

Script ffmpeg - Copie DVD-VIDÉO

- Objet : du tuto Ripper des dvd avec choix du titre et du bitrate
- Niveau requis :
[débutant](#)
- Commentaires : *Copies de dvd avancée via ffmpeg dans un script bash.*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :
[à-tester](#)
 - Création par [smolski](#) 25/04/2020 d'après un script de Golgot200
- Commentaires sur le forum : [Lien vers le forum concernant ce tuto](#) ¹⁾

Introduction

script utilisant ffmpeg pour copier des dvd video où en plus du titre, tu auras le choix du bitrate par défaut ("sans compression)" ou si tu ne veux pas, un autre choix te sera proposé pour le rentrer directement dans le terminal.

Installation

Installer [ffmpeg](#)

script - choix du titre et du bitrate

Se placer dans le bon répertoire de son user :

```
cd ~/usr/local/bin/
```

En root, créer un fichier :

```
gedit cdvd
```

copier le script suivant :

[ripdvd](#)

```
#!/bin/bash

## golgot200
## Testeur smolski
## 29/04/2020

START_TIME=$(date +%s)
```

```
PATH=$PATH:/sbin:/usr/sbin

## CONDITION : UN SEUL DVD-VIDÉO.
mounting_directory=$( mount | grep '/dev/sr' | grep -Po
'on\K.*(?:=type)' | sed 's|\\(.*)|.|\\1|' | uniq )
COUNT_VIDEO_TS=$(find ${mounting_directory} -maxdepth 3 -name
'VIDEO_TS.IFO' 2> /dev/null | wc -l)

DEVICE=$(blkid /dev/sr* | awk '{print $1}' | sed -e 's://g')
COUNT_DEVICE=$(blkid /dev/sr* | awk '{print $1}' | sed -e 's://g' | wc
-l)

if [[ "$COUNT_VIDEO_TS" -eq 0 ]]; then
    echo -e "\e[3;4;32m Aucun DVD-VIDEO monté : Abandon\e\n[0m"
    exit 0
elif [[ "$COUNT_VIDEO_TS" -gt 1 ]]; then
    echo -e "\e[3;4;32m Il y a plus d'un DVD-VIDEO détecté :
Abandon\e\n[0m"
    exit 0

fi

if [[ "$COUNT_VIDEO_TS" -eq 1 ]] && [[ "$COUNT_DEVICE" -eq 1 ]]; then
echo -e "\e[3;4;32m Les conditions sont réunies : On continue\e\n[0m"

TITLE=$(lsdvd /dev/sr0 | head -1 | awk '{print $3}')
TITLE_COUNT=$(lsdvd "$DEVICE" | grep -c ^'Title: ')

fi

echo "Proceed..."

echo -e "DVD-Vidéo : \e[32m$TITLE\e[0m"
read -r -p "Ce nom convient-il ? [Y/n] " input
case $input in
[yY][eE][sS]|[yY])

echo -e "\e[3;4;32m Réponse:\n\e[0m  \e[32mOui\e[0m"
echo ""
mkdir -p "$TITLE/IFOS"
mkdir -p "$TITLE/VOBSUBS"
```

```
;;
[nN][oO]||[nN])
echo -e "\e[3;4;32m Réponse:\n\e[0m \e[33mNon\e[0m"
read -r -p "Entrer un nouveau nom: " TITLE
if [[ -z "$TITLE" ]]; then
    echo "Invalid input..."
    echo "Quit"
    exit 1
else
    echo ""
    mkdir -p "$TITLE/IFOS"
    mkdir -p "$TITLE/VOBSUBS"
fi
;;
*)
echo "Invalid input..."
echo "Quit"
exit 1
;;
esac

default_limit=-1

echo -e "\e[3;4;32m La limite est fixée pour encoder tous les
titres.\e[0m"
read -r -p "Encoder tous les titres ? [Y/n] " input
case $input in
[yY][eE][sS]||[yY])

    echo -e "\e[3;4;32m Réponse:\n\e[0m \e[32mOui, on encode tous les
titres.\e[0m"
    echo "$default_limit" >> "$TITLE/time_limit.txt"
    echo ""
    ;;
[nN][oO]||[nN])
echo -e "\e[3;4;32m Réponse:\n\e[0m \e[33mNon, je ne veux encoder
que les titres d'une certaine durée.\e[0m"
read -r -p "Entrer une limite (ex:1800) en secondes: " time_limit
if [[ -z "$time_limit" ]]; then
    echo "Invalid input..."
    echo "Quit"
    rm -r "$TITLE"
    echo -e "Supression du dossier \e[32m$TITLE\e[0m"
    exit 1
else
    times=$((time_limit / 60))
    echo -e "Seul les titres de \e[94m$times minutes\e[0m et plus
```

```

seront pris en compte."
    echo "$time_limit" >> "$TITLE/time_limit.txt"
    echo ""
fi
;;
*)
    echo "Invalid input..."
    echo "Quit"
    rm -r "$TITLE"
    echo -e "Supression du dossier \e[32m$TITLE\e[0m"
    exit 1
;;
esac

echo -e "\e[3;4;32m la langue française est privilégiée par
défaut.\e[0m"
read -r -p "Privilégier cette option ? [Y/n] " input
case $input in
    [yY][eE][sS]|[yY])

        echo -e "\e[3;4;32m Réponse:\n\e[0m  \e[32mOui, on garde cette
option.\e\n[0m"
        echo "1" > "$TITLE/choix_langues.txt"
        ;;
    [nN][oO]||[nN])
        echo -e "\e[3;4;32m Réponse:\n\e[0m  \e[33mNon, on garde toutes les
langues.\e[0m"
        echo "0" > "$TITLE/choix_langues.txt"

        ;;
    *)
        echo "Invalid input..."
        echo "Quit"
        rm -r "$TITLE"
        echo -e "Supression du dossier \e[32m$TITLE\e[0m"
        exit 1
        ;;
esac

read -r -p "Conserver le bitrate d'origine ? [Y/n] " input
case $input in
    [yY][eE][sS]|[yY])

        echo -e "\e[3;4;32m Réponse:\n\e[0m  \e[32mOui\e[0m"

for (( i=1; i<=TITLE_COUNT; i++ ))
do

