2**

cela donne le répertoire **eigen-3.4.0** dans lequel on tape

**mkdir build**

**cd build**

**cmake .. -DCMAKE\_BUILD\_TYPE=Release -DCUDA\_SDK\_ROOT\_DIR=/usr/share/nvidia-cuda-toolkit/ -DCUDA\_TOOLKIT\_ROOT\_DIR=/usr/share/nvidia-cuda-toolkit/ -DEIGEN\_TEST\_CUDA=on**  
**make**

à noter les options **-DCUDA\_SDK\_ROOT\_DIR=/usr/share/nvidia-cuda-toolkit/ -DCUDA\_TOOLKIT\_ROOT\_DIR=/usr/share/nvidia-cuda-toolkit/ -DEIGEN\_TEST\_CUDA=on** sont pour les possesseurs de GPU nvidia, puis en tant que root

**make install**

**cp eigen3.pc /usr/local/lib/pkgconfig**

on installera également la biblio de google pour pouvoir faire des tests du code C++, le site officiel est par ici <https://code.google.com/p/googletest/> on décompresse l'archive en tapant

**tar xvfz googletest-release-1.12.1.tar.gz**

cela donne le répertoire **googletest-release-1.12.1** dans lequel on tape

**mkdir build**

**cd build**

**cmake .. -DCMAKE\_BUILD\_TYPE=Release**  
**make**

puis en tant que root

**make install**

on revient dans le répertoire **movit** et on tape

**./configure**

on édite ensuite le fichier **Makefile** et on modifie la première ligne (en adaptant le chemin) comme suit

**GTEST\_DIR ?= /usr/local/linux/multimedia/googletest-release-1.12.1/googletest**

on tape ensuite

**make**

puis en tant que root

**make install**

## 5.4.5 Installation de ffmpeg

**ffmpeg** est un codec vidéo compatible MPEG-4 son site officiel est <http://ffmpeg.org/download.html>. On décompresse la dernière archive en tapant

**tar xvf ffmpeg-5.1.2.tar.xz**

cela donnera le répertoire **ffmpeg-5.1.2** dans lequel on tape

```
./configure --enable-shared --enable-gpl --enable-postproc --enable-libbmp3lame --enable-libtheora --enable-libvorbis --enable-libxvid --enable-libx264 --enable-libx265 --enable-libfdk-aac --enable-nonfree --enable-frei0r --enable-libpulse --enable-version3 --enable-opengl --enable-opencl --disable-stripping --enable-libgsm --enable-libvpx --enable-avfilter --enable-chromaprint --enable-librubberband --enable-libvidstab --enable-vidpau --enable-libnpp --enable-cuda-nvcc --nvcc=nvcc --enable-cuvid --enable-ffnvcodec --enable-libopencv --enable-libdc1394 --extra-ldflags="-L/usr/lib64" --extra-cflags="-I/usr/include/cuda" --enable-libass
```

On installera préalablement les packages **lib64gsm-devel**, **lib64schroedinger-devel**, **lib64vpx-devel**, **lib64vidpau-devel**, **lib64sdl2.0-devel** et **lib64opencl-devel**

si vous rencontrez cette erreur pour les possesseurs de nvidia

**ERROR: failed checking for nvcc**

il faudra éditer le fichier **configure** et modifier la ligne ainsi

```
nvccflags_default="-gencode arch=compute_50,code=sm_50 -O2"
```

Si **OpenCV** n'est pas bien reconnu il faudra taper préalablement

```
export CFLAGS="-I/usr/local/include/opencv4"
export LDFLAGS=-L/usr/local/lib64:$LDFLAGS
export PKG_CONFIG_PATH=/usr/local/lib64/pkgconfig:$PKG_CONFIG_PATH
```

et modifier le fichier **/usr/local/include/opencv4/opencv2/core/cvdef.h** comme ceci

```
/*
 *          C++ 11          *
 */
#ifndef CV_CXX11
/*# if __cplusplus >= 201103L || (defined(_MSC_VER) && _MSC_VER >= 1800)
# define CV_CXX11 1
# endif
#else
# if CV_CXX11 == 0
# undef CV_CXX11
# endif
#endif
#error "OpenCV 4.x+ requires enabled C++11 support"*/
#endif

#define CV_CXX_MOVE_SEMANTICS 1
#define CV_CXX_STD_ARRAY 1
/*#include <array>*/
#ifndef CV_OVERRIDE
# define CV_OVERRIDE override
#endif
#ifndef CV_FINAL
# define CV_FINAL final
#endif
```

à noter les options **--enable-libnpp** **--enable-cuda-nvcc** **--enable-cuvid** **--enable-ffnvcodec** pour les possesseurs de GPU nvidia qui risquent de générer l'erreur suivante

**ERROR: cuda requested, but not all dependencies are satisfied: ffnvcodec**

on va récupérer la dépendance manquante en tapant

```
git clone https://git.videolan.org/git/ffmpeg/nv-codec-headers.git
```

dans le répertoire **nv-codec-headers/** on tape **make** puis en tant que root **make install**

on retape **configure** avec toutes ses options

si vous avez l'erreur suivante

```
libavcodec/rle.c: Dans la fonction « ff_rle_count_pixels »:  
libavcodec/rle.c:27:5: erreur interne du compilateur: Erreur de segmentation  
int ff_rle_count_pixels(const uint8_t *start, int len, int bpp, int same)
```

il faudra éditer le fichier **ffbuild/config.mak** et supprimer les options d'optimisation (O2 et O3)

on tape **make**

si vous tombez sur l'erreur suivante

```
makeinfo: warning: erreur de chargement de ./doc/t2h.pm : Undefined subroutine  
&Texinfo::Config::set_from_init_file called at ./doc/t2h.pm line 24.  
Compilation failed in require at /usr/share/texinfo/TeXinfo/Config.pm line 130.  
makeinfo: warning: erreur de chargement de ./doc/t2h.pm : Undefined subroutine  
&Texinfo::Config::set_from_init_file called at ./doc/t2h.pm line 24.
```

il faudra rajouter l'option **--disable-doc** à la commande **configure**

on retape **make** puis en tant que root

**make install**  
**ldconfig**

Vous remplacez le chemin en absolu de **ffmpeg** avec celui de votre configuration. On obtient la liste des formats supportés par **ffmpeg** en tapant

**ffmpeg -formats**

Pour voir si les accélérations matérielles sont bien en place si vous avez une NVIDIA, vous avez la liste des encoders en tapant

**ffmpeg -encoders**

(extrait)

```
V..... libx264          libx264 H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10 (codec h264)  
V..... libx264rgb       libx264 H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10 RGB (codec h264)  
V....D h264_nvenc       NVIDIA NVENC H.264 encoder (codec h264)  
V..... h264_v4l2m2m     V4L2 mem2mem H.264 encoder wrapper (codec h264)  
V....D h264_vaapi       H.264/AVC (VA-API) (codec h264)  
V..... nvenc           NVIDIA NVENC H.264 encoder (codec h264)  
V..... nvenc_h264      NVIDIA NVENC H.264 encoder (codec h264)  
V..... libx265          libx265 H.265 / HEVC (codec hevc)  
V..... nvenc_hevc       NVIDIA NVENC hevc encoder (codec hevc)  
V....D hevc_nvenc       NVIDIA NVENC hevc encoder (codec hevc)  
V..... hevc_v4l2m2m     V4L2 mem2mem HEVC encoder wrapper (codec hevc)  
V....D hevc_vaapi       H.265/HEVC (VA-API) (codec hevc)
```

la liste des méthodes d'accélération matérielle est obtenue en tapant

**ffmpeg -hwaccels**

voilà le résultat

#### Hardware acceleration methods:

**vdpa  
cuda  
vaapi  
opencl**

## 6 Les conteneurs vidéos

### 6.1 Présentation

Dans ce chapitre sont présentés les conteneurs ogm et matroska qui ont tout deux beaucoup d'avenir. Ces deux conteneurs sont vraiment très riches par rapport à l'avi, ce paragraphe n'effleure cependant même pas les fonctionnalités les plus intéressantes (chapitrage, plusieurs flux vidéo, sous titrage, ...).

### 6.2 Le conteneur avi

AVI est un conteneur vidéo développé par Microsoft, c'est l'acronyme de Audio Vidéo Interleaved. Chaque flux (vidéo ou audio) est défini par un codec, on peut donc avoir différentes combinaisons de codecs audio ou vidéo, voilà une liste des codecs utilisables:

#### Codec vidéo

- MPEG
- divx
- xvid
- raw (YUV, YV12, ...)
- autres (indeo, cinepak, ...)

#### Codec audio

- mp3
- mp2
- pcm/wav
- autres

Ce conteneur a la particularité de pouvoir contenir un flux vidéo et deux flux audio (deux langues par exemple).

Les combinaisons les plus utilisées sont divx/mp3 ou xvid/mp3. Ce conteneur ne permet pas le chapitrage, les sous titres et ne reconnaît pas certains codecs plus modernes comme le vorbis.

### 6.3 Ogm tools

#### 6.3.1 Installation

Préalablement vous devez installer **libvdpdread** comme indiqué dans le chapitre 7.1.2.

Le site officiel des ogm tools est <http://www.bunkus.org/videotools/ogmtools/index.html> on y récupère l'archive qu'on décompresse en tapant

```
tar xvfj ogmtools-1.5.tar.bz2
```

Cela donne le répertoire **ogmtools-1.5** dans lequel on tape

```
./configure  
make
```

Puis en tant que root

```
make install
```

#### 6.3.2 Utilisation

On va créer un fichier ogm à partir d'un avi. Ouvrez le fichier .avi avec **avidemux**, sauvegardez la bande son au format mp2 (**Audio -> Sauve la piste audio**). Ouvrez le fichier avec **xmms** et avec son module d'enregistrement, sauvegardez le au format wav. On le transforme maintenant au format ogg.



```
oggenc -q4 -o bandeson1.ogg video.wav
```

Voilà le résultat

```
Opening with wav module: WAV file reader
Encoding "video.wav" to
  "bandeson1.ogg"
at quality 4,00
  [100,0%] [ 0m00s remaining] -
```

```
Done encoding file "bandeson1.ogg"
```

```
File length: 2m 49,0s
Elapsed time: 0m 45,6s
Rate:        3,7212
Average bitrate: 99,3 kb/s
```

Maintenant on peut fusionner la vidéo et le son.

```
ogmmerge -o video.ogm video.avi bandeson1.ogg
```

A noter que le fichier obtenu aura deux bandes sons, celle d'origine et celle précédemment obtenue (à la condition que celle d'origine soit compatible avec ogm). Pour éviter cela et n'avoir qu'une bande son (il va supprimer la bande son d'origine) on tapera:

```
ogmmerge -o video.ogm -A video.avi bandeson1.ogg
```

Voilà le résultat

```
Using AVI demultiplexer for video.avi. Opening file. This may take some time depending on the file's size.
+> Using video output module for video stream.
Using OGG/OGM demultiplexer for bandeson1.ogg.
OGG/OGM demultiplexer (bandeson1.ogg): using Vorbis audio output module for stream 1.
progress: 4246/4246 frames (100%)
```

Dans l'hypothèse où vous voudriez rajouter une deuxième bande son (film en anglais ou en français par exemple). Pour reprendre l'exemple précédent on tapera

```
ogmmerge -o movie.ogm video.ogm bandeson2.ogg
```

On aurait aussi bien pu taper

```
ogmmerge -o movie.ogm -A video.avi bandeson1.ogg bandeson2.ogg
```

Si vos fichiers sont trop gros pour tenir sur un CD, vous avez la commande suivante pour le splitter en fichier de 700Mo max (les fichiers commenceront par film)

```
ogmsplit -o film -s 700 movie.ogm
```

avec **mplayer** pour avoir la première bande son il suffit de taper (le premier track est numéroté 1, le suivant 2, etc.).

```
mplayer -aid 1 movie.ogm
```

avec **gmpayer** en cliquant sur l'image vous avez la commande **Piste Audio**, vous n'avez qu'à choisir Piste 1 ou Piste 2.

## 6.4 Vorbis tools

Comme son nom l'indique **vorbis tools** permet d'installer un tas d'outil en ligne de commande pour manipuler les fichiers audio ogg. Avec **vorbis tools** on peut ainsi encoder, lire ou éditer les fichiers ogg le tout dans un shell. Le site officiel est <http://www.xiph.org/downloads/> on y récupèrera l'archive qu'on décompressera en tapant

```
tar xvfz vorbis-tools-1.4.2.tar.gz
```

Cela donne le répertoire **vorbis-tools-1.4.2** dans lequel on va taper

```
./configure  
make
```

puis en tant que root

```
make install
```

## 6.5 Matroska

### 6.5.1 Installation

Le site officiel est <http://www.matroska.org/> on y récupérera deux bibliothèques. On décompresse la première en tapant

```
tar xvfz libebml-release-1.4.4.tar.gz
```

Cela donne le répertoire **libebml-release-1.4.4**, dans lequel on tape

```
mkdir build  
cd build  
cmake ..  
make
```

et en tant que root

```
make install  
ldconfig
```

Maintenant on décompresse la deuxième archive

```
tar xvfz libmatroska-release-1.7.1.tar.gz
```

Cela donne le répertoire **libmatroska-release-1.7.1**, dans lequel on tape

```
mkdir build  
cd build  
cmake ..  
make
```

Et en tant que root

```
make install  
ldconfig
```

Maintenant on récupère les outils **matroska** sur le site <http://www.bunkus.org/videotools/mkvtoolnix/>. Il faudra installer les packages **ruby** et **lib64cmark-devel** préalablement. On décompresse l'archive en tapant

```
tar xvf mkvtoolnix-72.0.0.tar.xz
```

Cela donne le répertoire **mkvtoolnix-72.0.0** dans lequel on tape

```
./configure  
rake
```

puis en tant que root

```
rake install
```

si vous ne disposez pas de la commande **rake**, vous pouvez la récupérer en tapant en tant que root

```
gem install rake
```

## 6.5.2 Utilisation

La syntaxe est très similaire à celle des ogm tools (normal c'est le même auteur). On reprend les mêmes fichiers vidéo et audio qu'avant. La commande de création d'un conteneur matroska (mkv) avec le fichier avi video.avi, dont on a extrait la bande son comme vu précédemment, est la suivante

```
mkvmerge -o video.mkv video.avi bandeson1.ogg
```

Voilà le résultat

```
mkvmerge v72.0.0 ('Minuano (Six-eight)') 64-bit  
'video.avi': Using the AVI demultiplexer. Opening file. This may take some time depending on the file's size.  
'video.ogg': Using the OGG/OGM demultiplexer.  
'video.avi' track 0: Using the video output module for the video track.  
'video.avi' track 1: Using the MPEG audio output module.  
'video.ogg' track 0: Using the Vorbis output module.  
The file 'video.mkv' has been opened for writing.  
progress: 4246/4246 frames (100%)  
The cue entries (the index) are being written...  
Muxing took 17 seconds.
```

Comme avec les ogm tools, le fichier obtenu aura deux bandes sons, celle d'origine et celle précédemment obtenue (à la condition que celle d'origine soit compatible avec ogm). Pour éviter cela et n'avoir qu'une bande son (il va supprimer la bande son d'origine) on tapera:

```
mkvmerge -o video.mkv -A video.avi bandeson1.ogg
```

Voilà le résultat

```
mkvmerge v72.0.0 ('Minuano (Six-eight)') 64-bit  
'video.avi': Using the AVI demultiplexer. Opening file. This may take some time depending on the file's size.  
'video.ogg': Using the OGG/OGM demultiplexer.  
'video.avi' track 0: Using the video output module for the video track.  
'video.ogg' track 0: Using the Vorbis output module.  
The file 'video.mkv' has been opened for writing.  
progress: 4246/4246 frames (100%)  
The cue entries (the index) are being written...  
Muxing took 7 seconds.
```

Dans l'hypothèse où vous voudriez rajouter une deuxième bande son (film en anglais ou en français par exemple). Pour reprendre l'exemple précédent on tapera

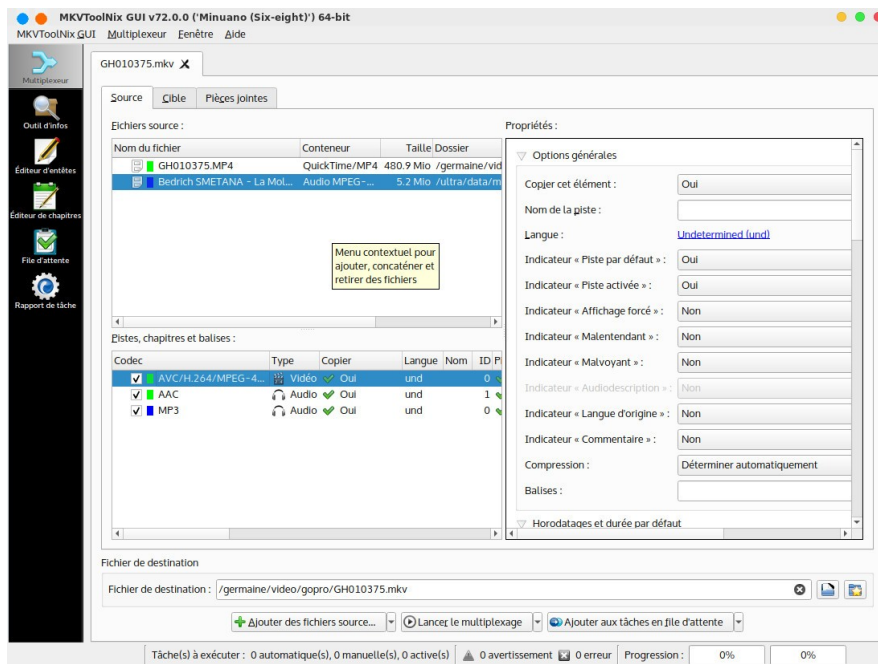
```
mkvmerge -o movie.ogm video.ogm bandeson2.ogg
```

On aurait aussi bien pu taper

```
mkvmerge -o movie.ogm -A video.avi bandeson1.ogg bandeson2.ogg
```

Le fichier obtenu est lu avec **mplayer** de la même manière qu'un fichier ogm.

A noter que **mkvtoolnix** est fourni avec une interface graphique qui a l'air vraiment très complète. Il suffit de taper **mkvtoolnix-gui**



A cet endroit <http://www.bunkus.org/videtools/mkvtoolnix/doc/mkvmerge-gui.html> vous trouverez une très bonne aide pour utiliser ce soft. Pour les allergiques à l'anglais, j'ai réalisé une [traduction littérale](#) de cette page.

