



# La aplicación **make**

Una ayuda en el proceso de  
compilación



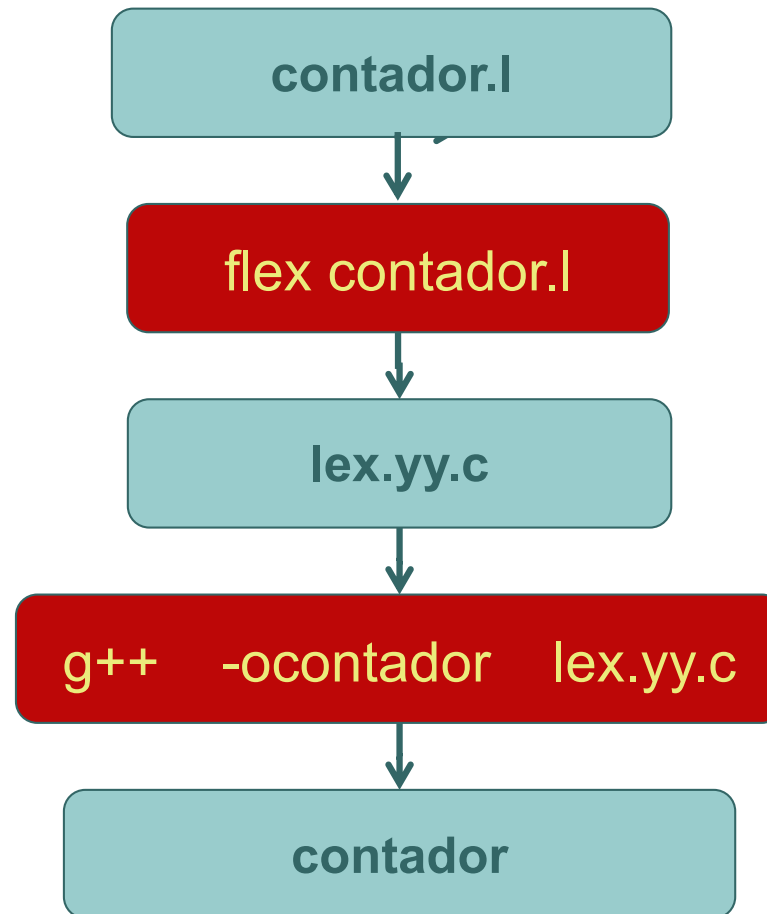
# Introducción

**make** es una herramienta que facilita las tareas asociadas a la compilación y ejecución de programas

En general, lleva a cabo procesos que nos permiten obtener unos ficheros (objeto) a partir de otros (fuente) mediante la ejecución de ciertas **órdenes** dirigidas al S.O. que están descritas en un fichero de texto llamado **makefile**



# Ejemplo sencillo (a. léxico)

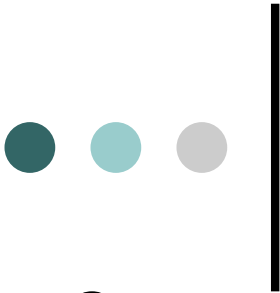




# Ejemplo sencillo de **makefile**

```
contador : lex.yy.c  
    g++ -ocontador lex.yy.c
```

```
lex.yy.c : contador.l  
    flex contador.l
```



# Make

## Comentarios

- En un fichero **makefile** todo el texto que se escriba detrás del símbolo #, hasta el fin de la línea, será considerado como un comentario.

Existen reglas en las que no se define la dependencia entre ficheros.

- Permiten realizar tareas rutinarias, como imprimir fuentes, generar documentación, borrar ficheros innecesarios al final del proceso, ...



