

## Fundamentos del Software: Ejercicio 1, Módulo II

### Ejercicio 1

Considerando los ficheros proporcionados (*main2.cpp*, *factorial.cpp*, *hello.cpp*, *sin.cpp*, *cos.cpp* y *tan.cpp*), realiza las siguientes operaciones por pasos (no te saltes ninguno y ve mostrando cada uno de los resultados requeridos):

1. Escribe en un archivo *makefile* las órdenes necesarias para que dé la posibilidad de crear el archivo compilado *hello.o*.

Creamos un archivo *makefileEj1* con el siguiente código:

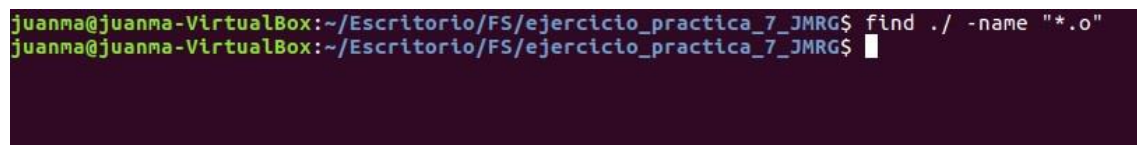


```
#Nombre del archivo: makefileEj1
#Uso: make -f makefileEj1
#Descripción: Crea un archivo compilado hello.o

hello.o: hello.cpp
    g++ -c hello.cpp
```

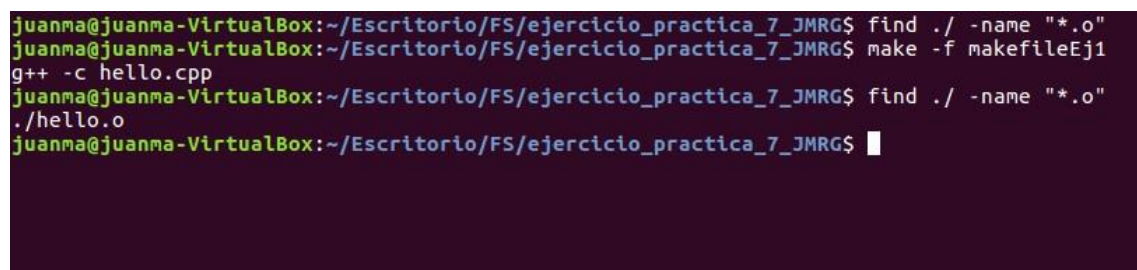
- a) Mediante el *makefile*, haz que se cree dicho fichero.

Observemos que si intentamos buscar un archivo con extensión “.o” con el uso del comando *find* no obtendremos ningún resultado de búsqueda.



```
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ find ./ -name "*.o"
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$
```

Tras ejecutar el *makefileEj1* sí que aparece el archivo *hello.o*:



```
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ find ./ -name "*.o"
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ make -f makefileEj1
g++ -c hello.cpp
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ find ./ -name "*.o"
./hello.o
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$
```

2. Modifica el mismo archivo para que se puedan compilar los archivos *tan.cpp*, *cos.cpp* y *sin.cpp* y obtener sus correspondientes archivos objeto.

Tras modificar el archivo makefile anterior, el resultado es el siguiente:



```
Abrir ▾ [icon]
#Nombre del archivo: makefileEj1
#Uso: make -f makefileEj1
#Descripción: Crea los archivos compilados tan.o, cos.o y sin.o

all: tan.o cos.o sin.o

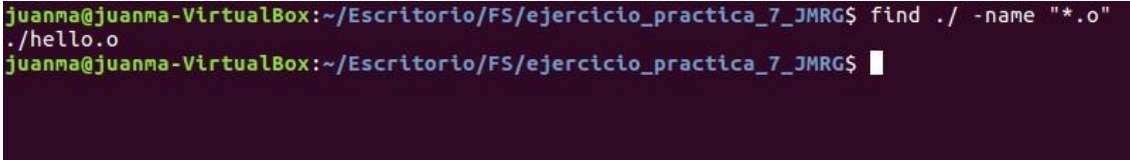
tan.o: tan.cpp
    g++ -c tan.cpp

cos.o: cos.cpp
    g++ -c cos.cpp

sin.o: sin.cpp
    g++ -c sin.cpp
```