## 7 Les lecteurs vidéo sous Linux

### 7.1 Installation des bibliothèques de base

#### 7.1.1 Installation de mad

MAD est un décodeur audio haute qualité MPEG. L'URL officiel est [www.underbit.com/products/mad/](http://www.underbit.com/products/mad/) vous y trouverez la dernière archive que vous décompressez en tapant :

```
tar xvfz libmad-0.15.1b.tar.gz
```

Cela donne le répertoire **mad-0.15.1b** dans lequel on tape successivement

```
./configure  
make
```

si vous rencontrez l'erreur suivante

```
cc1: erreur: option "-fforce-mem" de la ligne de commande non reconnue
```

Editez le fichier **configure** et supprimez **fforce-mem**

```
optimize="$Soptimize -fforce-mem"
```

relancez **configure** et **make** puis en tant que root

```
make install  
ldconfig
```

#### 7.1.2 Installation de libdvdrread

Cette bibliothèque offre des outils pour lire les DVD. L'URL est <http://download.videolan.org/pub/videolan/> où on récupérera l'archive qu'on décompresse en tapant

```
tar xvfj libdvdrread-6.1.3.tar.bz2
```

Cela donne le répertoire **libdvdread-6.1.3** dans lequel on tape

```
./configure  
make
```

Puis en tant que root

```
make install
```

### 7.1.3 Installation de libdvdnav

Cette bibliothèque de fonctions fournit des outils pour pouvoir "naviguer" dans un DVD, c'est à dire pouvoir utiliser les menus interactifs proposés dans la plupart des DVD. L'URL est <http://download.videolan.org/pub/videolan/> on y récupère l'archive qu'on décompresse en tapant

```
tar xvfj libdvdnav-6.1.1.tar.bz2
```

Cela donne le répertoire **libdvdnav-6.1.1** dans lequel on tape

```
./configure  
make
```

Puis en tant que root

```
make install
```

### 7.1.4 Installation de libdvdcss

**Libdvdcss** est une bibliothèque offrant des outils pour lire du flux MPEG2 d'un disque dur ou d'un DVD. Il permet aussi de lire les DVD cryptés. L'URL est <http://www.videolan.org/developers/libdvdcss.html> on y récupère l'archive qu'on décompresse en tant

```
tar xvfz libdvdcss-1.4.3.tar.gz
```

Cela donne le répertoire **libdvdcss-1.4.3** dans lequel on tape

```
./configure  
make
```

Puis en tant que root

```
make install
```

### 7.1.5 Installation de libdvbpsi

**libdvbpsi** dont le site est <http://www.videolan.org/developers/libdvbpsi.html> permet la diffusion de flux MPEG2, on y récupère l'archive qu'on décompresse en tapant

```
tar xvfj libdvbpsi-1.3.3.tar.bz2
```

Cela donne le répertoire **libdvbpsi-1.3.3** dans lequel on tape

```
./configure  
make
```

puis en tant que root

```
make install
```

cette bibliothèque est nécessaire à **vlc** pour faire du streaming vidéo.

### 7.1.6 Installation de live555

**live555** dont le site <http://www.live555.com/liveMedia/public/> permet de faire du streaming et de la réception vidéo, il est nécessaire à **vlc**. On décompresse l'archive en tapant

```
tar xvfz live555-latest.tar.gz
```

Cela donne le répertoire **live** dans lequel on édite le fichier **config.linux** pour rajouter l'option **-fPIC**

```
COMPILE_OPTS = $(INCLUDES) -I. -O2 -fPIC -DSOCKLEN_T=socklen_t -  
D_LARGEFILE_SOURCE=1 -D_FILE_OFFSET_BITS=64
```

on tape ensuite

```
./genMakefiles linux
```

on tape ensuite

```
make
```

on tape ensuite en tant que root

```
make install
```

```
ln -s /usr/local/linux/multimedia/live/liveMedia/include/ /usr/include/liveMedia
```

```
ln -s /usr/local/linux/multimedia/live/groupsock/include /usr/include/groupsock
```

```
ln -s /usr/local/linux/multimedia/live/UsageEnvironment/include/ /usr/include/UsageEnvironment
```

```
ln -s /usr/local/linux/multimedia/live/BasicUsageEnvironment/include/ /usr/include/BasicUsageEnvironment
```

## 7.2 Les lecteurs multiformats

### 7.2.1 Présentation

Sont présentés ici les lecteurs les plus courants à savoir **xine**, **MPlayer** et **vlc**. Ces différents players sont capables de lire les avi classiques (du style divx/xvid mp3) mais aussi les **ogm** et les **matroska**.

### 7.2.2 VLC

#### 7.2.2.1 Installation

VLC est un player vidéo particulièrement puissant, le site officiel est <http://www.videolan.org> je n'ai pas réussi à installer la dernière version stable 3.0.18 qui génère un tas d'erreurs, j'ai donc récupéré la version en développement avec **git**

```
git clone https://code.videolan.org/videolan/vlc.git
```

cela donne le répertoire **vlc**. On installera éventuellement les packages **lib64cddb-devel**, **qt5-devel**, **fribidi**, **fribidi-devel**, **lib64sndfile-devel**, **lib64gcrypt-devel**, **lib64qt5gui-devel**, **wayland-protocols-devel**, **lib64SDL\_image-devel** et **qt5svg-devel**. On revient dans **vlc** on y tape

```
./configure --enable-gles2 --enable-fdkaac --enable-qt
```

puis on tape

```
make
```

pour l'erreur suivante

```
/bin/ld: /usr/local/lib/libliveMedia.a(TLSState.o): in function `TLSState::read(unsigned char*, unsigned int)':  
TLSState.cpp:(.text+0x49): undefined reference to `SSL_read'  
/bin/ld: TLSState.cpp:(.text+0x5f): undefined reference to `SSL_get_error'
```

on modifie le fichier **modules/access/Makefile.am** en rajoutant à la fin de cette ligne

```
liblive555_plugin_la_LIBADD = $(LIBS_live555) $(SOCKET_LIBS) -lssl -lcrypto
```

on retape **make** puis en tant que root

**make install**

on rajoute dans le fichier **/etc/ld.so.conf** la ligne suivante

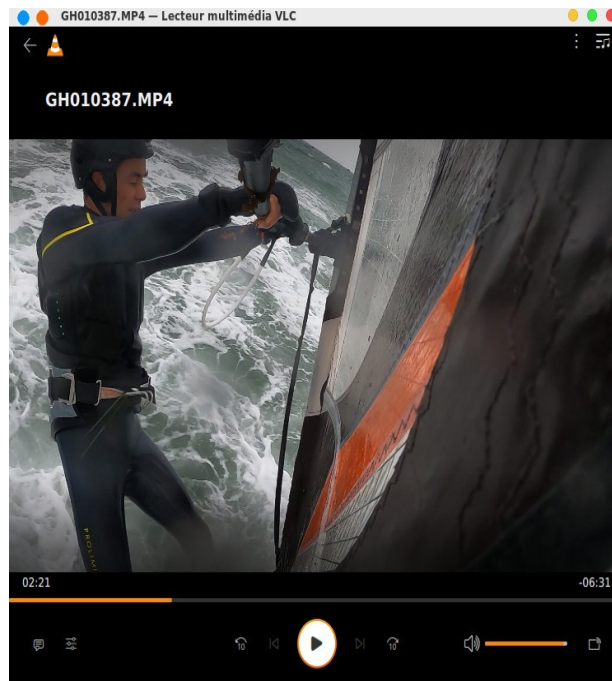
**/usr/local/lib/vlc**

puis on tape

**ldconfig**

### 7.2.2.2 Utilisation

Il suffit de taper **vlc**, l'interface est particulièrement riche (nettement plus que **xine** ou **mplayer** qui sont basiques à côté).



## 7.2.3 Xine

### 7.2.3.1 Installation

Dans ce paragraphe je détaille l'installation de la version tarball disponible sur le site officiel. Avant d'aller plus loin on pourra installer les packages suivants

**lib64cdio-devel**  
**lib64vcd-devel**  
**lib64xt-devel**  
**lib64xvnc-devel**  
**lib64graphicsmagick-devel**

j'ai dû créer préalablement ces deux liens

```
ln -s /usr/include/GraphicsMagick/wand/ /usr/include  
ln -s /usr/include/GraphicsMagick/magick/ /usr/include
```

On récupérera **xine** à l'URL [xine.sourceforge.net](http://xine.sourceforge.net). On décompresse la première archive en tapant

```
tar xvf xine-lib-1.2.11.tar.xz
```

Cela donne le répertoire **xine-lib-1.2.11** dans lequel on tape

```
./configure
```

Voici le résultat

**xine-lib summary:**

- 
- \* **input plugins:**
    - file - stdin\_fifo
  - \* **Network:**
    - rtsp - rtp
    - net - pnm
    - http - ftp
    - https - ftpes
    - gopher
    - tls
    - mms
    - IPv6

(disabled plugins: sftp scp nfs smb)
  - \* **Optical discs:**
    - cdda
    - vcd
    - dvd (external libs)
  - \* **Devices:**
    - dvb
    - v4l2

(disabled plugins: v4l)
  - \* **Misc:**
    - vdr
    - avio (libavformat)
    - test

(disabled plugins: gnome-vfs)
  
  - \* **demultiplexer plugins:**
    - 4xm - aac
    - ac3 - aiff
    - asf
    - avformat (with libavformat)
    - avi - cdda
    - ea wve - film
    - FLAC
    - FLAC (with libFLAC)
    - fli - flv
    - idcin - iff
    - image
    - interplay mve - ivf
    - matroska
    - mng
    - mod
    - mpeg - mpeg\_audio
    - mpeg\_block - mpeg\_elem
    - mpeg\_pes - mpeg\_ts
    - Nosefart (NSF)
    - nsf - nsv
    - ogg
    - psx str - pva
    - qt/mpeg-4 - raw dv
    - real/realaudio - roq
    - smjpeg - snd/au
    - vmd - voc
    - vox - vqa
    - wav
    - wc3 mve - ws aud
    - yuv4mpeg2

(disabled plugins: WavPack)
  
  - \* **video decoder plugins:**



- MPEG 1,2 (libmpeg2)
  - Amiga Bitplane
  - Raw RGB
  - Raw YUV
  - dxr3\_video
  - gdk-pixbuf
  - image
  - libjpeg
  - libpng
  - theora
  - vdpau
  - libvpx (VP8/VP9)
  - libaom (AV1)
  - ffmpeg
- (disabled plugins: w32dll mmal OpenHEVC dav1d)

**\* audio decoder plugins:**

- linear PCM
  - dvaudio
  - GSM 06.10
  - faad
  - Nosefart (NSF)
  - FLAC (with libFLAC)
  - speex
  - vorbis
  - MAD (MPG 1/2/3) (\*INTERNAL\* library)
  - DTS (\*INTERNAL\* library)
  - A52/ra-dnet (external library, float math)
  - Musepack (\*INTERNAL\* library)
  - ffmpeg
- (disabled plugins: w32dll WavPack)

**\* subtitle decoder plugins:**

- spu            - spucc
- spucmml       - sputext
- spudvb
- PGS (BluRay) - TextST (BluRay)
- dxr3\_spu

**\* post effect plugins:**

**\* planar video effects:**

- invert        - expand
- eq            - eq2
- boxblur      - denoise3d
- unsharp     - tvtime
- postproc
- vdr

**\* SFX:**

- goom         - oscscope
- fftscope    - mosaico
- tdaudioanalyzer

**\* Audio:**

- upmix        - upmix\_mono
- volnorm     - scretch

**\* video driver plugins:**

- OpenGL 2.0 (with bicubic scaling)
  - X11 (GLX) backend
  - X11 (EGL) backend
  - Wayland (EGL) backend
- XShm (X11 shared memory)
- Xv (XVideo \*shared\*)
- XxMC (XVideo extended motion compensation)
- OpenGL (with GLU support)
- vaapi (Video Acceleration (VA) API for Linux)

- vdpau (X11 Video Decode and Presentation API for Unix)
  - XvMC (XVideo motion compensation)
  - xcb-shm (X shared memory using XCB)
  - xcb-xv (XVideo using XCB)
  - aa (Ascii ART)
  - fb (Linux framebuffer device)
  - sdl (Simple DirectMedia Layer)
  - dxr3 (Hollywood+ and Creative dxr3, mpeg video only)
- (disabled plugins: sunfb caca directfb directx stk macos mmal vidix)

\* audio driver plugins:

- alsa (ALSA - Advanced Linux Sound Architecture)
  - Jack
  - oss (Open Sound System)
  - pulseaudio (PulseAudio sound server)
- (disabled plugins: CoreAudio directx esd fusionsound irixal sun sndio)

---

On tape ensuite

**make**

j'ai dû également dans le fichier `src/video_dec/Makefile` effacer dans la variable `CFLAGS` les occurrences de `-Werror-implicit-function-declaration`

puis en tant que root

**make install**

Dans le fichier `/etc/ld.so.conf` on rajoute les lignes

```
/usr/local/lib/xine/plugins/2.8
/usr/local/lib/xine/plugins/2.8/post
```

On tape alors, toujours en tant que root

**ldconfig**

Maintenant on décompresse la deuxième archive en tapant

**tar xvf xine-ui-0.99.12.tar.xz**

Cela donne le répertoire `xine-ui-0.99.12` dans lequel on tape successivement

```
./configure
make
```

Puis en tant que root

**make install**

### 7.2.3.2 Utilisation

On lance **xine** en tapant simplement

**xine**

Voilà le résultat avec le skin **CelomaChrome**



**Xine** a le look d'un lecteur de DVD de salon



Vous trouverez d'autres skins ici [http://rpm.pbone.net/index.php3/stat/4/idpl/12097414/dir/fedora\\_1/com/xine-skins-1.8-1.fr.noarch.rpm.html](http://rpm.pbone.net/index.php3/stat/4/idpl/12097414/dir/fedora_1/com/xine-skins-1.8-1.fr.noarch.rpm.html) il sera peut être nécessaire de l'installer en omettant les dépendances (**urpmi -allow-nodeps**).

Les skins vont se retrouver sous **/usr/share/xine/skins** il faudra les copier sous **/usr/local/share/xine/skins** il suffit de les décompresser sous **/usr/local/share/xine/skins**.

**xine** est capable de lire les fichiers vidéo utilisant les codes compatibles MPEG1 (VCD) ou MPEG2 (SVCD) et même MPEG4 (xvid ou DivX). Il lit les DVD dès lors que vous avez installé **libdvdcss** il n'est plus nécessaire d'installer un quelconque plugin (**xine-dvdnav**, **d4d**, **d5d**, ...) tout est inclus dans **xine-lib**

### 7.2.4 MPlayer

Le site officiel est <http://www.mplayerhq.hu/design7/news.html> On récupérera la dernière version stable **MPlayer** qu'on décompresse en tapant

```
tar xvf MPlayer-1.5.tar.xz
```

Cela donne le répertoire **MPlayer-1.5**. Auparavant on va décompresser les codecs windows qu'on récupérera dans la section download du site. On crée d'abord le répertoire d'accueil:

```
mkdir /usr/local/lib/win32
```

Dans lequel on copie les codecs

```
cp essential-20071007.tar.bz2 /usr/local/lib/win32/
```

Qu'on décompresse après s'être placé sous **/usr/local/lib/win32/**

```
tar xvfj essential-20071007.tar.bz2
```

Dans le répertoire **essential-20071007** ainsi obtenu on tape

```
mv * ..
```

Vous pouvez supprimer maintenant le répertoire **/usr/local/lib/win32/essential-20071007**.  
Revenons dans le répertoire **mplayer**, on tape alors

```
./configure --language=fr --enable-gui
```

Je n'ai pas activé **live555** car ça plante à la compilation. En fin de **./configure** on a les principales options qui seront utilisées pour la compilation de **MPlayer**

**Config files successfully generated by ./configure --language=fr --enable-gui !**

```
Install prefix: /usr/local
Data directory: /usr/local/share/mplayer
Config direct.: /usr/local/etc/mplayer
```

```
Byte order: little-endian
Optimizing for: native
```

```
Languages:
Messages/GUI: fr
Manual pages: fr
Documentation: fr
```

```
Enabled optional drivers:
Input: dvdnav ftp pvr tv-v4l2 tv librtmp cddb cdda dvdread vcd dvb networking
Codecs: libdav1d libvpx x264 xvid ffmpeg(internal) real xanim libopus faad2 faac libmpeg2 liba52 mpg123
libtheora libgsm speex libvorbis twolame libmad liblzo OpenJPEG
Audio output: alsa openal jack pulse oss v4l2 sdl mpegpes(dvb)
Video output: v4l2 matrixview opengl sdl pnm jpeg mng mpegpes(dvb) fbdev aa ggi xvixidix cvidix vdpau xv
x11 xover yuv4mpeg md5sum tga
```

```
Disabled optional drivers:
Input: vstream radio tv-v4l1 tv-dshow live555 nemesi bluray smb
Codecs: libaom crystalhd libdv libopencore_amrwb libopencore_amrnb qtx win32 ilbc musepack libdca
toolame gif
Audio output: sndio sun nas esd arts dxr2
Video output: zr zr2 dxr3 dxr2 vesa gif89a svga caca winvidix 3dfx xmga dga xvmc directfb dfbmga bl xvr100
tdfx_vid wii s3fb tdfxfb mga
```

On peut taper **make** si vous rencontrez l'erreur

```
/bin/ld: ffmpeg/libavcodec/libdav1d.o: in function `libdav1d_receive_frame':
libdav1d.c:(.text+0x2d8): undefined reference to `dav1d_get_event_flags'
```

```
./configure --language=fr --enable-gui --disable-libdav1d-lavc
```

```
undefined reference to `__pow_finite'
```

il faudra éditer le fichier **config.mak** et au niveau de la variable **CFLAGS** supprimer **--fast-math** on retape **make**

puis en tant que root **make install**

A présent on peut récupérer les thèmes dont le thème par défaut s'appelle **Blue**. On le copiera sous **/usr/local/share/mplayer/skins**

```
cp Blue-1.13.tar.bz2 /usr/local/share/mplayer/skins
```

Dans ce répertoire on tape ensuite successivement

```
cd /usr/local/share/mplayer/skins
tar xvfj Blue-1.13.tar.bz2
ln -s Blue default
```

C'est fini on peut taper **MPlayer** en tapant

```
gmplayer
```

Et voilà le résultat



La vidéo apparaît dans une fenêtre séparée et non pas dans la petite fenêtre. Autre vue de **MPlayer** en utilisant un skin copiant lecteur bien connu...

**MPlayer** est capable de lire à peu près tous les formats (MPEG1, MPEG2, MPEG4, ...), y compris le format DV brut ! Pour lire les DVD dans le menu (clique bouton droit de la souris sur la fenêtre d'affichage) DVD->Ouvrir un disque, puis DVD->Langues audio->French. Il reconnaît également les conteneurs ogm et les matroska.

Ca n'a pas grand chose à voir avec la vidéo, mais **MPlayer** permet de pouvoir écouter la radio ! Une liste des radios francophones peut se trouver ici [https://doc.ubuntu-fr.org/liste\\_radio\\_france](https://doc.ubuntu-fr.org/liste_radio_france) il suffit d'en choisir une et de taper le lien **mplayer** <http://direct.francebleu.fr/live/fbarmorique-midfi.mp3>

# 8 Streaming vidéo

## 8.1 Présentation

Nous allons nous servir de **vlc**.

## 8.2 Diffuser une vidéo vers un poste en particulier

Il suffit de taper la commande

```
vlc video.avi --sout '#std{access=udp,mux=ts,dst=192.168.26.50:1234}'
```

192.168.26.50 est l'adresse IP du poste client et 1234 le port de diffusion, sur le poste client pour recevoir le flux, il suffit de taper

```
vlc udp://@:1234
```

## 8.3 Faire du streaming sur un réseau

Le streaming permet la réception sur plusieurs postes clients d'un réseau

```
vlc video.avi --sout '#std{access=mms,h,dst=:1234}'
```

pour la réception de n'importe quel poste client, il suffit de taper

```
vlc mms://serveurdiffusion:1234
```

**serveurdiffusion** étant le nom du serveur de diffusion et 1234 le port de diffusion.

pour transcoder puis diffuser en UDP multicast à l'adresse 224.1.0.200 on tapera

```
vlc video.avi --sout '#transcode{vcode=mp4v,acodec=mpga,vb=800,ab=128,deinterlace}:standard{access=udp,mux=ts,dst=224.1.0.200,sap,name="monflux"}'
```

pour diffuser simplement en UDP multicast à l'adresse 224.1.0.200 on tapera

```
vlc video.avi --sout '#duplicate{dst=standard{access=udp,mux=ts,dst=224.1.0.200,sap,name="monflux"}}'
```

pour lire le fichier il suffit de taper

```
vlc udp:@224.1.0.200
```

le flux multicast pouvant être stoppé par les routeurs on pourra rajouter à la commande de diffusion l'option **--ttl X** avec X qui correspond au nombre de routeur.