```

```

BITRATE=$(mplayer -noconfig all -nocache -vo null -ao null -frames 0 -
identify -dvd-device "$DEVICE" dvdnav://" $i" 2>/dev/null | \
    grep "VIDEO:" | awk '{print$8}' | sed 's/\.[^ ]*/ /g')
echo "$BITRATE" >> "$TITLE/bitrate.txt"
done

;;
[nN][o0]||[nN])
echo -e "\e[3;4;32m Réponse:\n\e[0m \e[33mNon, on applique un
bitrate de son choix\e[0m"

read -r -p "Appliquer un nouveau Bitrate (ex 1500): " newbitrate
echo -e "\e[3;4;33mChoix du bitrate:\n\e[0m \e[3;6;94m$newbitrate
kbits/s\e[0m (pour tous les titres)"

if [[ -z "$newbitrate" ]]; then
echo -e " \e[3;6;94m Bitrate erroné\e[0m"
echo "Quit"
rm -r "$TITLE"
echo -e "Supression du dossier \e[32m$TITLE\e[0m"
exit 1
else
for (( i=1; i<=TITLE_COUNT; i++ ))
do
echo "$newbitrate" >> "$TITLE/bitrate.txt"
done
fi
*)
echo "Invalid input..."
echo "Quit"
rm -r "$TITLE"
echo -e "Supression du dossier \e[32m$TITLE\e[0m"
exit 1
;;
esac

for (( i=1; i<=TITLE_COUNT; i++ ))
do

bv=$(sed -n "$i p" "$TITLE/bitrate.txt")
bitrate_k=$((bv/1))k
bufsize_k=$((bv/2))k

LENGTH=$(mplayer -noconfig all -nocache -vo null -ao null -frames 0 -
identify -dvd-device "$DEVICE" dvdnav://" $i" 2>/dev/null | \

