

# Vim が勝手に作るファイルとその働き

*KusaReMKN*

## 概要

ここでは Vim が勝手に作るファイルであるスワップファイル、バックアップファイル、アンドゥファイルについて簡単に説明する。また、それらのファイルに関する設定も紹介する。

### 1. スワップファイル

このファイルには現在編集しているファイルの変更された箇所の情報が記録されている。このファイルは Vim が編集を終了すると即座に削除される。しかし、Vim が正常に終了しなかった場合 (例えば、コンピュータがクラッシュしたり、Vim の実行中に回線が切断されたりなどがこれにあたる) このファイルは削除されずに取り残される。

編集中のファイルが `.foo.txt` という名前だった場合、デフォルトのスワップファイルの名前は `./foo.txt.swp` となる。`./foo.txt.swp` が既に存在している場合は `./foo.txt.swo` や `./foo.txt.swn` のように末尾の文字から順に若くなる。若くなりすぎて `./foo.txt.saa` まで存在する場合はスワップファイルの生成を諦める。

スワップファイルの生成場所はオプション `directory` で設定できる。例えば、スワップファイルを `$HOME/temp` に生成させたい場合は `$HOME/.vimrc` に次のような行を加えれば良い。

```
set directory=$HOME/temp      " スワップファイルは $HOME/temp に生成
se dir=$HOME/temp             " 上の短縮形
```

スワップファイルの生成を抑制することもできる。`$HOME/.vimrc` に次のような行を加えればスワップファイルは生成されなくなる。この設定は、例えば、ディスクの上で作業をしている場合やディスクの容量が極端に限られている場合には有用かもしれない。

```
set noswapfile                 " スワップファイルを生成しない
se noswf                       " 上の短縮形
```

反対にスワップファイルを生成したい場合は `$HOME/.vimrc` に次のような行を加えれば良い。

```
set swapfile                   " スワップファイルを生成
se swf                         " 上の短縮形
```

より詳細な情報を必要とする場合は Vim で次のように実行すると良いだろう。

```
:help swap-file
:help 11.3
:help :rec
:help 'directory'
:help 'swapfile'
```

## 2. バックアップファイル

このファイルにはオリジナルファイルのバックアップが記録されている。

編集中のファイルが `.foo.txt` という名前だった場合、デフォルトのバックアップファイルの名前は `.foo.txt~` となる。`.foo.txt~` が既に存在している場合は上書きされる。この設定はオプション `backupext` で設定できる。

バックアップファイルの生成場所はオプション `backupdir` で設定できる。例えば、バックアップファイルを `$HOME/temp` に生成させたい場合は `$HOME/.vimrc` に次のような行を加えれば良い。

```
set backupdir=$HOME/temp      " バックアップファイルは $HOME/temp に生成
se bdir=$HOME/temp           " 上の短縮形
```

デフォルトの設定ではオリジナルファイルの書き込み処理の前にバックアップファイルを生成し、書き込み処理の完了後にバックアップファイルを削除するようになっている。書き込み処理の完了後にもバックアップファイルを保持したい場合は `$HOME/.vimrc` に次のような行を加えれば良い。

```
set backup                    " バックアップファイルを保持
se bk                         " 上の短縮形
```

反対にバックアップファイルを保持したくない場合は `$HOME/.vimrc` に次のような行を加えれば良い。

```
set nobackup                  " バックアップファイルを保持しない
se nobk                       " 上の短縮形
```

オリジナルファイルに対する書き込み処理の最中にもバックアップファイルを生成したくない場合は `$HOME/.vimrc` に次のような行を加えれば良い。しかし、バックアップファイルの生成を無効にした状態でオリジナルファイルの書き込みに失敗した場合、Vim を終了してしまうとオリジナルファイルと編集内容の両方を失うことになる。

```
set nowritebackup             " 書き込みの前にバックアップを生成しない
set nowb                      " 上の短縮形
```

反対にバックアップファイルを生成したい場合は `$HOME/.vimrc` に次のような行を加えれば良い。

```
set writebackup               " 書き込みの前にバックアップを生成する
se wb                         " 上の短縮形
```

より詳細な情報を必要とする場合は Vim で次のように実行すると良いだろう。

```
:help backup
:help 'backupext'
:help 'backupdir'
:help 'backup'
:help 'writebackup'
```

### 3. アンドゥファイル

このファイルにはアンドゥの履歴が記録されている。アンドゥの履歴をファイルに保存することで、あとで Vim を起動したときにアンドゥの履歴を復元することができる。

編集時のファイルが `.foo.txt` という名前だった場合、デフォルトのアンドゥファイルの名前は `./foo.txt.un~` となる。`./foo.txt.un~` が既に存在する場合は上書きされる。

アンドゥファイルの生成場所はオプション `undodir` で設定できる。例えば、アンドゥファイルを `$HOME/temp` に生成させたい場合は `$HOME/.vimrc` に次のような行を加えれば良い。

```
set undodir=$HOME/temp      " アンドゥファイルは $HOME/temp に生成
se udir=$HOME/temp          " 上の短縮形
```

デフォルトの設定ではアンドゥファイルは生成されない。アンドゥファイルを生成したい場合は `$HOME/.vimrc` に次のような行を加えれば良い。

```
set undofile                 " アンドゥファイルを生成
se udf                       " 上の短縮形
```

反対にアンドゥファイルを生成したくない場合は `$HOME/.vimrc` に次のような行を加えれば良い。

```
set noundofile              " アンドゥファイルを生成しない
se noudf                    " 上の短縮形
```

より詳細な情報を必要とする場合は Vim で次のように実行すると良いだろう。

```
:help undo
:help 'undodir'
:help 'undofile'
```