

Programación básica

Alma González
Octubre 2021



Argumentos de la función *main*

- Hasta el momento hemos usado la función `main` sin argumentos, i.e `main()`.
- La función principal puede tener argumentos de entrada, que nos ayuden a leer parámetros para la ejecución del programa desde la línea de comandos.
- Ejemplo: Supongamos que quiere tener un programa para calcular la conversión de temperaturas Celsius a Kelvin, en un rango dado y escribir el resultado en un archivo llamado “temperaturas.dat”, y que los parámetros de entrada sean especificados desde el momento de ejecutar el programa. Es decir, quisiéramos ejecutar el programa de la forma:

```
./programa 0 25 20 temperaturas.dat
```

Argumentos de la función *main*

- La sintaxis general será:

```
int main(int argc, char *argv[] ){  
    instrucciones...;  
}
```

- Donde,

argc: es el numero de argumentos que se dan desde la linea de comandos, Incluyendo el propio nombre del programa.

*argv[]: es un apuntador a un arreglo que contendrá todos los argumentos pasados al programa a partir de la linea de comandos.

Argumentos de la función *main*

Ejemplo: programa que recibe y checa los argumentos que se usaron

```
#include <stdio.h>

int main( int argc, char *argv[] ) {
    printf("El programa que estas ejecutando es: %s\n", argv[0]);

    if( argc == 2 ) {
        printf("El argumento que se dio es: %s\n", argv[1]);
    }
    else if( argc > 2 ) {
        printf("Mas argumentos de los necesarios \n");
    }
    else {
        printf("Se requiere de al menos 1 argumento \n");
    }

    return(0);
}
```

Argumentos tipo string (char)

```
#include <stdio.h>

int main( int argc, char *argv[] ) {
    char *nombre_archivo;
    FILE *fp;

    printf("El programa que estas ejecutando es: %s\n", argv[0]);

    if( argc == 3 ) {
        nombre_archivo=argv[1];
        printf("El primer argumento que se dio es: %s\n", argv[1]);
        printf("El nombre del archivo a abrir es: %s\n", nombre_archivo);
        fp = fopen(nombre_archivo, "r");
        ///Aquí se pondría todo lo que se quiere hacer....
        fclose(fp);

        printf("El segundo argumento que se dio es: %s\n", argv[2]);
        nombre_archivo=argv[2];
        printf("El nombre del archivo a abrir es: %s\n", nombre_archivo);
        fp = fopen(nombre_archivo, "w");
        ///Aquí se pondría todo lo que se quiere hacer....
        fclose(fp);
    }
    else if( argc > 3 ) {
        printf("Mas argumentos de los necesarios \n");
    }
    else {
        printf("Se requiere de al menos 1 argumento \n");
    }

    return(0);
}
```

Argumentos tipo int, float, etc...

atof :
conversion de
cadena a
flotante

atoi :
conversion de
cadena a
entero

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

int main( int argc, char *argv[] ) {
    printf("El programa que estas ejecutando es: %s\n", argv[0]);
    char *nombre_archivo;
    float Tin,Tfin;
    int num;
    FILE *fp;
    if( argc == 5 ) {
        Tin=atof(argv[1]);
        Tfin=atof(argv[2]);
        num=atoi(argv[3]);
        nombre_archivo= argv[4];
        printf("El nombre del archivo a abrir es: %s\n", nombre_archivo);
        fp = fopen(nombre_archivo, "w+");
        fprintf(fp,"Tin: %f, \t Tfin: %f, \t N: %i",Tin,Tfin,num);
        fclose(fp);
    }
    else if( argc > 5 ) {
        printf("Diste mas argumentos de los necesarios \n");
    }
    else {
        printf("Se requieren 4 argumentos: Tin, Tfin, num, nombre_archivo \n");
    }
    return 0;
}
```

Compilar un programa con mas de un archivo

- Es posible separar los archivos en dos o mas archivos.c y compilarlos al mismo tiempo. Por ejemplo la función principal está en archivo y las funciones definidas por el usuario, en otro...
- Ej. gcc cuadrado.c funciones.c -o cuadrado.o

cuadrado.c

```
#include <stdio.h>

void cuadrado();

int main(){
    cuadrado();
    return (0);
}
```

Contiene a
la función
principal

funciones.c

```
#include <stdio.h>

void cuadrado( ){
    float x,x2;
    printf("Introduce un numero\n");
    scanf("%f",&x);
    x2=x*x;
    printf("El cuadrado de %f es %f\n",x,x2);
}
```

Contiene
otras
funciones

Ejercicio.

- Cambiar al menos 3 de los códigos de clases pasadas para que la función principal utilice argumentos de entrada.
- Además dichos programas deberán utilizar funciones que estén definidas en otro archivo, de tal forma que haya un archivo que continúe a la función principal y otro que contiene a las funciones, pero se compilan en un solo programa.