```

```

sed -E -n 's/^. *ID_LENGTH=([0-9]+).*$/\1/p')

#time_limit=$(sed q "$TITLE/time_limit.txt") ## utilité ? ##
[[ "$LENGTH" -le "$time_limit" ]] && continue

mplayer -noconfig all -nocache -dvd-device "$DEVICE" dvdnav://" $i"
\
    -dumpstream -dumpfile "$TITLE/$i-$TITLE.vob"

INTERLEAVED_FRAMES=$(ffmpeg -filter:v idet -frames:v 200 -an -f
rawvideo \
    -y /dev/null -i "$TITLE/$i-$TITLE.vob" 2>&1 | \
    awk '/(Single|Multi) frame detection/ {sum += $8+$10}
    END { print sum }')

if [[ "$INTERLEAVED_FRAMES" -gt "100" ]]; then
    FILTERS='-filter:v
yadif=0:-1:0,hue=b=0.0:s=0.7,hqdn3d=1.5:1.5:4:4,unsharp=luma_msize_x=5:
luma_msize_y=5:luma_amount=0.5,noise=c0s=2:c1s=2:c2s=2:c0f=a+t'
else
    FILTERS='-filter:v
hue=b=0.0:s=0.7,hqdn3d=1.5:1.5:4:4,unsharp=luma_msize_x=5:luma_msize_y=
5:luma_amount=0.5,noise=c0s=2:c1s=2:c2s=2:c0f=a+t'

fi

COUNT_AUDIO=$(ffprobe "$TITLE/$i-$TITLE.vob" 2>&1 | grep -c "Audio:")
FPS=$(mplayer -noconfig all -nocache -vo null -ao null -frames 0 -
identify "$TITLE/$i-$TITLE.vob" 2>/dev/null | grep "VIDEO:" | awk
'{print$6}')

RATIO=$(lsdvd -x "$DEVICE" -t "$i" | grep -m1 "Aspect ratio:" | awk
'{print$11}' | sed -e 's/\\/\\/\\/g' | tr -d ',')

O_CHANNELS=$(ffprobe "$TITLE/$i-$TITLE.vob" 2>&1 | grep -c '(Audio:
ac3, 0 channels)')

REMOVE_O_CHANNELS=$(ffprobe "$TITLE/$i-$TITLE.vob" 2>&1 | awk '/Stream/
&& /Audio:/ && /kb\s|\\(LC\\),/ { print $2 }' | \
    cut -d "[" -f2 | cut -d "]" -f1 | awk '{ printf " -map i:"$0"

```

```

" }')

ALLSHOW_TAG=$(mplayer -vo null -ao null -frames 0 -identify
dvdnav://" $i" -dvd-device "$DEVICE" 2>/dev/null | \
    awk '/audio stream:/ {printf " -metadata:s:a:"$3"
language="$8" -metadata:s:a:"$3" title="$5$6"}')

MAP_ID_STREAMS=$(lsdvd -a "$DEVICE" -t "$i" 2> /dev/null | grep
"Language:" | sed -e 's/c/lc/g' | awk '{ printf " -map i:"$21" }')

MAP_ID_STREAMS_LPCM=$(lsdvd -a "$DEVICE" -t "$i" 2> /dev/null | grep
"lpcm" | awk '{ printf " -map i:"$22" }')

COUNT_LPCM=$(lsdvd -a "$DEVICE" -t "$i" | grep -c 'lpcm')

AIDS_LPCM=$(lsdvd -a "$DEVICE" -t "$i" | awk '/lpcm/ { print $22 }')

## VOB PRÉSENT.
## AUCUN AUDIO.
if [[ -e "$TITLE/$i-$TITLE.vob" ]] && [[ "$COUNT_AUDIO" -eq "0" ]];
then

echo -e "Le bitrate du titre n°\e[94m$i\e[0m sera donc de :
\e[32m$bitrate_k\e[0m"
sleep 2

    ffmpeg -hide_banner -analyzeduration 1000M -probesize 1000M -vsync
2 -r "$FPS" -i "$TITLE/$i-$TITLE.vob" \
    -map 0:v -metadata title="$TITLE" \
    -c:v libx264 -b:v "$bitrate_k" -maxrate "$bitrate_k" -bufsize
"$bufsize_k" \
    -loglevel repeat+verbose -movflags faststart \
    -x264opts 'keyint=300:min-keyint=25:8x8dct:sliced-threads=0' \
    -an -movflags +faststart -max_muxing_queue_size 9999 \
    -deblock 1:1 -flags +loop -qcomp 0.60 -qblur 0.5 -coder 1 -
me_range 6 -sc_threshold 42 -bf 10 -trellis 2 -mbtree 1 -qmin 0 -qmax
52 \
    $FILTERS \
    -preset veryfast -y "$TITLE/$i-$TITLE.mkv"

rm -f "$TITLE/$i-$TITLE.vob"

fi

