

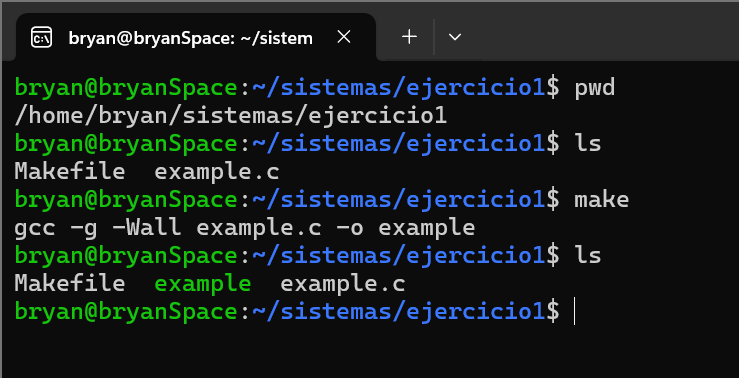
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
|  | | Memoria Laboratorio C | | |  | |
|  |  | | | | |  |
|  | | |  |  | | |
|  | | | Programación C |  | | |
|  | | | Linux default terminal window vector clip art | Public domain vectorsThe 40 Most-Used Linux Commands You Should Know22/01/2024—Redes y Sistemas Operativos—Bryan Xavier Quilumba Farinango |  | | |

**1.1: Repaso de aspectos sobre sistemas operativos y programación en C. Manejo de hilos.**

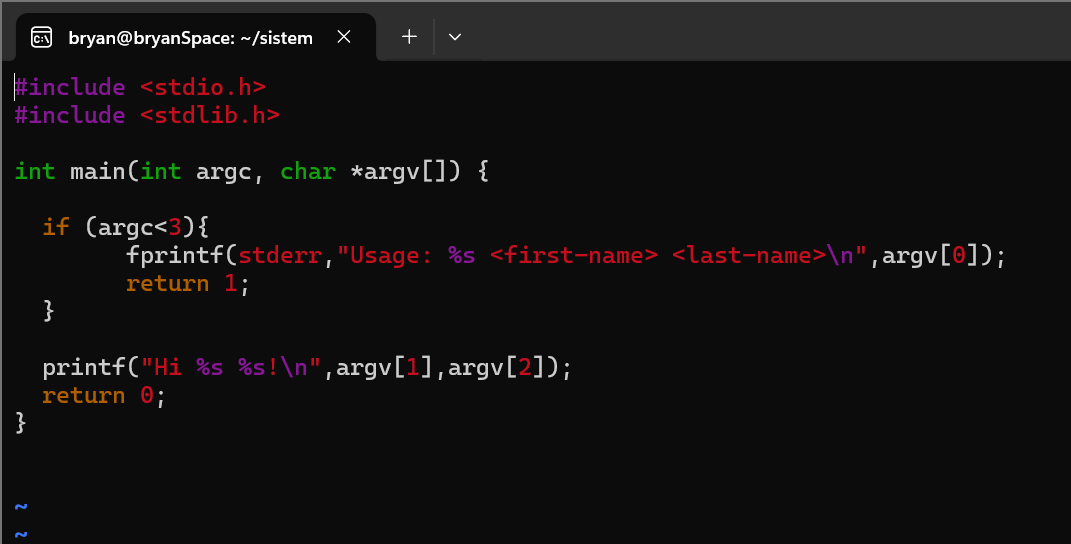
**Ejecuta make en la línea de comandos y comprueba las órdenes que ejecuta para construir el proyecto. ¿qué comando(s) se invoca(n)? ¿Se ha generado algún fichero ejecutable como resultado? Ejecuta ls para ver el contenido del directorio.**

Se invoca "gcc -g -Wall example.c -o example", que compila example. y crea el ejecutable example

Si, se ha creado el ejecutable example



Tras construir el proyecto, ejecuta los siguientes comandos, e indica la salida generada por pantalla. ¿A que se debe este comportamiento? Consulta el código fuente usando un editor de textos.



./example

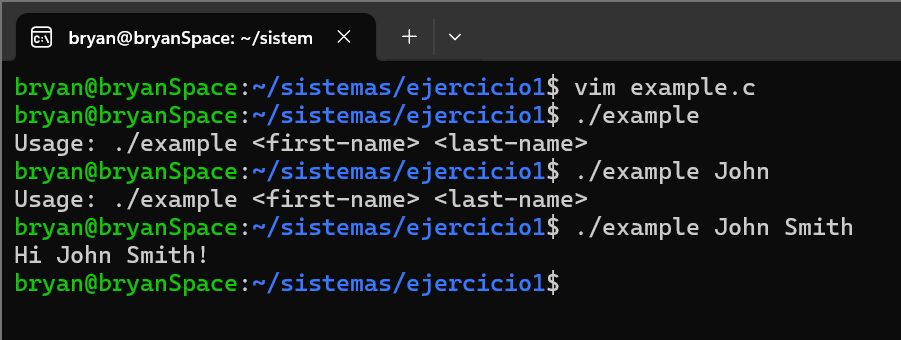
El programa muestra un mensaje en el que nos indican que la ejecución debe ir acompañada de 2 argumentos correspondientes al nombre y apellido.

