

make について

e.wakabayashi-aa

2019 年 9 月 2 日

概要

make は古くから使われているバッチ処理の仕組みです。C 言語のコンパイル (ビルド) に使われてきました。目的とするファイルとそのために必要なファイルとそのために実行すべきコマンドの三つをセットにしただけの単純な構造です。

アプリケーションの開発はその最終結果は 1 個のファイルであっても複数のソースファイル (プログラム) と多くの中間ファイルからなりいくつもの手順を必要とします。当然手作業では煩雑で間違いも生じるのでバッチ処理を行うこととなります。しかしただコマンドを並べただけのバッチでは毎回全ソースをコンパイルすることになりたった一か所の修正のたびに 1 時間以上も待たされることとなります。バッチに条件判定を追加するのも手ですがそれはエラーの一因ともなります。

make はファイルの更新日から行うべき作業を判断し実行します。1 か所の修正のために全行程を実行することはありません。間違いも起こりません。このように便利な make を C 言語のコンパイル用途だけにとどめる手はありません。プログラムを小さくモジュール化することにより開発が容易になりエラー時にも原因箇所特定 (そこで止まる) がすぐにできます。

他の用途にも使えるという記述は見ますがその具体的内容はあまり見ません。ここでその一例と make についての簡単な解説を紹介します。

1 基礎

まず基本的なことがら。

1.1 make と Makefile

make は実行ファイル (UNIX 系なので Windows の拡張子「.exe」は尽きません。)、**Makefile** はテキストファイルです。**Makefile** に記された内容を **make** が実行します。

2 nmake.exe

nmake.exe は Windows 版の make です。Visual Studio に含まれます。

2.1 有償か無償か

詳しくいえば nmake.exe は Visual Studio のコマンドライン・ビルドツール「Build Tools for Visual Studio 2019」の一部です。そしてそれは無償でダウンロードできます。よって nmake.exe も無償であると判断しました。

<https://visualstudio.microsoft.com/ja/downloads/>

2.2 どこにあるか

かなり深いところにあります。現 PC では以下です。ここからコピーして使っています。

C:\Program Files (x86)\Microsoft Visual Studio\2017\BuildTools\VC\Tools\MSVC\14.15.26726\bin\Hostx64\x64

64bit 版を使いしましたが 32bit 版もあります。他のフォルダにもあります。違いはわかりません。

3 Makefile

実行内容を記述したテキストファイルです。make と同じ場所になければなりません。ファイル名は「Makefile」でなければなりません。(オプションを指定すればその限りではありません。)

3.1 Description Blocks

Makefile の基本単位です。ファイルの依存関係を記述します。記述順序は任意です。make がその依存関係とファイル更新日から判断してやるべきことを決めます。

```
targets... : dependents...  
    commands...
```

- targets は目的とするファイル、dependents(依存) はそのために必要なファイル、commands はそのために実行しなければいけないコマンドです。
- targets と dependent は「:」(コロン) で区切ります。
- command の前はタブ 1 個

重要 command の前は必ずタブでなければなりません。「空白 4 個」ではいけません。make エラーの原因がの多くがこれです。

<https://docs.microsoft.com/en-us/cpp/build/reference/description-blocks?view=vs-2017>

3.2 Targets (ターゲット)

複数指定することができます。空白で区切ります。ターゲットを複数指定するとコマンドがターゲットの数だけ実行されるような気がしますがそうはならなかったように記憶しています。一回実行してすべてのターゲット・ファイルが作成されるとそれを検知してそのブロックはもう実行されません。

3.2.1 Pseudotargets

ターゲットにはファイル名を指定するのですが、存在しないファイル、絶対に作られることのないファイルも指定することができます。そうするとどういことが起こるかというとそのターゲットを持つブロックは毎回必ず実行されることになります。そのときのターゲットはファイル名というよりは「ラベル名」といった意味合いになります。UNIX 系ではそういう用途のとき「.」(ドット)で始まる「ファイル名」をし、そのようなターゲットを「ドット・ターゲット」と呼んでいるのですが Windows の nmake.exe はそれを「Pseudotargets」と称しています。

<https://docs.microsoft.com/en-us/cpp/build/reference/pseudotargets?view=vs-2017>

3.3 Dependents (依存)

ターゲットが依存するファイルです。空白で区切って複数指定できます。ここに前述のドット・ターゲットも指定できるのですがあまりお勧めできません。なぜならば、ドット・ターゲットは毎回必ず実行されますのでそれに「依存」するファイルも毎回作成のためのコマンドが実行されることになります。すでに最新版が存在するファイルの作成作業が再度行われることになり make を使う意味が半減します。