```

```

## VOB PRÉSENT.
## AU MOINS UN AUDIO OU PLUS.
## SANS 0 CHANNEL.
## SANS LPCM.
if [[ -e "$TITLE/$i-$TITLE.vob" ]] && [[ "$COUNT_AUDIO" -ge "1" ]] &&
[[ "$0_CHANNELS" -eq "0" ]] && [[ "$COUNT_LPCM" -eq "0" ]]; then

echo -e "Le bitrate du titre n°\e[94m$i\e[0m sera donc de :
\e[32m$bitrate_k\e[0m"
sleep 2

    ffmpeg -hide_banner -analyzeduration 1000M -probesize 1000M -vsync
2 -r "$FPS" -i "$TITLE/$i-$TITLE.vob" \
        -map 0:v -metadata title="$TITLE" \
        -c:v libx264 -b:v "$bitrate_k" -maxrate "$bitrate_k" -bufsize
"$bufsize_k" \
        -loglevel repeat+verbose -movflags faststart \
        -x264opts 'keyint=300:min-keyint=25:8x8dct:sliced-threads=0' \
        ${MAP_ID_STREAMS} -c:a copy ${ALLSHOW_TAG} -movflags
+faststart -max_muxing_queue_size 9999 \
        -deblock 1:1 -flags +loop -qcomp 0.60 -qblur 0.5 -coder 1 -
me_range 6 -sc_threshold 42 -bf 10 -trellis 2 -mbtree 1 -qmin 0 -qmax
52 \
        $FILTERS \
        -preset veryfast -y "$TITLE/$i-$TITLE.mkv"

#          ffmpeg -hwaccel auto -c:v mpeg2_cuvld -analyzeduration 1000M -
probesize 1000M -i "$TITLE/$i-$TITLE.vob" \
#          -map 0:v -metadata title="$TITLE" \
#          -c:v h264_nvenc -preset medium -b:v $BITRATE_K -bufsize
$BUFSIZE_K -profile:v high -bf 3 -b_ref_mode 0 -temporal-aq 1 -rc-
lookahead 20 -vsync 0 \
#          -loglevel repeat+verbose \
#          -filter:v
hue=b=0.0:s=0.7,hqdn3d=1.5:1.5:4:4,unsharp=luma_msize_x=5:luma_msize_y=
5:luma_amount=0.5,noise=c0s=2:c1s=2:c2s=2:c0f=a+t \
#          ${MAP_ID_STREAMS} -c:a copy $ALLSHOW_TAG -movflags
+faststart \
#          -y "$TITLE/$i-$TITLE.mkv"

rm -f "$TITLE/$i-$TITLE.vob"

fi

```



```

## VOB PRÉSENT.
## AU MOINS UN AUDIO OU PLUS.
## UN 0 CHANNEL OU PLUS.
## SANS LPCM.

if [[ -e "$TITLE/$i-$TITLE.vob" ]] && [[ "$COUNT_AUDIO" -ge "1" ]] &&
[[ "$0_CHANNELS" -ge "1" ]] && [[ "$COUNT_LPCM" -eq "0" ]]; then

echo -e "Le bitrate du titre n°\e[94m$i\e[0m sera donc de :
\e[32m$bitrate_k\e[0m"
sleep 2

    ffmpeg -hide_banner -analyzeduration 1000M -probesize 1000M -vsync
2 -r "$FPS" -i "$TITLE/$i-$TITLE.vob" \
        -map 0:v -metadata title="$TITLE" \
        -c:v libx264 -b:v "$bitrate_k" -maxrate "$bitrate_k" -bufsize
"$bufsize_k" \
        -loglevel repeat+verbose -movflags faststart \
        -x264opts 'keyint=300:min-keyint=25:8x8dct:sliced-threads=0' \
        ${REMOVE_0_CHANNELS} -c:a copy ${ALLSHOW_TAG} -movflags
+faststart -max_muxing_queue_size 9999 \
        -deblock 1:1 -flags +loop -qcomp 0.60 -qblur 0.5 -coder 1 -
me_range 6 -sc_threshold 42 -bf 10 -trellis 2 -mbtree 1 -qmin 0 -qmax
52 \
        $FILTERS \
        -preset veryfast -y "$TITLE/$i-$TITLE.mkv"

rm -f "$TITLE/$i-$TITLE.vob"

fi

## VOB PRÉSENT.
## AVEC OU SANS AUDIO(S).
## SANS 0 CHANNEL.
## AU MOINS UN LPCM OU PLUS.

if [[ -e "$TITLE/$i-$TITLE.vob" ]] && [[ "$COUNT_AUDIO" -ge "0" ]] &&
[[ "$0_CHANNELS" -eq "0" ]] && [[ "$COUNT_LPCM" -ge "1" ]]; then

echo -e "Le bitrate du titre n°\e[94m$i\e[0m sera donc de :
\e[32m$bitrate_k\e[0m"