## 8.4 Vidéo à la demande (VoD)

Il faut lancer le serveur manager de video à la demande accessible sur le serveur de diffusion par **telnet**

```
vlc --extraintf telnet --telnet-password videolan --rtsp-host 192.168.1.11:1234
```

le mot de passe au serveur de diffusion sera **videolan**, son adresse 192.168.1.11 et le port de diffusion 1234, on va maintenant rajouter des vidéos. On se connecte sur le serveur par **telnet**, le port par défaut étant 4212

```
telnet 192.168.1.11 4212
Trying 192.168.1.11...
Connected to 192.168.1.11.
Escape character is '^]'.
Password:
Welcome, Master
>
```

pour rajouter une vidéo après le prompt on tape

```
> new canal1 vod enabled setup canal1 input "/maurice/video/perso/tahiti/danses/danses-marquisiennes-xvid.avi"
```

pour en rajouter une deuxième

```
> new canal2 vod enabled setup canal2 input "/maurice/video/perso/tahiti/dansesmangareva.avi"
```

Maintenant d'un poste client pour lire le **canal1** on va taper

```
vlc rstp://192.168.1.11:1234/canal1
```

## 9 Logiciels de montage vidéo

### 9.1 Présentation

**Kdenlive** est certainement un des logiciels de montage les plus aboutis sous Linux, riche en fonctionnalité et qui permet mille fantaisies de montage mais qui du coup est assez complexe à utiliser, c'est le soft que j'utilise habituellement.

**Flowblade Movie Editor** est également basé sur **mlt** comme **kdenlive**, il est assez déroutant d'utilisation car il se démarque des autres sur le vocabulaire et sur certaines philosophie d'utilisation, c'est un logiciel qui comme **kdenlive** est riche en fonctionnalité et qui reste complexe à utiliser, il n'est pas conseillé pour des montages simples.

**Shotcut** est basé sur **mlt** également, c'est un très bon compromis entre facilité d'utilisation et richesse des fonctionnalités.

**OpenShot Video Editor** a contrario est plus simple d'utilisation et dispose de moins de fonctionnalités, notamment au niveau des effets, en revanche il intègre de base des fonctionnalités d'animation qu'on ne retrouve pas sous **kdenlive**, sur la base notamment de **blender** et **inkscape**.

**Pitivi Video Editor** est encore plus simple d'utilisation, ses fonctionnalités sont assez sommaires mais suffisantes pour débiter, on le privilégiera pour commencer le montage vidéo.

**Cinelerra** est un logiciel de montage évolué et relativement complexe et exigeant.

Cette page ne présente que l'installation de ces logiciels, les tutoriels d'utilisation de [Kdenlive](#), [Cinelerra](#), [OpenShot Video Editor](#), [Pitivi Video Editor](#), [Flowblade Movie Editor](#), [Shotcut](#) et [synfig studio](#) sont dédiées davantage à leur mise en œuvre. A noter également une page sur l'utilisation de [LiVES](#) logiciel conçu à l'origine pour faire des clips vidéo que je ne maintiens plus.

Pour une comparaison de tous ses logiciels vous pouvez consulter cette [autre page](#) et [cette page](#) qui présente les grandes notions du montage vidéo.

Il existe aussi **avidemux** décrit dans le chapitre Transcodage avec avidemux qui permet de faire quelques montages à partir d'une vidéo préexistante même si ce n'est pas sa vocation première.

Tous les outils précédemment cités ne permettent pas de pouvoir faire des arrangements audio pour cela il faudra utiliser les outils de MAO qui font l'objet d'[une page particulière](#).

## 9.2 Outils de montage vidéo

### 9.2.1 KDenlive

#### 9.2.1.1 Présentation

**KDEenlive** est un logiciel de montage vidéo relativement puissant et simple d'utilisation.

#### 9.2.1.2 Installation

Sous Mageia il faut installer les packages suivants

```
urpmi extra-cmake-modules qt5-devel lib64qt5svg-devel lib64kf5bookmarks-devel lib64kf5plotting-devel
```

lib64kf5notifyconfig-devel lib64kf5newstuff-devel lib64kf5notifications-devel lib64qt5qml-devel lib64qt5quick-devel lib64kf5parts-devel kded task-plasma5-minimal lib64kf5crash-devel lib64kf5filemetadata-devel lib64kf5xmlgui-devel lib64exif-devel lib64v4l-devel lib64qt5webkitwidgets-devel lib64kf5purpose-devel lib64kf5declarative-devel lib64qt5quickwidgets-devel lib64qt5multimedia-devel lib64qt5webengine-devel lib64ebur128-devel lib64qt5networkauth-devel lib64xpm-devel kdelibs4-devel lib64v4l-devel lib64qjson-devel ladspa-devel lib64gdk\_pixbuf2.0-devel swig lib64python-devel lib64samplerate-devel mediainfo lib64sox-devel lib64rtaudio-devel

on installera l'outil d'animation intégrée **glaxnimate** qu'on récupérera à l'adresse <https://glaxnimate.mattbas.org> et plus précisément ici <https://gitlab.com/mattbas/glaxnimate/-/releases> et on décompresse l'archive en tapant

```
tar xvfz glaxnimate-0.5.1.tar.bz2
```

cela donne le répertoire **glaxnimate-0.5.1** on installera préalablement **lib64potrace-devel** on se déplace sous **glaxnimate** et on tape

```
mkdir build
```

```
cd build
```

```
cmake ..
```

```
make puis en tant que root make install
```

On revient à **mlt** qu'on trouvera ici <https://www.mltframework.org/> on décompresse l'archive en tapant

```
tar xvfz mlt-7.12.0.tar.gz
```

cela donne le répertoire **mlt-7.12.0**

on tape dans le répertoire de **mlt** dans lequel on tape

```
mkdir build
```

```
cd build
```

```
cmake .. -DSWIG_PYTHON=ON -DMOD_OPENCV=ON -DMOD_GLAXNIMATE=ON
```

voilà ce que ça donne

```
-- The following features have been enabled:
```

- \* GPLv2
- \* GPLv3
- \* Module: avformat
- \* Module: DeckLink
- \* Module: Frei0r
- \* Module: GDK
- \* Module: Glaxnimate (Qt5)
- \* Module: JACKRack
- \* Module: Kdenlive
- \* Module: Normalize
- \* Module: Oldfilm
- \* Module: OpenCV
- \* Module: Movit
- \* Module: Plus
- \* Module: PlusGPL
- \* Module: Qt (Qt5)
- \* Module: Resample
- \* Module: RtAudio
- \* Module: Rubberband



- \* Module: SDL1
- \* Module: SDL2
- \* Module: SoX
- \* Module: vid.stab
- \* Module: Vorbis
- \* Module: XINE
- \* Module: XML
- \* SWIG: Python

-- The following OPTIONAL packages have been found:

- \* JACK
- \* OpenGL
- \* Freetype
- \* Fontconfig
- \* Qt5Svg
- \* Qt5Core
- \* Qt5Gui
- \* Qt5Widgets
- \* Qt5Xml
- \* Qt5
- \* SWIG

-- The following REQUIRED packages have been found:

- \* Threads
- \* OpenCV
- \* PkgConfig
- \* X11
- \* Python3
- \* LibArchive
- \* ZLIB

-- The following features have been disabled:

- \* Tests
- \* Doxygen
- \* Module: NDI
- \* Module: Qt6
- \* SWIG: C#
- \* SWIG: Java
- \* SWIG: Lua
- \* SWIG: Node.js
- \* SWIG: Perl
- \* SWIG: PHP
- \* SWIG: Ruby
- \* SWIG: Tcl

-- Configuring done

-- Generating done

-- Build files have been written to: /usr/local/linux/multimedia/mlt-7.12.0/build

puis on tape **make** et en tant que root

**make install**

dans le fichier `/etc/ld.so.conf` on rajoutera la ligne `/usr/local/lib64/mlt-7` et on tapera **ldconfig**. Il faudra veiller à ce qu'aucune ancienne installation de **mlt** soit présente, au besoin supprimez la. Pour voir si **mlt** est bien compilé avec toutes les options, vous pouvez taper ces commandes qui donnent

**melt -query "producers"**

**melt -query "filters"**

pour voir si frei0r a bien été intégré on tapera

```
melt -query | grep frei0r
```

et voilà une partie du résultat

```
- frei0r.3dflippo
- frei0r.B
- frei0r.G
- frei0r.IIRblur
- frei0r.R
- frei0r.aech0r
- frei0r.alpha0ps
- frei0r.alphagrad
- frei0r.alphaspot
- frei0r.balanc0r
- frei0r.baltan
- frei0r.bgssubtract0r
- frei0r.bluescreen0r
- frei0r.brightness
- frei0r.bw0r
- frei0r.c0rners
- frei0r.cairogradient
- frei0r.cairoimagegrid
- frei0r.cartoon
(...)
```

Revenons à **kdenlive**, l'URL officiel de **kdenlive** [kdenlive.org](http://kdenlive.org). A l'adresse <https://invent.kde.org/multimedia/kdenlive> on récupérera la dernière version qu'on décompresse en tapant

```
tar xvfj kdenlive-release-22.12.tar.bz2
```

cela donne le répertoire **kdenlive-release-22.12** dans lequel on tape

```
mkdir build;cd build
cmake ..
```

voilà le résultat

```
-- Found KF5: success (found suitable version "5.101.0", minimum required is "5.92.0") found components:
DBusAddons
-- Checking for module 'mlt+-7'
-- Found mlt+-7, version 7.12.0
-- Found MLT: /usr/local/lib64/libmlt-7.so (found suitable version "7.12.0", minimum required is "7.10.0")
-- Found MLT++: /usr/local/lib64/libmlt+-7.so
-- Looking for include file malloc.h
-- Looking for include file malloc.h - found
-- Looking for include file pthread.h
-- Looking for include file pthread.h - found
-- Found KF5 Purpose, filesharing enabled
-- Found LibV4L2: /usr/include
-- Looking for include file linux/input.h
-- Looking for include file linux/input.h - found
-- Found SharedMimeInfo: /bin/update-mime-database (found version "2.2")
-- The following RUNTIME packages have been found:

* LibV4L2, video4linux support libraries, <https://linuxtv.org/downloads/v4l-utils/>
  Required for better webcam support

-- The following OPTIONAL packages have been found:

* KF5JobWidgets (required version >= 5.101.0)
* Freetype
* Fontconfig
```

- \* KF5Service (required version >= 5.101.0)
- \* KF5Auth (required version >= 5.101.0)
- \* Qt5Gui (required version >= 5.15.2)
- \* KF5Sonnet (required version >= 5.101.0)
- \* KF5I18n (required version >= 5.101.0)
- \* Qt5Qml (required version >= 5.15.7)
- \* Qt5QmlModels (required version >= 5.15.7)
- \* KF5Package (required version >= 5.101.0)
- \* KF5DocTools (required version >= 5.92.0)
- \* KF5Crash (required version >= 5.92.0)
- \* KF5Purpose (required version >= 5.92.0)
- \* PkgConfig

-- The following REQUIRED packages have been found:

- \* ECM (required version >= 5.92.0)
- \* KF5Archive (required version >= 5.92.0)
- \* KF5KIO (required version >= 5.92.0)
- \* KF5NotifyConfig (required version >= 5.92.0)
- \* KF5NewStuff (required version >= 5.92.0)
- \* KF5Codecs (required version >= 5.101.0)
- \* KF5XmlGui (required version >= 5.92.0)
- \* KF5Notifications (required version >= 5.92.0)
- \* KF5GuiAddons (required version >= 5.92.0)
- \* Gettext
- \* KF5TextWidgets (required version >= 5.92.0)
- \* KF5IconThemes (required version >= 5.92.0)
- \* KF5Declarative (required version >= 5.92.0)
- \* KF5Solid (required version >= 5.92.0)
- \* KF5FileMetaData (required version >= 5.92.0)
- \* KF5CoreAddons (required version >= 5.101.0)
- \* Qt5Widgets
- \* Qt5Svg
- \* Qt5Quick
- \* Qt5QuickControls2
- \* Qt5Concurrent
- \* Qt5QuickWidgets
- \* Qt5Multimedia
- \* Qt5NetworkAuth
- \* KF5DBusAddons (required version >= 5.92.0)
- \* KF5 (required version >= 5.92.0)
- \* Qt5DBus
- \* Qt5
- \* MLT (required version >= 7.10.0), Multimedia framework, <<https://mltframework.org>>  
Required to do video processing
- \* Qt5Core
- \* Qt5UiPlugin  
Required to build Qt Designer plugins
- \* SharedMimeInfo, A database of common MIME types, <<https://freedesktop.org/wiki/Software/shared-mime-info/>>

-- The following features have been disabled:

- \* QCH, Source code documentation in QCH format (for e.g. Qt Assistant, Qt Creator & KDevelop)

-- Configuring done

-- Generating done

-- Build files have been written to: /usr/local/linux/multimedia/kdenlive-release-22.12/build

puis

make

puis en tant que root

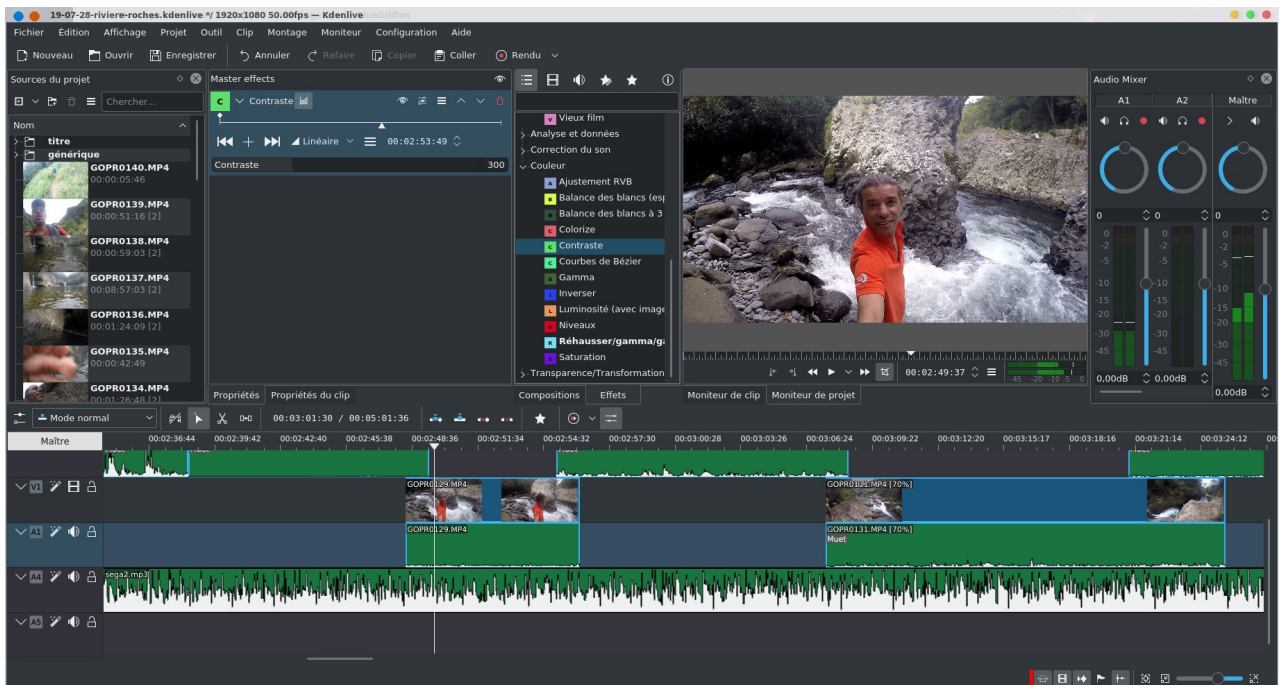
## make install

attention par défaut **kdenlive** est installé sous **/usr/bin**

Ce n'est pas tout à fait fini, c'est vous voulez activer l'algorithme **DaSiamRPN** pour le suivi de mouvement (motion tracking) il faudra récupérer les modèles sous [https://docs.kdenlive.org/en/effects\\_and\\_compositions/effect\\_groups/alpha\\_manipulation/motion\\_tracker.html?highlight=motion%20tracker#id2](https://docs.kdenlive.org/en/effects_and_compositions/effect_groups/alpha_manipulation/motion_tracker.html?highlight=motion%20tracker#id2) et les placer sous la homedirectory **~/local/share/kdenlive/opencvmodels**

### 9.2.1.3 Utilisation

Voici **kdenlive** lancé avec un projet de montage



Vous trouverez un tutoriel d'utilisation par [ici](#) pour les versions 19.X et supérieure. Pour les versions 18.X et antérieure, le tutorial est [là](#).

## 9.2.2 Pitivi Video Editor

### 9.2.2.1 Installation

Le site officiel est <http://www.pitivi.org/> on fera une installation en utilisant [flatpak](#). Sous une Mageia 8, il suffira de taper

**urpmi flatpak**

**flatpak remote-add --if-not-exists flathub https://flathub.org/repo/flathub.flatpakrepo**  
**flatpak install flathub org.pitivi.Pitivi**

A noter que la commande **flatpak** peut s'exécuter en tant que simple utilisateur et dans ce cas **Pitivi** sera installé sous **~/local/share/flatpak** et les paramètres de configuration sous **~/var/app/org.pitivi.Pitivi/config/**

C'est tout ! Trop facile...

