
6

Fases de la Traducción y Errores

6.1. Objetivos

- Identificar las fases de traducción y errores.

6.2. Temas

- Fases de traducción.
- Preprocesamiento.
- Compilación.
- Ensamblado.
- Vinculación (Link).
- Errores en cada fase.

6.3. Tareas

1. Investigar las funcionalidades y opciones que su compilador presenta para limitar el inicio y fin de las fases de traducción.
2. Para la siguiente secuencia de pasos:
 - a. Transcribir en `readme.md` cada **comando ejecutado** y
 - b. Describir en `readme.md` el **resultado** u **error** obtenidos para cada paso.

6.3.1. Secuencia de Pasos

1. Escribir `hello2.c`, que es una variante de `hello.c`:

```
#include <stdio.h>

int/*medio*/main(void){
    int i=42;
    printf("La respuesta es %d\n");
```

2. Preprocesar hello2.c, no compilar, y generar hello2.i. Analizar su contenido.
3. Escribir hello3.c, una nueva variante:

```
int printf(const char *s, ...);

int main(void){
    int i=42;
    printf("La respuesta es %d\n");
```

4. Investigar la semántica de la primera línea.
5. Preprocesar hello3.c, no compilar, y generar hello3.i. Buscar diferencias entre hello3.c y hello3.i.
6. Compilar el resultado y generar hello3.s, no ensamblar.
7. Corregir en el nuevo archivo hello4.c y empezar de nuevo, generar hello4.s, no ensamblar.
8. Investigar hello4.s.
9. Ensamblar hello4.s en hello4.o, no vincular.
- 10.Vincular hello4.o con la biblioteca estándar y generar el ejecutable.
- 11.Corregir en hello5.c y generar el ejecutable.
- 12Ejecutar y analizar el resultado.
- 13Corregir en hello6.c y empezar de nuevo.
- 14Escribir hello7.c, una nueva variante:

```
int main(void){
    int i=42;
    printf("La respuesta es %d\n", i);
}
```

15Explicar porqué funciona.

6.4. Restricciones

- El programa ejemplo debe enviar por stdout la frase La respuesta es 42, el valor 42 debe surgir de una variable.

6.5. Productos

```
DD-FasesErrores
|-- readme.md
|-- hello2.c
|-- hello3.c
|-- hello4.c
|-- hello5.c
|-- hello6.c
`-- hello7.c
```