audio, video, ffmpeg, code, shell, em

ffmpeg

(page créée le 15 juillet 2021)
Voir aussi : la fabrique gif / ffmpeg

ffmpeg est un logiciel en ligne de commande pour la manipulation de fichiers audiovisuels (video, audio, image, son, metadonnées). Il fait partie des logiciels “couteaux suisses” de manipulation de fichiers multimédia (avec imagemagick et sox). ffmpeg est utile pour :

- changer les caractéristiques d'une vidéo (définition, nombre d'images par seconde),
- changer les caractéristiques d'un fichier audio : par ex. conversion de wav vers mp3,
- transcoder (changer de format et de codec),
- extraire d'une vidéo : des images, une bande-son, une séquence, etc.,
- manipuler les sous-titres,
- créer des vidéos : en mixant images, séquences, bandes son,
- ajouter / manipuler les méta-données
- et une multitude d'autres choses!

ffmpeg est fourni avec deux autres utilitaires :

- **ffprobe** : affichage des caractéristiques d'un fichier (doc ffprobe)
- **ffplay** : lecteur de fichiers multimedia (doc ffplay)

Exemples d'utilisation

Conversion par lot en mp3

Ce script traite tous les fichiers .wav d'un répertoire pour les transformer en .mp3 en ajoutant des métadonnées (utilisé pour les podcasts des 26 heures de radio baleine)

```
#!/bin/bash
# ffmpeg 3.2.12-1 / debian 9.5 @ kirin / 20210715
# conversion de fichiers audio .wav en fichiers .mp3 podcastables
# la plupart des métadonnées sont ajoutées automatiquement, pour les autres il faut compléter à la main!
# ffmpeg et métadonnées : https://wiki.multimedia.cx/index.php/FFmpeg_Metadata

metadataArtist="(collectif)"
metadataAlbum="Marathon Radio La Baleine"
metadataGenre="Spoken & Audio"
metadataYear="2021"

for i in *.wav; do
    ffmpeg -i "$i" \
    -vn -ar 44100 -ac 2 -b:a 192k -f mp3 \
    -metadata artist="$metadataArtist" \
    -metadata album="$metadataAlbum" \
    -metadata genre="$metadataGenre" \
    -metadata date="$metadataYear" \
    "${i%wav}mp3"
done
```

Ajouter une image dans les méta-données

(Sans recompression la partie sonore)
Pas facile de trouver des infos récapitulatives sur la définition et le poids des fichiers recommandés pour des podcasts... (ici l'image fait 512×512 et < 200 ko

```
ffmpeg -i in.mp3 -i image.png -map 0:0 -map 1:0 -codec copy -id3v2_version 3 -metadata:s:v title="Album cover" -metadata:s:v comment="Cover (front)" out.mp3
```

`-codec copy` : pas de réencodage, ni pour l'image, ni pour le son

sous forme de script pour traiter tout un répertoire

```
#!/bin/bash
# ffmpeg 3.2.12-1 / debian 9.5 @ kirin / 20210715

for i in *.mp3; do
    ffmpeg -i "$i" -i image.png -map 0:0 -map 1:0 -codec copy -id3v2_version 3 -metadata:s:v title="Album cover" -metadata:s:v
comment="Cover (front)" "${i%wav}mp3"
done
```

Chercher les vidéos silencieuses d'un dossier

Comment savoir si une vidéo a une piste son audible ? Le script renvoie le nom de chaque fichier avec le volume moyen, en décibel.

```
#!/bin/bash
#  20230131 @ kirin / ffmpeg 3.2.12-1 / debian 9.5
#  Passer le chemin du dossier en argument : liste_volume_video.sh /chemin/vers/dossier
#  Ce script ne vérifie pas que les fichiers sont bien des vidéos!

# Définir le dossier selon l'argument passé

if [ -z "$1" ]; then
  dir="."
else
  dir=$1
fi

# Parcourir le dossier et traiter chaque fichier

for file in "$dir"/*; do
  echo -n "$file : "
  ffmpeg -i "$file" -af "volumedetect" -f null /dev/null 2>&1 | grep "mean_volume:" | sed 's/\[.*\]//g'
done
echo -e "\n"
```

Extraire les 20 premières secondes de chaque morceau d'un dossier de mp3

```
#!/bin/bash
# ffmpeg 3.2.5-1 debian 9.1 stretch @ urzhiata / 20180807
for i in *.mp3
do
  ffmpeg -i "$i" -t 10 -vn -acodec pcm_s16le -sample_fmt s16 -ar 44.1k -ac 1 mod_"$i".wav
done
```

Petits trucs pratiques

Convertir des fichiers realmedia (.rm) en mp3

Ça fonctionnerait aussi avec un autre format audio en entrée (wav, aiff, etc.)

```
# 20210927 ffmpeg 3.2.12-1 / debian 9.5 @ kirin
ffmpeg -i realmedia.rm -vn -ar 44100 -ac 2 -ab 192k -f mp3 audio.mp3
```