<https://docs.microsoft.com/en-us/cpp/build/reference/dependents?view=vs-2017>

3.4 長い行

複数ターゲットや依存を指定すると行が長くなります。行末に「\」(バックスラッシュ)を置くことで行の分割ができます。

3.5 マクロ (変数)

「マクロ」と言えばプログラムのことですが、make が作られたのは何十年も前。現在における「変数」のことを当時は「マクロ」と呼んでいたのでしょう。

マクロ名=文字列

マクロ名に使えるのは英数字と「_」(アンダースコア)、大文字小文字を区別します。マクロ名にマクロを含めることができます。マクロの参照は「\$(マクロ名)」とします。「\$」をマクロ文字列に含めたいときは「\$\$」とします。

<https://docs.microsoft.com/en-us/cpp/build/reference/defining-an-nmake-macro?view=vs-2017>

3.5.1 定義済みマクロ

\$@ 実行中のターゲット

\$MAKEDIR nmake.exe のディレクトリ

3.5.2 環境変数

環境変数もマクロのようにして参照することができます。

\$(環境変数名)

但し、環境変数値の中に「\$」が含まれているとそれをマクロと見なし変換しようと試みます。

4 実行例 1

4.1 Makefile の確認

echo コマンドで文字列「AAA」を書き出すことにことにより a.txt ファイルを作成しえいます。ターゲットは「a.txt」で依存はありません。コマンドは `echo AAA > a.txt` です。次は「b.txt」で「a.txt」をコピーして作成しています。「c.txt」も同様です。「clean」はドット・ターゲットです。ファイルをクリアするときに使います。

ブロックは分かりやすいように実行順に記述していますが実際には順不同です。

```
N:\about_make\sample>type Makefile
a.txt :
    echo AAA > a.txt

b.txt : a.txt
    copy a.txt b.txt

c.txt : b.txt
    copy b.txt c.txt

.clean :
    DEL /Q a.txt b.txt c.txt
```

4.2 c.txt を作成

ターゲットとして「c.txt」を指定して nmak.exe を実行しました。

c.txt とその途中に作られた b.txt と a.txt が確認できます。

```
N:\about_make\sample>nmake.exe c.txt

Microsoft (R) Program Maintenance Utility Version 14.14.26433.0
Copyright (C) Microsoft Corporation. All rights reserved.

    echo AAA > a.txt
    copy a.txt b.txt
    1 個のファイルをコピーしました。
    copy b.txt c.txt
    1 個のファイルをコピーしました。

N:\about_make\sample>
N:\about_make\sample>dir /b *.txt
a.txt
b.txt
c.txt

N:\about_make\sample>
```

4.3 b.txt を削除して再実行

「b.txt」を削除して、再度実行しています。

「/nologo」オプションでロゴを非表示にしています。

a.txt はすでに存在しているのでコマンドは実行されません。

c.txt も存在するのですが更新日が依存している b.txt より古いのでコマンドが実行されました。

```
N:\about_make\sample>del b.txt
```

```
N:\about_make\sample>dir /b *.txt
```

```
a.txt
```

```
c.txt
```

```
N:\about_make\sample>nmake.exe /nologo c.txt
```

```
copy a.txt b.txt
```

```
1 個のファイルをコピーしました。
```

```
copy b.txt c.txt
```

```
1 個のファイルをコピーしました。
```

```
N:\about_make\sample>dir /b *.txt
```

```
a.txt
```

```
b.txt
```

```
c.txt
```

```
N:\about_make\sample>
```

4.4 .clean

ターゲットに「.clean」を指定して実行しました。a.txt, b.txt, c.txt が削除されました。

```
N:\about_make\sample>dir /b *.txt
```

```
a.txt
```

```
b.txt
```

```
c.txt
```

```
N:\about_make\sample>nmake.exe /nologo .clean
```

```
DEL /Q a.txt b.txt c.txt
```

```
N:\about_make\sample>dir /b *.txt
```

```
ファイルが見つかりません
```

```
N:\about_make\sample>
```

4.5 コマンドの確認

「/n」オプションを付けて実行すると、コマンドは実行されず表示されるだけです。

その他のオプションは「/?」オプションで確認できます。

```

N:\about_make\sample>nmake.exe /nologo c.txt /n
    echo AAA > a.txt
    copy a.txt b.txt
    copy b.txt c.txt