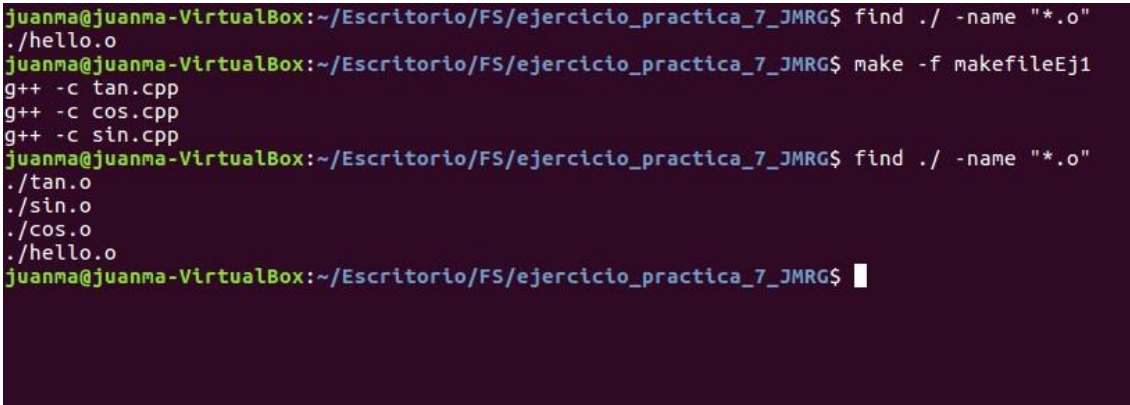
- a) Mediante el *makefile*, haz que se creen cada uno de los ficheros independientemente.

Al ejecutar el comando *find* para ver los archivos con extensión “.o”, solo nos aparece el archivo *hello.o*, el cual hemos creado en el apartado anterior.



```
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ find ./ -name "*.o"
./hello.o
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$
```

Tras ejecutar el makefile *makefileEj1*, se crean los archivos *tan.o*, *cos.o* y *sin.o*, tal y como se muestra en la siguiente imagen:



```
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ find ./ -name "*.o"
./hello.o
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ make -f makefileEj1
g++ -c tan.cpp
g++ -c cos.cpp
g++ -c sin.cpp
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ find ./ -name "*.o"
./tan.o
./sin.o
./cos.o
./hello.o
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$
```

b) Mediante el *makefile*, haz que se creen cada uno de los ficheros de 1 sola vez.

En este caso, el makefile sería de la siguiente forma:

```
Abrir ▾ [icon]
#Nombre del archivo: makefileEj1
#Uso: make -f makefileEj1
#Descripción: Crea los archivos compilados tan.o, cos.o y sin.o

all: tan.cpp cos.cpp sin.cpp
    g++ -c tan.cpp cos.cpp sin.cpp
```

A continuación se muestra la ejecución de dicho makefile (cabe destacar que se eliminaron los archivos objeto *tan.o*, *cos.o* y *sin.o*, ya creados anteriormente, con el fin de mostrar que el resultado de ejecución del makefile es correcto).

```
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ find ./ -name "*.o"
./hello.o
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$
```

```
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ find ./ -name "*.o"
./hello.o
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ make -f makefileEj1
g++ -c tan.cpp cos.cpp sin.cpp
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ find ./ -name "*.o"
./tan.o
./sin.o
./cos.o
./hello.o
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$
```

3. Seguidamente, modifica el *makefile* para que genere una librería llamada *libmath*.

El makefile modificado sería el siguiente:

```
Abrir ▾ [icon]
#Nombre del archivo: makefileEj1
#Uso: make -f makefileEj1
#Descripción: Genera la librería libmath.a

libmath.a: tan.cpp cos.cpp sin.cpp
    g++ -c tan.cpp cos.cpp sin.cpp
    ar rvs libmath.a tan.o cos.o sin.o
```

a) Ejecuta el *makefile* y comprueba qué resultado ha proporcionado.

Si ejecutamos ahora el comando *find* para ver los archivos con extensión “.a”, vemos que no obtenemos resultados:

```
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ find ./ -name "*.a"
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$
```

Si ejecutamos el makefile *makefileEj1*:

```
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ find ./ -name "*.a"
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ make -f makefileEj1
g++ -c tan.cpp cos.cpp sin.cpp
ar rvs libmath.a tan.o cos.o sin.o
ar: creando libmath.a
a - tan.o
a - cos.o
a - sin.o
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ find ./ -name "*.a"
./libmath.a
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$
```

4. Añade las órdenes necesarias para obtener el programa principal compilado y un fichero ejecutable denominado *Practica7*.

