

PRÁCTICA 7 - FUNDAMENTOS DE SOFTWARE

Apellidos y nombre autores:

Luque Framit, Rafael
Luque Salguero, Pablo
Martínez Díaz, David
Martínez Sáez, Néstor
Molina plaza, Salvador
Rico Ramos, José Luís
Rosado García, J. Miguel
Tirado Guzmán, Miguel

Grupo: A2

Curso: 1º

Titulación: Ingeniería informática + ADE

RESOLUCIÓN EJERCICIOS

- Ejercicio 1:

Enunciado:

Considerando el material proporcionado (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):

1. Escribe en un archivo makefile las órdenes necesarias para que dé la posibilidad de crear el archivo compilado hello.o.
 - a. Mediante el makefile, haz que se cree dicho fichero.
2. Modifica el mismo archivo para se puedan compilar los archivos tan.cpp, cos.cpp y sin.cpp y obtener sus correspondientes archivos objeto.
 - a. Mediante el makefile, haz que se creen cada uno de los ficheros independientemente (llamar 3 veces al script).
 - b. Mediante el makefile, haz que se creen cada uno de los ficheros de 1 sola vez.
3. Seguidamente, modifica el makefile para que genere la librería llamada libmatematicas.
 - a. Ejecuta el makefile y comprueba qué resultado ha proporcionado.
4. Añade las órdenes necesarias para obtener el programa principal compilado y obtener el ejecutable.
 - a. Que el ejecutable final se llame programaE7.

- Solución:

1. # Nombre archivo: makefile_hello
Uso: make

```
hello.o: hello.cpp
    g++ -I./includes -c hello.cpp
```

2. # Nombre archivo: makefile_hello
Uso: make

- Llamando 3 veces al script:

objetivo: tan.o cos.o sin.o

```
tan.o: tan.cpp
    g++ -I./includes -c tan.cpp
```

```
cos.o: cos.cpp
    g++ -I./includes -c cos.cpp
```

```
sin.o: sin.cpp
    g++ -I./includes -c sin.cpp
```

- Llamando a todas las dependencias de 1:

objetivo: tan.o cos.o sin.o
g++ -I./includes -c tan.cpp cos.cpp

3. # Nombre archivo: makefile_hello
Uso: make

objetivo: hello.o tan.o sin.o cos.o libmatematicas.a
g++ -I./includes -c hello.cpp tan.cpp sin.cpp cos.cpp

```
libmatematicas.a:
    ar -rvs libmatematicas.a sin.o cos.o tan.o
```

4. Modificamos el makefile y ponemos como objetivo programaE7:
programaE7: main2.o hello.o factorial.o
g++ -I./includes -L./ -o programaE7 main2.o hello.o factorial.o -lmatematicas

- Makefile final:

programaE7: main2.o hello.o factorial.o libmatematicas.a
g++ -I./includes -L./ -o programaE7 main2.o hello.o factorial.o -lmatematicas

libmatematicas.a: sin.o cos.o tan.o
ar -rvs libmatematicas.a sin.o cos.o tan.o

main2.o: main2.cpp
g++ -I./includes -c main2.cpp

tan.o: tan.cpp
g++ -I./includes -c tan.cpp

cos.o: cos.cpp
g++ -I./includes -c cos.cpp

sin.o: sin.cpp
g++ -I./includes -c sin.cpp

hello.o: hello.cpp
g++ -I./includes -c hello.cpp

factorial.o: factorial.cpp
g++ -I./includes -c factorial.cpp

- Pruebas:

1. Aquí tenemos una prueba de cómo se crea un objeto de “hello.o”, a través del comando “make -f”. Sin embargo, si este ya existe, nos avisará y nos lo creará de nuevo:

```
pablo@pablo-VirtualBox:~/FS/PRACTICA7/ArchivosModuloII/sesion07$ ls
cos.cpp      factorial.cpp  main2.cpp  makefileA  makefileE  sin.cpp
desktop.ini  functions.h   main.cpp  makefileB  makefileH  tan.cpp
dirprograma hello.cpp     makefile  makefileC  makefile_hello
dirprograma libmates.a   makefile2  makefileD  mates.h
pablo@pablo-VirtualBox:~/FS/PRACTICA7/ArchivosModuloII/sesion07$ make -f makefile_hello
g++ -I./includes -c hello.cpp
pablo@pablo-VirtualBox:~/FS/PRACTICA7/ArchivosModuloII/sesion07$ ls
cos.cpp      factorial.cpp  libmates.a  makefile2  makefileD  mates.h
desktop.ini  functions.h   main2.cpp  makefileA  makefileE  sin.cpp
dirprograma hello.cpp     main.cpp  makefileB  makefileH  tan.cpp
dirprograma hello.o     makefile  makefileC  makefile_hello
pablo@pablo-VirtualBox:~/FS/PRACTICA7/ArchivosModuloII/sesion07$
```

