

Informática II

Programación en C bajo GNU/Linux

Gonzalo F. Perez Paina



Universidad Tecnológica Nacional
Facultad Regional Córdoba
UTN-FRC

– 2024 –

Programar en lenguaje C

```
1  /* Primer programa en C */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hola mundo.\n");
7      return 0;
8  }
```



Programar en lenguaje C

```
1  /* Primer programa en C */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hola mundo.\n");
7      return 0;
8  }
```

► ¿Qué representa el texto de arriba? 🖱️

Programar en lenguaje C



```
1  /* Primer programa en C */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hola mundo.\n");
7      return 0;
8  }
```

- ▶ ¿Qué representa el texto de arriba? 
- ▶ ¿Qué hay que hacer para obtener lo sig. en la terminal? 

```
Hola mundo.
```

Programar en lenguaje C

```
1  /* Primer programa en C */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hola mundo.\n");
7      return 0;
8  }
```

- ▶ ¿Qué representa el texto de arriba? 
- ▶ ¿Qué hay que hacer para obtener lo sig. en la terminal? 

```
Hola mundo.
```

- ▶ ¿Qué herramienta(s) utilizan?

IDE – Integrated Development Environment

Entorno de desarrollo integrado

Incluye un conjunto de herramientas informáticas para facilitar el desarrollo de software.

IDE – Integrated Development Environment

Entorno de desarrollo integrado

Incluye un conjunto de herramientas informáticas para facilitar el desarrollo de software. Normalmente consiste en:

IDE – Integrated Development Environment

Entorno de desarrollo integrado

Incluye un conjunto de herramientas informáticas para facilitar el desarrollo de software. Normalmente consiste en:

1. Editor de código fuente

IDE – Integrated Development Environment

Entorno de desarrollo integrado

Incluye un conjunto de herramientas informáticas para facilitar el desarrollo de software. Normalmente consiste en:

1. Editor de código fuente
2. Herramientas de construcción

IDE – Integrated Development Environment

Entorno de desarrollo integrado

Incluye un conjunto de herramientas informáticas para facilitar el desarrollo de software. Normalmente consiste en:

1. Editor de código fuente
2. Herramientas de construcción
3. Depurador (debugger)

IDE – Integrated Development Environment

Entorno de desarrollo integrado

Incluye un conjunto de herramientas informáticas para facilitar el desarrollo de software. Normalmente consiste en:

1. Editor de código fuente
2. Herramientas de construcción
3. Depurador (debugger)

Algunas características de los IDE

IDE – Integrated Development Environment

Entorno de desarrollo integrado

Incluye un conjunto de herramientas informáticas para facilitar el desarrollo de software. Normalmente consiste en:

1. Editor de código fuente
2. Herramientas de construcción
3. Depurador (debugger)

Algunas características de los IDE

- Resaltado de sintaxis

IDE – Integrated Development Environment

Entorno de desarrollo integrado

Incluye un conjunto de herramientas informáticas para facilitar el desarrollo de software. Normalmente consiste en:

1. Editor de código fuente
2. Herramientas de construcción
3. Depurador (debugger)

Algunas características de los IDE

- ▶ Resaltado de sintaxis
- ▶ Indentado automático

IDE – Integrated Development Environment

Entorno de desarrollo integrado

Incluye un conjunto de herramientas informáticas para facilitar el desarrollo de software. Normalmente consiste en:

1. Editor de código fuente
2. Herramientas de construcción
3. Depurador (debugger)

Algunas características de los IDE

- ▶ Resaltado de sintaxis
- ▶ Indentado automático
- ▶ Plegado de código

IDE – Integrated Development Environment

Entorno de desarrollo integrado

Incluye un conjunto de herramientas informáticas para facilitar el desarrollo de software. Normalmente consiste en:

1. Editor de código fuente
2. Herramientas de construcción
3. Depurador (debugger)

Algunas características de los IDE

- ▶ Resaltado de sintaxis
- ▶ Indentado automático
- ▶ Plegado de código
- ▶ Autocompletado

IDE – Integrated Development Environment

Entorno de desarrollo integrado

Incluye un conjunto de herramientas informáticas para facilitar el desarrollo de software. Normalmente consiste en:

1. Editor de código fuente
2. Herramientas de construcción
3. Depurador (debugger)

Algunas características de los IDE

- ▶ Resaltado de sintaxis
- ▶ Indentado automático
- ▶ Plegado de código
- ▶ Autocompletado
- ▶ Administración de proyecto

IDE – Integrated Development Environment

Entorno de desarrollo integrado

Incluye un conjunto de herramientas informáticas para facilitar el desarrollo de software. Normalmente consiste en:

1. Editor de código fuente
2. Herramientas de construcción
3. Depurador (debugger)

Algunas características de los IDE

- ▶ Resaltado de sintaxis
- ▶ Indentado automático
- ▶ Plegado de código
- ▶ Autocompletado
- ▶ Administración de proyecto
- ▶ Terminal embebida

IDE – Integrated Development Environment

Entorno de desarrollo integrado

Incluye un conjunto de herramientas informáticas para facilitar el desarrollo de software. Normalmente consiste en:

1. Editor de código fuente
2. Herramientas de construcción
3. Depurador (debugger)

Algunas características de los IDE

- ▶ Resaltado de sintaxis
- ▶ Indentado automático
- ▶ Plegado de código
- ▶ Autocompletado
- ▶ Administración de proyecto
- ▶ Terminal embebida
- ▶ Etc.