Tras añadir las órdenes necesarias para obtener lo pedido, el makefile *makefileEj1* queda de la siguiente forma:

```
Abrir ▾ [icon]
#Nombre del archivo: makefileEj1
#Uso: make -f makefileEj1
#Descripción: Genera el programa principal compilado y un
#             fichero ejecutable denominado Practica7

Practica7: main2.o libmath.a libfunctions.a
        g++ -L./ -o Practica7 main2.o -lmath -lfunctions

main2.o: main2.cpp
        g++ -c main2.cpp

libmath.a: tan.cpp cos.cpp sin.cpp
        g++ -c tan.cpp cos.cpp sin.cpp
        ar rvs libmath.a tan.o cos.o sin.o

libfunctions.a: hello.cpp factorial.cpp
        g++ -c hello.cpp factorial.cpp
        ar rvs libfunctions.a hello.o factorial.o
```

Finalmente, se comprueba que no hay ningún archivo con el nombre *Practica7*, pero tras ejecutar el makefile *makefileEj1*, se crea dicho archivo y lo podemos ejecutar:

```
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ find ./ -name "Practica7"
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$
```

```
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ find ./ -name "Practica7"
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ make -f makefileEj1
g++ -c main2.cpp
g++ -c hello.cpp factorial.cpp
ar rvs libfunctions.a hello.o factorial.o
ar: creando libfunctions.a
a - hello.o
a - factorial.o
g++ -L./ -o Practica7 main2.o -lmath -lfunctions
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ find ./ -name "Practica7"
./Practica7
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ ./Practica7
Hello World!
The factorial of 5 is 120

sin(90)=1
cos(90)=-4.37114e-08
tan(90)=-2.28773e+07
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$
```




## Ejercicio 2

Haciendo uso del archivo anterior, genera un nuevo *makefile* añadiendo al código las siguientes variables:

1. Una variable que almacene dónde están los ficheros *include*.
2. Una variable que almacene el compilador.
3. Utiliza la variable en la que se guarda el nombre del objetivo de la regla actual para no tener que repetir siempre dicha cadena.
4. Modifica las reglas que puedas utilizando la variable donde se almacenan las dependencias de la regla actual.

Creemos un archivo *makefile* *makefileEj2* con el código del *makefile* *makefileEj1* y lo modificamos con las siguientes variables:

```
Abrir ▾ 
#Nombre del archivo: makefileEj2
#Uso: make -f makefileEj2|
#Descripción: Genera el programa principal compilado y un
#              fichero ejecutable denominado Practica7
#
#Variable que indica el compilador
CC=g++
#Variable que indica el directorio de los ficheros include
INCLUDE_DIR=./

Practica7: main2.o libmath.a libfunctions.a
$(CC) -L./ -o $@ $< -lmath -lfunctions

main2.o: main2.cpp
$(CC) -o $@ -I$(INCLUDE_DIR) -c $^

libmath.a: tan.cpp cos.cpp sin.cpp
$(CC) -I$(INCLUDE_DIR) -c $^
ar rvs $@ tan.o cos.o sin.o

libfunctions.a: hello.cpp factorial.cpp
$(CC) -I$(INCLUDE_DIR) -c $^
ar rvs $@ hello.o factorial.o
```

Al igual que hicimos en el final del ejercicio 1, comprobamos que no hay ningún archivo con el nombre *Practica7*, pero tras ejecutar el *makefile* *makefileEj2*, se crea dicho archivo y lo podemos ejecutar (cabe destacar que se eliminó el archivo objeto *Practica7*, ya creado anteriormente, con el fin de mostrar que el resultado de ejecución del nuevo *makefile* es correcto).

```
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ find ./ -name "Practica7"
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$
```

```

juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ find ./ -name "Practica7"
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ make -f makefileEj2
g++ -L./ -o Practica7 main2.o -lmath -lfunctions
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ find ./ -name "Practica7"
./Practica7
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$ ./Practica7
Hello World!
The factorial of 5 is 120

sin(90)=1

cos(90)=-4.37114e-08

tan(90)=-2.28773e+07
juanma@juanma-VirtualBox:~/Escritorio/FS/ejercicio_practica_7_JMRG$

```

Destaquemos finalmente el aumento del número de archivos del directorio en el que hemos trabajado tras haber realizado los ejercicios 1 y 2. Comenzamos trabajando con los siguientes archivos:



Tras realizar ambos ejercicios, el directorio contenía los archivos que se muestran a continuación:

