

FFmpeg



my recommendations on research & development tools.

筒口 拳†

キーワード：FFmpeg, 動画, ビデオ, オーディオ, コーデック, フォーマット変換

1. ま え が き

近年、インターネットを介して配信される動画画像をPCや携帯電話等で視聴するスタイルが定着してきており、これらに関する研究開発も活発に行われている。しかしながら、世の中には実に多くの動画画像フォーマットがあり、OS間のみならずアプリケーション間でも互換性がないことも多く、動画画像を利用する研究者にとっては頭の痛いところである。映像配信サービスにおいても、コンテンツ提供側、コンテンツ視聴側の両方において、動画画像のファイル形式や解像度(表示サイズ)、ビットレートの変換等へのニーズが高まっている。このような、多種多様な動画画像フォーマットへの対応を解決するツールの一つがFFmpegである。

FFmpegは、ビデオデータやオーディオデータを保存・変換・再生できるツールであり、ビデオ/オーディオコーデックをプログラムから利用するためのライブラリーを含んでいる。もう少し詳しく書くと、FFmpegは、多くのメンテナによって開発が進められているオープンソースソフトウェアであり、動画画像ファイル/ビデオファイル/オーディオファイル/静止画像ファイルのフォーマットを変換するコマンドラインツールであるffmpeg、ライブ放送向けのマルチメディアストリーミングサーバであるffserver、メディアプレーヤffplay、プログラミング用ライブラリーであるlibavutil、libavcodec、libavformat、libavdevice、libswscale等から構成されるツール/ライブラリー群の総称である。

動画画像のファイルフォーマット(コンテナという用語もあるが、ファイルフォーマットとコンテナとは同じ概念とみてよいだろう)には多くの種類があり、ほぼ、対応する拡張子で表される。この動画画像をエンコードあるいはデコードするには、ビデオデータのコーデック、および、オーディオデータのコーデックの両方が必要となる(コーデックはビデオデータやオーディオデータの符号化の方式、あるいはその方式を実装したプログラムを表す用語である。

通常、あるコーデックでエンコードしたデータをデコードするには、そのコーデックを必要とする)。ファイルフォーマットとコーデックは1:1の対応をしているわけではなく、同じ拡張子を持つ動画画像ファイルであっても、片方は「Aというビデオコーデックとaというオーディオコーデック」で符号化され、もう片方は「Bというビデオコーデックとbというオーディオコーデック」で符号化されている、ということもあり得る。

よって、より多くの種類の動画画像ファイルをエンコード/デコードするには、それに対応したより多くのコーデックを入手するか、あるいは、自身の環境が持つコーデックで利用できるように形式に変換することが必要となってくる。FFmpegは、多くのファイルフォーマットやコーデックに対応しており、再生や保存、相互の変換が可能である。

FFmpegがサポートするファイルフォーマット、およびコーデックは公式サイト¹⁾の「Documentation」にある「General Documentation」に記載されているが、主なファイルフォーマットだけでもASF, AVI, SWF, FLV, QuickTime, MPEG-1, MPEG-2, MPEG-4, 3GPP, WMV, WMA, MP3, WAV, TwinVQなどをサポートしており、静止画像もJPEG, YUV, PNG, GIF, TIFF, BMPなどをサポートしている。ただし、エンコード/デコードの両方をサポートしているものもあれば、デコードのみ、というものもあるので注意が必要である。

このように広範囲なファイル形式をサポートしていることや、無償で配布されていることもあり(ライセンスについては後述する)、多くのアプリケーションやインターネットを介した映像配信サービスにおいて利用されている。また、冒頭で述べた通り、コンテンツを視聴する側においても、ファイル形式の変換等への要望が多い。それゆえ、これらのニーズを満たすFFmpegへの関心は研究開発者以外にも高まっており、公式サイト¹⁾をはじめとしてインターネット上に多くの情報がある。また、FFmpegを本格的に解説した、日本語で書かれた書籍として²⁾がある。

2. 入 手 方 法

FFmpegはソース配布の形をとっており、ソースコードを入手して自力でコンパイルするか、OSに対応したパツ

† NTTサイバースリソリューション研究所
"FFmpeg" by Ken Tsutsuguchi (NTT Cyber Solutions Laboratory, Yokosuka)

パッケージ管理ツール(Linux環境におけるyumやapt-getなど、MacOS XにおけるMacPortsなど)を用いるか、あるいは、それぞれのプラットフォームにあった実行可能なプログラムを入手する(FFmpegを用いたアプリケーションは、公式サイト「About」→「Projects Using FFmpeg」に記載されている)。

自力でコンパイルする場合は、公式サイト「Download」にあるソースコードをダウンロードし(SVNスナップショットを入手することもできる)、展開後、必要なオプションをつけてコンパイルする。

オプションを知るには、展開したディレクトリに移動し、

```
% ./configure --help
```

を実行する。デフォルトの設定でよければ、以下を実行する。

```
% ./configure
% make
% make install
```

オプションのコーデックについては、あらかじめ事前にコンパイル・インストールしておく。Windows環境では、MinGWまたはCygwinを用いてコンパイルすることができる。詳細は公式サイト「Documentation」にある「General Documentation」を参照のこと。

3. 使用例

3.1 コマンドラインツールとして

最も多い使い方は、実際に動画画像フォーマットを変換することであろう。以下、いくつかの使用例を紹介する。これらは、公式サイト¹⁾の「Documentation」にある「Frequently Asked Questions」や、「Command Line Interface (CLI) and Related Usage Documentation」の「FFmpeg Documentation」に記載されている。