IDE – Integrated Development Environment

Entorno de desarrollo integrado

Incluye un conjunto de herramientas informáticas para facilitar el desarrollo de software. Normalmente consiste en:

1. Editor de código fuente
2. Herramientas de construcción
3. Depurador (debugger)

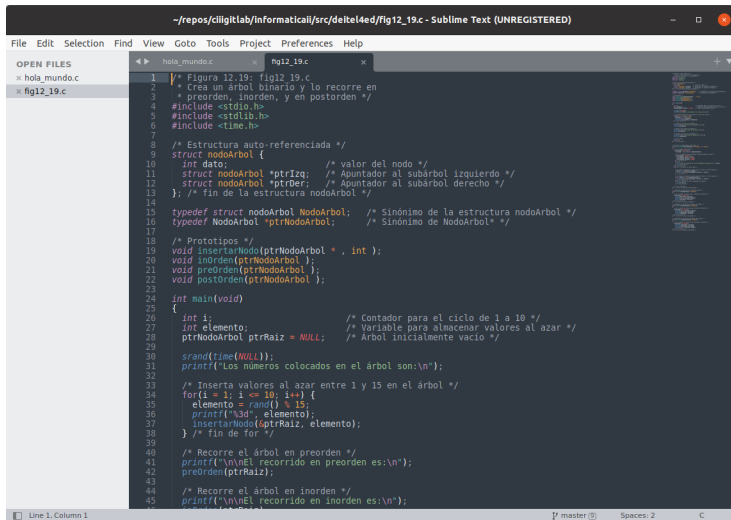
Algunas características de los IDE

- ▶ Resaltado de sintaxis
- ▶ Indentado automático
- ▶ Plegado de código
- ▶ Autocompletado
- ▶ Administración de proyecto
- ▶ Terminal embebida
- ▶ Etc.

¿Cuáles conocen?

IDE – Integrated Development Environment

Sublime Text



```
~/repos/ciiigitlab/informaticaii/src/deitel4ed/fig12_19.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

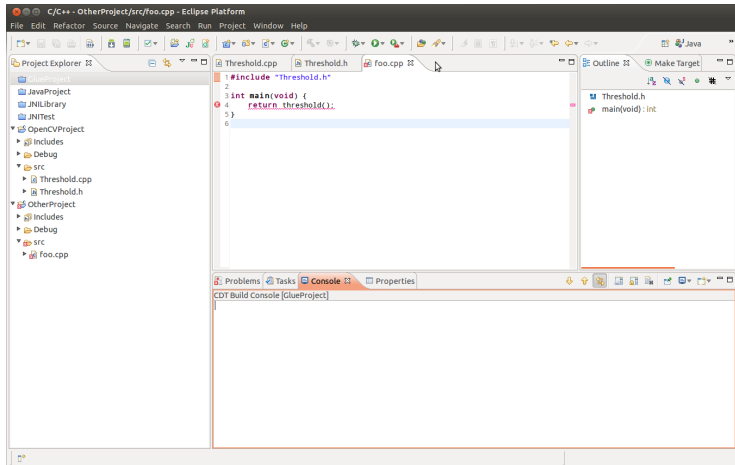
OPEN FILES
x hola_mundo.c
x fig12_19.c

1  /* Figura 12.19: fig12_19.c
2  * Crea un árbol binario y lo recorre en
3  * preorden, inorden, y en postorden */
4  #include <stdio.h>
5  #include <stdlib.h>
6  #include <time.h>
7
8  /* Estructura auto-referenciada */
9  struct nodoArbol {
10     int dato;           /* valor del nodo */
11     struct nodoArbol *ptrIzq; /* Apuntador al subárbol izquierdo */
12     struct nodoArbol *ptrDer; /* Apuntador al subárbol derecho */
13 }; /* fin de la estructura nodoArbol */
14
15 typedef struct nodoArbol NodoArbol; /* Sinónimo de la estructura nodoArbol */
16 typedef NodoArbol *ptrNodoArbol;   /* Sinónimo de NodoArbol* */
17
18 /* Prototipos */
19 void insertarNodo(ptrNodoArbol *, int );
20 void inOrden(ptrNodoArbol );
21 void preOrden(ptrNodoArbol );
22 void postOrden(ptrNodoArbol );
23
24 int main(void)
25 {
26     int i;               /* Contador para el ciclo de 1 a 10 */
27     int elemento;        /* Variable para almacenar valores al azar */
28     ptrNodoArbol ptrRaiz = NULL; /* Árbol inicialmente vacío */
29
30     srand(time(NULL));
31     printf("Los números colocados en el árbol son:\n");
32
33     /* Inserta valores al azar entre 1 y 15 en el árbol */
34     for(i = 1; i <= 10; i++) {
35         elemento = rand() % 15;
36         printf("%3d", elemento);
37         insertarNodo(&ptrRaiz, elemento);
38     } /* fin de for */
39
40     /* Recorre el árbol en preorden */
41     printf("\n\nEl recorrido en preorden es:\n");
42     preOrden(ptrRaiz);
43
44     /* Recorre el árbol en inorden */
45     printf("\n\nEl recorrido en inorden es:\n");
46     inOrden(ptrRaiz);
47 }
```

