Sumario

ED

Entornos de Desarrollo (ED)
Unidad de Trabajo: UT1
Tarea 0103: Análisis de la traducción de un programa en C

1.Descripción de la tarea:

En esta tarea, aprenderás a analizar y documentar el proceso de compilación de un programa escrito en C, desde la creación del archivo hasta la ejecución del programa final. Este proceso incluye varias fases: **preprocesado**, **compilación**, **ensamblado** y **enlazado**. A lo largo de la tarea, te guiaremos paso a paso en la creación y ejecución de un programa C en un entorno Linux, usando la terminal.

2.Instrucciones paso a paso:

2.1.Crear el archivo fuente en C

Lo primero que debes hacer es crear un archivo que contenga el código fuente en C. Para ello, utilizaremos el editor de texto `nano`, que puedes ejecutar desde la terminal. Sigue estos pasos:

- Abre la terminal de Linux.

- Para crear un archivo llamado `main.c`, escribe el siguiente comando y presiona **Enter**:

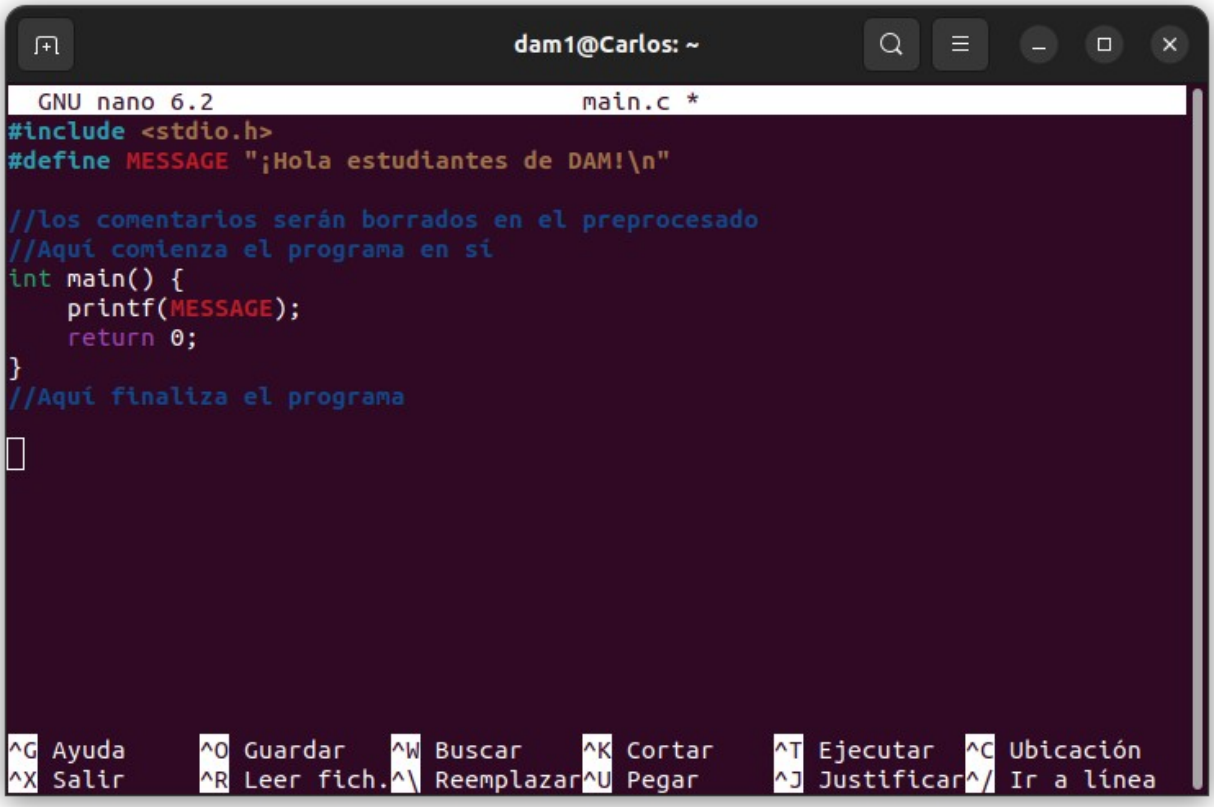
  ```
  nano main.c
  ```

- Dentro del editor de texto `nano`, escribe el siguiente código:

  ```
  #include <stdio.h>
  #define MESSAGE "¡Hola estudiantes de DAM!\n"

  //los comentarios serán borrados en el preprocesado
  //Aquí comienza el programa en sí
  int main() {
      printf(MESSAGE);
      return 0;
  }
  //Aquí finaliza el programa
  ```

- Para guardar el archivo y salir de `nano`, presiona **Ctrl + O** (luego Enter para confirmar el nombre del archivo), y después **Ctrl + X** para cerrar el editor.



