



Outils de développement logiciel

TP n° 6 : compilation multifichiers avec *make*

Alain Lebreton

2023-2024

Objectif

S'initier à la compilation multifichiers avec **make**.

Pré-requis : chapitre 10

Durée estimée : 1 séance

1 Introduction

On se propose de produire un programme permettant de chercher les années comprises entre 2022 et 2100 pour lesquelles le 1er janvier est un lundi. Cette recherche fait appel à la **congruence de Zeller** (du mathématicien allemand Christian Zeller — 1822-1899) permettant de calculer le jour de la semaine pour une année donnée, que ce soit sur le calendrier julien ou grégorien (le nôtre).

2 Travail à réaliser

Exercice n° 1

1. Depuis votre terminal, déplacez-vous dans votre dossier "odl" et créez le sous-dossier "tp06", puis placez-vous dans ce dernier.
2. Copiez l'ensemble des sous-dossiers du dossier "ressources-odl-fisa/tp06/" dans votre dossier de travail en respectant la structure arborescente.
3. Dans le sous-dossier "exercice01", complétez le fichier "Makefile" de manière à ce qu'il permette la production du programme "day_in_a_week" à partir du fichier "week_day.c".
Remarque : on se contentera d'écrire une seule règle de production.
4. Compilez et testez "day_in_a_week".

Exercice n° 2

En général, les programmes sont "fabriqués" à partir de plusieurs fichiers sources.

1. Placez-vous dans le sous-dossier "exercice02", puis complétez le fichier "Makefile" de manière à ce qu'il permette la production du programme "day_in_a_week" à partir des fichiers "week_day.c" et "main.c".
Remarque : on se contentera là encore de n'écrire qu'une seule règle de production.
2. Exécutez **make** puis testez "day_in_a_week".

Exercice n° 3

En analysant le fichier "main.c" de l'exercice précédent, vous pouvez constater que le prototype de la fonction "day_in_a_week()" est placé dans le fichier "main.c".

Mettre les prototypes dans les fichiers ".c" n'est pas une bonne pratique !. En général, les prototypes des fonctions, de même que les constantes et macros utilisées sont regroupés dans des fichiers d'**entête** (extension ".h" en C).

Dans cet exercice, le prototype de la fonction "day_in_a_week()" est déplacé dans le fichier "week_day.h".

Vous remarquerez de plus que ce fichier d'entête comporte des directives particulières placées au début et en fin de fichier.

```
#ifndef WEEK_DAY_H
#define WEEK_DAY_H

/* Contenu du fichier d'entête ... */

#endif
```

Ces directives de précompilation évitent l'insertion multiple de ce fichier.

1. Placez-vous dans le sous-dossier "exercice03", puis complétez le fichier "Makefile" de manière à ce qu'il permette la production du programme "day_in_a_week" à partir des fichiers "week_day.c" et "main.c". On affinera l'écriture de notre règle en utilisant les macros **CC**, **CPPFLAGS** et **CFLAGS** au début du fichier "Makefile".
2. Exécutez **make** puis testez "day_in_a_week".

Exercice n° 4

Jusqu'à présent nous n'avons produit qu'une règle dans le fichier "Makefile". Cette règle compile tous les fichiers à chaque fois, ce qui peut devenir prohibitif si notre projet comporte de nombreux fichiers. La solution consiste à effectuer une compilation modulaire (compilation où chaque fichier est compilé indépendamment des autres et qui s'arrête avant l'édition de liens — production du fichier objet ".o"). Ainsi, **make** n'exécutera que les règles de production des fichiers objets que pour les fichiers qui auront subi une modification entre temps.

1. Placez-vous dans le sous-dossier "exercice04", puis complétez le fichier "Makefile" de manière à ce qu'il permette la production du programme "day_in_a_week" à partir des fi-

chiers "week_day.o" et "main.o". Ces deux derniers fichiers seront produits par deux règles séparées.

2. Exécutez **make** puis testez "day_in_a_week".
3. Les fichiers objets (*.o) sont toujours présents dans le dossier. Ajoutez une règle "clean" permettant de supprimer ces fichiers du dossier.
4. Ajoutez aussi une règle "distclean" qui a pour dépendance la règle "clean" et supprime l'exécutable.
5. Exécutez **make** puis testez ces deux règles.

Exercice n° 5

La plupart du temps, les fichiers d'entête sont placés dans un dossier "include" à la racine du projet, les fichiers sources C sont placés dans un dossier "src" et l'exécutable dans un dossier "bin".

1. Placez-vous dans le sous-dossier "exercice05", puis complétez le fichier "Makefile" de manière à ce qu'il permette la production du programme "day_in_a_week" (dans "bin") à partir des fichiers "week_day.c" et "main.c" placés dans "src" et du fichier d'entête "week_day.h" placé dans "include".
2. Exécutez **make** puis testez vos règles.

3 Livrable

Déposez une archive contenant vos fichiers sur FOAD.

4 Résumé

Dans ce TP vous vous êtes initié à la production d'un programme multifichiers à l'aide de l'outil **make**, et ce en écrivant vos premiers fichiers "Makefile".