2. Comprobamos que los archivos se crean, de las dos maneras, tanto llamando 3 veces al script como llamándolos todos de una:

```
pablo@pablo-VirtualBox:~/FS/PRACTICA7/ArchivosModuloII/sesion07$ ls
cos.cpp      factorial.cpp  main2.cpp  makefileA  makefileE  sin.cpp
desktop.ini  functions.h   main.cpp  makefileB  makefileH  tan.cpp
dirprograma hello.cpp     makefile  makefileC  makefile_hello
dirprograma libmates.a   makefile2  makefileD  mates.h
pablo@pablo-VirtualBox:~/FS/PRACTICA7/ArchivosModuloII/sesion07$ make -f makefile_hello
g++ -c -o tan.o tan.cpp
g++ -c -o cos.o cos.cpp
g++ -I./includes -c tan.cpp cos.cpp sin.cpp
pablo@pablo-VirtualBox:~/FS/PRACTICA7/ArchivosModuloII/sesion07$ ls
cos.cpp      libmates.a  makefile2  makefileD  mates.h  tan.o
factorial.cpp  main2.cpp  makefileA  makefileE  sin.cpp
desktop.ini  functions.h  main.cpp  makefileB  makefileH  tan.cpp
dirprograma hello.cpp  makefile  makefileC  makefile_hello
pablo@pablo-VirtualBox:~/FS/PRACTICA7/ArchivosModuloII/sesion07$
```

3. Modificamos el makefile para crear la librería:

```
pablo@pablo-VirtualBox:~/FS/PRACTICA7/ArchivosModuloII/sesion07$ ls
cos.cpp      libmatematicas.a  makefile2  makefileD  mates.h  tan.o
cos.o        factorial.cpp     main2.cpp  makefileA  makefileE  sin.cpp
desktop.ini  functions.h      main.cpp   makefileB  makefileH  sin.o
libmatematicas.a  hello.cpp  makefile  makefileC  makefile_hello  tan.cpp
pablo@pablo-VirtualBox:~/FS/PRACTICA7/ArchivosModuloII/sesion07$ make -f makefile_hello
g++ -c -o hello.o hello.cpp
ar -rvs libmatematicas.a sin.o cos.o tan.o
ar: creando libmatematicas.a
a - sin.o
a - cos.o
a - tan.o
g++ -I./includes -c hello.cpp tan.cpp sin.cpp cos.cpp
pablo@pablo-VirtualBox:~/FS/PRACTICA7/ArchivosModuloII/sesion07$ ls
cos.cpp      libmatematicas.a  main.cpp  makefileB  makefileH  sin.o
cos.o        factorial.cpp     makefile  makefileC  makefile_hello  tan.cpp
desktop.ini  functions.h      libmatematicas.a  makefile2  makefileD  mates.h  tan.o
libmatematicas.a  hello.cpp  main2.cpp  makefileA  makefileE  sin.cpp
pablo@pablo-VirtualBox:~/FS/PRACTICA7/ArchivosModuloII/sesion07$
```

4. Procedemos a crear el programaE7 y lo ejecutamos:

```
ricco@RicoChet:~/Escritorio/FS/Practica 7/ArchivosModuloII/sesion07$ g++ -L./ -o programaE7 main2.o factorial.o hello.o -lmatematicas
Hello World!
The factorial of 5 is 120
sin(90)=1
cos(90)=-4.37114e-08
tan(90)=2.28773e+07
ricco@RicoChet:~/Escritorio/FS/Practica 7/ArchivosModuloII/sesion07$ ls
cos.cpp  desktop.ini  factorial.o  hello.cpp  libmatematicas.a  main2.o  main.o  makefile2  makefileB  makefileD  makefileH  mates.h  sin.cpp  tan.cpp
cos.o    libmatematicas.a  functions.h  hello.o    main2.cpp  main.cpp  makefile  makefileA  makefileC  makefileE  makefile_hello  programaE7  sin.o    tan.o
ricco@RicoChet:~/Escritorio/FS/Practica 7/ArchivosModuloII/sesion07$
```

- Ejercicio 2:

Enunciado:

Modifica el archivo anterior añadiéndole 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 todo el rato dicha cadena.

