

=== RESPUESTA 1 ===

Te puedo explicar **paso a paso** cómo tu código en C termina convirtiéndose en un binario ejecutable. En C, hay **varios pasos**, cada uno con su función:

1. Preprocesamiento

- Programa involucrado: **Preprocesador** (`cpp`).
- Qué hace:
- Sustituye macros definidas con `#define`.
- Incluye los archivos de cabecera con `#include`.
- Evalúa condiciones de compilación `#if`, `#ifdef`, etc.
- Resultado: un archivo C **expandido** listo para compilar.

Ejemplo:

```
#define PI 3.14
#include <stdio.h>
```

Se convierte en un archivo donde `PI` se reemplaza por `3.14` y se copia el contenido de `stdio.h`.

2. Compilación

- Programa involucrado: **Compilador** (por ejemplo `gcc`).
 - Qué hace:
 - Traduce el código C **preprocesado** a **código ensamblador**.
 - Verifica errores de sintaxis y tipos.
 - Resultado: un archivo `.s` (ensamblador).
-

3. Ensamblado

- Programa involucrado: **Ensamblador** (`as`).
 - Qué hace:
 - Convierte el código ensamblador en **código máquina** (instrucciones binarios entendibles por la CPU).
 - Resultado: un archivo **objeto** `.o`.
-

4. Enlazado (Linking)

- Programa involucrado: **Linker** (`ld`).
- Qué hace:

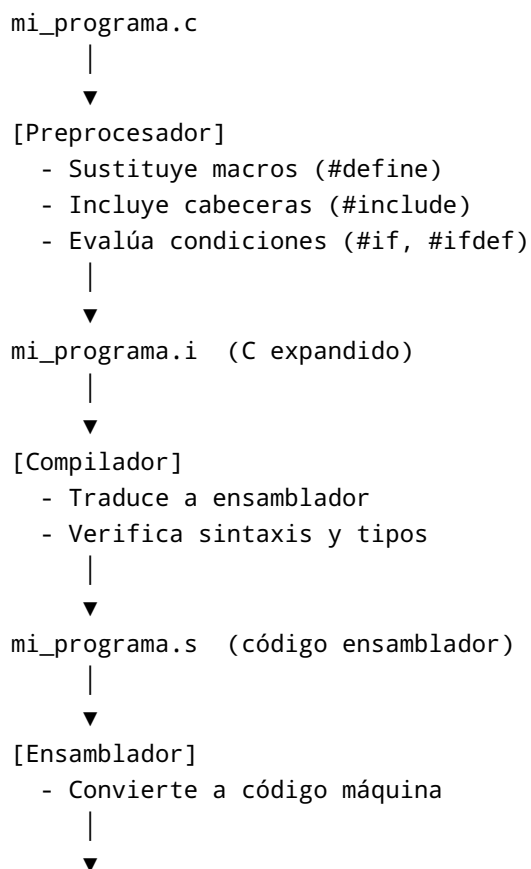
- Combina los archivos objeto `.o` con **librerías** necesarias (como `libc`).
- Resuelve referencias a funciones externas (por ejemplo, `printf`).
- Resultado: un **ejecutable binario** listo para correr.

Resumen de Archivos Intermedios

Paso	Entrada	Salida
Preprocesamiento	<code>.c</code>	<code>.i</code> o C expandido
Compilación	<code>.i</code>	<code>.s</code> (ensamblador)
Ensamblado	<code>.s</code>	<code>.o</code> (objeto)
Enlazado	<code>.o</code> + libs	Ejecutable (<code>a.out</code>)

=== RESPUESTA 2 ===

Diagrama visual del flujo completo desde tu código en C hasta el binario:



```
mi_programa.o  (archivo objeto)
  |
  ▼
[Linker / Enlazador]
- Combina archivos objeto
- Añade librerías (libc, etc.)
- Resuelve referencias externas
  |
  ▼
ejecutable    (por ejemplo: a.out)
```



Tip extra:

Con `gcc -save-temps mi_programa.c` puedes ver todos los archivos intermedios (`.i`, `.s`, `.o`) para entender cada etapa.