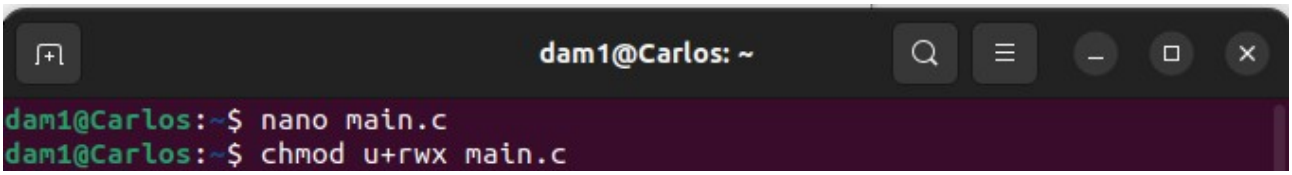
Inserte el contenido del main.c (Supongo que es un código con el lenguaje de programación .c que permite escribir un mensaje y que salga por la pantalla).

También lo hice desde el editor de texto de linux para comprobar varias maneras.

2.2.Cambiar los permisos del archivo para que sea ejecutable

En Linux, los archivos creados no son ejecutables de forma predeterminada. Antes de compilar y ejecutar el programa, deberás cambiar los permisos del archivo. En la terminal, escribe el siguiente comando para asegurarte de que tengas permisos de lectura y escritura:

```
chmod u+rwx main.c
```



Mediante este comando, lo que estoy haciendo es darle permisos al archivo para que me permita ejecutar el archivo que estoy creando, ya que predeterminadamente no es posible .

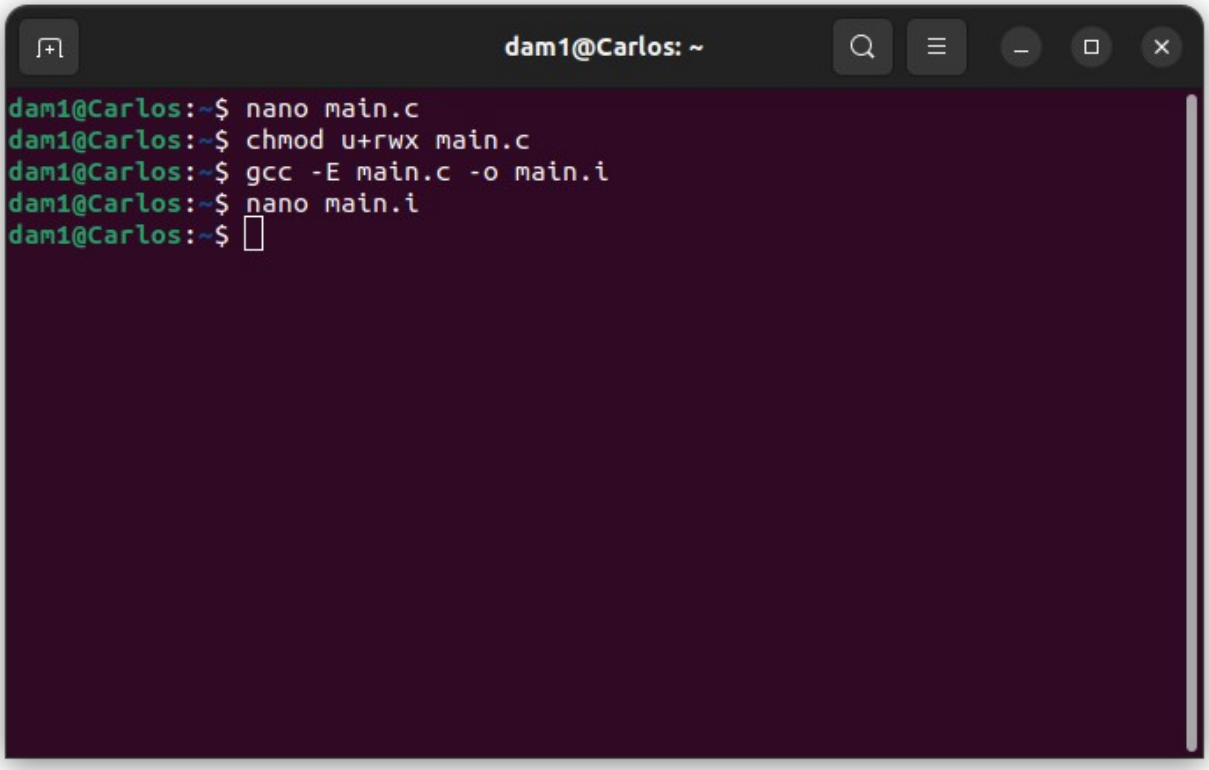
2.3.Preprocesado

Ahora que tienes el archivo fuente, el primer paso del proceso de compilación es el **preprocesado**. El preprocesador en C maneja directivas como `#include` y `#define`. Para realizar este paso, utiliza el siguiente comando en la terminal:

