

gnupack/cygwin

Windows 上で gnupack(cygwin) の gcc を使う方法。
MinGW ではなく、普通に gcc を使う。

メリット

- ・ 手軽
- ・ とりあえず、すぐ動かせる

デメリット

- ・ コンパイル後の exe を実行するには、cygwin.dll が必要になる

コンパイル例

```
gcc test.c
```

MinGW

gnupack で MinGW を使う

gnupack で MinGW を使う方法。
環境を作ってしまったら、結構使える。

メリット

- ・ exe 単体で実行可能。(cygwin.dll 等が不要)
- ・ コンパイラだけでなく、環境として整備しやすい

デメリット

- ・ 環境を整えるのがちょっと手間

コンパイル例

```
i686-w64-mingw32-gcc test.c
```

Msys2 で MinGW を使う

Msys2 に MinGW を使う方法。
gnupack とほぼ同じ。

Linux で MinGW を使う (クロスコンパイル)

MinGW は Linux で使えるので、Linux 上でクロスコンパイルする方法。
実行確認は wine を使うか、windows に exe をコピーして実行する。

メリット

- ・ 普段 Linux で開発している場合は、そのままの環境が使える。

デメリット

- ・ Windows での確認の際に exe をコピーする必要がある (wine 使ってもいいけど)
- ・ 環境を整えるのがちょっと面倒

コンパイル例

i686-w64-mingw32-gcc test.c

知っておいたほうがいいコンパイルオプション

オプション	意味
-s	strip。オブジェクトファイルのシンボルを削除することによってもファイルサイズを小さくする
-static	実行時に dll がありません。とか出たら static にして実行バイナリにライブラリを組み込む。
-mwindows	コンソールアプリではなく、ウインドウを使うようなアプリをコンパイルするときに必要