

## PRACTICA: Creación de Aplicaciones en C (II)

Objetivos:

- Manejo del Lenguaje C
- Uso de funciones de librería y ficheros cabecera
- Uso de compilador: gcc
- Crear librerías dinámicas y estáticas
- Gestionar los proyectos con make

1.- Dado los siguientes ficheros cabecera :

En /home/lsi/practicaC/include/ tenemos los siguientes .h:aritmetica.h:

```
#ifndef _ARITMETICA_H
#define _ARITMETICA_H
int producto (int a, int b);
int resta (int a, int b);
int suma (int a, int b);
#endif
```

constante.h

```
#ifndef _CONSTANTE_H
#define _CONSTANTE_H
#define PI 3.1415
#endif
```

area.h

```
#ifndef _AREA_H
#define _AREA_H
int rectangulo (int x, int y);
float circulo (int x);
#endif
```

Dado los siguientes módulos en C

En /home/lsi/practicaC/fuentes/ tenemosaritmetica.c:

```
int suma (int a, int b) { return a+b; }
int resta (int a, int b) { return a-b; }
int producto (int a, int b) {return a*b; }
```

area.c:

```
int rectangulo (int x,int y){return producto(x,y);}
float circulo (int x){return (PI * producto(x,x));}
```

a) Incluye todos los #include en los ficheros “aritmetica.c” y “area.c”

`#include "constante.h"``#include "aritmetica.h"`

b) Realiza la aplicación principal(main) en “/home/lsi/practicaC/fuentes/examen.c”, que pida 2 números enteros por teclado y nos muestre por pantalla la suma de los dos números y el área del rectángulo, utilizando los módulos de arriba.

`mkdir practicaC``mkdir practicaC/fuentes``cd practicaC/fuentes``gnome-text-editor examen.c`

```
#include <stdio.h>
#include "aritmetica.h"
#include "area.h"
int main(){
    int num1,num2,rdo;
    printf("Introduce 2 números separados por comas");
    scanf("%d,%d",&num1,&num2);
    rdo=suma(num1,num2);
    printf("La suma de %d y %d es %d",num1,num2,rdo);
    rdo=rectangulo(num1,num2);
    printf("\nEl área del rectángulo de lados %d y %d es %d\n",num1,num2,rdo);
    return 0;
}
```

c) Compila cada módulo por separado y finalmente crea el ejecutable utilizando dichos módulos objeto (suponiendo que estos módulos no están en ninguna librería).

```
cd practicaC/fuentes
gcc -o area.o -c area.c -I../include
gcc -o aritmetica.o -c aritmetica.c
gcc -o examen.o -c examen.c -I../include
gcc -o examen area.o aritmetica.o examen.o
```

d) Introduce área y aritmética en una librería estática que se llamará “libestatica.a” en un nuevo directorio /home/lsi/practicaC/librerias/ y vuelve a crear la aplicación.

```
cd ..
mkdir librerias
cd /home/$USER/practicaC/fuentes
gcc -o area.o -c area.c -I../include
gcc -o aritmetica.o -c aritmetica.c

ar -rv ../librerias/libestatica.a area.o aritmetica.o
ranlib ../librerias/libestatica.a
nm -s ../librerias/libestatica.a
gcc -I../include -L../librerias examen.c -lestatica -o examen -static
./examen
```

e) Introduce área y aritmética en una librería dinámica que se llamará “libdinamica.so” en el nuevo directorio /home/lsi/practicaC/librerias/ y vuelve a crear la aplicación.

```
cd /home/$USER/practicaC/fuentes
gcc -o area.o -c area.c -I../include -fPIC
gcc -o aritmetica.o -c aritmetica.c -fPIC

ld -o ../librerias/libdinamica.so area.o aritmetica.o -shared
gcc -I../include -L../librerias -o examen2 examen.c -ldinamica -Bdynamic
./examen2
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/$USER/practicaC/librerias
export LD_LIBRARY_PATH
./examen2
```

f) Crea el ejecutable (sin usar librerías) usando el comando make. Crea también los objetivos clean para borrar todos los módulos objeto (todos los .o) e install para instalar el ejecutable en /home/lxi/practicaC/bin . Se valorará el uso de VARIABLES.