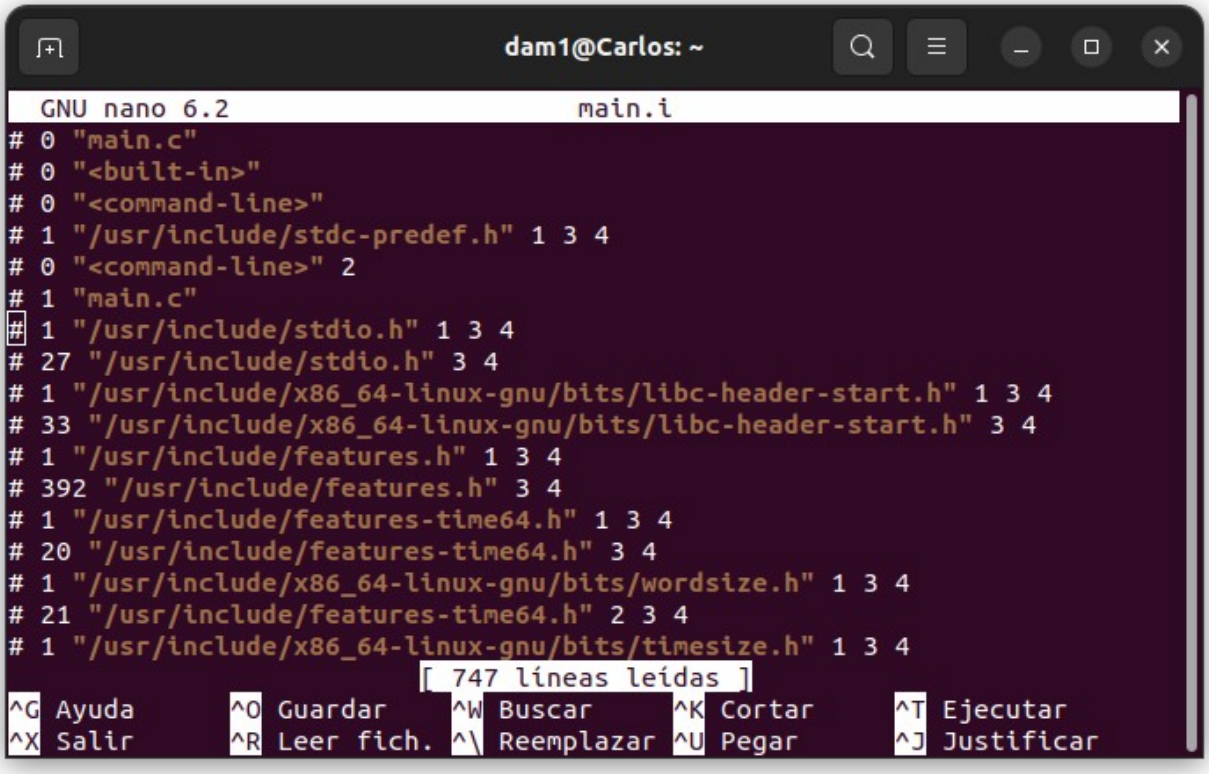
```
gcc -E main.c -o main.i

Este comando genera el archivo main.i, que contiene el código
preprocesado. Puedes abrirlo con el editor nano para ver cómo las macros y
las cabeceras han sido reemplazadas:

nano main.i
```



Mediante el comando **gcc -E main.c -o main.i**, estoy generando un nuevo archivo main.i con el cógido del anterior archivo preprocesado antes de pasar a la compilación.



