



Programación de Sistemas

CCPG1008

Federico Domínguez, PhD.

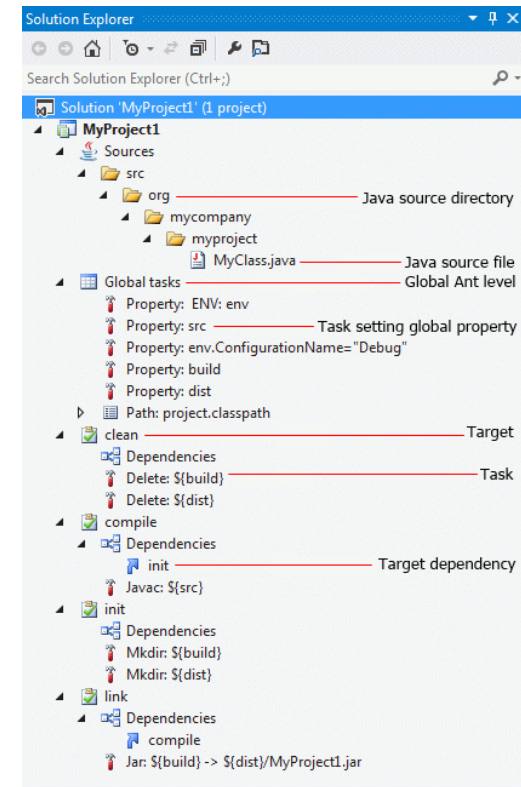
Unidad 2 – Sesión 2: Gestión de compilación con Make

Herramientas de gestión de compilación

Conocidos como *build tools*, son necesarios cuando los proyectos de desarrollo de software escalan en complejidad y tamaño.

Build tools típicamente automatizan:

- Compilación de código fuente
- Empaquetamiento
- Pruebas y despliegue



GNU Make es la herramienta de gestión de compilación *de facto* en Linux.

GNU Make nació por la necesidad, de programadores en el proyecto GNU, de actualizar rápidamente todos los binarios ejecutables una vez que se modificaba el código fuente.

Make está diseñado para actualizar un archivo automáticamente siempre y cuando otros en los que depende sean modificados.

Make utiliza archivos de configuración llamados *Makefiles* para guiar la compilación de un proyecto.

Make es parte de una cadena de herramientas conocida como el **GNU Build System** (*Autotools*).

- *Autoconf* es también parte de *Autotools* y es capaz de generar archivos *Makefile*.
- El objetivo de *Autotools* es gestionar la portabilidad de código fuente entre plataformas.



La compilación de estos tres archivos puede ser automatizada con *make*.

hellomake.c	hellofunc.c	hellomake.h
<pre>#include <hellomake.h> int main() { // call a function in another file myPrintHelloMake(); return(0); }</pre>	<pre>#include <stdio.h> #include <hellomake.h> void myPrintHelloMake(void) { printf("Hello makefiles!\n"); return; }</pre>	<pre>/* example include file */ void myPrintHelloMake(void);</pre>

`gcc -o hellomake hellomake.c hellofunc.c -I.`

Al ejecutar *make*, el programa busca un archivo con el nombre *Makefile* en el directorio local.

Archivo Makefile:

- Contiene una lista de reglas

```
target [target ...]: [component ...]  
Tab ↹ [command 1]  
.  
.  
.  
Tab ↹ [command n]
```

Ejemplo:

```
hellomake: hellomake.c hellofunc.c  
gcc -o hellomake hellomake.c hellofunc.c -I.
```

Al ejecutar *make*, el programa busca un archivo con el nombre *Makefile* en el directorio local.

Archivo Makefile:

- Contiene una lista de reglas

```
target [target ...]: [component ...]  
Tab [command 1]  
.  
.  
.  
Tab [command n]
```

Ejemplo:

Target → `hellomake:`

Dependencias (o componentes) → `hellomake.c hellofunc.c`

Comando → `gcc -o hellomake hellomake.c hellofunc.c -I.`

Regla (grouped by a bracket on the right)

```
hellomake: hellomake.c hellofunc.c  
    gcc -o hellomake hellomake.c hellofunc.c -I.
```

Una **regla** especifica un archivo (**target**) que debe ser creado usando un comando o comandos (**command**) y una dependencia o dependencias (**component**).

Ejemplo:

```
hellomake: hellomake.c hellofunc.c  
gcc -o hellomake hellomake.c hellofunc.c -I.
```

OUTPUT (points to `hellomake`)

INPUT (points to `hellomake.c` and `hellofunc.c`)

Regla (bracketed next to the command line)

PROCESS INPUT -> OUTPUT (points to the entire rule)

Demostración

Referencias uso de *make*

Documentación oficial: <https://www.gnu.org/software/make/>

Tutorial básico: <http://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/>