

ICF 122 - Fundamentos de Programación

Ingeniería Civil Informática

Unidad 1: Introducción

Facultad de Ingeniería



**Universidad
Andrés Bello**

¿Qué es un lenguaje de programación?



- Independencia con la arquitectura del computador.
- Instrucciones en lenguaje de máquina.
- Sentencias semejantes al lenguaje matemático o natural.

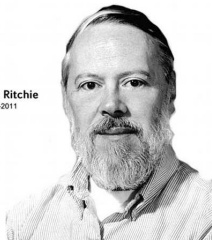
¿Qué lenguaje de programación veremos?

Introducción

Lenguaje de Programación C

- El **lenguaje de programación C** se diseñó en 1969.
- El lenguaje, su sintaxis y su semántica, así como el primer compilador de C fueron diseñados y creados por Dennis M. Ritchie, en los laboratorios Bell.
- En 1983, se definió el estándar **ANSI C**.
- El lenguaje tiene muy pocas reglas sintácticas, sencillas de aprender. Su léxico es muy reducido: tan solo **32 palabras**.

Dennis Ritchie
1941-2011



Introducción

Lenguaje de Programación C

- Es un lenguaje del paradigma imperativo, estructurado.
- Permite la programación modular.
- A menudo se le considera un lenguaje de medio y/o bajo nivel.
- Se utiliza mucho en la comunidad científica, en el desarrollo de software de sistemas y en aplicaciones.



Introducción

Máquina de Von Neumann

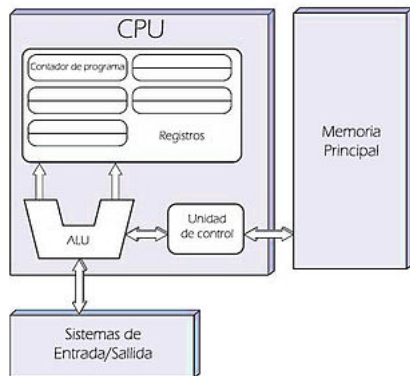
- John Von Neumann: científico y matemático húngaro - estadounidense.
- Se le atribuye “erróneamente” la creación de la arquitectura de computadores, que utiliza el mismo dispositivo tanto para las instrucciones como para los datos.
- Esta arquitectura originalmente fue diseñada por John Eckert, John Mauchly y otros.
- John Von Neumann en un informe de su propiedad describe esta arquitectura, creando la confusión.



Introducción

Máquina de Von Neumann

- CPU: unidad central de proceso (procesador).
- UC: la unidad de control.
- Memoria.
- Dispositivo de entrada y salida.
- Buses de interconexión.



Propósito de la programación en una frase:

Que el computador haga el trabajo por nosotros

- Los computadores son buenos para hacer tareas rutinarias.
- Idealmente, cualquier problema tedioso y repetitivo debería ser resuelto por un computador.
- Los seres humanos deberíamos dedicarnos solamente a los problemas más interesantes: los que requieren creatividad, pensamiento crítico y subjetividad.
- La programación es el proceso de transformar un método para resolver problemas en uno que pueda ser entendido por el computador.

- Diseño de un programa: crear y describir un procedimiento bien definido, que efectivamente resuelva el problema.
- El objeto de estudio de **la programación** no son los programas, sino **los algoritmos**.
- Un **algoritmo** es un procedimiento bien definido para resolver un problema.
- ¿Utilizamos algoritmos a diario?
 - Una receta de cocina es un algoritmo: la entrada son los ingredientes y otros datos como la cantidad de personas, el proceso es la serie de pasos para manipular los ingredientes y la salida es el plato terminado.
 - El método de multiplicación, aprendido en el colegio, es un algoritmo: la entrada son los dos factores, el proceso es la secuencia de pasos en donde los dígitos van siendo multiplicados, las reservas van siendo sumadas, y los productos intermedios son finalmente sumados, la salida del algoritmo es el producto obtenido.