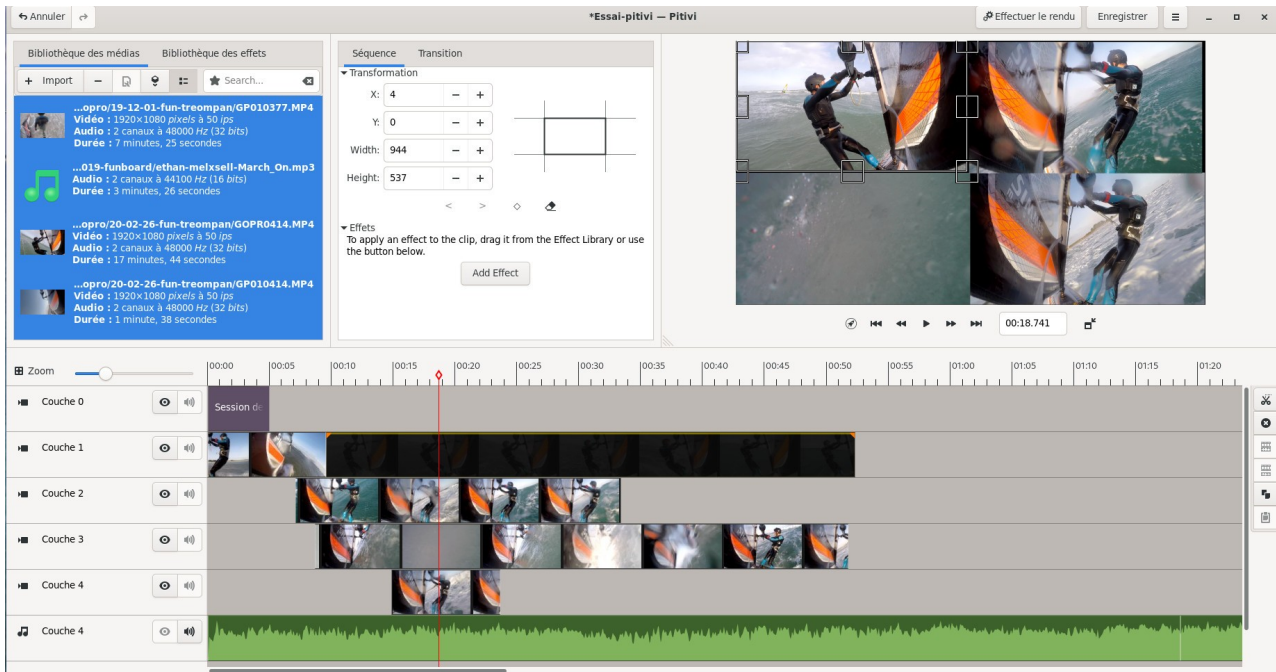
En cas de mise à jour on tapera

**flatpak update**

### 9.2.2.2 Utilisation

On retrouvera **Pitivi** intégré au menu de lancement des applications de votre environnement de bureau préféré (enfin du

moins c'est comme ça que se passe avec XFCE). Et voilà ce que ça donne avec un projet chargé.



Pour la suite, je vous invite à consulter le [tutoriel d'utilisation de Pitivi](#).

## 9.2.3 Flowblade Movie Editor

### 9.2.3.1 Installation

Le site officiel de **flowblade** est <https://jliljebl.github.io/flowblade/> mais on fera une installation via **flatpak** en tapant **flatpak --user install flathub io.github.jliljebl.Flowblade**

Voilà le résultat

Looking for matches...

Required runtime for io.github.jliljebl.Flowblade/x86\_64/stable (runtime/org.gnome.Platform/x86\_64/3.36) found in remote flathub

Do you want to install it? [Y/n]: y

io.github.jliljebl.Flowblade permissions:  
ipc pulseaudio x11 devices file access [1]

[1] host

ID	Arch	Branch	Remote	Download
1. [✓] org.gnome.Platform	x86_64	3.36	flathub	271,2 MB / 322,9 MB
2. [✓] org.gnome.Platform.Locale	x86_64	3.36	flathub	3,2 MB / 322,9 MB
3. [✓] org.freedesktop.Platform.GL.default	x86_64	19.08	flathub	94,9 MB / 94,9 MB
4. [✓] org.freedesktop.Platform.openh264	x86_64	2.0	flathub	918,1 kB / 916,8 kB
5. [✓] io.github.jliljebl.Flowblade	x86_64	stable	flathub	42,4 MB / 42,7 MB
6. [✓] io.github.jliljebl.Flowblade.Locale	x86_64	stable	flathub	8,3 kB / 6,3 MB

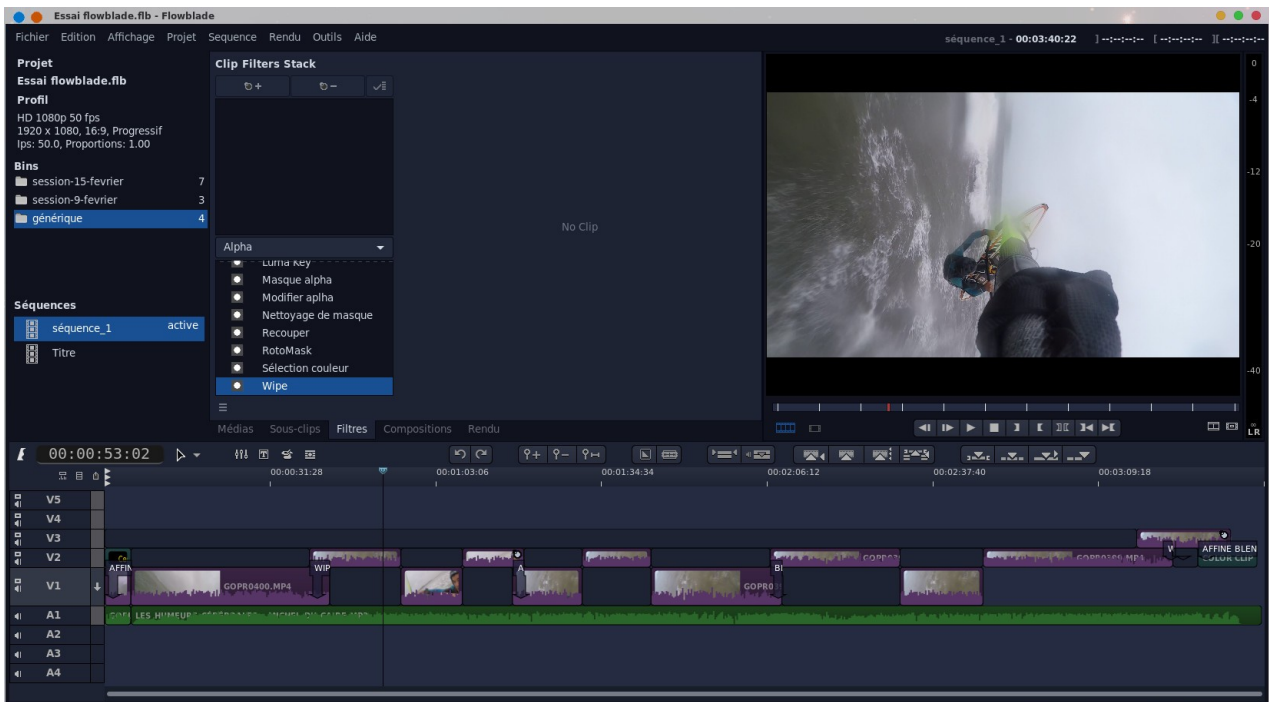
Installation complete

il faudra également installer le package **gmic**. Pour une mise à jour il faudra simplement taper

**flatpak update**

### 9.2.3.2 Utilisation

Le logiciel se lance à partir du menu graphique de votre distribution préférée et voilà le résultat:



Vous trouverez [ici](#) un tutoriel sur l'utilisation de **Flowblade**.

## 9.2.4 Shotcut

### 9.2.4.1 Installation

Le site officiel est <https://shotcut.org/> on fera une installation par **flatpak** en tapant

```
flatpak install flathub org.shotcut.Shotcut
```

voilà le résultat

Looking for matches...

Remote 'flathub' found in multiple installations:

- 1) system
- 2) user

Which do you want to use (0 to abort)? [0-2]: 2

Required runtime for org.shotcut.Shotcut/x86\_64/stable (runtime/org.kde.Platform/x86\_64/5.14) found in remote flathub

Do you want to install it? [Y/n]: y

org.shotcut.Shotcut permissions:

```
ipc network pulseaudio wayland x11 devices shm file access [1] dbus access [2]
```

[1] host, xdg-config/kdeglobals:ro

[2] com.canonical.AppMenu.Registrar

ID	Arch	Branch	Op	Remote	Download
1. [✓] org.kde.Platform		x86_64	5.14	i flathub	104,1 MB / 324,4 MB
2. [✓] org.freedesktop.Platform.GL.nvidia-440-64		x86_64		1.4 u	flathub 101,5 MB / 102,0 MB
3. [✓] org.kde.Platform.Locale		x86_64	5.14	i flathub	546,2 kB / 337,1 MB

4. [✓] org.shotcut.Shotcut

x86\_64

stable

i

flathub

35,7 MB / 38,6 MB

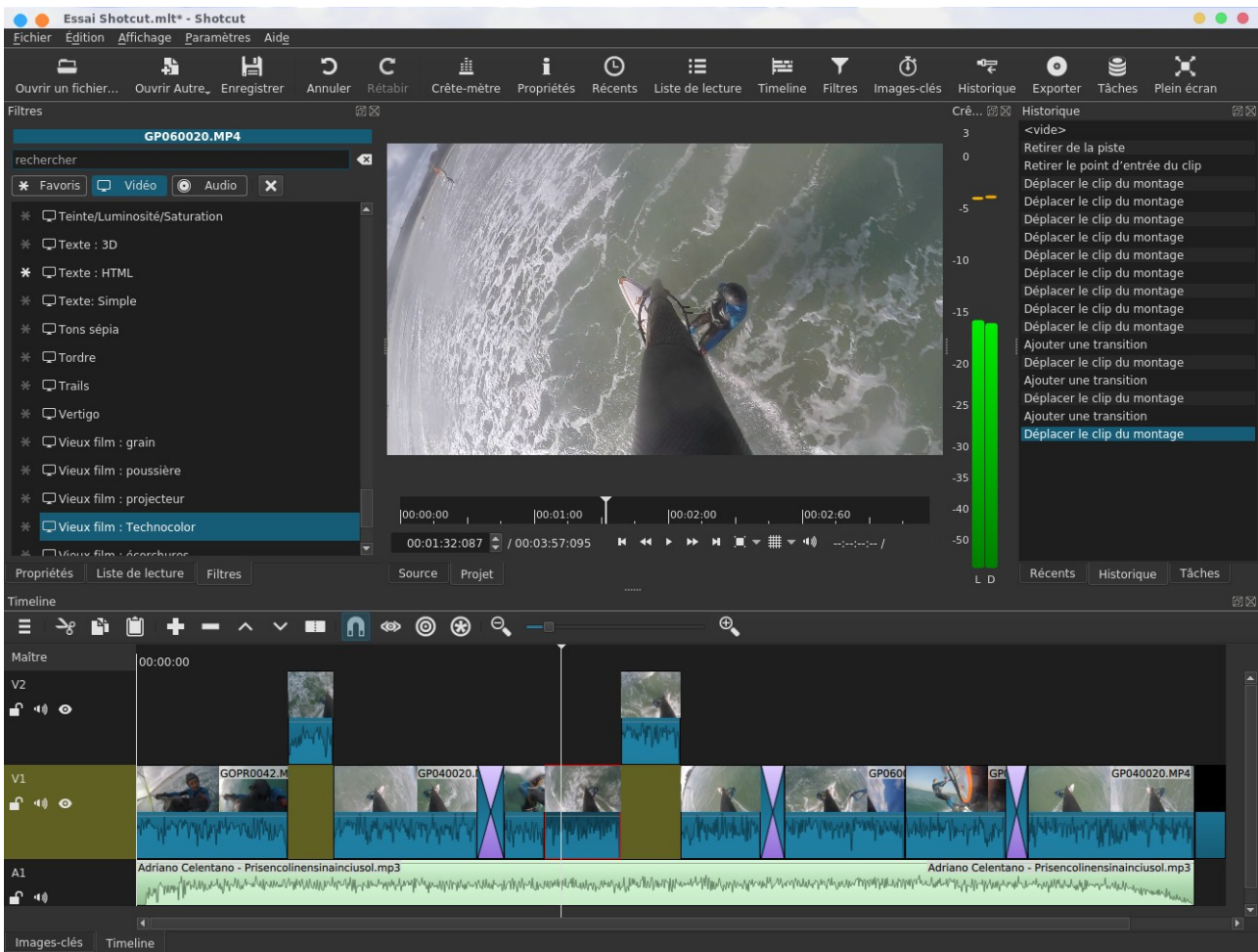
Changes complete.

Pour une mise à jour il faudra simplement taper

flatpak update

### 9.2.4.2 Utilisation

On lancera **shotcut** à partir du menu graphique de votre environnement préféré (enfin du moins c'est comme ça que ça se passe avec **XFCE**). Voilà le résultat avec un projet en cours.



On trouvera par [ici](#) un tutoriel d'utilisation de **Shotcut**.

## 9.2.5 Cinelerra

### 9.2.5.1 Installation

On récupérera **cinelerra** à l'URL <http://heroinewarrior.com> on décompresse l'archive en tapant

```
tar xvf cinelerra-8-src.tar.xz
```

Cela donne le répertoire **cinelerra-8** on doit installer en tant que root les packages

**texinfo**

**yasm**

**lib64xv1-devel**

j'édite le fichier configure et je modifie ainsi la ligne suivante

**CUDA\_DIR=/usr/lib64/cuda-toolkit-root-dir**

on tape ensuite

**./configure --enable-opengl**

j'ai édité le fichier **cinelerra-8/thirdparty/pulseaudio-13.0/src/Makefile** et j'ai supprimé la ligne

**@GSETTINGS\_RULES@**

je tape ensuite **make** et je tombe sur les erreurs

```
/bin/ld : x86_64/libquicktime.a(vaapi_encode_h264.o) : dans la fonction
« vaapi_encode_h264_write_identifier » :
/usr/local/linux/multimedia/cinelerra-7.4/quicktime/thirdparty/ffmpeg-3.3.3/libavcodec/vaapi_encode_h264.c:619
: référence indéfinie vers « vaQueryVendorString »
```

et

```
qth265.c:(.text+0x69c) : référence indéfinie vers « x265_api_get_179
```

ou bien encore

```
g++ -o ../bin/cinelerra `cat x86_64/objs`
/bin/ld: ../thirdparty/pulseaudio-13.0/src/.libs/libpulsecommon-13.0.a(libpulsecommon_13.0_la-x11prop.o):
référence au symbole non défini « xcb_get_setup »
/bin/ld : /usr/lib64/libxcb.so.1 : erreur lors de l'ajout de symboles : DSO manquant dans la ligne de commande
```

ou

```
g++ -o ../bin/cinelerra `cat x86_64/objs`
/bin/ld : ../thirdparty/libsndfile-1.0.28/src/.libs/libsndfile.a(libsndfile_la-flac.o) : dans la fonction
« sf_flac_meta_get_vorbiscomments » :
flac.c:(.text+0xcbd) : référence indéfinie vers « FLAC__metadata_object_vorbiscomment_find_entry_from »
/bin/ld : ../thirdparty/libsndfile-1.0.28/src/.libs/libsndfile.a(libsndfile_la-flac.o) : dans la fonction
« flac_write_strings » :
flac.c:(.text+0x1261) : référence indéfinie vers « FLAC__metadata_object_new »
/bin/ld : flac.c:(.text+0x137a) : référence indéfinie vers
« FLAC__metadata_object_vorbiscomment_entry_from_name_value_pair »
/bin/ld : flac.c:(.text+0x1396) : référence indéfinie vers
« FLAC__metadata_object_vorbiscomment_append_comment »
/bin/ld : ../thirdparty/libsndfile-1.0.28/src/.libs/libsndfile.a(libsndfile_la-flac.o) : dans la fonction « flac_close » :
flac.c:(.text+0x16db) : référence indéfinie vers « FLAC__metadata_object_delete »
/bin/ld : ../thirdparty/libsndfile-1.0.28/src/.libs/libsndfile.a(libsndfile_la-ogg_vorbis.o) : dans la fonction
« ogg_vorbis_open » :
ogg_vorbis.c:(.text+0xc37) : référence indéfinie vers « vorbis_version_string »
```

on modifie le fichier **quicktime/Makefile**

à la ligne suivante comme ceci

```
LIBS := $(OUTPUT) $(MPEG3_DIR)/$(OBJDIR)/libmpeg3.a -lpng -lz -lpthread -ldl -lm -llzma -lrt -lva -lX11 -lva-x11 -lva-drm -lvdpaudio -lnuma -lx265 -lxcb -ldbus-1 -lsystemd -lFLAC -lvorbis
```

on fera pareil dans le fichier **cinelerra/Makefile**

```
LIBS := \
-L./\
-L./guicast -lva -lX11 -lva-x11 -lva-drm -lvdpaudio -lnuma -lx265 -ldbus-1 -lsystemd -lFLAC -lvorbis
```

on retape **make** j'puis en tant que root



**make install**

j'ai du créer le lien suivant pour accéder à **cinelerra**

**ln -s /usr/local/linux/multimedia/cinelerra-8/bin/cinelerra /usr/local/bin/cinelerra**

si vous avez au lancement une erreur du style

**MWindow::init\_shm0: WARNING:/proc/sys/kernel/shmmax is 0x2000000.  
It should be a least 0x7FFFFFFF for cinelerra**

il faut éditer le fichier **/etc/sysctl.conf** et rajouter tout à la fin

**# neutralisation du message d'erreur cinelerra  
kernel.shmmax = 0x7ffffff**

Pour appliquer la modif il faut taper en tant que root

**sysctl -p**

Si ça plante au démarrage avec un message du genre

**signal\_entry: got SIGSEGV my pid=17430 execution table size=0:  
SigHandler::signal\_handler total files=0  
Abandon**

En effet au premier lancement de l'application il tente de créer un fichier sous **/usr/lib64/ladspa/** et comme un simple utilisateur n'a pas les droits, il plante. Pour le résoudre il faut d'abord lancer **cinelerra** en tant que root puis à nouveau en tant que simple utilisateur et là ça passe.

### 9.2.5.2 Utilisation

Voilà ce que ça donne en tapant **cinelerra**



L'ergonomie est particulièrement brouillonne, le logiciel n'a rien d'intuitif, reportez vous à cette [page](https://www.funix.org) sur le site <https://www.funix.org> pour avoir les bases de l'utilisation de **cinelerra**.

## 9.2.6 Fork de Cinelerra

### 9.2.6.1 Présentation

Il se trouve qu'il existe des forks de **cinelerra** qui sont très prometteurs, ils sont regroupés à cette adresse <https://cinelerra-cv.org/>. Pour être précis sur cette même page on va trouver deux variantes, **Cinelerra GG** ou la version Goodguy's qui est parti d'un développement personnel et **Cinelerra CV** qui est la version de la communauté (community version) qui apporte quelques fonctions supplémentaires et se veut plus stable, à noter que cette dernière version bien qu'étant un fork ne souhaite pas diverger du projet original et fusionne (merge) avec ses nouvelles évolutions, lire la page A propos qui explique un peu tout ça par là <https://cinelerra-cv.org/about.php>. A noter que la version **Cinelerra HV** (Heroin Virtual) n'est pas un fork mais est ni plus ni moins que la version originale présentée plus [haut](#).

Il semblerait que ce soit la version GG qui soit la plus active.

A noter qu'il existe également **Lumiera** (site web <http://lumiera.org/>), il est né de la réécriture du code de **Cinelerra HV** mais a divergé dans un projet distinct, il est aujourd'hui dans une version pre-alpha non utilisable.