# Make: Regla “clean”

- Ejemplo:

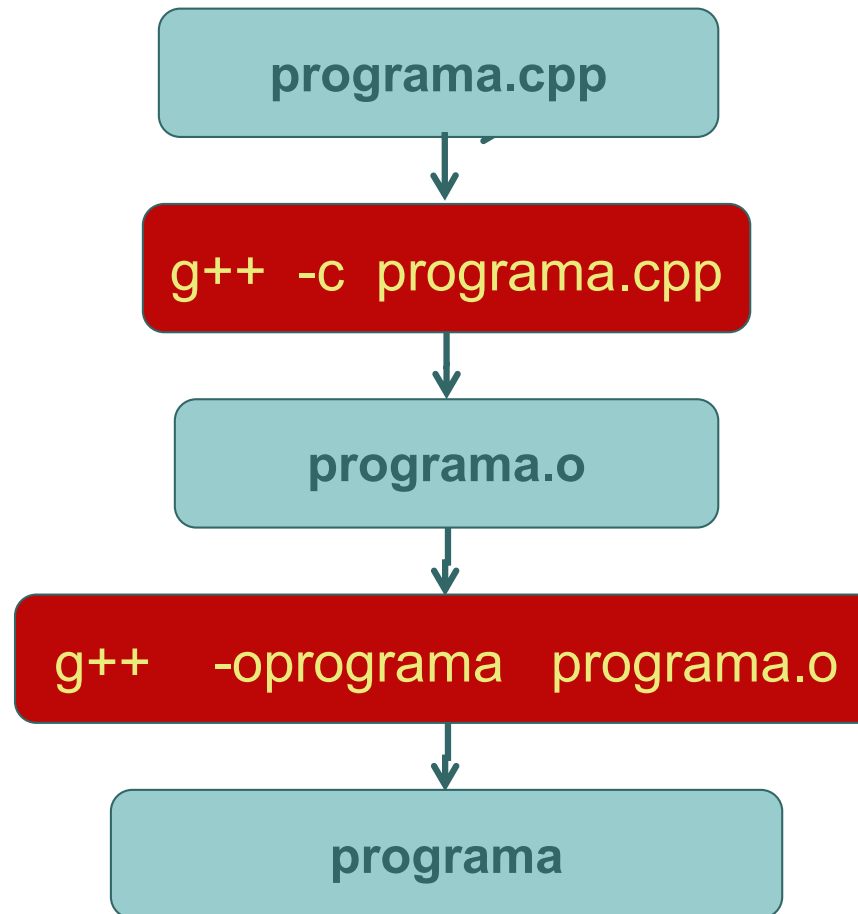
*clean:*

*rm -f \*.o*

- Es una regla para borrar ficheros y se suele colocar al final del fichero
- Para ejecutarla se debe dar la orden

*make clean*

# Ejemplo (compilación en dos fases)





# Ejemplo de **makefile**

#compilación de un programa en dos fases

```
programa : programa.o
```

```
    g++ -o programa programa.o
```

```
programa.o : programa.cpp
```

```
    g++ -c programa.cpp
```

```
clean :
```

```
    rm -f *.o
```