2.4.Compilación

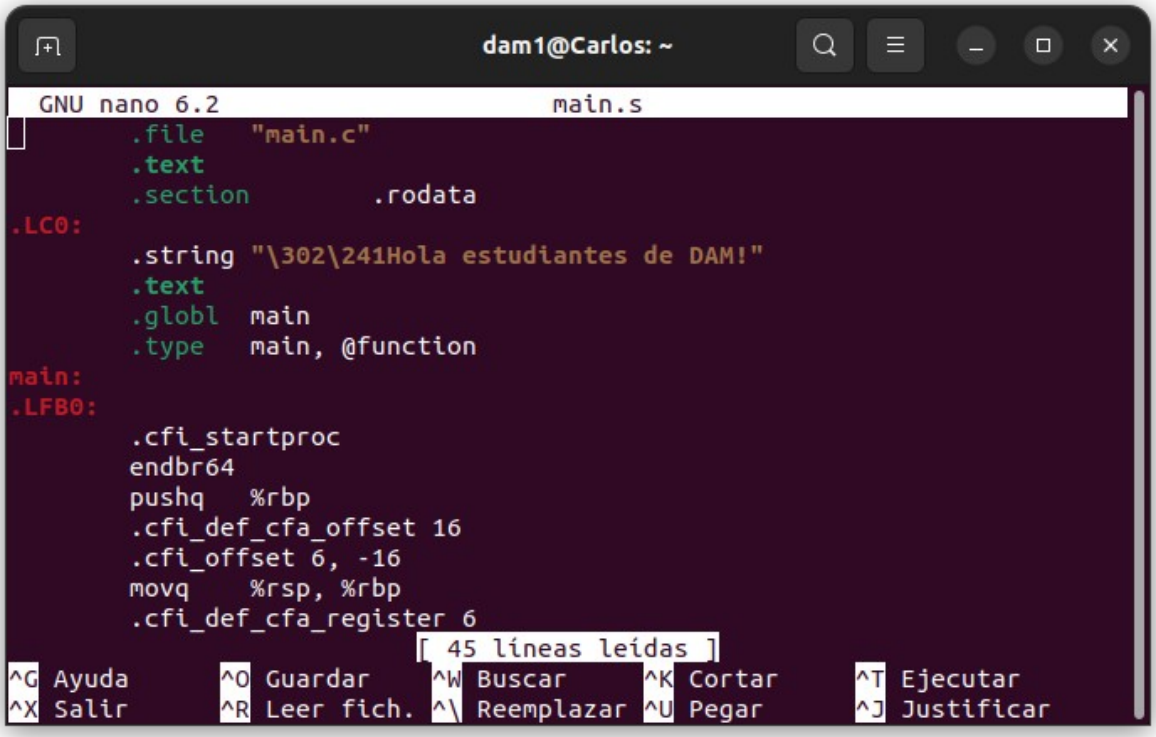
El siguiente paso es compilar el código preprocesado para convertirlo en código ensamblador. Ejecuta el siguiente comando para generar un archivo llamado `main.s`:

```
gcc -S main.i -o main.s
```

Abre el archivo ensamblador con `nano` para ver el resultado:

```
nano main.s
```

En este archivo podrás observar instrucciones en lenguaje ensamblador, que son más cercanas a las instrucciones que entiende el procesador.



En este paso, convierto el archivo main.i en un archivo con lenguaje ensamblador que esta contenido dentro de un nuevo archivo llamado main.s. Este lenguaje permite al ordenador comprender el contenido del archivo.

2.5.Ensamblado

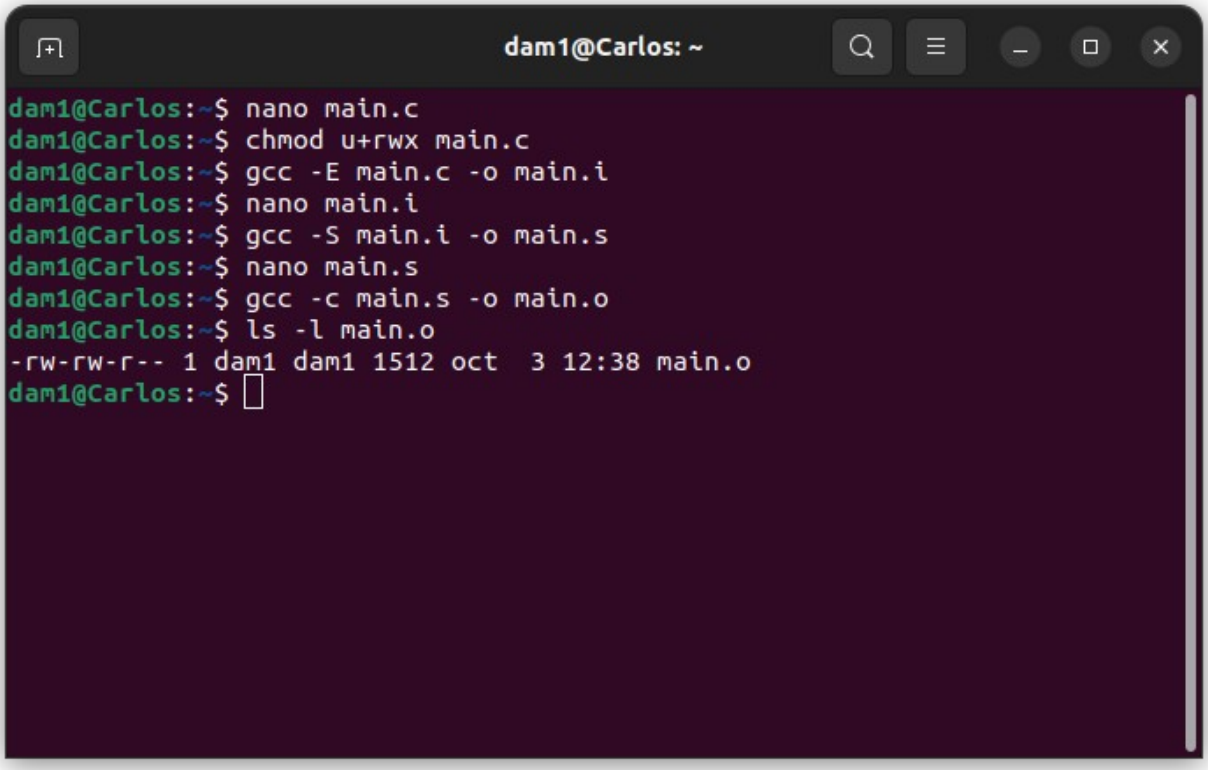
El código ensamblador ahora debe ser traducido a un archivo objeto (binario intermedio) que el sistema pueda manejar. Usa el siguiente comando para generar el archivo `main.o`:

```
gcc -c main.s -o main.o
```