### 9.2.6.2 Installation

Au final j'ai choisi la version GG qui a l'air de plus se distinguer de la version originale, la page de téléchargement est par là <https://www.cinelerra-gg.org/> on décompresse l'archive en tapant :

```
tar xvfz cin_5.1.20221231-src.tgz
```

cela donne le répertoire cinelerra-5.1, on y tape :

```
./autogen.sh  
./configure --with-opencv=git --with-gl
```

cela donne

Reason	Package
-----	-----
needed	libbthread
forced	a52dec
forced	djbbft
disabled	encore
disabled	audiofile
disabled	esound
forced	ffmpeg
forced	giflib
needed	ilmBase
needed	ladspa
forced	lame
forced	libraw1394
needed	libdv
forced	libogg
forced	libtheora
forced	libuuid
forced	libvorbis
forced	mjpegtools
needed	openExr
needed	openexr
forced	openjpeg
forced	twolame
forced	x264
forced	x265
needed	lv2
needed	lilv
needed	sratom
needed	serd
needed	sord
needed	suil

needed	ffnvcodec
needed	libdpx
shared	-ltheoraenc
shared	-ltheoradec
shared	-logg
shared	-lvorbis
shared	-lvorbisfile
shared	-logg
shared	-llmath
shared	-llmThread
shared	-llx
shared	-lpthread
shared	-lfftw3
shared	-lFLAC
shared	-lavc1394
shared	-lrom1394
shared	-licc61883
shared	-ljpeg
shared	-lsndfile
shared	-laom
shared	-ldav1d
shared	-lwebp
shared	-ltiff
shared	-lvp8
shared	-lopus
system	-lX11
system	-lXext
system	-lXinerama
system	-lXfixes
system	-lbz2
system	-lfontconfig
system	-lfreetype
system	-llzma
system	-lpng
system	-lpthread
system	-lz
system	-ljpeg
system	-lvp8
system	-lva
system	-lva-x11
system	-lva-drm
system	-lGL
system	-lGLU
system	-lXv
system	-lXft
system	-lasound
system	-lpulse-simple
system	-lpulse
system	-lusb-1.0
system	-ldl
system	-lnuma

**using: with-gl**  
**using: with-xft**  
**using: with-xxf86vm**  
**using: with-oss**  
**using: with-alsa**  
**using: with-firewire**  
**using: with-ogg**  
**using: with-dv**  
**using: with-dvb**  
**using: with-ladspa**  
**using: with-video4linux2**  
**using: without-esound**  
**using: with-pulse**

```
using: with-pactl
using: with-openexr
using: with-lv2
using: without-commercial
using: with-giflib
using: with-libzmpg
using: with-libdpx
using: with-shuttle
using: with-shuttle_usb
using: with-xv
using: with-vaapi
using: with-udpau
using: without-cuda
using: with-nv
using: with-wintv
using: with-x10tv
using: with-jobs = 14
using: exec-name = cin
using: with-cinlib = /usr/local/lib64/cin
using: with-cindat = /usr/local/share/cin
using: with-config-dir = $HOME/.bcast5
using: with-nested-dir = $HOME/Videos
using: with-snap-dir = $HOME/Pictures
using: with-browser = firefox
using: with-plugin-dir = /usr/local/lib64/cin/plugins
using: with-ladspa-dir = /usr/local/lib64/cin/ladspa
using: with-opencv = git
using: with-git-ffmpeg = no
using: with-noelision = auto
using: with-booby = no
using: with-clang = no

using: thirdparty build = yes
using: single-user = no
using: static-build = no
using: ladspa-build = yes
```

ensuite on tape comme d'habitude

**make**

ensuite en tant que root

**make install**

Si vous avez l'erreur suivante au lancement

```
PluginServer::open_plugin: load_obj /usr/local/lib64/cin/plugins/opencv/findobj.plugin =
/usr/local/lib64/cin/plugins/opencv/findobj.plugin: undefined symbol:
_ZN2cv14findHomographyERKNS_11_InputArrayES2_idRKNS_12_OutputArrayEid
int PluginServer::open_plugin(int, Preferences*, EDL*, Plugin*):
```

je ne vois pour l'instant comme possibilité que de désactiver l'option **OpenCV** en supprimant le répertoire `/usr/local/lib64/cin/plugins/opencv`

### 9.2.6.3 Utilisation

On lance le logiciel en tapant **cin**, l'interface est assez similaire à l'original, il y a quand même quelques nettes améliorations sur la présentation, c'est à surveiller, il devrait faire de l'ombre à la version originale.



## 9.2.7 OpenShot Video

### 9.2.7.1 Installation

**OpenShot** est un logiciel de montage récent assez similaire à **kdenlive** le site officiel est <https://www.openshot.org/>. On installera préalablement les packages suivantes **lib64qt5multimedia-devel**, **qtmultimedia5**, **lib64qt5multimediawidgets-devel**, **ruby-devel**, **python3-httpplib2**, **python3-qt5-core**, **python3-qt5-widgets**, **python3-qt5-webengine**, **python3-sip-devel**, **python3-qt5-webenginewidgets**, **lib64zmq-devel**, **lib64cppzmq-devel**, **inkscape**, **python3-pyzmq**, **python-pygoocanvas** et **python3-httpplib2**. il faudra avoir **python** avec une version supérieure à 3 et **blender**.

On installera **unittest** qu'on récupérera par là <https://github.com/unittest-cpp> on décompresse l'archive en tapant :

```
tar xvfz unittest-cpp-2.0.0.tar.gz
```

cela donne le répertoire **unittest-cpp-2.0.0** dans lequel on tape

```
mkdir build
cd build
cmake ..
make
make install
```

maintenant on revient aux bibliothèques d'**openshot**. On décompresse l'archive en tapant

```
tar xvfz libopenshot-audio-0.3.0.tar.gz
```

dans le répertoire **libopenshot-audio-0.3.0** on tape

```
mkdir build
cd build
cmake .. -DCMAKE_BUILD_TYPE=Release
make
```

et en tant que root

**make install**

on décompresse l'archive en tapant

**tar xvfz libopenshot-0.3.0.tar.gz**

dans le répertoire **libopenshot-0.3.0** on va taper

**mkdir build**

**cd build**

**cmake .. -DCMAKE\_BUILD\_TYPE=Release**

voilà le résultat

**-- The following OPTIONAL packages have been found:**

- \* ImageMagick
- \* cppzmq
- \* PkgConfig
- \* babl
- \* OpenCV (required version >= 4)
- \* Threads
- \* PythonInterp (required version >= 3)
- \* PythonLibs (required version >= 3)
- \* Ruby

**-- The following REQUIRED packages have been found:**

- \* OpenShotAudio (required version >= 0.3.0)
- \* jsoncpp
- \* Qt5Widgets
- \* Qt5Svg
- \* FFmpeg
- \* OpenMP
- \* ZeroMQ
- \* Protobuf (required version >= 3)
- \* Qt5Core (required version >= 5.15.7)
- \* Qt5Gui
- \* Qt5
- \* SWIG (required version >= 3.0)

**-- The following features have been disabled:**

- \* Coverage, analyze test coverage and generate report
- \* IWYU (include-what-you-use), Scan all source files with 'iwyu'
- \* Unit tests, Compile unit tests for library functions

**-- The following OPTIONAL packages have not been found:**

- \* Resvg
- \* Doxygen
- \* Catch2

**-- Configuring done**

**-- Generating done**

**-- Build files have been written to: /usr/local/linux/multimedia/libopenshot-0.3.0/build**

on tape ensuite

**make**

puis en tant que root **make install**

On revient enfin à **openshot** on y décompresse l'archive en tapant

```
tar xvfz openshot-qt-3.0.0.tar.gz
```

cela donne le répertoire **openshot-qt3.0.0** on tape en tant que root

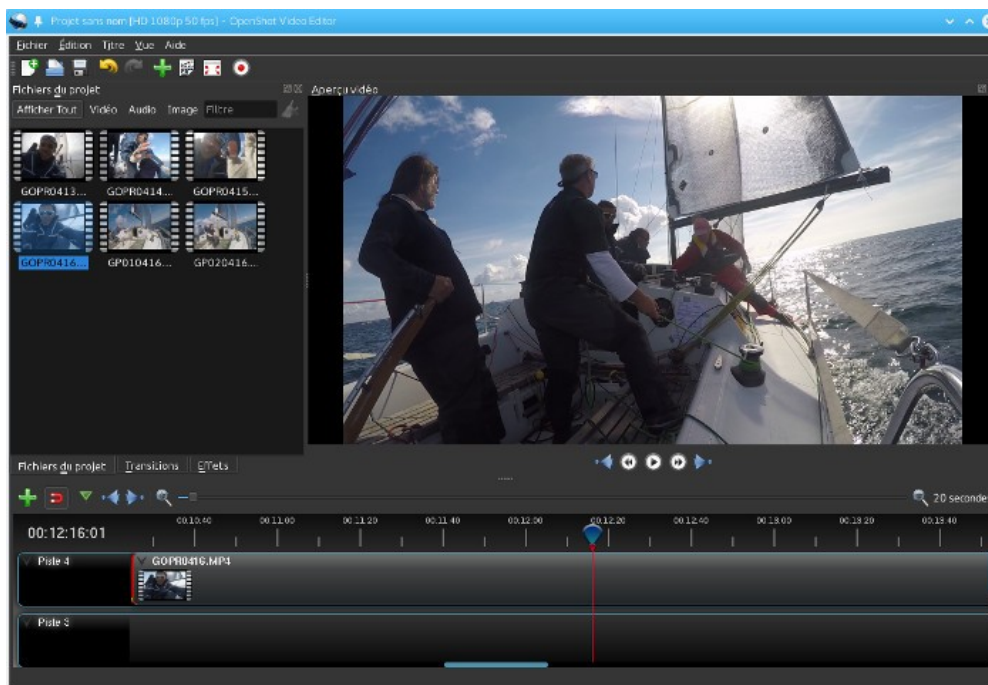
```
python setup.py install
```

### 9.2.7.2 Utilisation

Il suffit de taper dans un shell **openshot-qt**, au besoin il sera peut être nécessaire de taper préalablement

```
export PYTHONPATH=$PYTHONPATH:/usr/lib/python3.8/site-packages/
```

voilà la fenêtre principale avec des vidéos importées.



Pour plus d'info sur l'utilisation merci de consulter le [tutoriel](#).

## 9.3 Outils divers

### 9.3.1 Synfig Studio

**Synfig Studio** est un atelier de création d'animations qui peuvent ensuite être intégrées dans des logiciels de montage. C'est pour ça qu'on le retrouve dans ce document.

#### 9.3.1.1 Installation

Le site officiel est <http://www.synfig.org/>

Avant d'aller plus loin il faudra sans doute installer les packages suivants **lib64magick-devel**, **lib64xml++2.6-devel**, **lib64graphicsmagick-devel**, **lib64mng-devel** la dernière version 1.4.4 n'est pas compatible avec la dernière version de **ffmpeg** on installera donc la version de développement qu'on récupérera en tapant

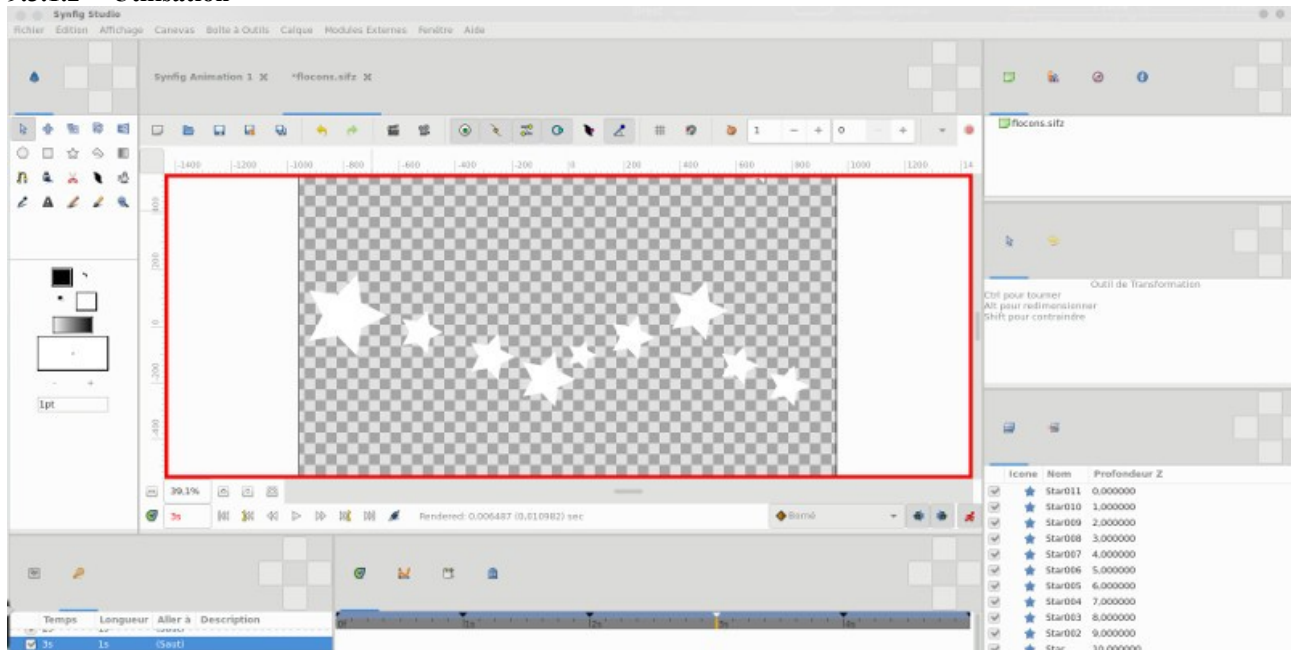
```
git clone https://github.com/synfig/synfig.git synfig
```

cela donne le répertoire **synfig** dans lequel on tape

```
./2-build-production.sh
```

l'exécutable se trouve sous **\_production/build/bin/synfigstudio** on pourra en faire un lien sous **/usr/local/bin**

### 9.3.1.2 Utilisation



Voilà ce que ça donne quand on lance **synfigstudio**, à gauche la "boite à outils", à droite les propriétés des outils, au milieu l'écran principal et en bas la partie pour les images clés (keyframes) et l'animation.

Pour le reste je vous invite à consulter le tutorial par [ici](#).

### 9.3.2 Blender

**Blender** est un logiciel de création 3D, il est nécessaire pour **OpenShot Video** pour créer les animations. Le site officiel est <http://www.blender.org/> avant d'aller plus loin il faudra installer les packages **scons**, **openimageio**, **lib64spnav-devel**, **lib64OpenEXR-devel**, **lib64ilmbase-devel**, **lib64freealut-devel**, **lib64openal-devel**, **lib64fftw-devel**, **lib64gomp-devel** et **python3-devel**, **python3-cython**

On récupérera d'abord **libdecor** à l'adresse <https://gitlab.gnome.org/jadahl/libdecor> on décompresse l'archive en tapant

```
tar xvfz libdecor-0.1.1.tar.xz
```

cela donne le répertoire **libdecor-0.1.1** dans lequel on tape

```
meson build --buildtype release
```

puis en tant que root

```
meson install -C build
```

on revient à **blender** qu'on récupérera en tapant

```
git clone --branch v3.4.1 git://git.blender.org/blender.git
```

cela donne le répertoire **blender** j'ai également dû récupérer les fichiers **languages** et **fr.po**, on placera le premier sous **blender/release/datafiles/locale** et le deuxième sous **blender/release/datafiles/locale/po**, on revient à blender dans lequel on tape

```
mkdir build
```

```
cd build
```

```
cmake .. -DFFMPEG=/usr/local -DWITH_CYCLES=ON -DWITH_CYCLES_STANDALONE=ON -DWITH_CYCLES_STANDALONE_GUI=ON -DWITH_CYCLES_EMBREE=OFF
```

s'il ne trouve pas **OpenImageIO**, il sera peut être nécessaire d'installer le package **lib64openimageio-devel**, comme ça m'installe un tas de dépendances qui vont écraser mes personnalisations, j'ai choisi d'installer strictement les packages nécessaires à **blender** avec l'option **--nodeps** pour ne pas vérifier les dépendances.



```
rpm -ivh --nodeps lib64openimageio-devel-2.4.6.1-6.mga9.x86_64.rpm
```

les options du **cmake** permettent de compiler le moteur de rendu **CYCLES** qui est jugé très supérieur à l'ancien, en tout cas il semblerait que par défaut **OpenShot Video Editor** qui utilise **blender** fait appel à ce moteur, car sinon par défaut ce moteur n'est pas compilé, revenons à la compilation de **blender**, on tape maintenant **make**

```
/usr/local/linux/multimedia/blender/intern/locale/boost_locale_wrapper.cpp:112:10: erreur: « cout » n'est pas un membre de « std »
```

dans le fichier j'ai rajouté dans les includes

```
#include <iostream>
```

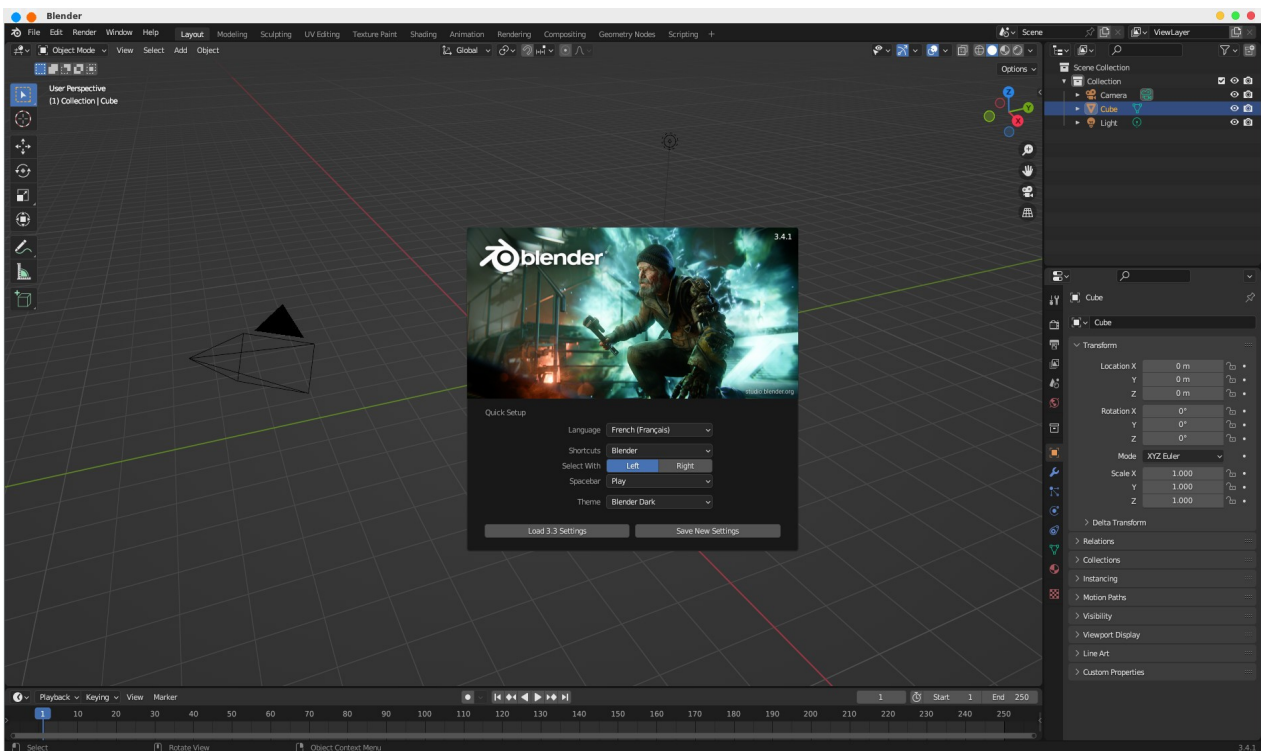
on retape **make** puis en tant que **root**

```
make install
```

j'ai créé ce lien pour qu'il soit accessible de partout (à adapter suivant votre config)

```
ln -s /usr/local/linux/multimedia/blender/build/bin/blender /usr/local/bin
```

et voilà ce que ça donne quand on le lance



## 10 Les outils de transcodage et de rippage de DVD

### 10.1 Présentation

Les outils présentés dans cette page permettent de transcoder des fichiers vidéo, c'est à dire changer le format (vidéo/audio) d'une vidéo, et de ripper des DVD pour en faire des fichiers vidéo de moindre taille. Un cas pratique est de prendre un fichier au format MPEG2 (DVD par exemple) et de le convertir en utilisant un codec vidéo HEVC dans un conteneur MP4 ou vice-versa.