コマンドラインのオプションを調べる

```
% ffmpeg -h
```

サポートしているフォーマットを調べる

```
% ffmpeg -formats
```

複数の静止画像を動画画像に変換する

まず、静止画像を連番をつけたファイル名に変更する。例えば、img.jpg, img.jpg, img3.jpg, ... といった具合である。そして次のコマンドを実行する：

```
ffmpeg -f image2 -i img%d.jpg /tmp/a.mpg
```

"%d"の部分は、ファイル名の番号の部分として解釈される。また、img001.jpg, img002.jpg, ... のようなファイル名の場合は、img%03d.jpgのように指定する。ffmpegが読み込む任意の静止画像フォーマットに対し、同じよう

な使い方ができる。

動画画像を静止画像に変換する

```
% ffmpeg -i movie.mpg movie%d.jpg
```

入力ファイルmovie.mpgは、連番静止画像movie1.jpg, movie2.jpg, ... に変換される。指定した拡張子でファイル形式を解釈するが、静止画像フォーマットを陽に指定することもできる：

```
-vcodec ppm
-vcodec png
-vcodec mjpeg
```

さきほどの例に適用すると：

```
% ffmpeg -i movie.mpg -f image2 -vcodec mjpeg
menu%d.jpg
```

単純なコピー

```
ffmpeg -i input.wmv -vcodec copy -acodec copy
output.wmv
```

copyオプションを使えば、映像だけ、もしくは、音声だけのコーデックを変換したいときに無駄な再エンコードをしなくてすむため、高速に実行できる。

先頭から6秒の位置から長さ3秒だけ切出して出力する

```
% ffmpeg -i input.wmv -vcodec wmv2 -acodec
wmav2 -sameq -ss 6 -t 3 output.wmv
```

あるフォーマットの動画画像を他のフォーマットに変換する

```
% ffmpeg -i input.mpg -sameq output.avi
```

ビデオのビットレートを64kbpsに変更する

```
% ffmpeg -i input.avi -b 64k output.avi
```

フレームレートを24fpsに変更する

```
% ffmpeg -i input.avi -r 24 output.avi
```

動画画像や静止画像を表示/再生する

FFmpegにはSDLとFFmpegに含まれるライブラリーをベースとするメディアプレーヤーffplayが含まれている：

```
% ffplay filename
```

3.2 開発ツールとして

FFmpegにはコマンドラインツールのみならず、C言語用のライブラリーが含まれている。いくつかの種類のライブラリーが用意されており、中にはデフォルト設定ではビルドされないものや、ライセンス形態が異なるものもあるので、コンパイル/インストールの際には注意が必要である。特に利用されると思われるものが、エンコード/デコードを行うコーデックを含むlibavcodecと、ファイルフ



フォーマットを扱うlibavformatであろう。

これらの使い方については、ダウンロードしたFFmpegのソースコードを展開したディレクトリに、ffmpeg.cやffplay.c、output_example.cといったCプログラムそのものが置かれており、参考になる。特に有用なサンプルプログラムとして、libavcodecはlibavcodec/apixample.c、libavformatはffplay.cやoutput_example.cを眺めてみるとよいだろう。

詳細は公式サイト「Documentation」にある「Developer Documentation」や「Public API Documentation」を参照のこと。

4. ライセンス

FFmpegのライセンス形態はLGPL (GNU Lesser General Public License) となっている。しかし、FFmpegが含むいくつかの付加的な部分や最適化部分において、GPL (GNU General Public License) のライセンス形態をとっているものもある。これらのGPLを採用している部分については、利用しないようにコンパイル時のオプションを指定することもできる。GPLを採用している部分を使うのであれば、FFmpeg全体にGPLが適用されることになるので、FFmpegを組み込んだアプリケーション等を配布する際には注意が必要である。また、ビデオデータやオーディオデータの符号化/圧縮方式は、日々、技術革新が進められている分野であるが、公式サイトでは、FFmpegは第三者が特許化したアルゴリズムを用いている可能性が

ないとは断言できないとしており、商用ソフトに組み込むことについては警告を発している。詳細は公式サイト「About」にある「License and Legal Issues」、あるいはソースコードに同梱して配布されるドキュメントを参照されたい。

5. む す び

以上、FFmpegについて簡単ではあるが解説を行った。動画像を利用する研究者や動画像を簡単に変換したい利用者にとって、FFmpegは非常に便利で有用なツールとなり得る。FFmpegを開発し続けているメンテナの方々には頭が下がる思いである。FFmpegに関するさらに詳しい情報は、本家サイトをはじめとして多くのリソースから得ることができる。本稿がこのツールの有効な活用や、読者の研究開発の一助となれば幸いである。(2009年11月30日受付)

〔文 献〕

- 1) FFmpeg, <http://ffmpeg.org/> (2009年11月現在)
- 2) 原一浩ほか：「FFmpegで作る動画共有サイト」、毎日コミュニケーションズ (2008)



筒口 拳 1989年、京都大学理学部卒業。1991年、同大学院工学研究科修士課程修了。同年、日本電信電話(株)(現NTT)ヒューマンインタフェース研究所に入社。コンピュータグラフィックス/アニメーション、画像処理等の研究に従事。現在、NTTサイバーソリューション研究所主任研究員。博士(情報学)。