Definición 1

*Un **algoritmo** es una secuencia de pasos para resolver un problema. Los pasos deben estar muy bien definidos, y se tienen que describir sin ambigüedades, de cómo llegar desde el inicio hasta el final.*

Componentes de un algoritmo:

- **entrada:** son los datos sobre los que el algoritmo opera.
- **proceso:** son los pasos que hay que seguir, utilizando la entrada. Es una secuencia de sentencias, que debe ser realizada en orden. También puede tener ciclos (grupos de sentencias que son ejecutadas varias veces) y condicionales (grupos de sentencias que sólo son ejecutadas bajo ciertas condiciones).
- **salida:** es el resultado que entrega el algoritmo.

Ejemplo 1

Algoritmo para resolver ecuaciones cuadráticas:

- $ax^2 + bx + c = 0$, con $a \neq 0$ y $a, b, c \in \mathbb{R}$.
- Sea $2x^2 - 5x + 2 = 0 \implies a = 2, b = -5, c = 2$. Soluciones: $x_1 = \frac{1}{2}$ y $x_2 = 2$ ¿Cómo obtenemos estos valores?
- *Problema computacional: dados a, b y c , encontrar los valores de x se satisfacen la ecuación $ax^2 + bx + c = 0$.*
- **entrada:** los valores a, b y c .
- **salida:** las raíces reales x .

Introducción

Algoritmos / Lenguaje natural

- Durante el proceso mental de diseñar un algoritmo, es común pensar y describir los pasos en la misma manera en que hablamos a diario.

Teniendo los valores de a , b y c , calcular el discriminante $\Delta = b^2 - 4ac$. Si el discriminante es negativo, entonces la ecuación no tiene soluciones reales.

Si el discriminante es igual a cero, entonces la ecuación tiene una única solución real, que es $x = -b/2a$.

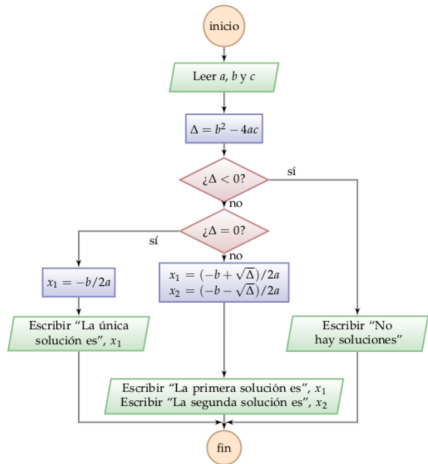
Si el discriminante es positivo, entonces la ecuación tiene dos soluciones reales, que son $x_1 = \sqrt{(-b - \Delta)/2a}$ y $x_2 = \sqrt{(-b + \Delta)/2a}$.

- **impreciso:** puede tener ambigüedades.
- **no universal:** personas distintas describirán el proceso de maneras distintas.
- **no estructurado:** la descripción no está expresada en función de componentes simples.

Introducción

Algoritmos / Diagrama de Flujo

Un **diagrama de flujo** es una representación gráfica de un algoritmo. Los pasos son representados por varios tipos de bloques, y el flujo de ejecución es indicado por flechas que conectan los bloques.



El **pseudocódigo** es una descripción estructurada de un algoritmo basada en ciertas convenciones notacionales. Si bien es muy parecido al código que finalmente se escribirá en el computador, el pseudocódigo está pensado para ser leído por humanos.

leer a

leer b

leer c

$\text{discriminante} = b^2 - 4ac$

si $\text{discriminante} < 0$:

escribir "La ecuación no tiene soluciones reales"

o si no, si $\text{discriminante} = 0$:

$x = -b/2a$

escribir "La solución única es", x

o si no:

$x_1 = (-b - \sqrt{\text{discriminante}})/2a$

$x_2 = (-b + \sqrt{\text{discriminante}})/2a$

escribir "La primera solución real es:", x_1

escribir "La segunda solución real es:", x_2

- El producto final de la programación siempre debe ser un **código** que pueda ser ejecutado en el computador, es decir, describir los algoritmos en un **lenguaje de programación**.
- Cualquier lenguaje de programación consiste en conjunto limitado de conceptos básicos, en función de los cuales uno puede expresar cualquier algoritmo.
- Existen otros lenguajes que Ud. aprenderá en el transcurso de su carrera, los cuales serán utilizados dependiendo del problema al cual se enfrente y el tipo de solución que requiera.