./example John

De nuevo el programa nos indica que debemos agregar nombre y apellido, siendo el ultimo el que nos faltaría.

./example John Smith

El programa lanza un mensaje saludando al nombre y apellido pasados por argumento.

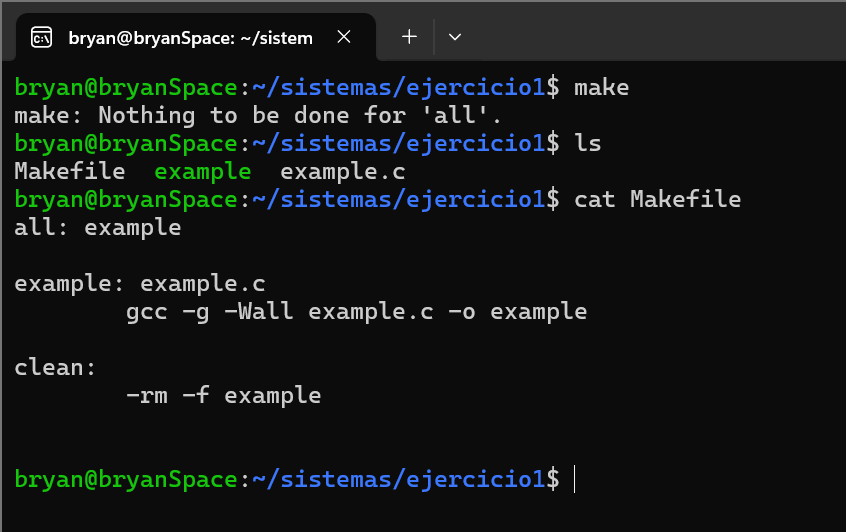


Este comportamiento se debe a que el código del programa captura el nímero de argumentos y cuando este es menor a tres avisa de la falta de argumentos, y cuando es superior lanza un saludo usando el primer y segundo argumento (realmente el segundo y tercero, pues el primer argumento siempre es el nombre del programa).

**Vuelve a ejecutar el comando make en más ocasiones. ¿Se vuelve a generar el fichero ejecutable invocando al compilador gcc? ¿Por qué?**

Me parece que no se genera uno nuevo, pues make lanza el aviso de que no hay nada por hacer para ‘all’.

Supongo que se debe a que dentro del fichero Makefile hay un etiqueta llamada all donde se indican los ficheros necesarios para compilar el programa, y cuando todos están ya presentes (en este caso solo era necesario example) entonces el Makefile no tiene nada por hacer.

****

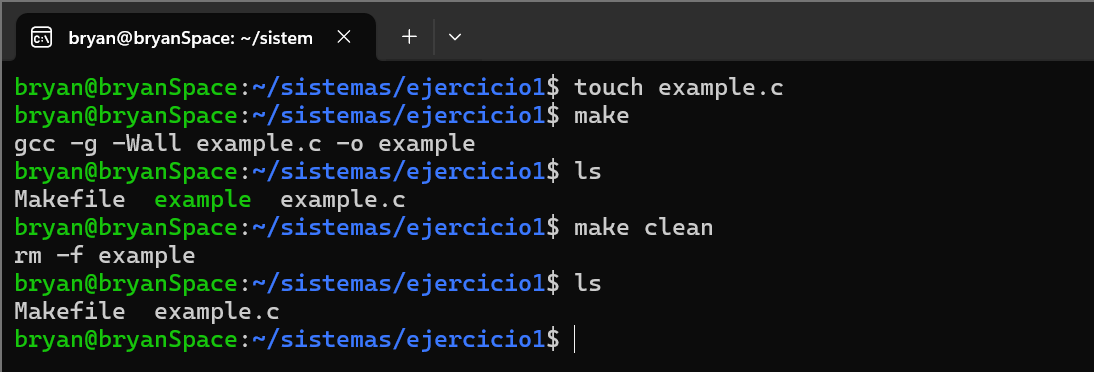
**Marca el fichero example.c como modificado ejecutando touch example.c. Después ejecuta de nuevo make . ¿Qué diferencia hay con la última vez que lo ejecutaste? ¿Por qué?**

Esta vez si que se vuelve a compilar y crear el ejecutable. Esto se debe a que la etiqueta example de Makefile requiere de example.c, y ha detectado que este último ha sido modificado por lo que realiza las tareas implementadas en esa etiqueta, en este caso gcc …

Esto tiene todo el sentido, ya que Makefile nos ayudará a compilar una y otra vez nuestros programas, los cuales tendrán ficheros que irán cambiando y tener estas etiquetas que están atentas es esencial para volver a compilar los archivos necesarios modificados.

**Ejecuta la orden make clean. ¿Qué ha sucedido?**

Se ha ejecutado el comando implementado en la etiqueta clean, el cual corresponde al remove del ejecutable.

****