

SO sesión 2: el lenguaje C

domingo, 8 de noviembre de 2020

19:17

<code>gcc</code>	Compilador de C de GNU	<code>-c,-o, -l</code> (i r L,-l (ele min
<code>sprintf</code>	Conversión de formato almacenándola en un búffer	
<code>atoi</code>	Convierte un string a un número entero	
<code>indent</code>	Indentación de ficheros fuente	

Cout = sprintf (función librería C) + write (llamada a sistema)

La salida estándar por defecto es la pantalla (canal 1) y la salida estándar de errores es el canal 2

Strlen, para conocer la longitud del texto

```
all: listaParametros
listaParametros: listaParametros.c
                gcc -o listaParametros listaParametros.c
clean:
                rm listaParametros|
```

Makefile: make

`\0` = espacio

`\n` = salto de línea (endl)

`%d` = int

`%c` = char

`%s` = string

mayúscula), –
úscula)

r de

```

#include "misfunciones.h"

int main(int argc, char *argv[]) {
    char buf[80];
    int suma = 0;
    if (argc < 3) {
        usage();
        return 0;
    }
    for (int i = 0; i < argc; i++){
        int mac;
        if (i == 0) {
            mac = sprintf(buf, "El argumento %d es %s (es el nombre del ejecutable)\n", i, argv[i]);
            write (1, buf, mac);
        }
        else {
            if (esNumero(argv[i])) {
                suma += mi_atoi(argv[i]);
            }
            else {
                mac = sprintf(buf, "%s no es numero\n", argv[i]);
                return 0;
            }
        }
    }
    int pmac;
    pmac = sprintf(buf, "La suma es %d\n", suma);
    write (1, buf, pmac);
}

```

```

#include "misfunciones.h"

#define MAX_SIZE 8

unsigned int char2int (char c) {
    return c - '0';
}

int mi_atoi (char *s) {
    int n = 0;
    int negativo = 0;
    if (*s == '-') {
        ++s;
        negativo = 1;
    }
    n = char2int(*s);
    ++s;
    while (*s != '\0') {
        n = (n*10) + char2int(*s);
        ++s;
    }
    if (negativo) return -n;
    else return n;
}

```

```

#include <stdio.h>
#include <string.h>
#include <unistd.h>

#define MAX_SIZE 8

//Asegura que el programa tiene almenos 2 inputs, en caso contrario, muestra un mensaje por pantalla
void usage();
//Convierte char a un entero
unsigned int char2int(char c);
//Convierta un string a un entero
int mi_atoi(char *s);
//Comprueba si el char es un numero o no
int esNumero (char *str);

```

```
all: suma
```

```
suma: suma.c misfunciones.c misfunciones.h  
      gcc -c misfunciones.c -o misfunciones.o  
      gcc -o suma suma.c misfunciones.o
```

```
clean:  
      rm suma  
      rm misfunciones.o
```

make -f Makefile2 (si no se tiene el makefile como 'Makefile' se ha de poner -f

```
#include <stdio.h>  
#include <string.h>  
#include <unistd.h>  
  
void usage() {  
    int len;  
    char buff[150];  
    len = sprintf(buff, "Usage: listaParametros arg1 [arg2..argn]\nEste programa cuenta el número de palabras que hay  
    write(1, buff, len);  
}  
  
int main(int argc, char *argv[]) {  
    if (argc == 1) {  
        usage();  
        return 0;  
    }  
    char buff[100];  
    int numeropalabras = 0;  
    for (int i = 0; i < strlen(argv[1]); ++i) {  
        if (argv[1][i] == ' ' || argv[1][i] == '.' || argv[1][i] == ',' || argv[1][i + 1] == '\0') ++numeropalabras;  
    }  
    int len = sprintf(buff, "%d palabras\n", numeropalabras);  
    write(1, buff, len);  
}
```

```
void usage() {  
    char buf[150];  
    int mac;  
    mac = sprintf(buf, "Usage: listaParametros arg1 [arg2..argn]\n  
    Este programa escribe por su salida la lista de argumentos que recibe\n");  
    write(1, buf, mac);  
}
```