Pour ce faire on utilisera les outils [Handbrake](#) ou [mencoder](#). Pour mémoire pendant longtemps `dvd::rip` basé sur

**transcode** a été la référence sous linux, c'est maintenant un projet mort. Je présente également dans cette page [avidemux](#) qui me sert à traiter les enregistrements de ma box internet en supprimant les pubs et en recadrant.

Dernière précision, cette page n'a pas pour vocation de faire l'apologie de la copie pirate de DVD, personnellement j'ai le besoin de faire une copie de mes DVD pour d'une part faire une sauvegarde en cas de détérioration du disque et d'autre part pour pouvoir visualiser le film sur un lecteur embarqué (dans la voiture par exemple). Je rajoute que le droit à la [copie privée](#) existe, en contrepartie de ce droit nous payons une taxe sur les consommables informatiques (disque dur, DVD, etc.) et appareils multimédia.

## 10.2 Transcodage avec avidemux

### 10.2.1 Présentation

**avidemux** est un logiciel permettant de convertir des fichiers vidéo, mais il est bien plus que cela, il permet de réaliser certains traitements vidéo et audio ainsi que de récupérer des fichiers avi "cassés".

### 10.2.2 Installation

Le site officiel est <http://www.avidemux.org>, on y récupère l'archive qu'on décompresse en tapant

```
tar xvzf avidemux_2.8.1.tar.gz
```

Cela donne le répertoire **avidemux\_2.8.1** dans lequel on tape

```
bash bootstrap.bash
```

Dans le répertoire **install/usr** on va trouver les fichiers utiles qu'il faudra copier manuellement en tant que root

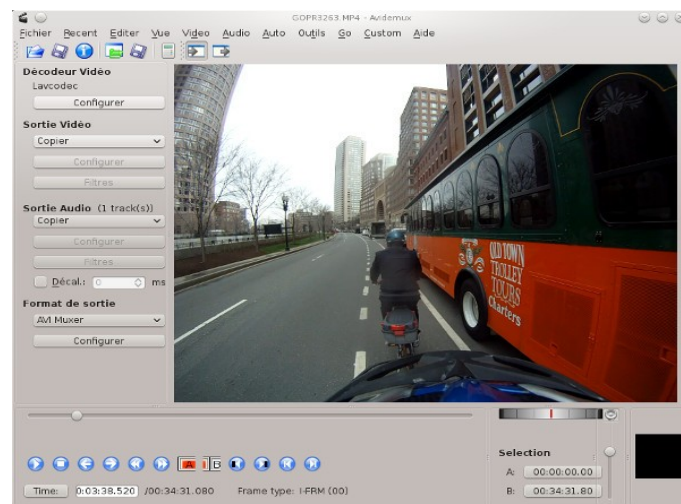
```
cp -f install/usr/bin/* /usr/bin
```

```
cp -Rf install/usr/lib64/* /usr/lib64
```

```
cp -Rf install/usr/share/* /usr/share
```

### 10.2.3 Utilisation

Il suffit de taper **avidemux3\_qt5**



L'étendue de toutes ses fonctionnalités est large en voici un très bref aperçu.

#### Fusionner deux vidéos

- Ouvrez la première vidéo
- Ouvrez la deuxième vidéo, Fichier->Ajouter
- Sauvegardez la vidéo obtenue, Fichier->Sauver

#### Couper des scènes

- Avec le curseur qui défile horizontalement, placez vous au début de la scène à couper
- Cliquez sur le bouton >A

- placez vous à la fin de la scène à couper
- Cliquez sur le bouton B>
- vous voyez le numéro des frames de début et de fin s'affichaient en bas à droite au niveau du champ Selection
- Editer-> Couper
- Sauvegardez la vidéo obtenue, Fichier->Sauver

### Sauvegarder la bande son

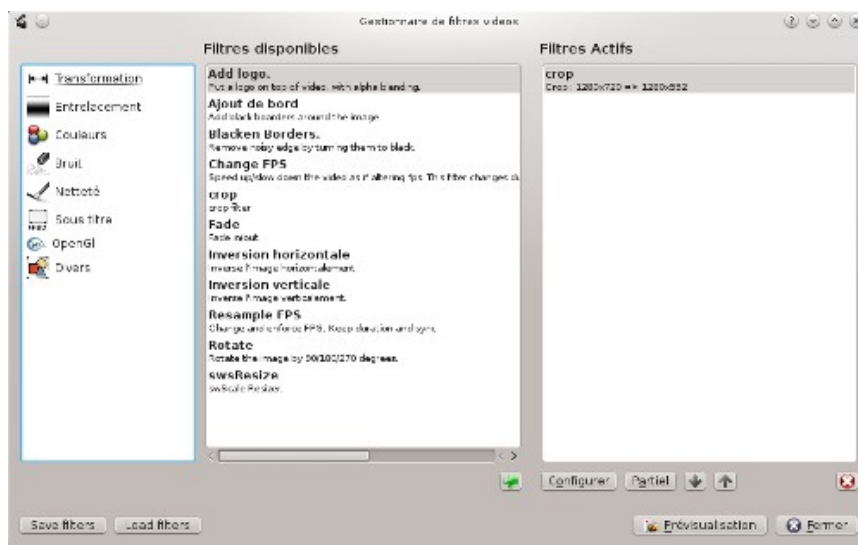
- Audio->Sauver la piste audio
- saisissez le nom d'un fichier terminé par .mp3

### Rajoutez une bande son

- Audio->Choisir les pistes puis Add audio track
- choisissez votre fichier MP3, la bande son va commencer en début de vidéo, si la musique se termine alors que la vidéo n'est pas encore terminée, il n'y a plus de son..., vous avez donc intérêt à prendre un fichier MP3 qui a une durée supérieure à la vidéo. (il y a peut être un moyen de faire autrement mais je ne l'ai pas vu).

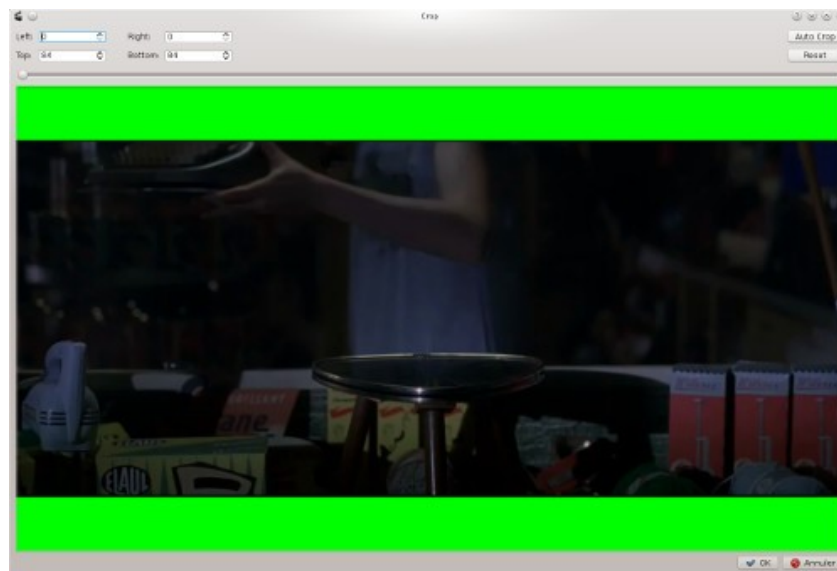
### Recadrer une vidéo

Pour recadrer une vidéo (crop), si vous avez par exemple des bandes noires disgracieuses, il faut rajouter un filtre de traitement accessible via le menu **Vidéo->Filtres**, on rajoute ensuite le traitement **crop**

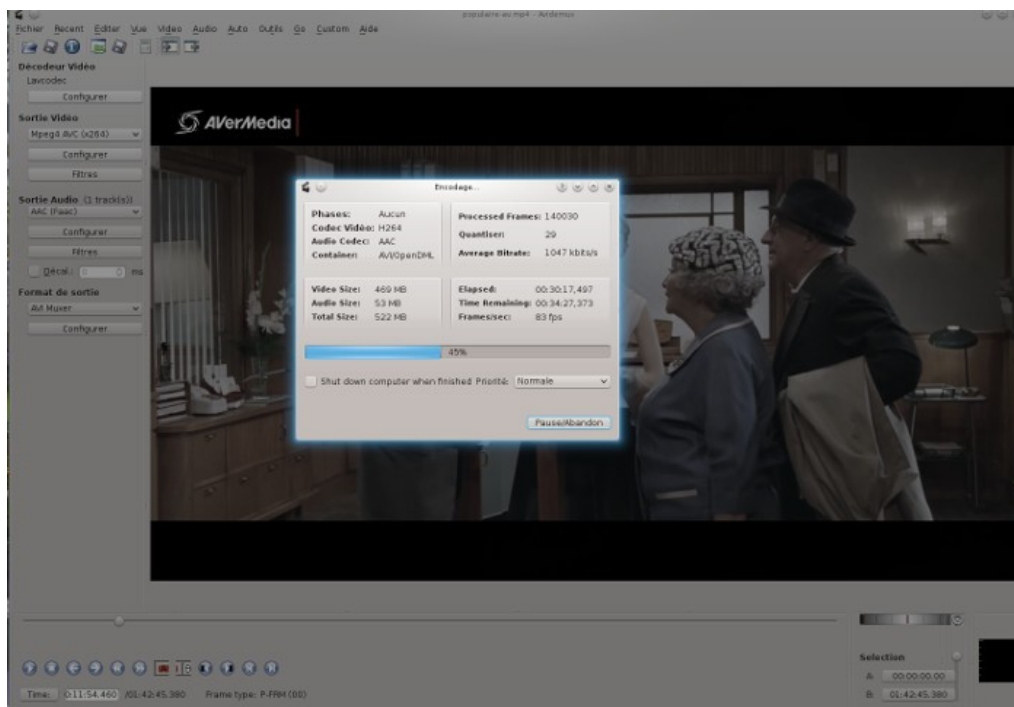


on retaille la vidéo comme il va bien, ici on place des bandes vertes qui correspondent au recadrage faisant disparaître les bandes noires indésirables.

On règle les bandes vertes avec les paramètres Left, Right, Top et Bottom.



Il suffit ensuite de sauvegarder la vidéo après avoir choisi les codecs audio et vidéo. Je vous conseille le format HEVC (x265) qui prend au moins 30% de moins de taille par rapport au MPEG4 AVC (x264) pour une qualité au moins identique, à vous de assurer que vos lecteurs vidéos reconnaissent le format x265. Par contre le temps de traitement est plus long, au moins 30% de plus, vous avez intérêt à avoir une machine plutôt puissante.



La documentation se trouve par ici [http://www.avidemux.org/admWiki/index.php?title=Main\\_Page](http://www.avidemux.org/admWiki/index.php?title=Main_Page)

## 10.3 Copier des DVD

### 10.3.1 Mencoder

**mencoder** est lié à **mplayer**. Pour ripper un DVD tout se fait en ligne de commande, on accède à la table des matières en tapant

**lsdvd**

voilà ce que ça peut donner

**libdvdread: Using libdvdcss version 1.2.13 for DVD access**

**Disc Title: CDRom**

**Title: 01, Length: 00:57:54.180 Chapters: 06, Cells: 06, Audio streams: 01, Subpictures: 00**

**Title: 02, Length: 01:13:17.120 Chapters: 08, Cells: 08, Audio streams: 01, Subpictures: 00**

**Title: 03, Length: 01:05:13.100 Chapters: 07, Cells: 07, Audio streams: 01, Subpictures: 00**

**Longest track: 02**

il indique la piste la plus longue, attention sur les DVD protégés ce n'est pas forcément la bonne piste ! Pour savoir quelle piste est la bonne, le plus simple est de lire simplement le DVD sur votre PC ou un lecteur de salon et de repérer la piste en cours de lecture. Ainsi on peut trouver jusqu'à 99 pistes sur un DVD et la piste lue peut être la 51 même si la piste la plus longue est la 2.

Admettons que la bonne piste à extraire soit la 1, pour la visualiser on tapera

**mplayer dvd://1**

voilà le résultat

**MPlayer SVN-r35233-4.6.3 (C) 2000-2012 MPlayer Team**

**Lecture de dvd://1**

libdvdread: Using libdvdcss version 1.2.10 for DVD access  
Il y a 17 titres sur ce DVD.  
Il y a 1 angles sur ce titre DVD.

libdvdread: Attempting to retrieve all CSS keys  
libdvdread: This can take a long time, please be patient

libdvdread: Get key for /VIDEO\_TS/VIDEO\_TS.VOB at 0x00089030  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO\_TS/VTS\_01\_1.VOB at 0x000897d0  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO\_TS/VTS\_02\_1.VOB at 0x000a0e08  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO\_TS/VTS\_03\_0.VOB at 0x000e3be5  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO\_TS/VTS\_03\_1.VOB at 0x0010a0af  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO\_TS/VTS\_04\_1.VOB at 0x002df94a  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO\_TS/VTS\_05\_1.VOB at 0x0037ea84  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO\_TS/VTS\_06\_0.VOB at 0x0038679d  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO\_TS/VTS\_06\_1.VOB at 0x0038687b  
libdvdread: Elapsed time 0  
libdvdread: Found 6 VTS's  
libdvdread: Elapsed time 0  
Flux audio : 0 format : ac3 (5.1) langue : en aide : 128.  
Flux audio : 1 format : ac3 (5.1) langue : fr aide : 129.  
Flux audio : 2 format : ac3 (5.1) langue : unknown aide : 130.  
Nombre du canaux audio sur le disque : 3.  
Sous-titre ( sid ) : 0 langue : en  
Sous-titre ( sid ) : 1 langue : ar  
Sous-titre ( sid ) : 2 langue : nl  
Sous-titre ( sid ) : 3 langue : fr  
Sous-titre ( sid ) : 4 langue : en  
Nombre de sous-titres sur le disque : 5

Fichier de type MPEG-PS détecté.  
VIDEO: MPEG2 720x576 (aspect 3) 25.000 fps 8000.0 kbps (1000.0 kbyte/s)

---

Ouverture du décodeur vidéo : [ffmpeg] FFmpeg's libavcodec codec family  
libavcodec version 54.64.100 (internal)  
Codec vidéo choisi : [ffmpeg2] vfm : ffmpeg (FFmpeg MPEG-2)

---

---

Ouverture décodeur audio : [ffmpeg] FFmpeg/libavcodec audio decoders  
AUDIO: 48000 Hz, 2 ch, s16le, 384.0 kbit/25.00% (ratio: 48000->192000)  
Codec audio sélectionné : [ffac3] afm : ffmpeg (FFmpeg AC-3)

---

[AO OSS] audio\_setup: Impossible ouvrir périphérique audio /dev/dsp : No such file or directory  
AO: [alsa] 48000Hz 2ch s16le (2 bytes per sample)  
Démarré la lecture...  
L'aspect du film est 1.78:1 - pré-redimensionnement à l'aspect correct.  
VO: [xv] 720x576 => 1024x576 Planar YV12

On repère quelques informations intéressantes, la piste audio est repéré par le numéro 129 et le sous titre en français par le 3. Pour visualiser le film en français avec un sous titre en français on tapera

**mplayer dvd://1 -aid 129 -sid 3**

si on ne veut pas de sous titre, on tapera

**mplayer dvd://1 -aid 129 -nosub**

vous constaterez que souvent il y a des bandes noires en haut et en bas du film, on peut les supprimer à l'encodage. Pour cela il faut les repérer en tapant

```
mplayer dvd://1 -vf cropdetect
```

au bout d'un certain temps de lecture, ça défile et on obtient

```
A: 230.5 V: 230.5 A-V: 0.000 ct: -0.088 1239/1239 4% 2% 0.9% 0 0  
[CROP] Aire coupée: X: 0..719 Y: 0..575 (-vf crop=720:576:0:0).
```

pour visualiser maintenant le film en français sans sous titre et sans les bandes noires, on tapera

```
mplayer dvd://1 -aid 129 -nosub -vf crop=720:576:0:0
```

c'est bon on peut passer maintenant à l'encodage. Tout d'abord on se place dans un répertoire vierge avec aucun fichier. On commence d'abord à extraire la bande son au format mp3 qualité 192kbit/s.

```
mencoder dvd://1 -aid 129 -nosub -oac mp3lame -lameopts mode=2:cbr:br=192:vol=0 -ovc frameno -o  
frameno.avi
```

voilà le résultat

```
MEncoder 1.1-4.6.3 (C) 2000-2012 MPlayer Team  
libdvdread: Using libdvdcss version 1.2.10 for DVD access  
Il y a 17 titres sur ce DVD.  
Il y a 1 angles sur ce titre DVD.
```

```
libdvdread: Attempting to retrieve all CSS keys  
libdvdread: This can take a _long_ time, please be patient
```

```
libdvdread: Get key for /VIDEO_TS/VIDEO_TS.VOB at 0x00089030  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO_TS/VTS_01_1.VOB at 0x000897d0  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO_TS/VTS_02_1.VOB at 0x000a0e08  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO_TS/VTS_03_0.VOB at 0x000e3be5  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO_TS/VTS_03_1.VOB at 0x0010a0af  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO_TS/VTS_04_1.VOB at 0x002df94a  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO_TS/VTS_05_1.VOB at 0x0037ea84  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO_TS/VTS_06_0.VOB at 0x0038679d  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO_TS/VTS_06_1.VOB at 0x0038687b  
libdvdread: Elapsed time 0  
libdvdread: Found 6 VTS's  
libdvdread: Elapsed time 0  
Flux audio : 0 format : ac3 (5.1) langue : en aide : 128.  
Flux audio : 1 format : ac3 (5.1) langue : fr aide : 129.  
Flux audio : 2 format : ac3 (5.1) langue : unknown aide : 130.  
Nombre du canaux audio sur le disque : 3.  
Sous-titre ( sid ) : 0 langue : en  
Sous-titre ( sid ) : 1 langue : ar  
Sous-titre ( sid ) : 2 langue : nl  
Sous-titre ( sid ) : 3 langue : fr  
Sous-titre ( sid ) : 4 langue : en  
Nombre de sous-titres sur le disque : 5
```

```
succès : format : 2 data : 0x851000 - 0xeac35800  
Fichier de type MPEG-PS détecté.  
VIDEO: MPEG2 720x576 (aspect 3) 25.000 fps 8000.0 kbps (1000.0 kbyte/s)  
[V] filefmt:2 fourcc:0x10000002 taille:720x576 fps:25.000 ftime:=0.0400
```

---

---

Ouverture décodeur audio : [ffmpeg] FFmpeg/libavcodec audio decoders  
libavcodec version 54.23.100 (internal)  
AUDIO: 48000 Hz, 2 ch, s16le, 384.0 kbit/25.00% (ratio: 48000->192000)  
Codec audio sélectionné : [ffac3] afm : ffmpeg (FFmpeg AC-3)

---

---

Audio MP3 sélectionné

Écriture de l'entête...