Convertir des fichiers audio en wav mono 16 bits et avec une fréquence d'échantillonnage de 16000 Hz

```
$ ffmpeg -i in.mp3 -acodec pcm_s16le -ac 1 -ar 16000 out.wav
```

Concaténer plusieurs fichiers son en un seul

```
$ ffmpeg -f concat -safe 0 -i <(for f in ./*.wav; do echo "file '$PWD/$f'"; done) -c copy out.wav
```

Vidéo à partir d'une image fixe (sans son)

```
ffmpeg -loop 1 -i image.png -c:v libx264 -t 10 -pix_fmt yuv420p -vf scale=768:768 video.mp4
```

Vidéo à partir d'une image fixe et d'un son, la vidéo fait la durée du son

```
ffmpeg -loop 1 -r 30 -i 5054.png -i 5054.wav -c:v libx264 -tune stillimage -c:a aac -b:a 250k -shortest 5054.mp4
```

Enregistrer une video du bureau (sur Debian)

```
# ffmpeg 3.2.12-1 / debian 9.5 @ kirin / 20181203
ffmpeg -video_size 1600x900 -framerate 10 -f x11grab -i :0.0+0,0 output.mp4
```

options : https://trac.ffmpeg.org/wiki/Capture/Desktop

Extraction d'un son depuis une vidéo en mp3

```
# ffmpeg 3.2.12-1 / debian 9.5 stretch @ kirin / 20190327
ffmpeg -i video.mp4 -vn -ar 44100 -ac 2 -ab 320k -f mp3 audio.mp3
```

Extraction d'un son depuis un VOB en .wav 16bits

```
# ffmpeg 3.2.5-1 / debian 9.1 @ urzhiata / 20180415
ffmpeg -i ./VTS_01_1.VOB -vn -acodec pcm_s16le -sample_fmt s16 -ar 44.1k -ac 2 ./son.wav
```

Extraction d'images de film toutes les x secondes

```
# ffmpeg 3.2.5-1 / debian 9.1 @ urzhiata / 20180831
ffmpeg -i film.avi -r 1 -f image2 "image-du-film-%4d"
# pour des versions de ffmpeg plus récentes, essayer
ffmpeg -i video.mp4 -vf fps=1 out%d.png       # extraction d'une image par seconde
ffmpeg -i video.mp4 -vf fps=1/60 out%02d.jpg  # extraction d'une image par minute
```

Extraction de toutes les images d'une vidéo

```
# testé 1( juin 2023 / ffmpeg version 3.2.18-0+deb9u1
ffmpeg -i video.mp4 images_%04d.png
```

Vidéo à partir d'une image fixe et d'un son, pour publication en post instagram (source)

```
# testé 21 fev 2023 / ffmpeg version 3.2.18-0+deb9u1
ffmpeg -loop 1 -i image.png -i son.wav -map 0:v -map 1:a -r 25 -c:v h264 -tune stillimage -crf 18 -c:a aac -b:a 128k -ac 2 -ar 44100 -
pix_fmt yuv420p -max_muxing_queue_size 1024 -shortest video.mp4
```

-loop 1 : joue le flux vidéo en boucle, ici une image qui se répète indéfiniment (mais sera adaptée à la longueur du son, voir plus bas -shortest)

-map 0:v -map 1:a: comment les entrées sont mappées sur les flux de sortie, ici le flux 0 sera utilisé pour la vidéo principale et le flux 1 pour le son principal

-r 25: 25 images par seconde

-c:v h264 -tune stillimage -crf 18: encodage h264 basique, avec réglage optionnel pour image fixe

-c:a aac -b:a 128k -ac 2 -ar 44100: encodage audio selon les recommandations d'instagram : format AAC, 128Kbps, stereo, 44.100 kHz.

-pix_fmt yuv420p: encodage colorimétrique, cf. https://en.wikipedia.org/wiki/YUV

-max_muxing_queue_size 1024: instruction magique pour repousser le mauvais sort et les bugs

-shortest: adapte à la longueur du flux le plus court, ici le son car les mages sont en boucle

Vidéo à partir de plusieurs images fixes, sans son, pour publication en story instagram

```
# testé 31 mai 2023 / ffmpeg version 3.2.18-0+deb9u1
ffmpeg -f concat -i input.txt -map 0:v -c:v h264 -tune stillimage -crf 18 -an -pix_fmt yuv420p -max_muxing_queue_size 1024 -vsync vfr
output.mp4
```

Les images sont à la même définition : 1080×1920
Elles sont listés dans le fichier texte **input.txt** avec leur durée en seconde (la dernière image est répété dans le fichier texte, c'est comme ça!) :

```
file 'image1.png'
duration 5
file 'image2.png'
duration 1
file 'image3.png'
duration 3
file 'image3.png'
```

