

m4 の基礎 の基礎 v1

2012/07/28 オープンCAE勉強会@富山

中川慎二

m4 とは？

- 汎用マクロプロセッサ
- マクロプロセッサ（あるいはプリプロセッサ）は、テキスト置換ツールである。
- テキストを、指定されたとおりに、置換してくれる

- [http://ja.wikipedia.org/wiki/M4_\(プログラミング言語\)](http://ja.wikipedia.org/wiki/M4_(プログラミング言語))
- Super Technique 講座 m4 チュートリアル
 - <http://www.nurs.or.jp/~sug/soft/super/m4.htm>
- GNU macro processor
 - http://www.bookshelf.jp/texti/m4/m4-ja.html#SEC_Top

OpenFOAM例題での使われ方

m4 < constant/polyMesh/blockMeshDict_Pipe-4.m4 > constant/polyMesh/blockMeshDict

- constant/polyMeshの中に、拡張子.m4というファイルが置いてある.
- この中に、blockMeshDictの生成に必要な情報が書いてある
- m4を使って、blockMeshDictを生成する

m4を使うメリット

- 修正が容易になる
 - 半径, 長さなどを定義して, そこから節点の座標を計算することができる
 - 寸法の変更の際, 定義部分だけを変更すると, 関連する情報を一気に修正することができる

円柱モデル作成例

<http://mogura7.zenno.info/~et/wordpress/ocse/?p=220>

<http://mogura7.zenno.info/~et/xoops/modules/wordpress/index.php?p=408>

練習 1

- 元

```
define(name, WATASHI) dnl  
I am name.
```

- 結果

```
I am WATASHI.
```

練習2

- 元

```
define(dvert1, (dx$1 dy$1)) dnl
```

```
Call `dvert1' with 1
```

```
// ` ' causes no replacement
```

```
dvert1(1)
```

- 結果

```
Call dvert1 with 1
```

```
// causes no replacement
```

```
(dx1 dy1)
```


練習3

- 元

```
define(dvert1, (dx$1 dy$1)) dnl  
define(dx1, 1000) dnl  
define(dy1, 2000) dnl  
`dx1' is defined as 1000  
Calling `dvert1' with 1  
dvert1(1)
```

- 結果

```
dx1 is defined as 1000  
Calling dvert1 with 1  
(1000 2000)
```

esyscmd

UNIXコマンドの出力をm4に読み込ませたいときは esyscmdを使ってください。このマクロはシェルコマンド`shell-command`の標準出力の内容に展開されます。

- esyscmd(shell-command)

OpenFOAM例題でのesyscmd活用

- `define(calc, [esyscmd(perl -e 'print ($1)'))]`
- `define(ca0, calc(cos((pi/180)*a0))`
- calcの定義でesyscmdを使う。ここではperlを起動して、変数を表示させる。実際には、cosやsinというPerlの関数を使っているので、正弦や余弦を求めた結果の数値が得られる。
- perlの-eオプションは次に続く'..'をperlスクリプトとして実行することを意味する。

おまけ: perlの関数

数学関数

abs, atan2, cos, exp, hex, int, log, oct,
rand, sin, sqrt, srand

http://rfs.jp/sb/perl/05/function_list.html

引用符 (quote characters) を変更する

- デフォルトの引用符は組み込みマクロchangequoteによって変更できます。
- `changequote(opt start, opt end)`
- `start`は新しい開始引用符で`end`は新しい終了引用符です。もし欠けている引数があるときは、デフォルトの引用符(`と')がその欠けている引数の代わりに使用されます。
- `changequote`は展開されると消滅します。

OpenFOAM例題でのchangequote

- OpenFOAM例題では, changequoteを使って, 引用符を変更している。
 - 変更前: 開始`, 終了' ←これが標準
 - 変更後: 開始[, 終了]
- changequote([,])

コメントデリミタ (comment delimiters) を変更する

- デフォルトのコメントデリミタ (comment delimiters) は組み込みマクロ `changecom` で変更できます。
- `changecom(opt start, opt end)`
- `start` が新しいコメント開始デリミタ (start-comment delimiter) で、`end` が新しいコメント終了デリミタ (end-comment delimiter) です。欠けている引数があるときは、デフォルトのコメント区切り記号 (`#` と改行文字) がその欠けている引数の代わりに使用されます。コメント区切り記号は任意の長さにすることができます。
- `changecom` は展開されると消滅します。

OpenFOAM例題でのchangeom

- OpenFOAM例題では, changeomを使って, コメント デリミタを変更している。
 - 変更前: 開始#, 終了↓(改行) ←これが標準
 - 変更後: 開始//, 終了↓(改行) ←開始のみ変更
- changeom(//)

入力中の空白 (whitespace) を削除する

- 組み込みマクロ `dnl` は、最初の改行文字までにある文字をすべて読み込んでから改行文字も含めてそれらを捨てる。
- `define` の呼び出しの後に続く改行を取り除くために、`define` と一緒によく使われる。
- OpenFOAM 例題で、`m4` で再生した `blockMeshDict` に多くの空行が含まれるのを防ぐのに使える。
- `define` 文の文末に `dnl` と追記すればよい。`dnl` から行末まではコメントと同じ扱いになる。
 - `define(dx1, 1000) dnl` こんな感じに