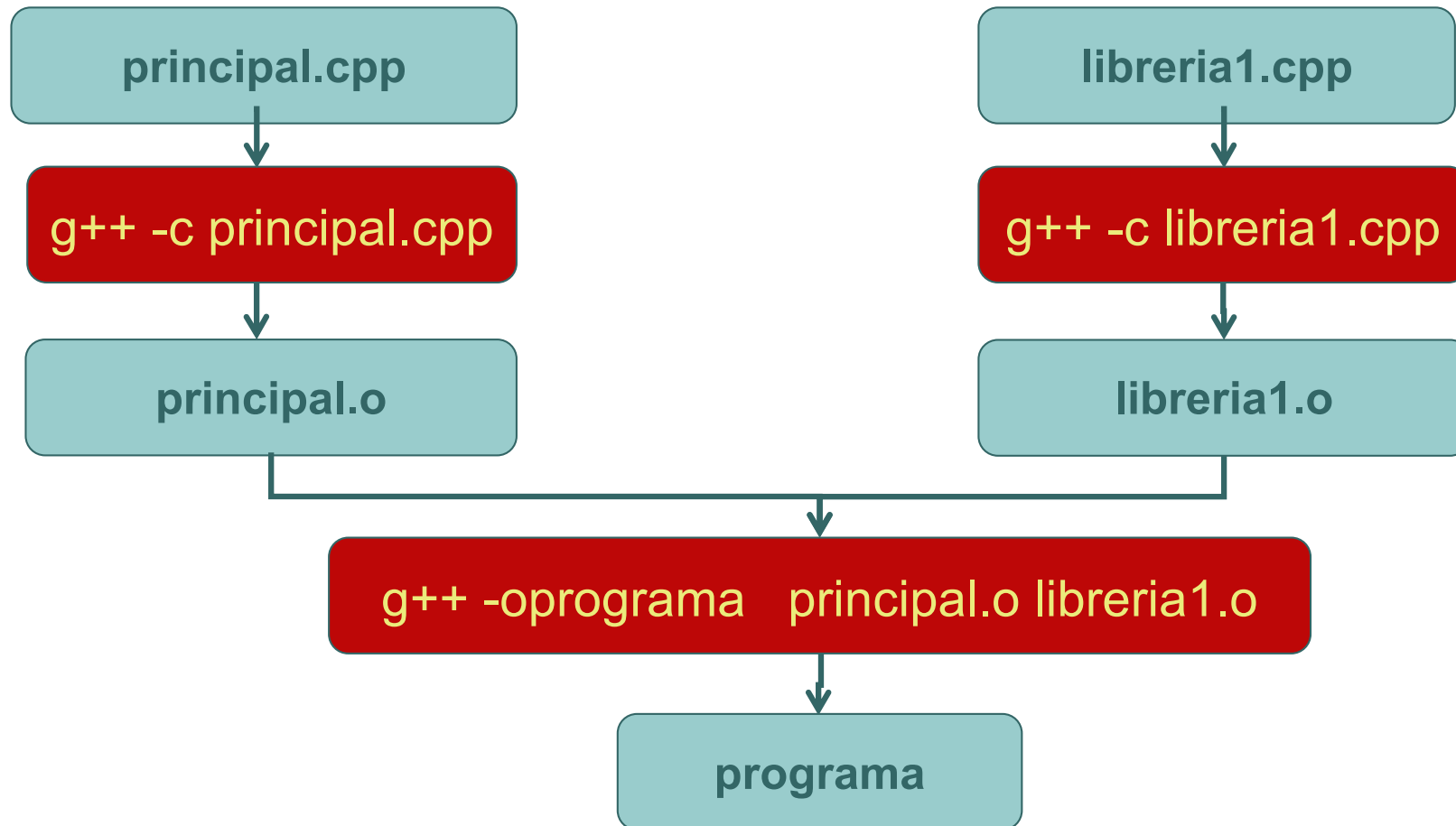




# Make: Variables o macros

- Una variable o macro es un nombre que representa una cadena de texto llamada el valor de la macro. Este valor puede sustituirse en cualquier parte de un fichero makefile.
- Una variable se define al asignarle el texto al que sustituirá `nombre = valor`
- Para utilizar la variable deberemos escribir su nombre entre paréntesis y precedido por el símbolo \$.

# Ejemplo (proyecto con varios ficheros)





# Ejemplo de **makefile**

OBJ = principal.o libreria1.o

programa : \$(OBJ)

    g++ -o programa \$(OBJ)

principal.o : principal.cpp

    g++ -c principal.cpp

libreria1.o : libreria1.cpp

    g++ -c libreria1.cpp



# Ejemplo de makefile

```
#ejemplo de fichero Makefile
```

```
LEX = flex
```

```
CXX = g++
```

```
simple : simple.o
```

```
    $(CXX) -o simple simple.o
```

```
simple.o : simple.cpp
```

```
    $(CXX) -c simple.cpp
```

```
simple.cpp : simple.l
```

```
    $(LEX) -o simple.cpp simple.l
```

```
clean :
```

```
    rm -f *.cpp *.o
```



# Usar make con diferentes ficheros **makefile**

Con make, se utiliza la opción **-f** para indicar que se va a utilizar un fichero de tipo “makefile” con otro nombre.

Por ejemplo:

```
make -f miCompilacion
```

Se suele utilizar cuando es necesario tener varios ficheros de tipo “makefile” en el mismo directorio



# La aplicación **make**

Una ayuda en el proceso de  
compilación