

ターミナルからファイルを開く

著者：関勝寿 公開日：2015 年 10 月 27 日 ソース：<https://sekika.github.io/2015/10/27/open-command/>

端末エミュレータ から指定したファイルを開連付けられているアプリケーション（あるいは指定したアプリケーション）で開く方法について、Mac, Linux, Windows それぞれの方法をメモ。

1. Mac の場合

ターミナルで open コマンドを使う。

open Readme.txt

とすると、Readme.txt に開連付けられているアプリケーションで Readme.txt が開かれる。

open <http://google.com>

とすれば、ブラウザで Google を開くことができる。また、ディレクトリを指定すると

open .

とすれば、Finder でカレントディレクトリが開かれる。

開連付けられているアプリケーションとは異なるアプリケーションを使いたいことがある。たとえば、html ファイルをエディタで編集したい時など。-t オプションを使うと、デフォルトのテキストエディタで開くことができる。

open -t filename

また、open コマンドの -a オプションを使って -a アプリケーションで、起動するアプリケーションを指定できる。

CotEditor で指定したファイルを開くコマンド cot を作るには、

```
#!/bin/sh
open $1 -a /Applications/CotEditor.app
```

といったシェルスクリプトを作っておき、実行可能にして cot コマンドとしてパスが通る場所に置いておく。

cot filename

として、使うことができる。

open --help で次のヘルプが表示される。

他にも、Finder と Terminal の連携をするための技がこの記事に解説されている。

- ・ [Finder と Terminal の連携を考える](#)

2. Linux の場合

xdg-open または gnome-open コマンドを使う。

- ・ [Linux でコマンドラインから URL や ファイルを開ける xdg-open/gnome-open のメモ](#)
- ・ [xdg-open\(1\) - Linux man page](#)

3. Windows の場合

コマンドプロンプトから start コマンドを使う。

- ・ [意外に知らない start コマンドの使い方](#)
- ・ [別ウィンドウでコマンドを実行 \(START\)](#)

以下は、Windows 10 で start /? として表示されるヘルプ。

指定されたプログラムまたはコマンドを実行するためにウィンドウを開きます。

```
START ["タイトル"] [/D パス] [/I] [/MIN] [/MAX] [/SEPARATE|/SHARED]
[/LOW|/NORMAL|/HIGH|/REALTIME|/ABOVENORMAL|/BELOWNORMAL]
[/NODE <NUMA ノード>] [/AFFINITY <16 進数の関係マスク>] [/WAIT]
[/B]
[ コマンド / プログラム ] [ パラメーター ]
```

"タイトル" ウィンドウのタイトル バーに表示するタイトル。

パス 開始するディレクトリ。

B 新しいウィンドウを作成せずにアプリケーションを起動します。アプリケーションは Ctrl + C を無視します。アプリケーションで Ctrl + C を有効にしていない場合、Ctrl + Break がアプリケーションを中断する唯一の方法です。

I 新しい環境は、現在の環境ではなく、cmd.exe に渡された元の環境になります。

MIN ウィンドウを最小化の状態で起動します。

MAX ウィンドウを最大表示の状態で起動します。

SEPARATE 16 ビットの Windows プログラムを別メモリ領域で起動します。

SHARED 16 ビットの Windows プログラムを共有メモリ領域で起動します。

LOW IDLE 優先度クラスでアプリケーションを起動します。

NORMAL NORMAL 優先度クラスでアプリケーションを起動します。

HIGH HIGH 優先度クラスでアプリケーションを起動します。

REALTIME REALTIME 優先度クラスでアプリケーションを起動します。

ABOVENORMAL ABOVENORMAL 優先度クラスでアプリケーションを起動します。

BELOWNORMAL BELOWNORMAL 優先度クラスでアプリケーションを起動します。

NODE 優先 NUMA (Non-Uniform Memory Architecture) ノードを 10 進の整数で指定します。

AFFINITY プロセッサの関係マスクを 16 進数で指定します。

プロセスはこれらのプロセッサで実行されるように制限されます。

/AFFINITY と /NODE を組み合わせると、関係マスクは異なって解釈されます。NUMA ノードのプロセッサ マスクを右にシフトしてビット 0 で始まるかのように関係マスクを指定します。