```

```

sleep 2

ffmpeg -hide_banner -analyzeduration 1000M -probesize 1000M -vsync
2 -r "$FPS" -i "$TITLE/$i-$TITLE.vob" \
    -map 0:v -metadata title="$TITLE" \
    -c:v libx264 -b:v "$bitrate_k" -maxrate "$bitrate_k" -bufsize
"$bufsize_k" \
    -loglevel repeat+verbose -movflags faststart \
    -x264opts 'keyint=300:min-keyint=25:8x8dct:sliced-threads=0' \
    ${MAP_ID_STREAMS_LPCM} -acodec pcm_s16be ${ALLSHOW_TAG} -
movflags +faststart -max_muxing_queue_size 9999 \
    -deblock 1:1 -flags +loop -qcomp 0.60 -qblur 0.5 -coder 1 -
me_range 6 -sc_threshold 42 -bf 10 -trellis 2 -mbtree 1 -qmin 0 -qmax
52 \
    $FILTERS \
    -preset veryfast -y "$TITLE/$i-$TITLE.mkv"

rm -f "$TITLE/$i-$TITLE.vob"

mkdir -p "$TITLE/WAVE"

for n in $AIDS_LPCM; do
    mplayer -noconfig all -nocache -benchmark -vc null -vo null \
        -ao pcm:fast:file="$TITLE/WAVE/[$i-($n)]-$TITLE.wav" \
        -identify -dvd-device "$DEVICE" dvdnav://" $i -ni -aid
"$n"
done

fi

choix_languages=$(sed q "$TITLE/choix_languages.txt")
COUNT_AUDIO_FR=$(mplayer -noconfig all -nocache -vo null -ao null -
frames 0 -identify "$TITLE/$i-$TITLE.mkv" 2>/dev/null |awk '/-alang
fr,/ { print $9 }' | wc -w)

## AU MOINS UN AUDIO FRANÇAIS OU PLUS.
if [[ -e "$TITLE/$i-$TITLE.mkv" ]] && [[ "$COUNT_AUDIO_FR" -ge "1" ]]
&& [[ "$choix_languages" -eq "1" ]] ; then

    mkvmerge -o "$TITLE/$i-$TITLE-[Language: Français].mkv" -a fre
"$TITLE/$i-$TITLE.mkv"

rm -f "$TITLE/$i-$TITLE.mkv"
mv "$TITLE/$i-$TITLE-[Language: Français].mkv" "$TITLE/$i-
$TITLE.mkv"

```

```

fi

## ON INDIQUE LA LANGUE DE L'AUDIO DANS LE RENOMAGE DU TITRE ET LE
NOMBRE.
INFO=$(mplayer -vo null -ao null -frames 0 -identify "$TITLE/$i-
$TITLE.mkv" 2>/dev/null | \
    awk '/alang/ { x = x $9 " "}
    END { sub(/,*$/, "", x); print x }')
COUNT_INFO=$(mplayer -vo null -ao null -frames 0 -identify "$TITLE/$i-
$TITLE.mkv" 2>/dev/null | grep -c "alang")
COUNT=$(mplayer -vo null -ao null -frames 0 -identify "$TITLE/$i-
$TITLE.mkv" 2>/dev/null | grep -c "aid")

## UNE OU PLUSIEURS INFOS SUR L'AUDIO ET UN OU PLUSIEURS AUDIOS.
## INTÉGRATION DES CHAPITRES.
if [[ -e "$TITLE/$i-$TITLE.mkv" ]] && [[ "$COUNT_INFO" -ge "1" ]] && [[
"$COUNT" -ge "1" ]] ; then
    mv "$TITLE/$i-$TITLE.mkv" "$TITLE/$i-$TITLE-[Audio: $INFO].mkv"
    ddxchap -t "$i" "$DEVICE" > "$TITLE/$i-$TITLE-Chapters.txt"
    mkvmerge "$TITLE/$i-$TITLE-[Audio: $INFO].mkv" --chapters
"$TITLE/$i-$TITLE-Chapters.txt" -o "$TITLE/$i-$TITLE-[Audio: $INFO +
Chapters].mkv"
    rm -f "$TITLE/$i-$TITLE-[Audio: $INFO].mkv"
    rm -f "$TITLE/$i-$TITLE-Chapters.txt"
    mediainfo "$TITLE/$i-$TITLE-[Audio: $INFO + Chapters].mkv" >>
"$TITLE/IFOS/$i-$TITLE-[Audio: $INFO + Chapters].log"

fi

SIDFR=$(lsdvd -s "$DEVICE" -t "$i" 2> /dev/null | awk '/Subtitle:/ &&
/Language: fr/ { sub(/,$/ , "", $2);
                                if ($2 ~ /^[0-9]+$/) print (--$2)
}')

COUNT_SID_FR=$(echo "$SIDFR" | wc -w)

## AU MOINS UN SUB FRANÇAIS OU PLUS (ON EXTRAIT TOUT ÇA).