ODML: Aspect information not (yet?) available or unspecified, not writing vprp header.

Écriture de l'entête...

ODML: Aspect information not (yet?) available or unspecified, not writing vprp header.

Pos: 0.8s 22f (0%) 0.00fps Trem: 0min 0mb A-V:0.082 [0:192]

Image sautée !

Pos: 1.2s 32f (0%) 0.00fps Trem: 0min 0mb A-V:0.082 [0:192]

Image sautée !

Pos: 2.0s 53f (0%) 0.00fps Trem: 0min 0mb A-V:0.081 [0:192]

Image sautée !

Pos:4440.0s 111003f (100%) 136.09fps Trem: 0min 104mb A-V:0.026 [0:192]

Écriture de l'index...

Écriture de l'entête...

ODML: Aspect information not (yet?) available or unspecified, not writing vprp header.

Débit binaire (bitrate) vidéo recommandé pour le CD 650MB : 1022

Débit binaire (bitrate) vidéo recommandé pour le CD 700MB : 1116

Débit binaire (bitrate) vidéo recommandé pour le CD 800MB : 1305

Débit binaire (bitrate) vidéo recommandé pour le CD 2 x 650MB : 2250

Débit binaire (bitrate) vidéo recommandé pour le CD 2 x 700MB : 2439

Débit binaire (bitrate) vidéo recommandé pour le CD 2 x 800MB : 2817

Flux vidéo : 0.800 kbit/s (100 B/s) taille : 444000 octets 4440.000 secs 111003 images

Flux audio : 192.000 kbit/s (24000 B/s) taille : 106562304 octets 4440.096 secs

A noter quelques informations importantes pour connaître le bitrate à définir suivant la taille du fichier à obtenir. Pour un encodage mpeg4 en 2 passes avec un bitrate de 1800 (excellente qualité, mais gros fichier 1,5Go pour un film d'1h30) on tapera d'abord la commande suivante

```
mencoder dvd://1 -aid 129 -nosub -vf crop=720:576:0:0 -oac copy -mc 0 -o /dev/null -ovc lavc -lavcopts vcodec=mpeg4:vbitrate=1800:vhq:autoaspect:vpass=1
```

voilà le résultat

**MEncoder 1.1-4.6.3 (C) 2000-2012 MPlayer Team**

libdvdread: Using libdvdcss version 1.2.10 for DVD access

Il y a 17 titres sur ce DVD.

Il y a 1 angles sur ce titre DVD.

libdvdread: Attempting to retrieve all CSS keys

libdvdread: This can take a \_long\_ time, please be patient

libdvdread: Get key for /VIDEO\_TS/VIDEO\_TS.VOB at 0x00089030

libdvdread: Elapsed time 0

libdvdread: Get key for /VIDEO\_TS/VTS\_01\_1.VOB at 0x000897d0

libdvdread: Elapsed time 0

libdvdread: Get key for /VIDEO\_TS/VTS\_02\_1.VOB at 0x000a0e08

libdvdread: Elapsed time 0

libdvdread: Get key for /VIDEO\_TS/VTS\_03\_0.VOB at 0x000e3be5

libdvdread: Elapsed time 0

libdvdread: Get key for /VIDEO\_TS/VTS\_03\_1.VOB at 0x0010a0af

libdvdread: Elapsed time 0

libdvdread: Get key for /VIDEO\_TS/VTS\_04\_1.VOB at 0x002df94a

libdvdread: Elapsed time 0

libdvdread: Get key for /VIDEO\_TS/VTS\_05\_1.VOB at 0x0037ea84

libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO\_TS/VTS\_06\_0.VOB at 0x0038679d  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO\_TS/VTS\_06\_1.VOB at 0x0038687b  
libdvdread: Elapsed time 0  
libdvdread: Found 6 VTS's  
libdvdread: Elapsed time 0  
Flux audio : 0 format : ac3 (5.1) langue : en aide : 128.  
Flux audio : 1 format : ac3 (5.1) langue : fr aide : 129.  
Flux audio : 2 format : ac3 (5.1) langue : unknown aide : 130.  
Nombre du canaux audio sur le disque : 3.  
Sous-titre ( sid ) : 0 langue : en  
Sous-titre ( sid ) : 1 langue : ar  
Sous-titre ( sid ) : 2 langue : nl  
Sous-titre ( sid ) : 3 langue : fr  
Sous-titre ( sid ) : 4 langue : en  
Nombre de sous-titres sur le disque : 5

succès : format : 2 data : 0x851000 - 0xeac35800  
Fichier de type MPEG-PS détecté.  
VIDEO: MPEG2 720x576 (aspect 3) 25.000 fps 8000.0 kbps (1000.0 kbyte/s)  
[V] filefmt:2 fourcc:0x10000002 taille:720x576 fps:25.000 ftime:=0.0400

---

---

Ouverture décodeur audio : [ffmpeg] FFmpeg/libavcodec audio decoders  
libavcodec version 54.23.100 (internal)  
AUDIO: 48000 Hz, 2 ch, s16le, 384.0 kbit/25.00% (ratio: 48000->192000)  
Codec audio sélectionné : [ffac3] afm : ffmpeg (FFmpeg AC-3)

---

---

Ouverture du filtre vidéo : [expand osd=1]  
Expand: -1 x -1, -1 ; -1, osd: 1, aspect: 0.000000, round: 1

---

---

Ouverture du décodeur vidéo : [ffmpeg] FFmpeg's libavcodec codec family  
Codec vidéo choisi : [ffmpeg2] vfm : ffmpeg (FFmpeg MPEG-2)

---

---

codec audio : copie img (format=2000 canaux=2 taux=48000 bits=16 B/s=48000 échant-1)  
L'aspect du film est 1.78:1 - pré-redimensionnement à l'aspect correct.  
videocodec: libavcodec (720x576 fourcc=34504d46 [FMP4])  
[VE\_LAVC] Codage haute qualité sélectionné (non temps réel) !  
[mpeg2video @ 0xe39c40]ac-tex damaged at 23 7  
[mpeg2video @ 0xe39c40]Warning MVs not available  
[mpeg2video @ 0xe39c40]concealing 1305 DC, 1305 AC, 1305 MV errors  
Pos: 0.0s 1f (0%) 0.00fps Trem: 0min 0mb A-V:0.000 [0:0]  
Écriture de l'entête...  
ODML: vprp aspect is 16:9.  
Écriture de l'entête...  
ODML: vprp aspect is 16:9.

1 image(s) répétée(s) !  
Pos:4440.1s 111004f (100%) 133.70fps Trem: 0min 1154mb A-V:0.000 [1792:384]

Abandonne des trames vidéo.  
Écriture de l'index...  
Écriture de l'entête...  
ODML: vprp aspect is 16:9.

Flux vidéo : 1792.090 kbit/s (224011 B/s) taille : 994636818 octets 4440.120 secs 111004 images

Flux audio : 384.000 kbit/s (48000 B/s) taille : 213124608 octets 4440.096 secs

c'est parti pour la deuxième passe en tapant

mencoder dvd://1 -aid 129 -nosub -vf crop=720:576:0:0 -oac copy -mc 0 -o film.avi -ovc lavc -lavcopts  
vcodec=mpeg4:vbitrate=1800:vhq:autoaspect:vpass=2

voilà le résultat



**MEncoder 1.1-4.6.3 (C) 2000-2012 MPlayer Team**  
libdvdread: Using libdvdcss version 1.2.10 for DVD access  
Il y a 17 titres sur ce DVD.  
Il y a 1 angles sur ce titre DVD.

libdvdread: Attempting to retrieve all CSS keys  
libdvdread: This can take a `_long_` time, please be patient

libdvdread: Get key for /VIDEO\_TS/VIDEO\_TS.VOB at 0x00089030  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO\_TS/VTS\_01\_1.VOB at 0x000897d0  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO\_TS/VTS\_02\_1.VOB at 0x000a0e08  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO\_TS/VTS\_03\_0.VOB at 0x000e3be5  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO\_TS/VTS\_03\_1.VOB at 0x0010a0af  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO\_TS/VTS\_04\_1.VOB at 0x002df94a  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO\_TS/VTS\_05\_1.VOB at 0x0037ea84  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO\_TS/VTS\_06\_0.VOB at 0x0038679d  
libdvdread: Elapsed time 0  
libdvdread: Get key for /VIDEO\_TS/VTS\_06\_1.VOB at 0x0038687b  
libdvdread: Elapsed time 0  
libdvdread: Found 6 VTS's  
libdvdread: Elapsed time 0  
Flux audio : 0 format : ac3 (5.1) langue : en aide : 128.  
Flux audio : 1 format : ac3 (5.1) langue : fr aide : 129.  
Flux audio : 2 format : ac3 (5.1) langue : unknown aide : 130.  
Nombre du canaux audio sur le disque : 3.  
Sous-titre ( sid ) : 0 langue : en  
Sous-titre ( sid ) : 1 langue : ar  
Sous-titre ( sid ) : 2 langue : nl  
Sous-titre ( sid ) : 3 langue : fr  
Sous-titre ( sid ) : 4 langue : en  
Nombre de sous-titres sur le disque : 5

succès : format : 2 data : 0x851000 - 0xeac35800  
Fichier de type MPEG-PS détecté.  
VIDEO: MPEG2 720x576 (aspect 3) 25.000 fps 8000.0 kbps (1000.0 kbyte/s)  
[V] filefmt:2 fourcc:0x10000002 taille:720x576 fps:25.000 ftime:=0.0400

---

Ouverture décodeur audio : [ffmpeg] FFmpeg/libavcodec audio decoders  
libavcodec version 54.23.100 (internal)  
AUDIO: 48000 Hz, 2 ch, s16le, 384.0 kbit/25.00% (ratio: 48000->192000)  
Codec audio sélectionné : [ffac3] afm : ffmpeg (FFmpeg AC-3)

---

Ouverture du filtre vidéo : [expand osd=1]  
Expand: -1 x -1, -1 ; -1, osd: 1, aspect: 0.000000, round: 1

---

Ouverture du décodeur vidéo : [ffmpeg] FFmpeg's libavcodec codec family  
Codec vidéo choisi : [ffmpeg2] vfm : ffmpeg (FFmpeg MPEG-2)

---

codec audio : copie img (format=2000 canaux=2 taux=48000 bits=16 B/s=48000 échant-1)  
L'aspect du film est 1.78:1 - pré-redimensionnement à l'aspect correct.  
videocodec: libavcodec (720x576 fourcc=34504d46 [FMP4])  
[VE\_LAVC] Codage haute qualité sélectionné (non temps réel) !  
[mpeg2video @ 0xe39c40]ac-tex damaged at 23 7  
[mpeg2video @ 0xe39c40]Warning MVs not available  
[mpeg2video @ 0xe39c40]concealing 1305 DC, 1305 AC, 1305 MV errors  
Pos: 0.0s 1f (0%) 0.34fps Trem: 0min 0mb A-V:0.000 [0:0]  
Écriture de l'entête...

ODML: vprp aspect is 16:9.  
Écriture de l'entête...  
ODML: vprp aspect is 16:9.

1 image(s) répétée(s) !  
Pos:3941.3s 98532f (89%) 121.01fps Trem: 1min 1146mb A-V:0.000 [1790:384]  
ODML: Starting new RIFF chunk at 1023MB.  
Pos:4440.1s 111003f (100%) 120.37fps Trem: 0min 1158mb A-V:0.000 [1799:384]

Image sautée !  
Pos:4440.1s 111004f (100%) 120.37fps Trem: 0min 1158mb A-V:0.000 [1799:384]

Abandonne des trames vidéo.  
Écriture de l'index...  
Écriture de l'entête...  
ODML: vprp aspect is 16:9.

Flux vidéo : 1799.993 kbit/s (224999 B/s) taille : 999023307 octets 4440.120 secs 111004 images

Flux audio : 384.000 kbit/s (48000 B/s) taille : 213124608 octets 4440.096 secs

A noter l'option **-mc 0** qui est nécessaire si vous constatez un décalage progressif entre la vidéo et le son, vous devriez avoir un minimum d'images sautées ou répétées sinon ce n'est pas bon signe. Un bon indicateur est la dernière mention de la commande.

Flux vidéo : 1799.993 kbit/s (224999 B/s) taille : 999023307 octets 4440.120 secs 111004 images

Flux audio : 384.000 kbit/s (48000 B/s) taille : 213124608 octets 4440.096 secs

Les flux vidéo et audio doivent avoir à peu près la même durée.

A noter que **mencoder** dispose d'une foule d'options, je n'en présente qu'une ici, mais elle est convient parfaitement quand on n'a pas trop de soucis de place disque et qu'on veut privilégier la qualité.  
Autre méthode en passant par l'étape intermédiaire de création d'un fichier VOB, récupération ici de la piste 52

**mplayer dvd://52 -dumpstream -dumpfile fichier.vob**

Tous les langues et sous titres disponibles se trouvent alors dans le fichier .vob.

A noter que si le DVD est endommagé (rayé), il faut le monter avec la commande **mount** et copier le répertoire **VIDEO\_TS** sur le disque dur

**mount /dev/sr0 /media/cdrom**

puis

**cp -Rf /media/cdrom/VIDEO\_TS/ /emile/video/dvdrip**

Il suffit ensuite de compresser le fichier en utilisant **ffmpeg** par exemple, le codec H264 avec la qualité 20.

**ffmpeg -i fichier.vob -acodec libmp3lame -aq 4 -vcodec libx264 -crf 20 -threads 0 fichier.mkv**

## 10.3.2 HandBrake

### 10.3.2.1 Installation

**Handbrake** est une interface graphique permettant de copier les DVD. Le site officiel est <http://handbrake.fr> où on récupèrera l'archive. Préalablement pour une Mageia il faudra installer les packages suivants **intltool lib64notify-devel lib64gudev1.0-devel dbus-glib-devel lib64webkitgtk1.0-devel lib64webkitgtk3.0-devel lib64gtk+2.0-devel lib64ass-devel lib64sample-rate-devel lib64jansson-devel meson lib64numa-devel** et **lib64mp4v2-devel**

On décompresse maintenant l'archive de **Handbrake** en tapant

**tar xvfj HandBrake-1.6.0-source.tar.bz2**

Cela donne le répertoire **HandBrake-1.6.0** dans lequel on tape successivement

```
./configure --enable-fdk-aac --enable-nvenc --enable-x265
```

(l'option **nvenc** est pour les NVIDIA) voilà le résultat

```
-----  
Build system:   x86_64-pc-linux-gnu  
Host system:    x86_64-pc-linux-gnu  
Target platform: linux  
Harden:        True  
Sandbox:       False (not supported on target platform)  
Enable FDK-AAC: True  
Enable FFmpeg AAC: True (required on target platform)  
Enable MediaFound.: False (not supported on target platform)  
Enable NVENC:   True  
Enable NVDEC:   False  
Enable QSV:     False  
Enable VCE:     False  
-----
```

**Build is configured!**

on tape ensuite

```
cd build  
make
```

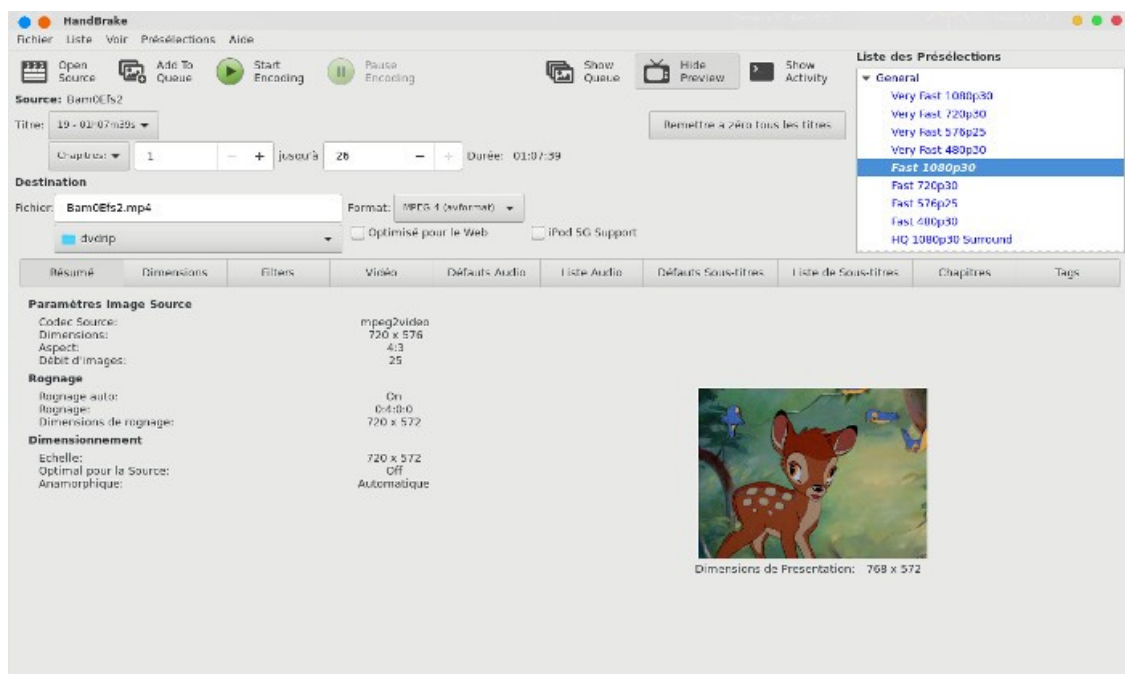
puis en tant que root

```
make install
```

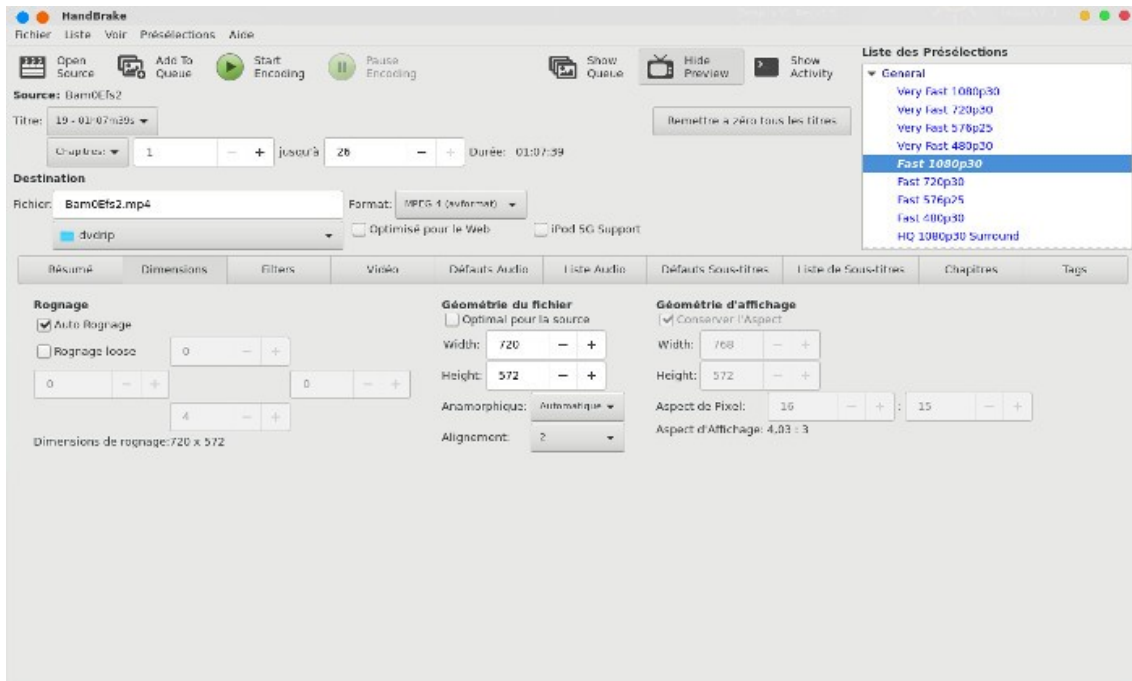
### 10.3.2.2 Utilisation

A noter qu'il existe une version commande en ligne **HandBrakeCLI** qui est très utile si vous souhaitez utiliser **HandBrake** en mode batch dans un script.