プロセスは、指定した関係マスクと NUMA ノードの間で共通するプロセッサ上で実行されるように制限されます。共通するプロセッサがない場合は、プロセスは指定した NUMA ノード上で実行されるように制限されます。

WAIT アプリケーションを起動し、終了するまで待ちます。

コマンド / プログラム

内部コマンドまたはバッチ ファイルの場合、コマンド プロセッサは cmd.exe の /K オプションを使用して実行されます。これはコマンドの後でもウィンドウが残ることを意味します。

内部コマンドまたはバッチ ファイルではない場合、そのプログラムはウィンドウ モードのアプリケーションまたはコンソールアプリケーションとして動作します。

パラメーター

コマンド / プログラムに渡すパラメーターです。

注意：SEPARATE および SHARED オプションは 64 ビット プラットフォームではサポートされません。

/NODE を指定すると、NUMA システム上のメモリ局所性を利用する方法でプロセスが作成されるようになります。たとえば、共有メモリ経由で互いに頻繁に通信する 2 つのプロセスを、メモリ待ち時間を最小限に抑えるために同じ優先 NUMA ノードを共有するように作成できます。これらのプロセスは可能であれば同じ NUMA ノードからメモリを割り当て、指定したノード外のプロセッサ上で実行されることもあります。

```
start /NODE 1 application1.exe
```

```
start /NODE 1 application2.exe
```

これら 2 つのプロセスは、さらに、同じ NUMA ノード内の特定のプロセッサ上で実行されるように制限できます。次の例では、application1 がノードの低順位の 2 つのプロセッサ上で実行されるのに対し、application2 はノードの次の 2 つのプロセッサ上で実行されます。この例では、指定したノードに少なくとも 4 つの

論理プロセッサがあることを想定しています。ノード番号は、関係マスクを変更しなくとも、そのコンピューターの任意の有効なノード番号に変更できることに注意してください。

```
start /NODE 1 /AFFINITY 0x3 application1.exe  
start /NODE 1 /AFFINITY 0xc application2.exe
```

コマンド拡張機能を有効にすると、コマンドラインまたは START コマンドによる外部コマンドの起動は、次のように変更されます：

非実行可能ファイルは、ファイル名をコマンドとして入力することによって、ファイルの関連付けを使って開くことができます (例：WORD.DOC は .DOC ファイル拡張子に関連付けられているアプリケーションを起動します)。コマンド スクリプト内でファイルの関連付けを作成する方法については、ASSOC と FTYPE コマンドを参照してください。

32 ビット GUI アプリケーションを実行する場合、CMD.EXE は、アプリケーションの終了を待たずにコマンド プロンプトに戻ります。コマンド スクリプト内で実行する場合は、この動作は発生しません。

最初のトークンが拡張子やパス修飾子を持たない文字列 "CMD" であるコマンドラインを実行する場合、"CMD" が COMSPEC 変数の値で置き換えられます。これにより現在のディレクトリの CMD.EXE が使われないようにします。

最初のトークンが拡張子を含まないコマンドラインを実行する場合、CMD.EXE は、PATHEXT 環境変数の値を使って拡張子の種類と順序を判断します。PATHEXT 変数の既定値は、次のとおりです：

```
.COM;.EXE;.BAT;.CMD
```

この構文は PATH 変数と同じであり、各要素はセミコロンで区切られていることに注意してください。

実行可能なファイルを検索するときにどの拡張子でも一致するファイルが見つからない場合は、拡張子なしの名前がディレクトリ名と一致するかどうかを調べます。

一致する場合は、START コマンドがそのパスでエクスプローラーを起動します。コマンドラインから実行した場合は、そのパスに対する CD /D の実行と同じになります。