```

```
if [[ "$COUNT_SID_FR" -ge "1" ]]; then

mkdir -p "$TITLE/VOBSUBS/$i-$TITLE"

for n in $SIDFR; do mencoder -dvd-device "$DEVICE" dvdnav://" $i" -
nosound -ovc frameno -force-avi-aspect "$RATIO" -o /dev/null \
        -ifo "/run/media/$USER/$TITLE/VIDEO_TS/VTS_01_0.IFO" -
sid "$n" -vobsubout "$TITLE/$n-vobsubs-fr" -vobsuboutindex "$n"

## ON CORRIGE LES ID VIDES SI BESOIN.
sed -i 's/id:./,/id: fr,/g' "$TITLE/$n-vobsubs-fr.idx"

taille_idx=$(stat -c%s "$TITLE/$n-vobsubs-fr.idx")
taille_sub=$(stat -c%s "$TITLE/$n-vobsubs-fr.sub")

if [[ "$taille_idx" -gt 1000 ]] && [[ "$taille_sub" -gt 0 ]] ; then

echo -e "\e[32m$n-vobsubs-fr.idx et $n-vobsubs-fr.sub sont bons.\e[0m"

else

## ON ISOLE LES VOBSUBS INVALIDES DANS LE DOSSIER VOBSUBS
echo -e "\e[32m$n-vobsubs-fr.idx ou $n-vobsubs-fr.sub incorrect ...
déplacement des deux.\e[0m"

mv "$TITLE/$n-vobsubs-fr.idx" "$TITLE/$n-vobsubs-fr.sub"
"$TITLE/VOBSUBS/$i-$TITLE"

fi

done

fi

## ON COMPTE CE QU'IL RESTE DE BON EN IDX/SUB.
COUNT_IDX_FR=$(find "$TITLE" -maxdepth 1 -name '*.idx' | wc -l)
COUNT_SUB_FR=$(find "$TITLE" -maxdepth 1 -name '*.sub' | wc -l)

## CONDITION : UN IDX/SUB OU PLUS.
if [[ -e "$TITLE/$i-$TITLE-[Audio: $INFO + Chapters].mkv" ]] && [
"$COUNT_IDX_FR" -ge "1" ] && [ "$COUNT_SUB_FR" -ge "1" ]
then
```

```
mkvmerge "$TITLE/$i-$TITLE-[Audio: $INFO + Chapters].mkv" --language
"0:fre" "$TITLE"/*.idx -o "$TITLE/$i-$TITLE-[Audio: $INFO + Chapters +
Sub].mkv"

## ON DÉPLACE LE RESTANT DES VOBSUBS UNE FOIS MIXÉS AVEC LES INVALIDES.
mv "$TITLE"/*.idx "$TITLE"/*.sub "$TITLE/VOBSUBS/$i-$TITLE"
rm -f "$TITLE/$i-$TITLE-[Audio: $INFO + Chapters].mkv"

INFO_SUB=$(mplayer -vo null -ao null -frames 0 -identify "$TITLE/$i-
$TITLE-[Audio: $INFO + Chapters + Sub].mkv" 2>/dev/null | \
    awk '/slang/ && /fre/ { x = x $9 "," }
    END { sub(/,*$/, "", x); print x }')

    mv "$TITLE/$i-$TITLE-[Audio: $INFO + Chapters + Sub].mkv"
"$TITLE/$i-$TITLE-[Audio: $INFO + Chapters + Sub: $INFO_SUB].mkv"
    mediainfo "$TITLE/$i-$TITLE-[Audio: $INFO + Chapters + Sub:
$INFO_SUB].mkv" >> "$TITLE/IFOS/$i-$TITLE-[Audio: $INFO + Chapters +
Sub: $INFO_SUB].log"

rm -f "$TITLE/IFOS/$i-$TITLE-[Audio: $INFO + Chapters].log"

fi

## AUCUN IDX ET SUB. ON TAG EN NOSUB
if [[ -e "$TITLE/$i-$TITLE-[Audio: $INFO + Chapters].mkv" ]] && [
"$COUNT_IDX_FR" -eq "0" ] && [ "$COUNT_SUB_FR" -eq "0" ]
then

mv "$TITLE/$i-$TITLE-[Audio: $INFO + Chapters].mkv" "$TITLE/$i-$TITLE-
[Audio: $INFO + Chapters NoSub].mkv"
mediainfo "$TITLE/$i-$TITLE-[Audio: $INFO + Chapters NoSub].mkv" >>
"$TITLE/IFOS/$i-$TITLE-[Audio: $INFO + Chapters NoSub].log"

rm -f "$TITLE/IFOS/$i-$TITLE-[Audio: $INFO + Chapters].log"

fi

## PAS D'INFO ET UN SEUL AUDIO. ON TAG EN UND
if [[ -e "$TITLE/$i-$TITLE.mkv" ]] && [[ "$COUNT_INFO" -eq "0" ]] && [[
```

```

"$COUNT" -eq "1" ]] ; then
    mv "$TITLE/$i-$TITLE.mkv" "$TITLE/$i-$TITLE-[Audio: Und].mkv"
    mediainfo "$TITLE/$i-$TITLE-[Audio: Und].mkv" >> "$TITLE/IFOS/$i-$TITLE-[Audio: Und].log"

fi

## PAS D'AUDIO DONC PAS D'INFO. ON TAG EN NOSOUND
if [[ -e "$TITLE/$i-$TITLE.mkv" ]] && [[ "$COUNT_INFO" -eq "0" ]] && [[ "$COUNT" -eq "0" ]] ; then
    mv "$TITLE/$i-$TITLE.mkv" "$TITLE/$i-$TITLE-[Audio: NoSound].mkv"
    mediainfo "$TITLE/$i-$TITLE-[Audio: NoSound].mkv" >> "$TITLE/IFOS/$i-$TITLE-[Audio: NoSound].log"

fi

done

rm -f "$TITLE/bitrate.txt"
rm -f "$TITLE/time_limit.txt"
rm -f "$TITLE/choix_langues.txt"
echo -e "\e[32mJob terminé.\e[0m"

echo -e "\e[32mTemps total écoulé:\e[0m \e[3;6;94m$(date -ud "@$(( $(date +%s) - START_TIME))" +%T)\e[0m (HH:MM:SS)"