Line 1, Column 1 master Spaces: 2 C

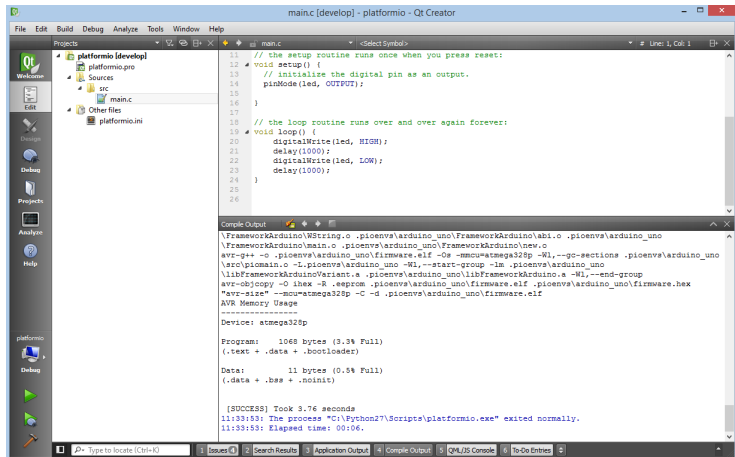
IDE – Integrated Development Environment

Eclipse CDT

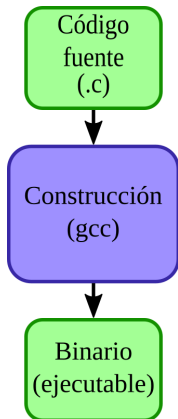


IDE – Integrated Development Environment

Qt Creator

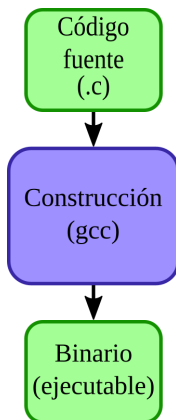


Construcción de un programa



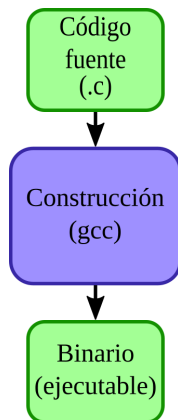
Construcción de un programa

1. Escribir el código fuente



```
1  /* Primer programa en C */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hola mundo.\n");
7      return 0;
8  }
```

Construcción de un programa

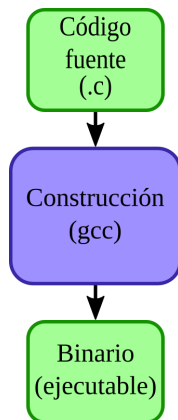


1. Escribir el código fuente

```
1  /* Primer programa en C */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hola mundo.\n");
7      return 0;
8  }
```

2. Guardar con extensión .c (nombre sin espacios), p.e. hola.c

Construcción de un programa



1. Escribir el código fuente

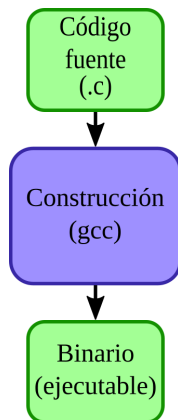
```
1  /* Primer programa en C */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hola mundo.\n");
7      return 0;
8  }
```

2. Guardar con extensión .c (nombre sin espacios), p.e. hola.c

3. Construir el programa

```
> gcc hola.c
```

Construcción de un programa



1. Escribir el código fuente

```
1 /* Primer programa en C */
2 #include <stdio.h>
3
4 int main(void)
5 {
6     printf("Hola mundo.\n");
7     return 0;
8 }
```