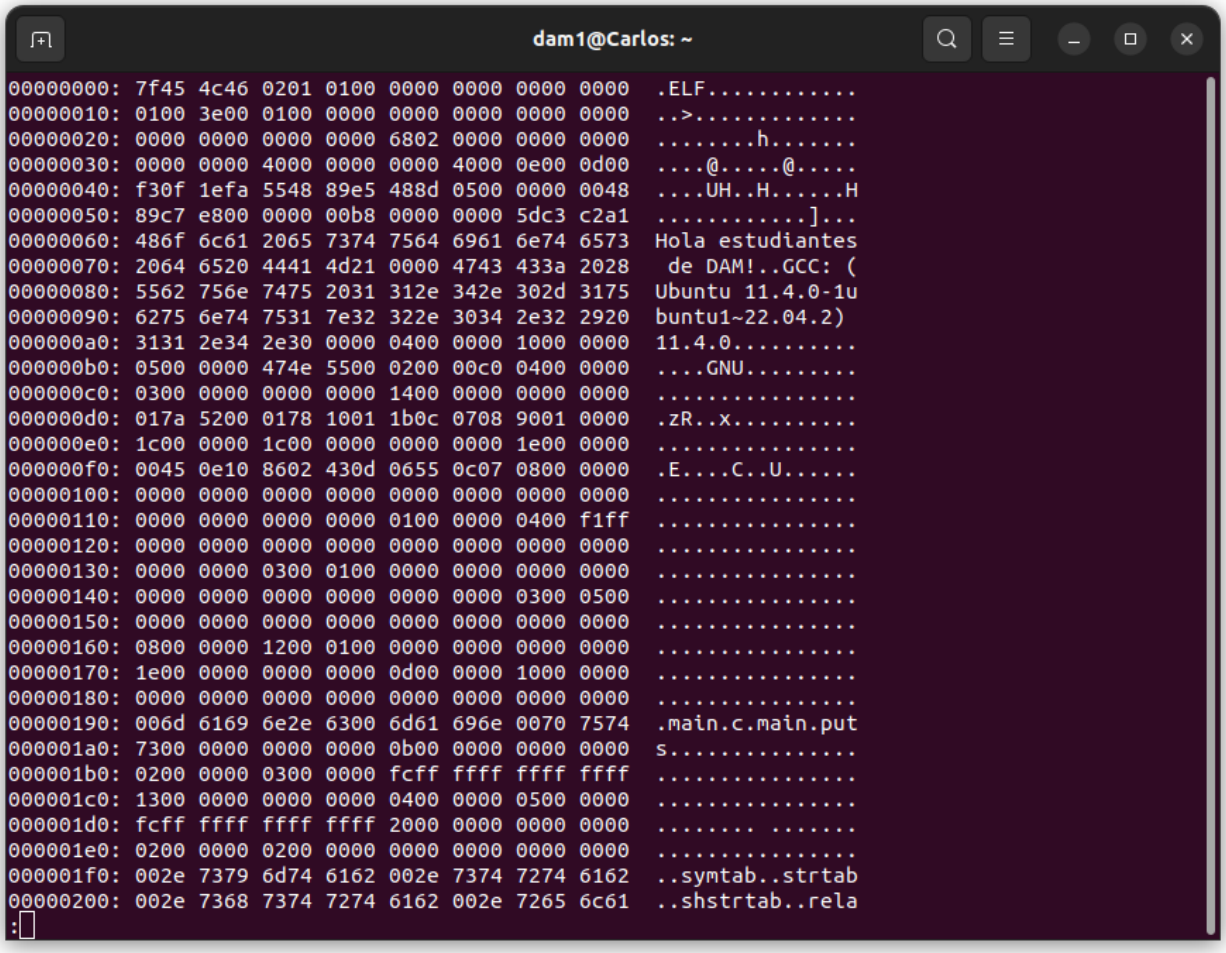
El archivo `main.o` es un archivo objeto que contiene el código en formato binario. Aunque no es necesario abrir este archivo, puedes verificar que se ha generado correctamente utilizando el siguiente comando:

```
ls -l main.o
```