Introducción

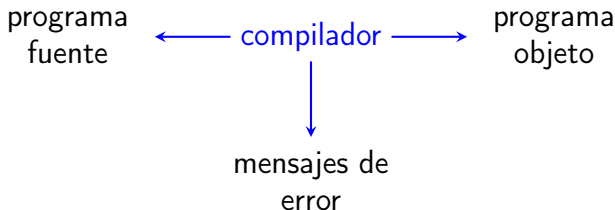
Algoritmos / Código

```
int main (){
    float a,b,c,delta,x1,x2;
    printf("Ingresa a:\n");
    scanf("%f",&a);
    printf("Ingresa b:\n");
    scanf("%f",&b);
    printf("Ingresa c:\n");
    scanf("%f",&c);
    delta = b*b - 4*a*c;
    if(delta < 0)
        printf("La ecuacion no tiene soluciones reales\n");
    else{
        if(delta == 0){
            x1 = -b/(2*a);
            printf("La solucion unica es %f\n",x1);
        }
        else{
            x1 = (-b - sqrt(delta))/(2*a);
            x2 = (-b + sqrt(delta))/(2*a);
            printf("Las soluciones son x1 = %f y x2 = %f\n",x1,x2);
        }
    }
    return 0;
}
```


Introducción

Compilación

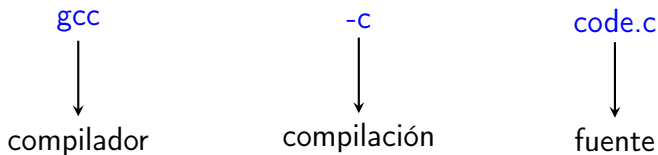
Un **compilador** es un programa que lee un programa escrito en un lenguaje (fuelle) y lo traduce a un programa equivalente en otro lenguaje (objeto).



Introducción

Compilación

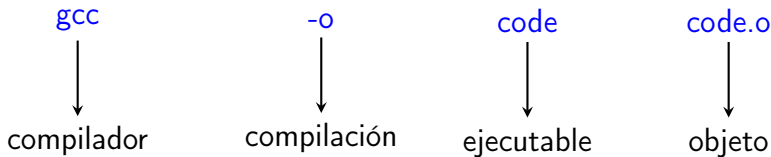
```
@progra:/$ ls  
code.c  
@progra:/$ gcc -c code.c
```



```
@progra:/$ ls  
code.c code.o
```

Introducción

Compilación



```
@progra:/$ gcc -o code code.o
@progra:/$ ls
code code.c code.o
@progra:/$ ./code
```

- Los lenguajes **interpretados** necesitan de un programa que traduzcan inmediatamente, línea a línea, el código fuente escrito.
- Cada línea de código es analizada y ejecutada al mismo tiempo.
- A diferencia de la compilación, el proceso de interpretación es más lento, ya que siempre se está verificando el código.
- No se genera ningún objeto, se traduce al lenguaje de máquina sin pasar por algún intermediario.
- Si se quiere volver a ejecutar el programa, realiza el análisis y la ejecución nuevamente, línea a línea.

Introducción

Sistema Binario

- Sistema de numeración, donde se utilizan solamente dos cifras: 0 y 1.
- Debido a que los computadores trabajan con dos niveles de voltaje, el sistema binario es utilizado (1: encendido, 0: apagado).
- Un número binario entonces está en base 2, ya que solamente se utilizan dos cifras.
- Para convertir un número binario a decimal, simplemente se multiplica por 2^i , donde i es la posición del bit, para luego sumar todas las cantidades.

1	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---

$2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$

= 133

Introducción

Sistema Binario

Ejemplo 2

$$11001110_2 = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ \hline \end{array} = 206$$

$$01011010_2 = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ \hline \end{array} = 90$$

$$10010011_2 = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ \hline \end{array} = 147$$

$$11110101_2 = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ \hline \end{array} = 245$$

Introducción

Sistema Binario

Para convertir un número decimal a binario, simplemente se divide repetidamente por 2 y se almacenan los restos de las divisiones.

