

## 参考

<https://www.yokoweb.net/2016/08/29/msys2-install/>

<https://yujiorama.hatenablog.com/entry/2019/07/22/095214>

<https://udon.little-pear.net/msys2-pacman-gpg-update/>

<https://blog.bachi.net/?p=5035>

## 概要

MSYS2 の Portable 環境を作る。

普通にインストールしてパスを変更するだけで良さそうな気もするけど。

## MSYS2 Portable を使う

### MSYS2 Portable

ちょっと古いけど、pacman でアップデートすれば特に問題ない。

```
.mintyrc
```

```
FontHeight=12
Locale=C
Charset=UTF-8
Blue = #7D7DFF
Term=xterm-256color
# FontSmoothing=partial
```

### アップデート

```
pacman -Suy
```

### アップデートで PGP 鍵エラーが出る場合

```
wget http://repo.msys2.org/msys/x86_64/msys2-keyring-r21.b39fb11-1-any.pkg.tar.xz
wget http://repo.msys2.org/msys/x86_64/msys2-keyring-r21.b39fb11-1-any.pkg.tar.xz.sig
pacman-key --verify msys2-keyring-r21.b39fb11-1-any.pkg.tar.xz.sig
pacman -U --config <(echo) msys2-keyring-r21.b39fb11-1-any.pkg.tar.xz
```

### pacman( パッケージマージャ ) の使い方

```
pacman -h でヘルプ。
pacman -S -h で -S オプションのヘルプ。
pacman -Sy でパッケージリストのアップデート
pacman -Su でパッケージのアップデート ( 事前に update-core が必要 )。
pacman -Ss [pattern] でパッケージを検索。
pacman -Sl | grep [pattern] でもだいたい分かる。
pacman -S [package] でインストール。
pacman -Rs [package] でアンインストール。
pacman -Sg でパッケージグループの一覧を表示。
pacman -Qqe でインストール済みのパッケージ一覧を表示。
pacman -Si [package] で詳しい情報を表示。
pacman -Sc で使っていないキャッシュを削除。
```

### 色々インストール

```
pacman -S git
pacman -S tmux
pacman -S mingw32/mingw-w64-i686-gcc
pacman -S mingw64/mingw-w64-x86_64-gcc
```

### nkf インストール

```
pacman -S make
wget https://osdn.jp/projects/nkf/downloads/64158/nkf-2.1.4.tar.gz
tar zxvf nkf-2.1.4.tar.gz
cd nkf-2.1.4
make install
```

## .bashrc 変更

```
export PATH=${PATH}:/mingw64/bin:/mingw32/bin
# git
GIT_PS1_SHOWDIRTYSTATE=true
GIT_PS1_SHOWUNTRACKEDFILES=true
GIT_PS1_SHOWSTASHSTATE=true
GIT_PS1_SHOWUPSTREAM=auto
source /usr/share/git/completion/git-prompt.sh
export PS1='%[%033]0;$TITLEPREFIX:$PWD%007%`%n%[%033[32m%]u@%h %[%033[35m%]$MSYSTEM %[%033[33m%]w%[%033[36m%]`__git_ps1`%[%033[0m%]n$ '
```

## Git for Windows Portable を使う

Git for Windows Portable は Msys2 を Git 用にカスタマイズしたものだ。

基本的に git を動作させる以外に不要なものは削除されている。

そのため、通常の Msys のように利用するためには、様々なパッケージを再インストールする必要がある。

git-bash.exe を起動すると、mingw64.exe の起動と同様に mingw64/bin にパスが通った状態で起動する。

Git for Windows の git は、Windows のパスを普通に使えるようにするため、MinGW でコンパイルされた git を利用しているため

```
mingw64\bin\git.exe
```

であり、/usr/bin/git.exe ではないので注意。

## HOME 変更

以下のようなバッチで HOME ディレクトリを変える

```
@echo off
set HOME=% dp0\home\portable
start "" git-bash.exe
```

## pacman を使えるようにする

```
git clone --depth=1 https://github.com/git-for-windows/git-sdk-64 gfw-sdk
cp gfw-sdk/usr/bin/pacman* /usr/bin/
cp -a gfw-sdk/etc/pacman.* /etc/
mkdir -p /var/lib/
cp -a gfw-sdk/var/lib/pacman /var/lib/
cp -a gfw-sdk/usr/share/makepkg/util* /usr/share/makepkg/
pacman --database --check
```

```
curl -L https://raw.githubusercontent.com/git-for-windows/build-extra/master/git-for-windows-keyring/git-for-windows.gpg % | pa
```

## 色々インストール

git for windows は軽量化のため余計なファイルが削除されている。

pacman で必要なパッケージを再インストールしておく

## コンパイラとかヘッダファイルとか

```
pacman -S gcc mingw-w64-i686-gcc mingw-w64-x86_64-gcc
pacman -S msys2-w32api-headers mingw-w64-i686-headers-git mingw-w64-x86_64-headers-git
pacman -S binutils mingw-w64-i686-binutils mingw-w64-x86_64-binutils
pacman -S mingw-w64-i686-crt-git mingw-w64-x86_64-crt-git
pacman -S mingw-w64-i686-winpthreads-git mingw-w64-x86_64-winpthreads-git
pacman -S windows-default-manifest mingw-w64-i686-windows-default-manifest mingw-w64-x86_64-
-windows-default-manifest
pacman -S msys2-runtime-devel msys2-w32api-runtime
```

## wget

```
pacman -S libpcre2_8 libmetalink libgpgme wget
```

## Msys2 起動用の exe 復活

```
pacman -S msys2-launcher
```

## ツール実行でエラーになったら

```
strace
```

を使ったり、[Linux でファイルアクセスを監視する](#)を参考に関連ライブラリを調べる

## peco を使う

msys2 では、[peco](#) ではなく [percol](#) を使う。

使い方もオプションも同じなので、percol で問題ない。

## peco を使う 非推奨

<https://github.com/peco/peco/releases>

ダウンロードして適当なパスに配置 or ソースを落としてコンパイル、インストール

winpty で実行する必要があるためちょっと細工が必要。

パイプで連携せずに一時ファイルで連携する。

### .bashrc

```
peco-select-history() {
    HISTTIMEFORMAT= history | sort -k1,1nr | perl -ne 'BEGIN { my @lines = (); } s/^\s*\d+\s*//;
    $in=$_; if (!(grep {$in eq $_} @lines)) { push(@lines, $in); print $in; } > /tmp/peco-select-history
    winpty bash -c "${HOME}/bin/peco-select-history.sh"
    READLINE_LINE=$(cat /tmp/peco-select-history_2)
    READLINE_POINT=$(cat /tmp/peco-select-history_3)
}
bind -x '"%C-r": peco-select-history'
```

### ~/bin/peco-select-history.sh

```
declare l=$(peco --query "$READLINE_LINE" /tmp/peco-select-history)
echo $l > /tmp/peco-select-history_2
echo ${#l} > /tmp/peco-select-history_3
```