Atout de son à une vidéo pour publication instagram

```
# testé 6 juin 2023 / ffmpeg version 3.2.18-0+deb9u1
# sans réencodage de la vidéo
ffmpeg -i video.mp4 -i son.wav -map 0:v -map 1:a -c:v copy -shortest video3.mp4
# avec réencodage
ffmpeg -i video.mp4 -i son.wav -map 0:v -map 1:a -r 25 -c:v h264 -tune stillimage -crf 18 -c:a aac -b:a 128k -ac 2 -ar 44100 -pix_fmt
yuv420p -max_muxing_queue_size 1024 -shortest video2.mp4
```

Extraire une image d'une vidéo

```
ffmpeg -i video.mp4 -vframes 1 image.png
ffmpeg -i video.mp4 -ss 00:23:45 -vframes 1 image.png   # momen de capture de l'image défini par -ss
```

Ajouter du délai à un sous-titre externe (au format .srt)

```
# testé 27 avril 2023 / ffmpeg version 3.2.18-0+deb9u1
# cas ou les sous-titres sont en avance de 3s, on les retarde
ffmpeg -itsoffset 3 -i sous-titre.srt -c copy sous-titre_retard.srt
```

“Hardsub” (incruster des sous-titres dans une vidéo)

```
# testé 27 avril 2023 / ffmpeg version 3.2.18-0+deb9u1
ffmpeg -i film.mp4 -c:v libx264 -crf 18 -preset slow -c:a copy -vf "\subtitles=sous-titre.srt" film_sub.mp4
```

(-c:a copy : copie du flux audio, -crf : 18 qualité visuellement proche du lossless)
cf. infos sur les qualité du h264 : https://trac.ffmpeg.org/wiki/Encode/H.264

Changer le framerate et modifier la vitesse de lecture, sans réencoder (source)
En 2 étapes, en partant d'une vidéo à 60 fps pour la passer à 30 fps, et la ralentir (2 fois plus lente). Dans ce cs, je n'ai pas besoin de son mais l'exemple de la source concerne aussi le son.

```
# testé 12 juin 2023 / ffmpeg version 3.2.18-0+deb9u1
ffmpeg -i video_60fps.mp4 -c copy -f h264 output_raw_bitstream.h264
ffmpeg -r 30 -i output_raw_bitstream.h264 -c copy video_30fps.mp4
```

Vidéo sans son en ping-pong / yoyo (source)
Une fois à l'endroit, une fois à l'envers

```
# testé 12 juin 2023 / ffmpeg version 3.2.18-0+deb9u1
ffmpeg -i input.mp4 -filter_complex "[0:v]reverse,fifo[r];[0:v][r] concat=n=2:v=1 [v]" -map "[v]" output.mp4
```

Réencoder en changeant les dimensions

```
# testé 14 juin 2023 / ffmpeg version 3.2.18-0+deb9u1
ffmpeg -i video.mp4 -vf scale="360:360" -c:a aac -b:a 128k -ac 2 -ar 44100 -pix_fmt yuv420p output.mp4
```

Assembler une vidéo à partir d'une série d'images fixes, sans son

```
# testé 6 juin 2024 / ffmpeg version 3.2.18-0+deb9u1
ffmpeg -r 25 -an -i image-%03d.png -s 768x960 video.mp4 # image-001.png à image-999.png
```

Répéter une séquence x fois dans une même vidéo

```
# testé 7 juin 2024 sans son / ffmpeg version 3.2.18-0+deb9u1
ffmpeg -f concat -i sequences.txt -c copy video.mp4
```

Au préalable il faut avoir préparé un fichier texte comme ceci :

```
# sequences.txt
file 'scene.mp4'
file 'scene.mp4'
file 'scene.mp4'
```

Assembler une vidéo de 10 images à partir d'une image unique, sans son

```
# testé 10 juin 2024 sans son / ffmpeg version 3.2.18-0+deb9u1
ffmpeg -loop 1 -i image.png -c:v libx264 -t 1 -r 10 -pix_fmt yuv420p -vf scale=1536:1920 video.mp4
```

-t 1 : durée de la vidéo = 1 seconde

-r 10 : nombre d'images par secondes

Ressources

Pour **éditer des métadonnées avec une interface graphique**, on peut utiliser picard : https://picard.musicbrainz.org/ , des alternatives existent!
Pour **éditer des métadonnées en ligne de commande** : lltag ou id3v2 ou id3tool

Pour en apprendre plus sur ffmpeg :

- documentation : https://ffmpeg.org/ffmpeg.html
- https://ffmpegfromzerotohero.com/blog/ (c'est aussi un livre)
- mixage audio / vidéo : https://stackoverflow.com/a/11783474

Article extrait de : http://www.lesporteslogiques.net/wiki/ - **WIKI Les Portes Logiques**
Adresse : **http://www.lesporteslogiques.net/wiki/ressource/logiciel/ffmpeg/start?rev=1719242283**
Article mis à jour: **2024/06/24 17:18**