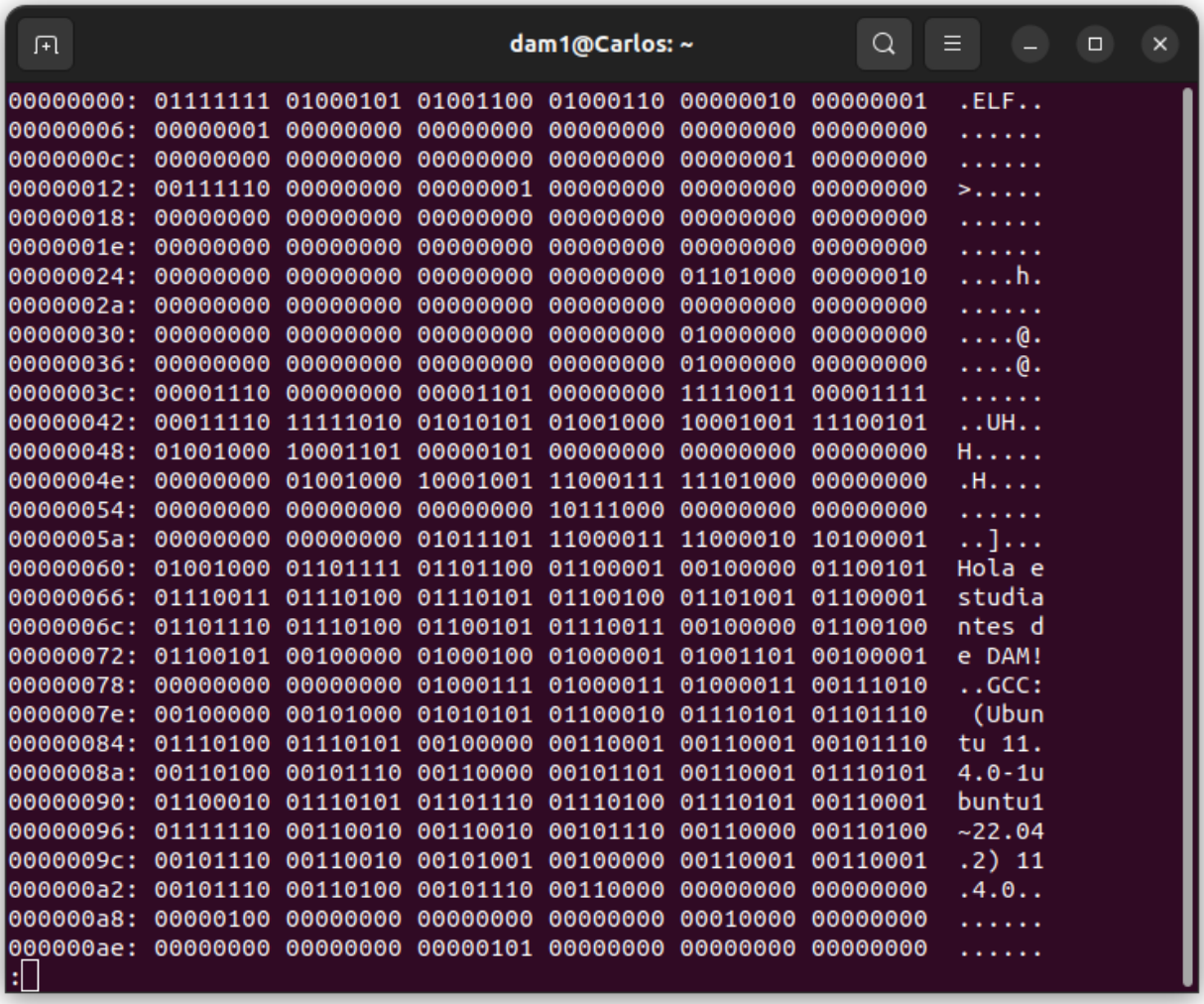
También puedes investigar cómo mostrar el contenido en binario.



