

ソースからソフトをインストールする

- [ソースからソフトをインストールする](#)
 - [ソースからソフトをインストールしたい](#)
 - [はじめに](#)
 - [適当な場所でアーカイブを展開する](#)
 - [アーカイブの中の README, INSTALL などの文書をよく読み構成設定をする。\(configure\)](#)
 - [ビルド \(make\)](#)
 - [インストール \(make install\)](#)
 - [こちらどうぞ。](#)
 - [共有ライブラリが存在するのにライブラリが見つからないとエラーが出る \(ld.so.conf\)](#)
 - [パッチってどうやってあてるの? \(patch\)](#)
 - [ソースからインストールしたソフトをアンインストールしたい。](#)
 - [インストールを監視する](#)
 - [make で実行される処理を確認してみる](#)
 - [make uninstall](#)
 - [ソフト別のディレクトリを作っておく](#)
 - [rpm や deb に変換](#)
 - [ソースからインストールしたソフトをバージョンアップしたい。](#)

ソースからソフトをインストールしたい

はじめに

ローカル環境向けディレクトリ `/usr/local` にインストールすれば、全てのユーザーがソフトウェアを使用できます。

- `/usr` に入れると RPM みたいなパッケージを展開して入るファイルとごっちゃになるのでオススメしない。
- `/usr/[アプリケーション名]` みたいに新しいディレクトリを作ってもOK。作ったら `PATH` に入れておこうね。

または、ユーザのホームディレクトリなどを使いましょう。

通常は、下の項目の手順で実行できます。が、異なる場合もあります。

- `autoconf`を使わず独自のMakefileを使っている場合など。
- ビルド中に他のライブラリのヘッダ、`pkg-config`(依存関係調査ツール)などのツールを必要とするものもある。

事前にソフト付属のドキュメント (README, INSTALL など) やオンラインドキュメントを見て実行しましょう。

適当な場所でアーカイブを展開する

ホームディレクトリとかでアーカイブを展開。

- 拡張子が `.tar.gz` または `.tgz` の場合

- tar zxvf [圧縮されたファイル]
- 拡張子が .tar.bz2 または .tbz の場合

tar jxvf [圧縮されたファイル]

アーカイブの中の README, INSTALL などの文書をよく読み構成設定をする, (configure)

configure スクリプトがある場合は,

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

と入力してオプションを確認して、オプションを設定する。

```
./configure [オプション]
```

/usr/local にインストールするなら --prefix=/usr/local、ホームディレクトリなら --prefix=\$HOME という感じでオプションをつける。

システムを一通り調査したあとでMakefileができます。

Makefile のみの場合は、Makefile の中を確認して必要なら編集する。

ビルド (make)

ビルドは README などでは指示がなければ、make と入力するだけです。ソフトの規模(とマシンの速度)によって数分～数時間と時間にバラつきがあるので気長に待ちましょう。

ビルドが終了するとシェルに戻ります。シェルに戻っていても、直前に ***** Error ***** など表示されていれば、途中でエラーが出て中断しているので、README を読み直したり、ソースや Makefile などを見て原因を探ってみてください。

インストール (make install)

/usr/local に保存するなら root で作業する必要があるので、su で root になって make install でインストール。

- chown hoge /usr/local で /usr/local ごとオーナーを変更してしまうという手もあり。

ホームディレクトリならそのまま make install と入力します。

こちらもどうぞ。

- [ソフトウェアのコンパイル&インストール](#)

「ソースファイルの取得」「Archie で検索」のここはちょっと古いけど。

- [Linux From Scratch](#)

ソフトウェアのコンパイル事例が載ってて参考になるよ。

- configureをデバッグする

<http://www.ibm.com/developerworks/jp/linux/library/l-debcon/>

共有ライブラリが存在するのにライブラリが見つからないとエラーが出る (ld.so.conf)

以下のようなエラーが出る場合。

```
hoge: error while loading shared libraries: libfuga.so.0
: cannot open shared object file: No such file or directory
```

依存関係や設定問題を自動で解決できるパッケージシステムではこういった問題はあまり発生しませんが、[ソースからコンパイル](#)してライブラリを入れた場合に起きることがたまにあります。

/etc/ld.so.conf にライブラリの存在するディレクトリ (ソースから入れた場合は /usr/local/lib) が記述されていますか? Red Hat系のディストリビューションでは記述がない場合があります。

記述がない場合は追加して、root で ldconfig と入力しましょう。ライブラリのキャッシュが再構築されて /usr/local/lib にある共有ライブラリも見つけられるようになります。

まあ、以下の内容が書いてあれば大抵の場合は大丈夫でしょう。

```
/lib
/usr/lib
/usr/X11R6/lib
/usr/local/lib
```

- [共有ライブラリをシステムに認識させるには \(@IT\)](#)
- [Manpage of ldconfig](#)

パッチってどうやってあてるの? (patch)

ソースを展開したディレクトリで patch -p1 < パッチのファイル名 と入力。これでだいたいok、gzipやbzip2で圧縮されていたら

```
zcat パッチ | patch -p1 #拡張子*.gz
bzip2 パッチ | patch -p1 #拡張子*.bz
```

で、たぶん大丈夫。

- [Patch でバージョンアップ](#)

ソースからインストールしたソフトをアンインストールしたい。

ソースから入れたソフトはインストールしたファイルを一つ一つ手で消すのがキホソ。がんがね。;)

楽するための Tips

インストールを監視する

[paco](#) を使うと、make install でインストールしたソフトを削除できます。

参考: [paco——インストールの監視](#)

参考: [paco](#)

make で実行される処理を確認してみる

make に -n オプションをつけて make -n install と入力するとインストールで何が実行されるかわかる。

make uninstall

ソースの Makefile に uninstall という行があれば、make uninstall が使えるかもしれない。

[ITmedia の解説](#) を参考に Makefile を読んでみてください。

ソフト別のディレクトリを作っておく

./configure のときに --prefix=/usr/local/ソフト名 と指定して、ソフト毎のディレクトリを作っておく。ディレクトリごと消してアンインストール。

/opt/[ソフト名] とかでもよし。

rpm や deb に変換

rpm / deb アーカイブ化し、パッケージシステムを使ってインストールする。

[RPMファイルの作り方](#) など参照。

簡単にパッケージ化してくれる [CheckInstall](#) というのもあります。

ソースからインストールしたソフトをバージョンアップしたい。

やり方は場合による。

そのまま新バージョンを上書きインストールすれば問題なく動作する場合も多いが、旧バージョンと新バージョンの変更点が大きい場合は旧バージョンのアンインストールが必要かもしれない。