[gnome-text-editor makefile](#)

```

examen:area.o aritmetica.o examen.o
    gcc -o examen area.o aritmetica.o examen.o
area.o:area.c
    gcc -o area.o -c area.c -I./include
aritmetica.o:aritmetica.c
    gcc -o aritmetica.o -c aritmetica.c
examen.o:examen.c
    gcc -o examen.o -c examen.c -I./include
clean:
    rm *.o
install:
    cp examen ..../bin

```

2. Dado el programa principal incompleto longitudcadena.c

```

#define N 255
int main()
{
char cad[N];
printf("\nIntroduce una cadena: \n");
__fpurge(stdin);
fgets(cad, sizeof(cad), stdin);
printf("\nLa longitud de la cadena %s es %d", cad, strlen(cad));
return 0;
}

```

a)Crea la función “mystrlen” que sustituya a la función strlen(cad). Esta función devuelve igual que strlen la longitud de la cadena que se le pasa como parámetro. Introduce la función mystrlen en un módulo cuyo nombre será “cadenas”. Completa/Modifica el programa principal para que funcione correctamente con la función mystrlen. Realiza el ejecutable usando dicho módulo ( sin utilizar librerías ) y explica los pasos que has realizado para construir el ejecutable.

[mkdir practicaC](#)

[cd practicaC](#)

[mkdir fuentes bin include librerias](#)

[cd include](#)

[cadenas.h](#)

```
int mystrlen(char[] cad);
```

[cd ./fuentes](#)

[cadenas.c](#)

```
#include <stdio.h>
```

```
int mystrlen(char cad[]){
    int contador=0;
```

```

while(cad[contador] != '\0'){
    contador=contador+1;
}
return contador;
}

```

longitudcadena.c

```

#define N 255
#include <stdio.h>
#include "cadenas.h"
int main()
{
char cad[N];
printf("\nIntroduce una cadena: \n");
__fpurge(stdin);
fgets(cad, sizeof(cad), stdin);
printf("\nLa longitud de la cadena %s es %d", cad, strlen(cad));
return 0;
}

```

```

gcc -o cadenas.o -c cadenas.c
gcc -o longitudcadena.o -c longitudcadena.c -I./include
gcc -o longitudcadena -c longitudcadena.o cadenas.o

```

b)Crea el ejecutable (sin usar librerias) usando el comando make. Crea también los objetivos clean para borrar todos los módulos objeto (todos los .o) e install para instalar el ejecutable en /home/lisi/practicaC/bin . Se valorará el uso de VARIABLES.

**gnome-text-editor makefile**

```

longitudcadena:longitudcadena.o cadenas.o
    gcc -o longitudcadena -c longitudcadena.o cadenas.o
cadenas.o:cadenas.c
    gcc -o cadenas.o -c cadenas.c
longitudcadena.o:longitudcadena.c
    gcc -o longitudcadena.o -c longitudcadena.c -I./include
clean:
    rm *.o
install:
    cp longitudcadena ..bin

```

c)Crea la librería estática libcadenas.a con el módulo “cadenas”. Crea el ejecutable usando dicha librería

```

cd /home/$USER/practicaC/fuentes
gcc -o cadenas.o -c cadenas.c

ar -rv ..librerias/libcadenas.a cadenas.o
ranlib ..librerias/libcadenas.a
nm -s ..librerias/libcadenas.a
gcc -I./include -L..librerias longitudcadena.c -lcadenas -o longitudcadena -static
./longitudcadena

```

Nota: Dentro de la carpeta /home/lsi/practicaC/, crea las siguientes carpetas: include, librerias, fuentes y bin siguiendo la estructura del primer ejercicio