2. Guardar con extensión .c (nombre sin espacios), p.e. hola.c

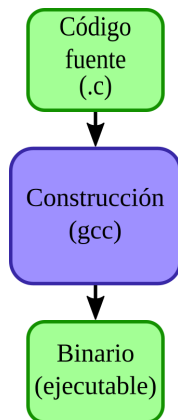
3. Construir el programa

```
> gcc hola.c
```

4. Ejecutar el programa

```
> ./a.out
Hola mundo.
```

Construcción de un programa



1. Escribir el código fuente

```
1 /* Primer programa en C */  
2 #include <stdio.h>  
3  
4 int main(void)  
5 {  
6     printf("Hola mundo.\n");  
7     return 0;  
8 }
```

2. Guardar con extensión .c (nombre sin espacios), p.e. hola.c

3. Construir el programa

```
> gcc hola.c
```

4. Ejecutar el programa

```
> ./a.out  
Hola mundo.
```

En la cátedra se utilizará este procedimiento

Introducción a gcc

hola.c

```
1  /* Primer programa en C */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hola mundo.\n");
7      return 0;
8  }
```

Introducción a gcc

hola.c

```
1  /* Primer programa en C */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hola mundo.\n");
7      return 0;
8  }
```

Compilación

```
> gcc hola.c
```

Introducción a gcc

hola.c

```
1  /* Primer programa en C */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hola mundo.\n");
7      return 0;
8  }
```

Compilación

```
> gcc hola.c
```

(salida a.out)

Introducción a gcc

hola.c

```
1 /* Primer programa en C */  
2 #include <stdio.h>  
3  
4 int main(void)  
5 {  
6     printf("Hola mundo.\n");  
7     return 0;  
8 }
```

Compilación

```
> gcc hola.c
```

(salida a.out)

Cómo cambiar el nombre a binario?

Introducción a gcc

hola.c

```
1 /* Primer programa en C */
2 #include <stdio.h>
3
4 int main(void)
5 {
6     printf("Hola mundo.\n");
7     return 0;
8 }
```

Compilación

```
> gcc hola.c
```

(salida a.out)

Cómo cambiar el nombre a binario?

```
> gcc hola.c -o hola
```

Introducción a gcc

hola.c

```
1  /* Primer programa en C */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hola mundo.\n");
7      return 0;
8  }
```

Compilación

```
> gcc hola.c
```

(salida a.out)

Cómo cambiar el nombre a binario?

```
> gcc hola.c -o hola
```

```
> gcc -Wall hola.c -o hola
```

Introducción a gcc

hola.c

```
1  /* Primer programa en C */
2  #include <stdio.h>
3
4  int main(void)
5  {
6      printf("Hola mundo.\n");
7      return 0;
8  }
```

Compilación

```
> gcc hola.c
```

(salida a.out)

Cómo cambiar el nombre a binario?

```
> gcc hola.c -o hola
```

```
> gcc -Wall hola.c -o hola
```

(habilita todas las advertencias/warnings)

Introducción a gcc

- ▶ Si el código fuente presenta errores de sintaxis el compilador (C o C++) reportará los errores y no producirá el archivo binario de salida.

Introducción a gcc

- ▶ Si el código fuente presenta errores de sintaxis el compilador (C o C++) reportará los errores y no producirá el archivo binario de salida.
- ▶ Los compiladores tiene la capacidad de advertirnos sobre situaciones inesperadas en el código fuente.

Introducción a gcc

- ▶ Si el código fuente presenta errores de sintaxis el compilador (C o C++) reportará los errores y no producirá el archivo binario de salida.
- ▶ Los compiladores tiene la capacidad de advertirnos sobre situaciones inesperadas en el código fuente.
- ▶ Se puede modificar este comportamiento utilizando *flags* de compilación.

Introducción a gcc

- ▶ Si el código fuente presenta errores de sintaxis el compilador (C o C++) reportará los errores y no producirá el archivo binario de salida.
- ▶ Los compiladores tiene la capacidad de advertirnos sobre situaciones inesperadas en el código fuente.
- ▶ Se puede modificar este comportamiento utilizando *flags* de compilación.
- ▶ El compilador puede considerarse como una herramienta útil para verificar el código fuente y aprender.

Introducción a gcc

- ▶ Si el código fuente presenta errores de sintaxis el compilador (C o C++) reportará los errores y no producirá el archivo binario de salida.
- ▶ Los compiladores tiene la capacidad de advertirnos sobre situaciones inesperadas en el código fuente.
- ▶ Se puede modificar este comportamiento utilizando *flags* de compilación.
- ▶ El compilador puede considerarse como una herramienta útil para verificar el código fuente y aprender.
- ▶ Los *flags* (banderas) del compilador permiten cambiar o alterar el comportamiento del compilador.

Ejemplo de configuración del IDE Geany