```

Puis collez-le dans le fichier créé. Enregistrer

Modifier les droits de ce fichier pour votre user. [Le droit d'appartenance](#) :

```
chown user:user cdvd
```

Ajouter [le droit d'exécution](#) :

```
chmod u+x cdvd
```

Revenir à son home/user :

```
cd
```

Et quitter root avec le raccourci ; Ctrl d

Et voilà, plus qu'à insérer un dvd et à lancer la commande en user.

```
cdvd
```

```
Les conditions sont réunies : On continue  
Proceed...  
DVD-Vidéo : DVD_VIDEO_RECORDING  
Ce nom convient-il ? [Y/n]
```

On valide n pour modifier le titre

```
Ce nom convient-il ? [Y/n] n
```

```
Réponse : Non  
Entrer un nouveau nom:  
nouveau_nom
```

```
Garder le bitrate d'origine ? [Y/n]  
n  
Réponse : Non  
Entrer un nouveau Bitrate:
```

Par exemple

```
1000
```

Plus il est élevé, plus la qualité sera grande.

```
Le bitrate du titre n°1 sera donc de : 1000k
```

Et c'est parti...



Si vous avez plusieurs vidéos, chacune pourra être régler à sa façon !

¹⁾

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

From:
<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:
<http://debian-facile.org/doc:programmation:bash:ffmpeg>

Last update: **29/04/2020 20:26**

