

Autotest.mk

GNU Make を利用したテスト自動化ツール

j8takagi

Copyright © 2010 j8takagi

Table of Contents

1	Autotest.mk の概要	1
	要件.....	1
2	Autotest.mk のインストール	2
	2.1 ダウンロード.....	2
	2.2 ファイルの検証.....	2
	2.3 ファイルの展開.....	2
3	Autotest.mk チュートリアル	3
	3.1 サンプルプログラムの準備.....	3
	3.2 グループディレクトリーの作成.....	3
	3.3 テストディレクトリーの作成.....	3
	3.4 テスト用コマンドファイルの作成.....	3
	3.5 テスト想定結果の作成と検証.....	3
	3.6 テストの実行とログの確認.....	3
4	Autotest.mk マニュアル	4
	4.1 テンプレートディレクトリー.....	4
	4.1.1 <code>make</code> または <code>make group</code> グループの作成.....	4
	4.1.2 テンプレートのファイル.....	4
	4.2 グループディレクトリー.....	4
	4.2.1 <code>make create</code> テストの作成.....	4
	4.2.2 テスト名の変更.....	4
	4.2.3 テストの削除.....	4
	4.2.4 <code>make</code> または <code>make check</code> グループ内にあるすべてのテストの実行.....	4
	4.2.5 <code>make checkall</code> グループ内にあるすべてのテストの実行、計時.....	4
	4.2.6 <code>make clean</code> グループ内にあるすべてのテストで、実行結果をクリア.....	4
	4.2.7 <code>make time</code> グループ内にあるすべてのテストを計時.....	4
	4.2.8 <code>make cleantime</code> グループ内にあるすべてのテストで、計時結果をクリア.....	4
	4.2.9 グループのファイル.....	4
	4.3 テストディレクトリー.....	4
	4.3.1 <code>make set</code> テスト想定を作成.....	4
	4.3.2 <code>make set</code> テスト想定を再作成.....	4
	4.3.3 <code>make</code> または <code>make check</code> テストの実行.....	4
	4.3.4 <code>make time</code> テストの計時.....	4
	4.3.5 <code>make cleantime</code> テスト計時結果のクリア.....	4
	4.3.6 <code>make cleantime</code> テスト結果のクリア.....	4
	4.3.7 <code>make cleanall</code> テストの想定と結果をクリア.....	4
	4.3.8 テストのファイル.....	4

1 Autotest.mk の概要

Autotest.mk は、GNU Make を利用したテスト自動化ツールです。何回か `make` コマンドを実行し、プログラムのテスト結果を表す実行ファイルを作成すれば、テストの準備は完了です。あとは `make` コマンドを 1 回実行すれば、いつでもテストを実行できます。テストの実行結果は、ログに出力されます。

Autotest.mk では、次の 3 つの概念を使います。

‘テンプレート’

テストグループとテストのテンプレートです。

‘テストグループ’

複数のテストを含むグループです。

‘テスト’

テスト個々の内容と結果を表します。

要件

Autotest.mk は、Linux や Mac OS X など動作します。現在動作を確認しているのは、次のディストリビューションです。

- Ubuntu Linux 10.10
- Mac OS X バージョン 10.6.5

Mac OS X では、GNU Findutils (<http://www.gnu.org/software/findutils/>) をインストールしてください。

2 Autotest.mk のインストール

Autotest.mk をインストールするには、次の作業をします。

1. Autotest.mk ファイルのダウンロード
2. Autotest.mk ファイルの検証
3. Autotest.mk ファイルの展開

2.1 ダウンロード

GitHub/j8takagi/autotest_mk (http://github.com/j8takagi/autotest_mk) のダウンロードページ (http://github.com/j8takagi/autotest_mk/downloads) から最新版の tgz ファイルをダウンロードします。

続けて、対応する 'md5sum' ファイルをダウンロードします。

2.2 ファイルの検証

ダウンロードが完了したら、`md5sum` と `diff` で正しくファイルがダウンロードできているかを検証します。

```
$ md5sum autotest_mk.tar.gz | diff -s autotest_mk.tar.gz.md5sum -  
Files autotest_mk.tar.gz.md5sum and - are identical
```

2.3 ファイルの展開

'autotest_mk' ファイルをダウンロードしたら、次のコマンドで展開します。

```
$ tar xvzf autotest_mk.tar.gz
```

展開したら、Autotest.mk は使える状態です。多くのプログラムをインストールするときに必要なコンパイルやビルド、システムディレクトリーへのインストール (`./configure` や `make`、`make install`) は必要ありません。

3 Autotest.mk チュートリアル

Autotest.mk を使ったテストの手順は、次のとおりです。

1. グループディレクトリーの作成
2. テストディレクトリーの作成
3. テスト用コマンドファイルの作成
4. テスト想定結果の作成と検証
5. テストの実行とログの確認

ここでは、簡易電卓プログラム `calc` をテストする手順を取り上げます。

3.1 サンプルプログラムの準備

まず、Autotest.mk のパッケージに含まれている `'sample/calc'` を任意のフォルダーへコピーします。

```
$ cd ~/autotest.mk
$ cp -r ~/sample/calc ~/calc
```

3.2 グループディレクトリーの作成

テンプレートディレクトリーで、`make` を実行します。このとき、テストグループのディレクトリーを変数 `GROUPDIR` で指定します。

指定したディレクトリーが作成されます。

```
$ cd ~/autotest.mk/template
$ make GROUPDIR=~/calc/calc_test
```

次のコマンドでグループディレクトリーに移動し、その中を確認します。

テストを実行するための Makefile が作成されます。

```
$ cd ~/autotest.mk/template
$ make GROUPDIR=~/calc/calc_test
```

3.3 テストディレクトリーの作成

3.4 テスト用コマンドファイルの作成

3.5 テスト想定結果の作成と検証

3.6 テストの実行とログの確認

4 Autotest.mk マニュアル

4.1 テンプレートディレクトリー

テンプレートディレクトリーは、Autotest.mk をインストールするときに展開により作成された `template` ディレクトリーです。このディレクトリーに移動すると次の操作ができます。

4.1.1 `make` または `make group` グループの作成

変数 `GROUPDIR` で指定されたディレクトリーをテストグループディレクトリーとして作成し、必要な次のファイルをコピーします。

4.1.2 テンプレートのファイル

4.2 グループディレクトリー

4.2.1 `make create` テストの作成

4.2.2 テスト名の変更

4.2.3 テストの削除

4.2.4 `make` または `make check` グループ内にあるすべてのテストの実行

4.2.5 `make checkall` グループ内にあるすべてのテストの実行、計時

4.2.6 `make clean` グループ内にあるすべてのテストで、実行結果をクリア

4.2.7 `make time` グループ内にあるすべてのテストを計時

4.2.8 `make cleantime` グループ内にあるすべてのテストで、計時結果をクリア

4.2.9 グループのファイル

4.3 テストディレクトリー

4.3.1 `make set` テスト想定を作成

4.3.2 `make set` テスト想定を再作成

4.3.3 `make` または `make check` テストの実行

4.3.4 `make time` テストの計時

4.3.5 `make cleantime` テスト計時結果のクリア

4.3.6 `make cleantime` テスト結果のクリア

4.3.7 `make cleanall` テストの想定と結果をクリア

4.3.8 テストのファイル