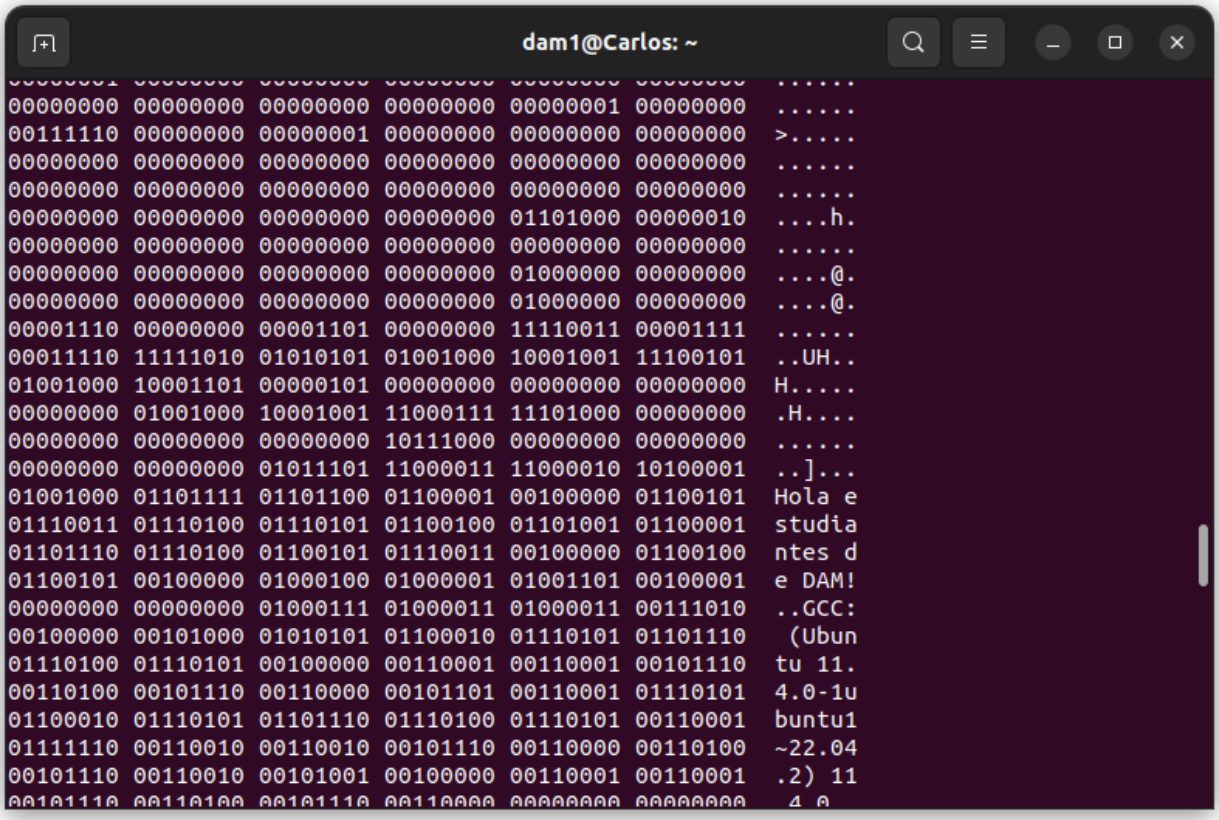
HEXADECIMAL (xxd main.o | less)



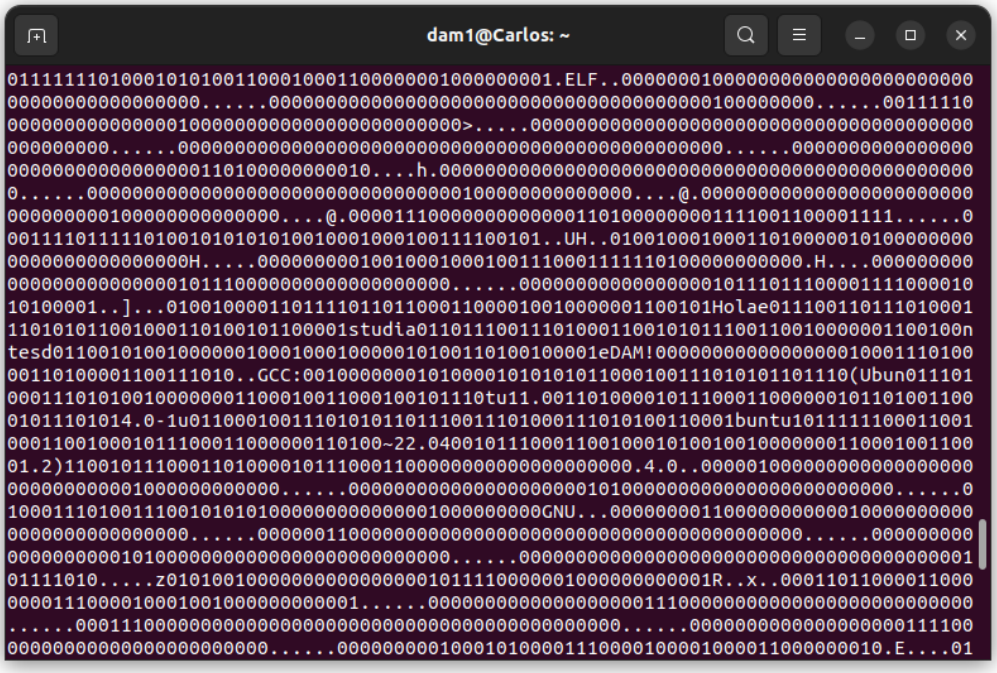
BINARIO (xxd -b main.o | less)



