

# ソフトウェア第三 — Makefile 復習 —

機械情報工学科 水内郁夫

<http://www.jsk.t.u-tokyo.ac.jp/~ikuo/lec/soft3/>

(ID:soft3, pass:MechanOI)

2008 年 10 月 20 日 (月)

## 1 Makefile

Makefile はコンパイルの手間を減らしてくれる効果だけでなく、自分で書いた構造を忘れてしまうことを防ぐ効果もある。夏学期計算機演習でも取り扱ったが、Makefile の基本を改めて載せておく。Makefile の基本的な仕組みについては「Mech 環境利用の手引き<sup>1</sup>」の 6.5 節も参照すること。

### 1.1 Makefile 基本

Makefile のルールの基本は次の形である。

```
ターゲット:依存するファイル
          コマンド
```

コマンドラインから

```
make (ターゲット)
```

とすることで、「ターゲット」が存在しない、あるいはターゲットよりも新しい「依存するファイル」場合にコマンドが実行される。また、コマンド行の始まりは必ずタブコードを書く。したがって test1 が main.c と hello.c からコンパイルされる場合は以下のように記述する。

```
test1.exe: main.c hello.c goodbye.c
          gcc -o test1.exe main.c hello.c goodbye.c
```

これでコマンドラインから以下のように打ち込めばコンパイルができる。

```
% make test1.exe
gcc -o test1.exe main.c hello.c goodbye.c
```

### 1.2 依存関係の記述

上の例では hello.c が更新された場合でも main.c や goodbye.c のコンパイルが行われてしまい、非効率である。これを回避するためには以下のように main.c から main.o を作るルールと、hello.c から hello.o を作るルールと、goodbye.c から goodbye.o を作るルールを Makefile に書き、test2 は hello.o と goodbye.o と main.o から作られるようにする。そうすると、hello.c が更新された場

<sup>1</sup><http://www.mech.t.u-tokyo.ac.jp/appliance/MechBook/enshu2008-ver10.pdf> . 計算機演習のホームページからリンクされている

合, `make test2` とすると `hello.c` のみ再コンパイルされ, 不要な `main.c` や `goodbye.c` の再コンパイルは行われない.

```
test2.exe: main.o hello.o goodbye.o
    gcc -o test2.exe main.o hello.o goodbye.o
main.o: main.c
    gcc -o main.o -c main.c
hello.o: hello.c
    gcc -o hello.o -c hello.c
goodbye.o: goodbye.c
    gcc -o goodbye.o -c goodbye.c
```

### 1.3 自動変数

ルールを書くときに同じことを何回も書くことを避けるために, 自動変数が用意されている.

```
test3.exe: main.o hello.o goodbye.o
    gcc -o $@ $^
main.o: main.c
    gcc -o $@ -c $^
hello.o: hello.c
    gcc -o $@ -c $^
goodbye.o: goodbye.c
    gcc -o $@ -c $^
```

`$@` ターゲットのファイル名, `$^` は依存するファイル名であるため, この `makefile` は上の例と同じことになる.

`$$` ターゲットがアーカイブメンバだったときのターゲットメンバ名

`$<` 最初の依存するファイルの名前

`$$?` ターゲットより新しいすべての依存するファイル名

`$$^` すべての依存するファイルの名前

`$$+` `Makefile` と同じ順番の依存するファイルの名前

`$$*` サフィックスを除いたターゲットの名前

さらに, ファイル名 `.c` からはファイル名 `.o` を作るルールは, 次のように書ける.

```
%.o: %.c
    gcc -o $@ -c $<
```

こうすると汎用なルールになり, `Makefile` は以下のように4行でシンプルに書くことができる.

```
test4.exe: main.o hello.o goodbye.o
    gcc -o $@ $^
%.o: %.c
    gcc -o $@ -c $<
```

`Makefile` は奥が深いので, 色々調べてみると良い. たとえば, 岡田慧先生のホームページ <http://www.jsk.t.u-tokyo.ac.jp/~k-okada/makefile/> は参考になるだろう.