```

5 実行例 2

実際にペア予測マクロプログラムで使した Makefile。

5.1 コード

```

1
2 LSPP   = \\bsu01119\WKB\LS-PrePost-4.5-x64\lsprepost4.5_x64.exe
3 RS     = \\bsu01119\WKB\R\R-3.5.3\bin\x64\Rscript.exe
4 PANDOC = \\bsu01119\WKB\Pandoc\pandoc.exe
5
6
7 .usage :
8 @type <<
9 Usage:
10  nmake.exe .clean                work フォルダ内消去
11  nmake.exe work\log.yaml
12  nmake.exe work\intfor_nodelist.csv ノード・リスト作成
13  nmake.exe .chk_nodelist         ノード・リスト確認
14  nmake.exe work\mold_node_orderd.csv モールド・ノード・リスト作成
15  nmake.exe .chk_moldnode         モールド・ノード・リスト確認
16  nmake.exe work\report.html      結果レポート作成
17  nmake.exe work\press12vh.csv    圧力グラフ作成
18  nmake.exe .save                 work フォルダを results フォルダにコピー
19
20 <<
21
22
23 .clean:
24 del /Q work\*
25 IF EXIST work\report_files RD /S /Q work\report_files
26 echo dummy > work\.gitkeep
27
28
29 #.chkvhline :
30 # powershell.exe -File prog\check_vhline.ps1
31
32
33 .chk_nodelist : work\intfor_nodelist.csv work\element.dyn
34 @echo #
35 @echo # ノード並びの確認

```

```

36 @echo #
37 $(RS) prog\checkorder.R work\intfor_nodelist.csv work\element.dyn
38
39
40 .chk_moldnode : work\mold_node_orderd_201.csv \
41                 work\mold_node_orderd_202.csv \
42                 work\mold_node_orderd_203.csv \
43 work\mold_elem.dyn
44 @echo #
45 @echo #     モールド・ノード並びの確認
46 @echo #
47 $(RS) prog\checkorder.R work\mold_node_orderd_201.csv work\mold_elem.dyn
48 $(RS) prog\checkorder.R work\mold_node_orderd_202.csv work\mold_elem.dyn
49 $(RS) prog\checkorder.R work\mold_node_orderd_203.csv work\mold_elem.dyn
50
51
52 .save :
53 @echo #
54 @echo #     Copy work to results
55 @echo #
56 powershell.exe -File prog\copy_work_to_result.ps1 work results
57
58
59 #
60 # 個別ファイル
61 #
62
63 work\log.yaml      :
64 @echo #
65 @echo #     $@
66 @echo #
67 powershell.exe -File prog\logyaml.ps1
68
69
70 work\VH_line.cfile : work\log.yaml
71 @echo #
72 @echo #     $@
73 @echo #
74 powershell.exe -File prog\vhline.ps1
75
76
77 work\edgenodes.csv : work\element.dyn \
78                     work\node.dyn
79 @echo #
80 @echo #     $@
81 @echo #
82 $(RS) prog\get_edgenodes5.R work\element.dyn \

```

```

83                                work\node.dyn      \
84                                work\edgenodes.csv
85
86
87 work\intfor_nodelist.csv : work\VH_line.cfile \
88                                work\edgenodes.csv \
89                                work\node_last.dyn
90 @echo #
91 @echo #      $@
92 @echo #
93 $(RS) prog\trim_nodes.R work\VH_line.cfile \
94                                work\edgenodes.csv \
95                                work\node_last.dyn \
96                                work\intfor_nodelist.csv
97
98
99 work\debug_disp_edge.cfile : work\edgenodes.csv
100 @echo #
101 @echo #      $@
102 @echo #
103 powershell.exe -File prog\cfile_for_disp_nodes.ps1 work\edgenodes.csv work\debug_disp_edge.cfile
104
105
106 work\debug_disp_intfor.cfile : work\intfor_nodelist.csv
107 @echo #
108 @echo #      $@
109 @echo #
110 powershell.exe -File prog\cfile_for_disp_nodes.ps1 work\intfor_nodelist.csv work\debug_disp_intfor.
111
112
113 work\debug_area_node.cfile : work\area_node.csv
114 @echo #
115 @echo #      $@
116 @echo #
117 powershell.exe -File prog\cfile_for_area_node.ps1 work\area_node.csv work\debug_area_node.cfile
118
119
120 work\debug_intfor.png : work\node_last.dyn \
121                                work\edgenodes.csv \
122                                work\intfor_nodelist.csv \
123 work\VH_line.cfile
124 @echo #
125 @echo #      $@
126 @echo #
127 $(RS) prog\debug_intfor.R work\node_last.dyn      \
128                                work\edgenodes.csv      \
129                                work\intfor_nodelist.csv \