(xxd -b main.o | cut -d" " -f2-)



O (xxd -b main.o | cut -d" " -f2- | tr -d ' \n'



2.6.Enlazado

El último paso es enlazar el archivo objeto con las bibliotecas necesarias para crear el ejecutable. Usa el siguiente comando:

```
gcc main.o -o main
```

Esto generará el ejecutable `main`. Sin embargo, si no puedes ejecutarlo directamente, deberás asegurarte de que tiene los permisos correctos. Cambia los permisos del ejecutable utilizando:

```
chmod u+x main
```

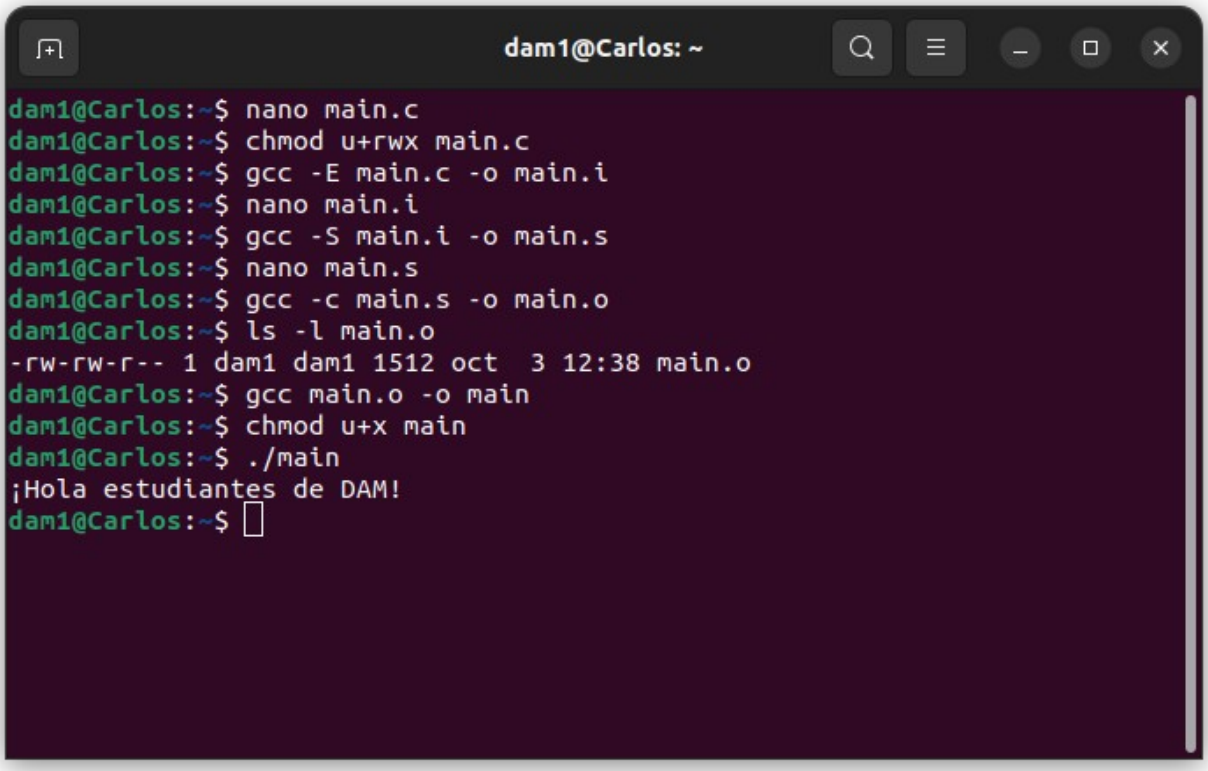
Por último, el archivo binario main.o lo paso o genero un ejecutable capaz de realizar la acción que necesito (en este caso, que muestre por la pantalla ¡Hola estudiantes de DAM!.

2.7.Ejecutar el programa

Una vez generado el ejecutable, puedes correr el programa usando el siguiente comando:

```
./main
```

Si todo ha salido bien, verás que el programa imprime el mensaje **Hello, World!** en la terminal.


dam1@Carlos: ~
dam1@Carlos:~$ nano main.c
dam1@Carlos:~$ chmod u+rwx main.c
dam1@Carlos:~$ gcc -E main.c -o main.i
dam1@Carlos:~$ nano main.i
dam1@Carlos:~$ gcc -S main.i -o main.s
dam1@Carlos:~$ nano main.s
dam1@Carlos:~$ gcc -c main.s -o main.o
dam1@Carlos:~$ ls -l main.o
-rw-rw-r-- 1 dam1 dam1 1512 oct 3 12:38 main.o
dam1@Carlos:~$ gcc main.o -o main
dam1@Carlos:~$ chmod u+x main
dam1@Carlos:~$ ./main
¡Hola estudiantes de DAM!
dam1@Carlos:~$