4. Modifica las reglas que puedas utilizando la variable donde se almacenan las dependencias de la regla actual.

Para cada uno de los puntos y de los ejercicios anteriormente señalados, indicar cada una de las soluciones por separado con una captura indicando el resultado y cómo quedaría el makefile. Solamente es necesario adjuntar 2 .sh uno para el ejercicio 1 y otro para el ejercicio 2.

- Solución:

Apartado 1, 2, 3 y 4:

Para el apartado 1, simplemente hay que añadir una variable Lib_Dir, que guarda el lugar de los includes.

Para el apartado 2, solo hay que crear una variable CC e igualarlo a g++.
Para el apartado 3, podemos hacer referencia al objetivo cambiándolo por el comando \$@.

Y, por último, el apartado 4 podemos hacer referencia a las reglas de un objetivo separados por espacio en blanco a través de \$^, mientras que lo señalamos con \$<.

Variable que indica el directorio en donde se encuentran los archivos de cabecera

```
INCLUDE_DIR=. /includes
```

Variable que indica el compilador que se va a utilizar

```
CC=g++
```

```
programaE7: main2.o hello.o factorial.o libmatematicas.a  
$(CC) -I$(INCLUDE_DIR) -L./ -o $@ $^
```

```
libmatematicas.a: sin.o cos.o tan.o  
ar -rvs libmatematicas.a sin.o cos.o tan.o
```

```
main2.o: main2.cpp  
$(CC) -I$(INCLUDE_DIR) -c main2.cpp $<
```

```
tan.o: tan.cpp  
$(CC) -I$(INCLUDE_DIR) -c tan.cpp $<
```

```
cos.o: cos.cpp  
$(CC) -I$(INCLUDE_DIR) -c cos.cpp $<
```

```
sin.o: sin.cpp  
$(CC) -I$(INCLUDE_DIR) -c sin.cpp $<
```

```
hello.o: hello.cpp  
$(CC) -I$(INCLUDE_DIR) -c hello.cpp $<
```

```
factorial.o: factorial.cpp  
$(CC) -I$(INCLUDE_DIR) -c factorial.cpp $<
```

- Pruebas:

Apartado 1 y 2: Simplemente creamos las variables Lib_Dir y la variables CC, para poder conocer su valor:

```
rico@RicoChet:~/Escritorio/Fs/Practica 7/ArchivosModuloII/sesion07$ make -f makefile_hello
make: 'programaE7' está actualizado.
rico@RicoChet:~/Escritorio/Fs/Practica 7/ArchivosModuloII/sesion07$ ./programaE7
Hello World!
The factorial of 5 is 120

sin(90)=1
cos(90)=-4.37114e-08
tan(90)=-2.28773e+07
```

Apartado 3: Comprobamos que funciona de nuevo el programa:

```
rico@RicoChet:~/Escritorio/Fs/Practica 7/ArchivosModuloII/sesion07$ make -f makefile_hello
make: 'programaE7' está actualizado.
rico@RicoChet:~/Escritorio/Fs/Practica 7/ArchivosModuloII/sesion07$ ./programaE7
Hello World!
The factorial of 5 is 120

sin(90)=1
cos(90)=-4.37114e-08
tan(90)=-2.28773e+07
```

Apartado 4: Finalmente comprobamos que funciona, otra vez:

```
rico@RicoChet:~/Escritorio/Fs/Practica 7/ArchivosModuloII/sesion07$ make -f makefile_hello
make: 'programaE7' está actualizado.
rico@RicoChet:~/Escritorio/Fs/Practica 7/ArchivosModuloII/sesion07$ ./programaE7
Hello World!
The factorial of 5 is 120

sin(90)=1
cos(90)=-4.37114e-08
tan(90)=-2.28773e+07
```

*Parece la misma foto, simplemente porque comprobamos si funciona el programa, haciendo chequeos o pruebas mientras realizamos la práctica.