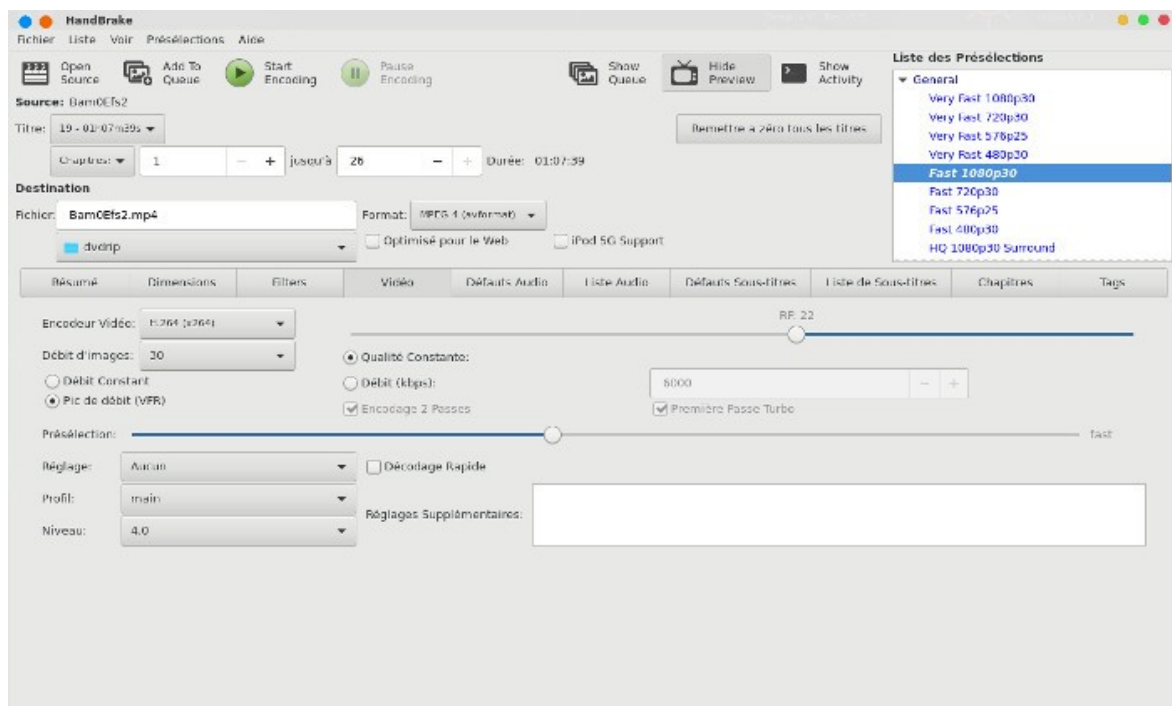
Quand on lance **HandBrake**, et qu'on veut indiquer le lecteur DVD, il faudra cliquer sur **Source**, il choisira par défaut la piste la plus longue, attention ce n'est pas forcément la bonne piste, vous avez la possibilité de choisir la piste au niveau du champ **Titre**.



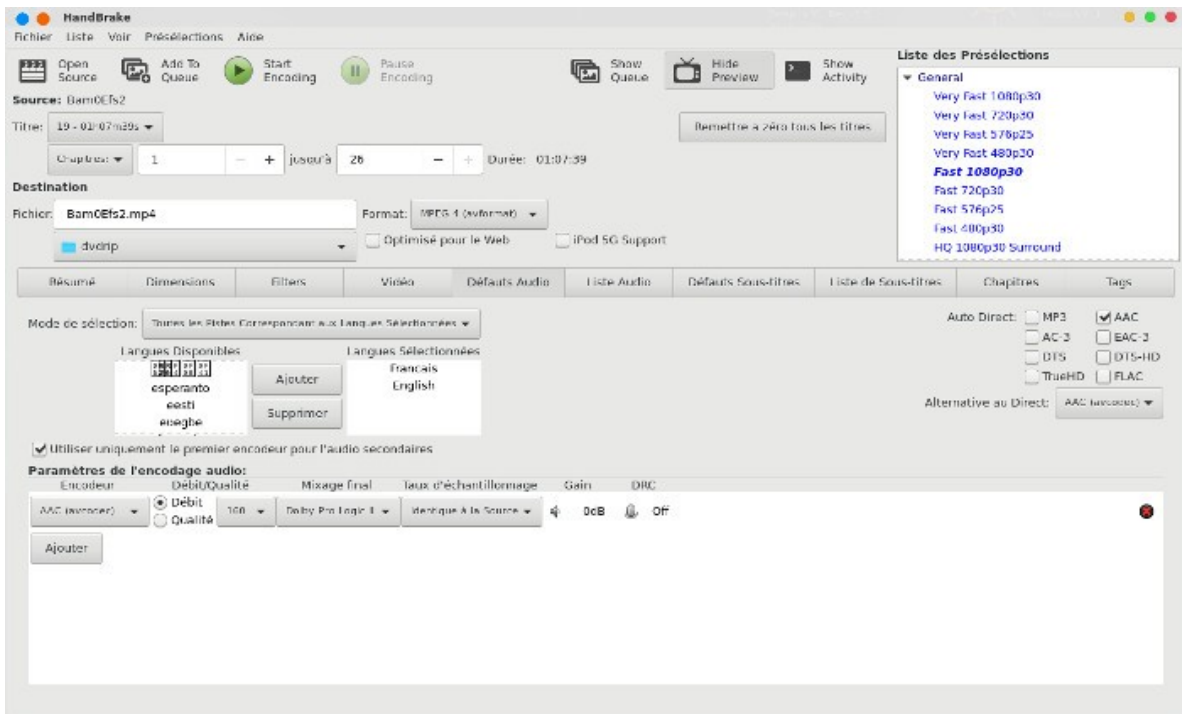
si on clique sur le bouton **Dimensions** on peut jouer sur les paramètres de cropping, personnellement j'y touche jamais



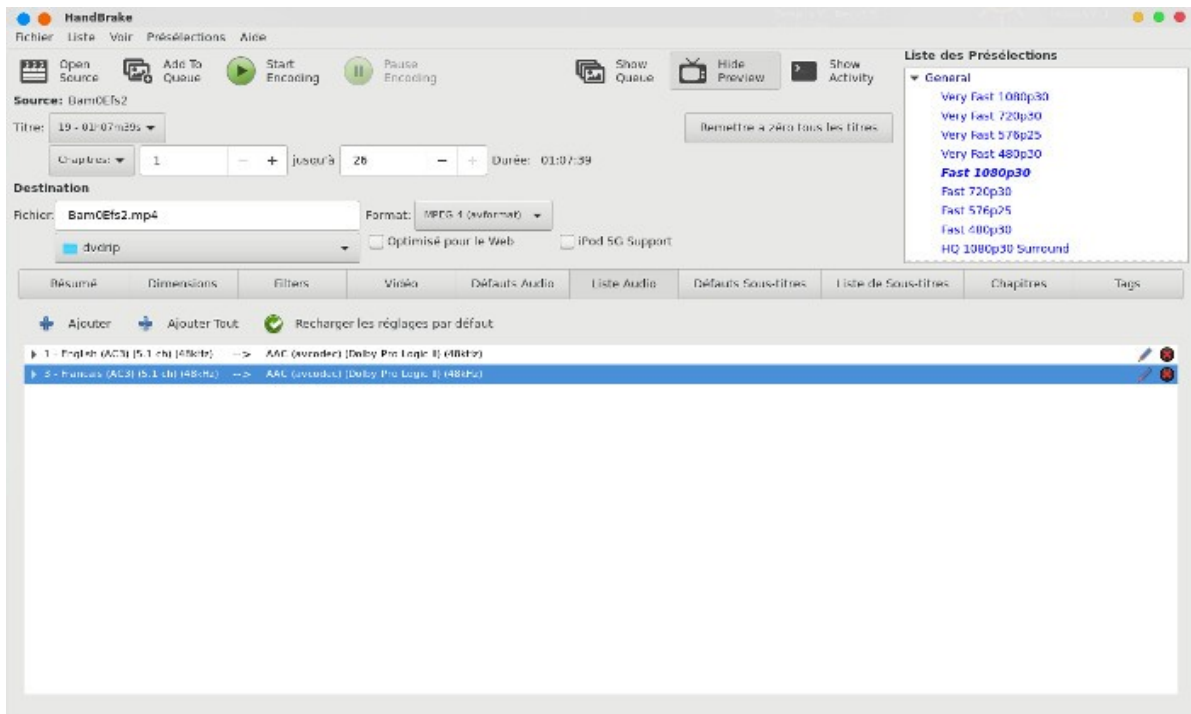
si on clique sur le bouton **Vidéo** on accède au choix de paramétrage, par défaut c'est du H264, on peut éventuellement fixer le bitrate si on veut améliorer la qualité (qui est déjà plus que correcte par défaut), personnellement je mets 2000kbps qui est un bon compromis taille/qualité. Il existe maintenant également la possibilité de sauvegarder en utilisant H265/HEVC qui a un bien meilleur ratio qualité/taille que le H264.



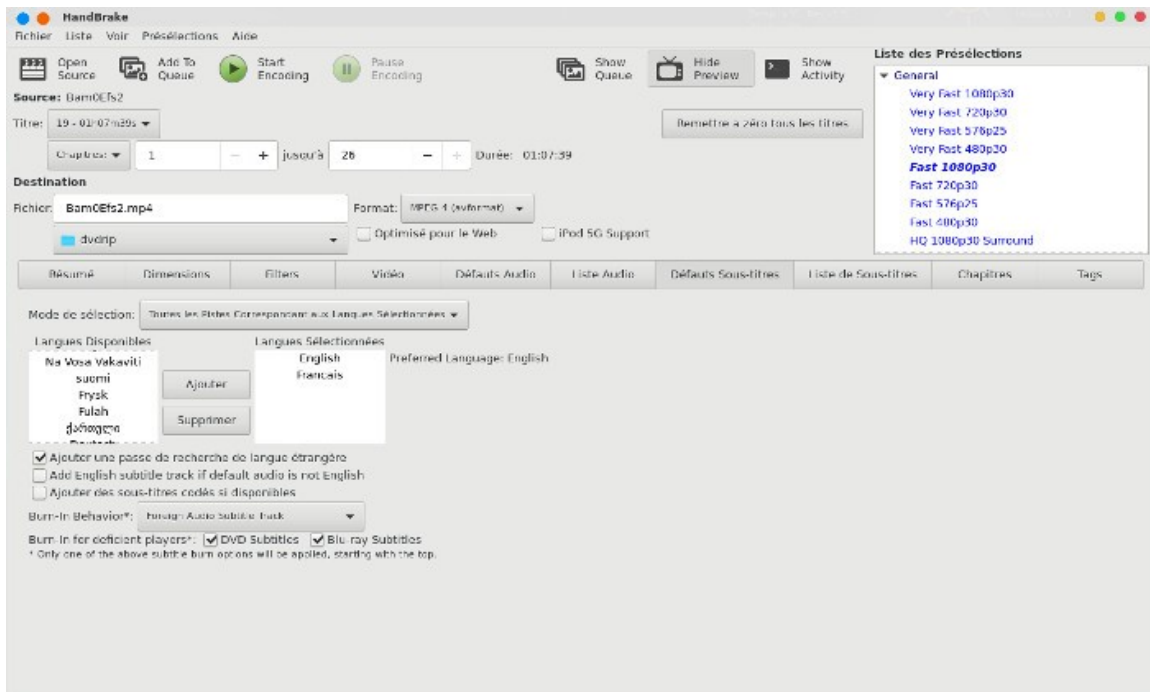
quand on clique sur le bouton **Defaults Audio**, on peut choisir les pistes audio et l'encodeur (par défaut AAC), attention tous les lecteurs ne lisent pas forcément l'AAC, vous pouvez choisir l'encodeur mp3 qui est plus universel.



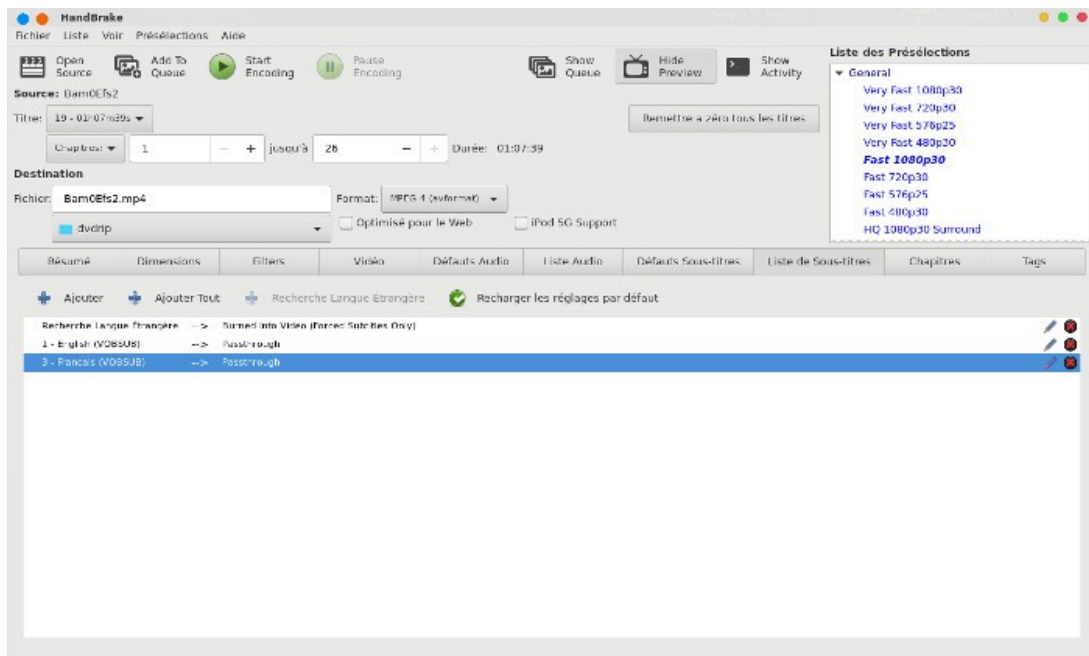
Sur le bouton **Audio list**, on obtient les pistes qui seront sélectionnées



Pour les sous titres on clique sur le bouton **Defaults Sous-titres**, on peut choisir les sous titres par défaut



Via le bouton **Liste de Sous-titres**, on sélectionne en cliquant sur la croix bleue les sous titres à rajouter.

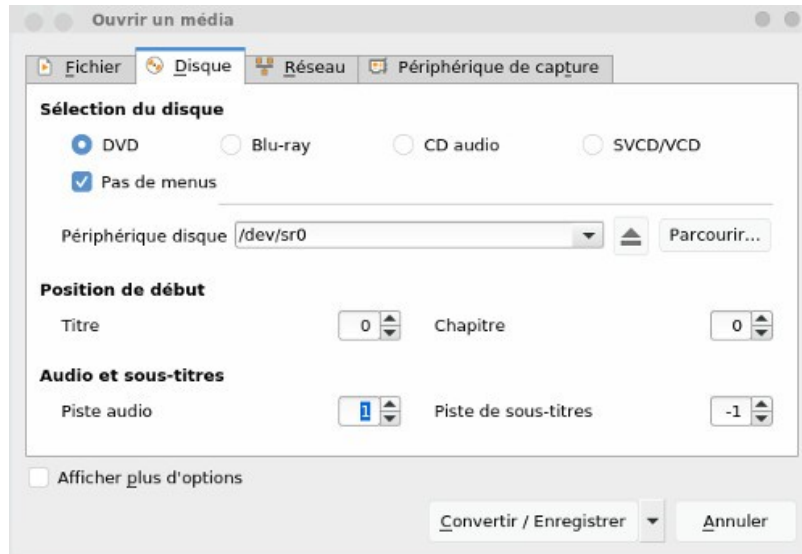


Vous disposez ensuite d'un tas de possibilités de configuration mais personnellement j'ai tout laissé avec la config par défaut et ça marche très bien. Idem pour les boutons **Filters**, **Chapitres** et **Tags**. On clique ensuite sur **Start Encoding** pour lancer l'encodage.

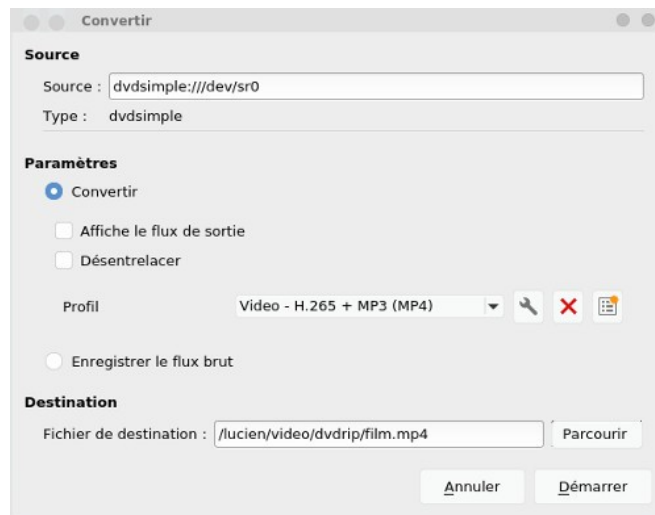
### 10.3.3 VLC

VLC dispose d'une fonction bien pratique pour pouvoir également copier des DVD et assez simple et rapide d'utilisation, seul inconvénient il n'est pas possible de récupérer toutes les bandes sons et les sous titres, il faudra choisir. Il suffit de lancer VLC, d'insérer son DVD, et d'aller dans le menu Média→Convertir/Enregistrer...

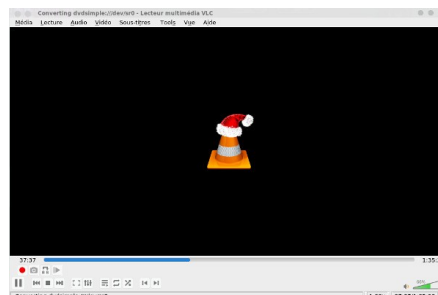
On choisit d'abord le support DVD dans l'onglet Disque, puis la piste audio et de sous titre (on laisse à -1 pour en choisir aucune), il est préférable de lancer une lecture classique préalablement pour identifier la bonne piste et éventuellement sous titre.



Ensuite on choisit le format d'encodage, il vaut mieux ne pas cocher la case **Affiche le flux de sortie** qui va ralentir le process. On a le choix également d'enregistrer le flux brut pour un traitement particulier ensuite sur **avidemux** par exemple.



On saisit le nom du fichier et c'est parti en cliquant sur **Démarrer**.



L'état d'avancement est donné par la barre de progression.