

# Makefile, options spécifiques

M Billaud

9 octobre 2018

## Résumé

Comment indiquer des options spécifiques pour certaines cibles d'un Makefile

## Table des matières

<b>1</b>	<b>Problématique</b>	<b>1</b>
<b>2</b>	<b>Solution : “target-specific variable values”</b>	<b>2</b>
<b>3</b>	<b>Autre cas d'utilisation, bibliothèques</b>	<b>3</b>
<b>4</b>	<b>Documentation</b>	<b>3</b>

**Résumé.** Comment indiquer des options spécifiques pour certaines cibles d'un Makefile.

**Dernière révision** 15 novembre 2021.

## 1 Problématique

On a un programme principal `main.c`

```
#include <stdio.h>
```

```
void foo();
```

```
void bar();
```

```
int main(void)
```

```
{
```

```

    foo();
    bar();
    return 0;
}

```

qui fait appel à deux fonctions compilées séparément, dans des fichiers sources `foo.c` et `bar.c`.

Un `Makefile` pour la compilation séparée pourrait être

```
CFLAGS = -Wall
```

```
main: main.o foo.o bar.o
```

**Mais** la compilation des sources nécessite des options de compilation différentes. Par exemple les sources font référence à une variable `TRACE` du pré-processeur. et on veut que *l'un* des sources (`foo.c`) soit compilé avec l'option `-DTRACE`, et lui seul.

## 2 Solution : “target-specific variable values”

Une solution simple est l'affectation spécifique pour une cible. La dernière ligne de ce `Makefile`

```
CFLAGS = -Wall
```

```
main: main.o foo.o bar.o
```

```
foo.o: CFLAGS += -DTRACE
```

indique que l'option `-DTRACE` doit être ajoutée à `CFLAGS` lors de la construction de la cible `foo.o`.

Déroulement du `make` :

```

$ make
cc -Wall -c -o main.o main.c
cc -Wall -DTRACE -c -o foo.o foo.c
cc -Wall -c -o bar.o bar.c
cc  main.o foo.o bar.o -o main

```

### 3 Autre cas d'utilisation, bibliothèques

Dans le cas où un Makefile construit plusieurs exécutables, il se peut que certains nécessitent des bibliothèques particulières.

```
all: premier second troisieme

premier:  premier.o  a.o b.o
second:   second.o      b.o c.o
troisieme: troisieme.o a.o b.o c.o

premier:  LDFLAGS =-lm
troisieme: LDFLAGS =-lm
```

Les deux dernières lignes peuvent aussi être groupées

```
premier troisieme:  LDFLAGS =-lm
```

### 4 Documentation

- manuel GNU make, **6.11 Target-specific Variable Values**

[https://www.gnu.org/software/make/manual/html\\_node/Target\\_002dspecific.html](https://www.gnu.org/software/make/manual/html_node/Target_002dspecific.html)