

Regarder les films débiles de YouTube avec uniquement du logiciel libre

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 21 octobre 2009. Dernière mise à jour le 4 novembre 2010

<https://www.bortzmeyer.org/youtube-en-logiciel-libre.html>

Aujourd'hui, l'Internet sert presque uniquement à regarder les vidéos publiées sur YouTube. Mais le mode de publication par défaut est sous forme d'animation Flash, une technologie fermée, contrôlée par un seul acteur, Adobe. Peut-on voir les mêmes films débiles que tout le monde avec uniquement du logiciel libre ?

Si on utilise une machine qui ne fait tourner que du logiciel libre, dans mon cas une Debian, version « lenny » qui n'a que la section « *main* », celle qui ne contient que du logiciel libre, et donc pas celui de la section « *non-free* », que voit-on sur YouTube ? « *Hello, you either have JavaScript turned off or an old version of Adobe's Flash Player. Get the latest Flash player.* » Eh oui, YouTube impose l'utilisation de la technologie fermée et privatrice Flash.

Peut-on utiliser un autre logiciel pour voir ses animations ? Gnash, équivalent d'Adobe Flash en logiciel libre n'a jamais vraiment atteint un niveau suffisant pour être utilisé couramment <<https://signal.eu.org/blog/2007/10/04/gnash-le-clone-libre-de-flash/>>.

Mais il existe une autre solution. La page renvoyée par YouTube contient le nom du film (au format FLV). On peut donc récupérer celui-ci avec un logiciel comme wget ou curl. Il y a toutefois plus pratique, utiliser l'excellent logiciel youtube-dl <<http://github.com/rg3/youtube-dl/>> qui automatise tout cela. Il analyse le source HTML de la page pour vous et récupère le film. Une fois celui-ci stocké sur votre disque, vous pourrez le regarder à loisir avec des logiciels comme mplayer ou vlc (FLV est lisible par beaucoup de logiciel libres, il n'y a même pas besoin de codecs non-libres).

Voici un exemple d'utilisation :

```
% youtube-dl -t 'http://www.youtube.com/watch?v=ro6sTexFQbU'
[youtube] Setting language
[youtube] ro6sTexFQbU: Downloading video info webpage
[youtube] ro6sTexFQbU: Extracting video information
[download] Destination: A_la_belle_viiiie-ro6sTexFQbU.flv
[download] 100.0% of 6.24M at 65.24k/s ETA 00:00
% mplayer A_la_belle_viiiie-ro6sTexFQbU.flv
...
```

Les apostrophes autour de l'URL sont là parce que le point d'interrogation est un caractère spécial pour le shell. L'option `-t` est là pour que le fichier prenne comme nom le titre du film (autrement, le fichier a un nom inhumain, l'identifiant YouTube du film.)

`youtube-dl` fonctionne par "*scraping*" de la page Web. Une telle technique est très fragile : il suffit que YouTube change son format et tout est à recommencer. En pratique, lorsque cela arrive, une nouvelle version de `youtube-dl` sort en général rapidement.

Enfin, pour venir, quelques bons films choisis tout à fait subjectivement et arbitrairement sur YouTube :

- Une bonne représentation de la fracture numérique sous forme d'une course de voitures entre les RIR : <<http://www.youtube.com/watch?v=-g-TOkMqufI>>.
- Tous les épisodes de « Avez-vous déjà vu..? », par exemple <<http://www.youtube.com/watch?v=n8twu3hpILs>>.
- L'immortel éléphant sur le trampoline : <<http://www.youtube.com/watch?v=TK27aknWVI4>>.
- Le formidable "*Food fight*" ou la seconde guerre mondiale où les camps en présence sont de la nourriture : <<http://www.youtube.com/watch?v=e-yldqNkGfo>>.

Je trouve personnellement que `youtube-dl` manque de quelques options pratiques comme la possibilité de lancer un programme de visualisation une fois le chargement terminé. Le petit script shell suivant répond à ce problème. On l'utilise ainsi :

```
% yt 'http://www.youtube.com/watch?v=7sjlX8ueFfU'
[youtube] Setting language
[youtube] 7sjlX8ueFfU: Downloading video info webpage
[youtube] 7sjlX8ueFfU: Extracting video information
[download] Destination: UN_VILLAGE_FRANCAIS_SAISSON_2-7sjlX8ueFfU.flv
[download] 100.0% of 1.19M at 284.76k/s ETA 00:00
MPlayer 1.0rc2-4.3.2-DFSG-free (C) 2000-2007 MPlayer Team
CPU: Intel(R) Pentium(R) Dual CPU E2140 @ 1.60GHz (Family: 6, Model: 15, Stepping: 13)
...
Playing UN_VILLAGE_FRANCAIS_SAISSON_2-7sjlX8ueFfU.flv.
```

et on a à la fois le film enregistré en un endroit précis et sa visualisation tout de suite. Le script est simplement :

```
#!/bin/sh

if [ -z "$1" ]; then
    echo "Usage: $0 YouTube-URL" >&2
    exit 1
fi

url=$1
LOGFILE=youtube-dl-$$$.log

set -e
cd ~/tmp
```

```
youtube-dl -t $url | tee $LOGFILE
file=$(awk '/Destination:/ {print $3}' < $LOGFILE)
if [ -z $file ]; then
    echo "$url: invalid YouTube URL" >&2
    exit 1
fi
mplayer $file
rm -f $LOGFILE
```

Comme `youtube-dl` n'a apparemment pas d'option pour indiquer simplement le nom du fichier enregistré sur la sortie standard (les options `-g` et `-e` n'affichent pas exactement ce nom, et elles inhibent le téléchargement), il faut utiliser `awk` pour analyser le résultat de `youtube-dl`.

Bien sûr, comme le fait de pouvoir regarder YouTube fait partie des droits de l'homme fondamentaux, plein de gens ont développé d'autres mécanismes pour le faire avec du logiciel libre. Ainsi, Yannick Roehly, me signale `get-flash-videos` <<http://code.google.com/p/get-flash-videos/>>, quasiment l'équivalent de `youtube-dl` mais écrit en Perl (et qui gère d'autres services que YouTube). Christophe Narbonne me recommande YouTube Perfect <<http://userscripts.org/scripts/show/38074>>, un script Greasemonkey pour voir YouTube. Marc Hertzog suggère un truc pour obtenir de YouTube <<http://googlesystem.blogspot.com/2008/04/download-youtube-videos-as-mp4-files.html>> un fichier MP4. Et Yannick Palanque préfère utiliser `cclive` <<http://code.google.com/p/cclive/>> (il existe aussi un `clive` <<http://clive.googlecode.com/>>).

Et pour Dailymotion ? Pas de problème, `youtube-dl` le gère depuis peu mais, apparemment, `clive` et `cclive` le font aussi. Notez que Dailymotion, contrairement à son concurrent, a en test une distribution de vidéos ne dépendant pas de logiciel non-libre : <<http://openvideo.dailymotion.com/>>. (Voir le communiqué de la FSF, « *"Dailymotion's support for Ogg is a big deal"* <<http://www.fsf.org/blogs/community/dailymotion-theora>> ».) Pour le service Vimeo, en revanche, il y a <<http://gist.github.com/146789>> (suggestion de Marc Hertzog).