Arquitectura de sistemas

Abelardo Pardo

University of Sydney
School of Electrical and Information Engineering
NSW, 2006, Australia
Autor principal del curso de 2009 a 2012

Iria Estévez Ayres

Damaris Fuentes Lorenzo

Pablo Basanta Val

Pedro J. Muñoz Merino

Hugo A. Parada

Derick Leony

Universidad Carlos III de Madrid
Departamento de Ingeniería Telemática
Avenida Universidad 30, E28911 Leganés (Madrid), España

Tabla de contenidos

8.1. Actividades

8.1.1. La función `printf`

Recursos

- Fichero `printf.c`

Plan de trabajo

1. Compila y ejecuta el fichero `printf.c`.
2. Modifica las cadenas de formato utilizadas en la función `printf` para que imprima con el siguiente formato (hazlo sin escribir más de dos espacios en blanco consecutivos):

```
La letra es               m
El número es              60345698
El número hexadecimal es  0X3FA
El número real es         3.10e+33
El string es              ABCDE
```

8.1.2. Operaciones de entrada salida con caracteres

Recursos

- Subcarpeta "`Using_getc`" en la carpeta compartida.

Plan de trabajo

Modifica el programa llamado `using_getc.c` en la carpeta `Using_getc` para que lea caracteres del teclado y los imprima por pantalla, hasta que el usuario pulse el carácter 'q', que terminará la aplicación. Si los caracteres son 'A','B' o 'C', imprime los valores numéricos. Pista: Usa una sentencia `switch` para implementar el ejercicio. Actualiza el contenido en el depósito **Subversion** cuando termines.

8.1.3. Operaciones de entrada/salida y gestión de memoria dinámica

Recursos

- Subcarpeta "`Using_scanf_and_memory`" en la carpeta compartida.

Plan de trabajo

Vamos a realizar un programa sencillo que permite introducir palabras y mostrarlas. El programa debe pedir por pantalla el tamaño de la palabra a escribir. Una vez obtenido, reservará la memoria adecuada y entonces pedirá la palabra en concreto. Tras introducirla, el programa la mostrará por pantalla. Estos pasos los hará dentro de un bucle hasta que el usuario introduzca que el tamaño de la palabra siguiente a introducir es 0. Modifica para ello el programa de la carpeta `Using_scanf_and_memory`.

Usa la función `scanf` para la recogida de datos, `printf` para mostrarlos y, para la gestión de memoria, usa la que creas más conveniente.

Al terminar, sube la versión actualizada a **Subversion**.

8.1.4. Operaciones de entrada salida con tipos de datos

Recursos

- Subcarpeta "`Using_scanf_with_numbers`" en la carpeta compartida.

Plan de trabajo

Modifica el programa llamado `using_scanf.c` de la carpeta `Using_scanf_with_numbers` para que lea 5 cifras de teclado y los imprima por pantalla en forma de pirámide simétrica. Por ejemplo, si tenemos los números 1,14,124,3456 y 12345, que la salida sea:

```
    1    1
   14   14
  124   124
 3456   3456
12345   12345
```

Al terminar, sube la nueva versión a **Subversion**.

8.1.5. Funciones `gets` y `fgets`

Recursos

- Subcarpeta "`Using_fgets`" en la carpeta compartida.

Plan de trabajo

Usando `fgets`, modificad el programa llamado `using_fgets.c`, almacenado en la carpeta

`Using_fgets`, para que permita al usuario introducir una línea de texto. El programa sólo cogerá hasta 80 caracteres. Tras recibirlos, que cuente la frecuencia de aparición de cada letra de esa línea introducida. El carácter 'A' será igual que 'a' a efectos de cómputo de esa frecuencia. Nota: Te será útil normalizar los caracteres a minúscula antes de contarlos, con la función `int tolower(int c)` de la librería `ctype.h`.

Cuando acabes, sube el fichero al repositorio con `svn commit`.

8.1.6. La función `getline`

Recursos

- Subcarpeta "`Using_getline`" en la carpeta compartida.

Plan de trabajo

Escribe, en el fichero bajo `Using_getline` llamado `using_getline.c`, un programa con los siguientes aspectos:

1. Una estructura denominada `struct survey` que pueda almacenar el nombre de una persona (de longitud variable) y un campo que me permita saber bien la calle de Leganés donde vive (si es de Leganés), o bien el distrito de Madrid donde reside (si no es de Leganés). Pista: Necesitarás en esta estructura un campo que te permita saber si es de Leganés o no.
2. Función `void data_enter(struct survey *ptr)`: Esta función debe preguntar al usuario por su nombre. Después, que le pregunte si es de Leganés y, si la respuesta es Sí (S), que le pregunte la calle donde vive. Si no es de Leganés (respuesta N), que le pregunte el distrito de donde viene.
3. Función `void data_display(struct survey *ptr)`: Que imprima el nombre del usuario y su calle si es de Leganés, o su distrito si no lo es.

Rellena la función `main` para que primero recoja los datos del usuario y que luego los muestre por pantalla con las funciones anteriores.

Cuando acabes, sube el fichero al repositorio con `svn commit`.