

Compilation séparée et Makefile

Cécile Braunstein
cecile.braunstein@lip6.fr

Separate compilation

Why ?

- Modular programming \Rightarrow code easy to understand
- Separation in multiple files \Rightarrow code easy to read
- Creation of library \Rightarrow code may be re-use
- Compile one part of the code at a time \Rightarrow increase maintainability

Principles

We use different files for :

- 1 Declaration of the names (types, functions, class, methods ...)
- 2 Implementation
- 3 Application

Header files (.h)

File .h

- Contains stuff people **calling** the grade functions need to know.
- Can be common to different source files

Utility

When we separate a set of code into a self-contained entity, we call this a **module**.

- This is the **interface** to the module
- Because of them we can compile separately the source files

Header file example

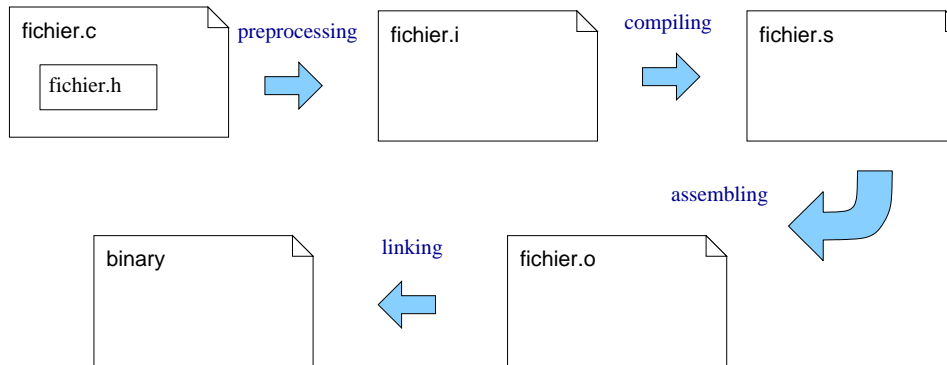
```
#ifndef GUARD_grade_h
#define GUARD_grade_h

#include <vector>
#include "Student_info.h"

double grade(double, double, double);
double grade(double, double, const std::vector<double>&);
double grade(const Student_info&);

#endif
```

Compilation steps



Details

Preprocessing - `g++ -E`

- Removing comments
- Including header `.h` in the source code `.cpp` (`#include`)
- Resolving the preprocessor directive `#define`, `#endif` ...
- Controlling syntax

Compiling - `g++ -S`

Produce the assembly code

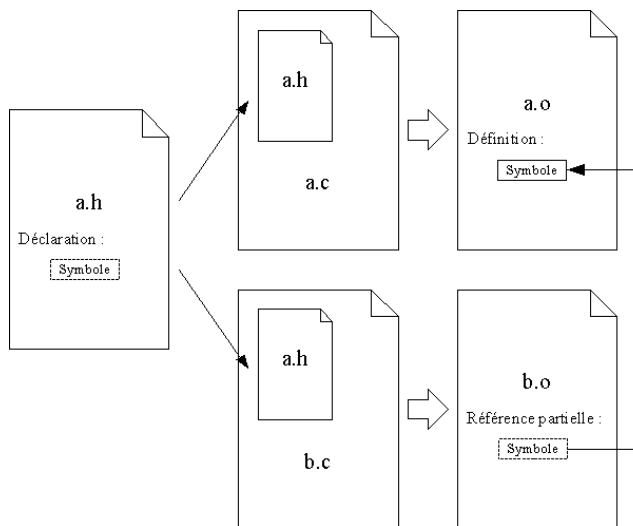
Assembling - `g++ -c`

Transform the assembly code into a binary code

Linking - `g++`

Group the objects files and the library functions to produce an executable

Resume



Compilation

Let a program separate in differents files

- student_info.h / student_info.cpp
- median.h / median.cpp
- grade.h / grade.cpp
- main.cpp

The executable is obtained by the command :

```
g++ -Wall student_info.cpp median.cpp grade.cpp main.cpp  
-o exec
```

When a file is modified, we need to compile everything again

Makefile

```
CC = g++
OPTION = -Wall
all: main
student_info.o: student_info.cpp student_info.h
    ${CC} ${OPTION} -c student_info.cpp -o student_info.o
grade.o: grade.cpp grade.h median.h student_info.h
    ${CC} ${OPTION} -c grade.cpp -o grade.o
median.o: median.cpp median.h
    ${CC} ${OPTION} -c median.cpp -o median.o
main: main.cpp grade.o median.o Student_info.o
    ${CC} ${OPTION} main.cpp grade.o median.o Student_info
    .o -o main
clean :
    rm *.o main
```

Makefile rules

```
target_name : dependences list  
<TAB>command  
<TAB>command
```