

ST0244-032

Clase 11

J.F. Cardona

Universidad EAFIT

26 de agosto de 2015

Agenda

- 1 Capítulo 3. Programación orientada a objeto con C++
 - Utilidad make

Programación orientada a objetos con C++

Utilidad make

- La utilidad make está diseñada para mantener un grupo de programas.
- Es también un lenguaje de programación un objetivo debe ser actualizado si algunas de sus dependencias cambia y se debe obtener de la invocación de uno o más comandos.

```
objetivo: [dependencia] ...  
\t [comando] ...
```

Donde \t Es un tabulador.

- En el siguiente ejemplo se ve un proyecto simple:

```
programa: programa.cpp programa.h  
\t g++ -o programa programa.cpp
```

El lenguaje de programación C++

Utilidad make

- El anterior proyecto puede simplificarse ya que si se observa el objetivo del proyecto que es construir programa se repite tanto en el objetivo, como en parte del comando.
- La utilidad make es como un lenguaje de programación, que tiene variables, que el usuario puede definir, pero también hay variables definidas internamente, una de ellas es \$@ que tiene el nombre de un objetivo

```
programa: programa.cpp programa.h  
\t g++ -o $@ programa.cpp
```

Programación orientada a objetos con C++

Utilidad make

- Se puede simplificar aun más, teniendo en cuenta que las dependencias siguen siempre un orden: primero el fichero fuente y luego los ficheros de encabezados:

```
programa: programa.cpp programa.h  
\t g++ -o $@ programa.cpp
```

- El comando se puede simplificar al tener en cuenta que la variable \$< significa la primera dependencia y se puede reducir así:

```
programa: programa.cpp programa.h  
\t g++ -o $@ $<
```

Programación orientada a objetos con C++

Utilidad make

- En algunas circunstancias es necesario introducir opciones del compilador para una tarea específica como por ejemplo depuración -g.

```
programa: programa.cpp programa.h  
\t g++ -g -o $@ $<
```

- El problema con estas opciones deben ser activadas y desactivadas según el proyecto: desarrollo o producción.
- Para evitar cometer errores, la utilidad de make permite el uso de variables:

```
CPPFLAGS=-g # Declara e inicializa una variable
```

```
programa: programa.cpp programa.h  
\t g++ ${CPPFLAGS} -o $@ $<
```