```



```

130                work\VH_line.cfile          \
131                work\debug_intfor.png
132
133
134 work\intfor1.cfile : work\intfor_nodelist.csv work\log.yaml
135 @echo #
136 @echo #      $@
137 @echo #
138 powershell.exe -File prog\intfor1_cfile.ps1
139
140
141 work\intfor2.cfile : work\intfor_nodelist.csv work\log.yaml
142 @echo #
143 @echo #      $@
144 @echo #
145 powershell.exe -File prog\intfor2_cfile.ps1
146
147
148 work\getdyn.cfile : work\log.yaml
149 @echo #
150 @echo #      $@
151 @echo #
152 powershell.exe -File prog\getdyn.ps1
153
154
155 work\element.dyn      \
156 work\node.dyn         \
157 work\node_last.dyn   \
158 work\mold_elem.dyn   \
159 work\mold_node.dyn   \
160 work\mold_node_last.dyn : work\getdyn.cfile
161 @echo #
162 @echo #      $@
163 @echo #
164 $(LSPP) c=$(MAKEDIR)\work\getdyn.cfile -nographics
165
166
167 work\mold_node_orderd_201.csv : work\mold_elem.dyn \
168                                work\mold_node.dyn
169 @echo #
170 @echo #      $@
171 @echo #
172 $(RS) prog\order_mold_node.R "201" work\mold_elem.dyn work\mold_node.dyn work\mold_node_orderd_201.
173
174
175 work\mold_node_orderd_202.csv : work\mold_elem.dyn \
176                                work\mold_node.dyn

```

```

177 @echo #
178 @echo #    $@
179 @echo #
180 $(RS) prog\order_mold_node.R "202" work\mold_elem.dyn work\mold_node.dyn work\mold_node_orderd_202.
181
182
183 work\mold_node_orderd_203.csv : work\mold_elem.dyn \
184                                work\mold_node.dyn
185 @echo #
186 @echo #    $@
187 @echo #
188 $(RS) prog\order_mold_node.R "203" work\mold_elem.dyn work\mold_node.dyn work\mold_node_orderd_203.
189
190
191 work\mold_node_orderd.csv : work\mold_node_orderd_201.csv \
192                             work\mold_node_orderd_202.csv \
193 work\mold_node_orderd_203.csv
194 @echo #
195 @echo #    $@
196 @echo #
197 powershell.exe -File prog\order_mold_node.ps1 work\mold_node_orderd_201.csv \
198                                             work\mold_node_orderd_202.csv \
199                                             work\mold_node_orderd_203.csv \
200                                             work\mold_node_orderd.csv
201
202
203 work\vhnode.csv : work\mold_node_last.dyn      \
204                  work\node_last.dyn           \
205                  work\mold_node_orderd.csv     \
206                  work\intfor_nodelist.csv     \
207                  work\VH_line.cfile
208 @echo #
209 @echo #    $@
210 @echo #
211 $(RS) prog\vhnode.R  work\mold_node_last.dyn      \
212                  work\node_last.dyn              \
213                  work\mold_node_orderd.csv       \
214                  work\intfor_nodelist.csv        \
215                  work\VH_line.cfile              \
216                  work\vhnode.csv
217
218
219 work\range.yaml : work\vhnode.csv work\intfor_nodelist.csv work\mold_node_orderd.csv
220 @echo #
221 @echo #    $@
222 @echo #
223 $(RS) prog\range.R  work\vhnode.csv work\intfor_nodelist.csv work\mold_node_orderd.csv work\ran

```

```

224
225
226 work\press12.csv : work\intfor1.cfile work\intfor2.cfile
227 @echo #
228 @echo #    $@
229 @echo #
230 start          $(LSPP)  c=$(MAKEDIR)\work\intfor1.cfile -nographics
231 start /WAIT $(LSPP)  c=$(MAKEDIR)\work\intfor2.cfile -nographics
232 $(RS) prog\press_merge.R work\pressure1.csv work\pressure2.csv work\press12.csv
233
234
235 work\press12vh.csv : work\press12.csv          \
236                    work\VH_line.cfile          \
237                    work\intfor_nodelist.csv
238 @echo #
239 @echo #    $@
240 @echo #
241 $(RS) prog\press_insert_vhline.R work\press12.csv work\vh.csv work\press12vh.csv
242
243
244 work\vh.csv : work\press12.csv work\VH_line.cfile work\intfor_nodelist.csv work\node_last.dyn
245 @echo #
246 @echo #    $@
247 @echo #
248 $(RS) prog\vh.R work\press12.csv work\VH_line.cfile work\intfor_nodelist.csv work\node_last.dyn
249
250
251 work\press.pgm : work\press12.csv work\vh.csv
252 @echo #
253 @echo #    $@
254 @echo #
255 $(RS) prog\press_pgm.R work\press12.csv work\vh.csv work\press.pgm
256
257
258 work\press.png : work\press.pgm
259 @echo #
260 @echo #    $@
261 @echo #
262 magick.exe convert -scale "500%x50%" work\press.pgm work\press.png
263
264
265 work\tire_mold_states.cfile : work\vh.csv work\log.yaml
266 @echo #
267 @echo #    $@
268 @echo #
269 powershell.exe -File prog\tire_mold_states_cfile.ps1
270

```

```

271
272 work\tire_mold_states_list.csv : work\tire_mold_states.cfile
273 @echo #
274 @echo #      $@, tire_state_xxxx.dyn, mold_state_xxxx.dyn
275 @echo #
276 powershell.exe -File prog\tire_mold_states_list.ps1
277
278
279 work\area2.csv : work\range.yaml          \
280                work\tire_mold_states_list.csv
281 @echo #
282 @echo #      $@
283 @echo #
284 powershell.exe prog\area2.ps1
285
286
287 work\areavh.csv : work\vh.csv work\log.yaml work\area2.csv
288 @echo #
289 @echo #      $@
290 @echo #
291 $(RS) prog\areavh.R
292
293
294 work\farthest_list.csv : work\range.yaml
295 @echo #
296 @echo #      $@
297 @echo #
298 powershell.exe -File prog\farthest.ps1
299
300
301 work\farthest.csv : work\farthest_list.csv
302 @echo #
303 @echo #      $@
304 @echo #
305 powershell.exe -File prog\farthest_bind.ps1
306
307
308 work\area_node.csv : work\range.yaml
309 @echo #
310 @echo #      $@
311 @echo #
312 $(RS) --encoding="UTF-8" prog\area_node.R work\range.yaml work\area_node.csv
313
314
315 work\report.html : work\areavh.csv          \
316                  work\area_node.csv        \
317                  work\vh.csv                \

```

```

318     work\farthest.csv      \
319                             work\log.yaml
320 @echo #
321 @echo #     range_sdtate_xxxx.png
322 @echo #
323 powershell.exe -File prog\rangepng.ps1 work\vh.csv
324 @echo #
325 @echo #     $$
326 @echo #
327 $(RS) <<
328 setwd('work')
329 knitr::knit(input='../prog/report.Rmd', output='report.md', encoding='UTF-8')
330 <<
331 $(PANDOC) -t html -o work\report.html -M charset:UTF-8 -M pagetitle:BareY2 -s work\report.md
332
333
334
335

```

5.2 解説

```

2 LSPP    = \\bsu01119\WKB\LS-PrePost-4.5-x64\lsprepost4.5_x64.exe
3 RS      = \\bsu01119\WKB\R\R-3.5.3\bin\x64\Rscript.exe
4 PANDOC  = \\bsu01119\WKB\Pandoc\pandoc.exe

```

LS-PREPOST などの実行に必要なアプリケーションは通常はローカル PC にインストールしておくのだが、共有サーバーに置くことによってインストールを不要にしている。「=」の両端に空白がありクォーティングも行っていないが、マクロ値に空白を含む場合はこういうやり方はできない。

7 .usage :

「.usage」ターゲットで簡単な使い方を表示するようにしている。make を引数なし (ターゲットを指定しないで) で実行すると Makefile に書かれた最初のターゲットが実行される。

```

8 @type <<
...
20 <<

```

インライン・ファイル (ヒア・ドキュメント) を使用しています。コマンド名に「@」を付けるとそのコマンドは実行時に表示されません。

<https://docs.microsoft.com/en-us/cpp/build/reference/specifying-an-inline-file?view=vs-2017>

```

63 work\log.yaml      :
64 @echo #
65 @echo #     $$
66 @echo #
67 powershell.exe -File prog\logyaml.ps1

```

「\$\$」現在のターゲットを表すマクロです。現在どんな作業 (作ろうとしているファイル名) をしているかを表示しています。