Número	Resultado	Resto
141	70	1
70	35	0
35	17	1
17	8	1
8	4	0
4	2	0
2	1	0
1	0	1

$$141 = 10001101_2$$

Introducción

Sistema Binario

- También los números fraccionarios se pueden representar en el sistema binario.
- Veamos un ejemplo con el número 5.3125
- Primero notamos que $5 = 101_2$, luego nos preocupamos de la parte decimal.

$$0.3125 \times 2 = 0.625 \quad \longrightarrow 0$$

$$0.625 \times 2 = 1.25 \quad \longrightarrow 1$$

$$0.25 \times 2 = 0.5 \quad \longrightarrow 0$$

$$0.5 \times 2 = 1 \quad \longrightarrow 1$$

$$5.3125 = 101.0101_2$$

Veamos otro ejemplo con el número 3.1

$$0.1 \times 2 = 0.2 \longrightarrow 0$$

$$0.2 \times 2 = 0.4 \longrightarrow 0$$

$$0.4 \times 2 = 0.8 \longrightarrow 0$$

$$0.8 \times 2 = 1.6 \longrightarrow 1$$

$$0.6 \times 2 = 1.2 \longrightarrow 1$$

$$0.2 \times 2 = 0.4 \longrightarrow 0$$

$$0.4 \times 2 = 0.8 \longrightarrow 0$$

$$0.8 \times 2 = 1.6 \longrightarrow 1$$

$$0.6 \times 2 = 1.2 \longrightarrow 1$$

$$3.1 = 11.000110011\dots = 11.0\overline{0011}_2$$

Introducción

Sistema Binario

- Para convertir un número binario fraccionario a decimal, simplemente multiplicamos el valor por 2^{-i} , donde i es la posición del bit, para luego sumar todas las cantidades.
- Por ejemplo veamos a cuál número decimal corresponde 0.101001_2

1	0	1	0	0	1
---	---	---	---	---	---

$= 0.640625$

2^{-1} 2^{-2} 2^{-3} 2^{-4} 2^{-5} 2^{-6}

0.5 0.125 0.15625

Introducción

Sistema Octal

- El sistema octal, considera la utilización de 8 dígitos.
- Podemos convertir un número binario en octal:
 - Agrupe en grupos de a 3 iniciando por el lado derecho. Si al terminar de agrupar no completa 3 dígitos, entonces agregue ceros a la izquierda.
 - Posteriormente vea el valor que corresponde.
 - La cantidad correspondiente en octal se agrupa de izquierda a derecha.

Binario	000	001	010	011	100	101	110	111
Octal	0	1	2	3	4	5	6	7

Introducción

Sistema Octal

Ejemplo 3

$$110111_2 = \begin{array}{|c|c|c|} \hline 1 & 1 & 0 \\ \hline \end{array} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline \end{array} = 67_8$$

6 7

$$10001111_2 = \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline \end{array} \begin{array}{|c|c|c|} \hline 0 & 0 & 1 \\ \hline \end{array} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline \end{array} = 217_8$$

2 1 7

$$1000011_2 = \begin{array}{|c|c|c|} \hline 0 & 0 & 1 \\ \hline \end{array} \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline \end{array} \begin{array}{|c|c|c|} \hline 0 & 1 & 1 \\ \hline \end{array} = 103_8$$

1 0 3

Introducción

Sistema Octal

Para convertir un número octal a binario, simplemente cada dígito octal se convierte a su equivalente de 3 dígitos binarios.

Ejemplo 4

$$247_8 = \begin{array}{|c|} \hline 2 \\ \hline \end{array} \begin{array}{|c|} \hline 4 \\ \hline \end{array} \begin{array}{|c|} \hline 7 \\ \hline \end{array} = 10100111_2$$

010 100 111

$$150_8 = \begin{array}{|c|} \hline 1 \\ \hline \end{array} \begin{array}{|c|} \hline 5 \\ \hline \end{array} \begin{array}{|c|} \hline 0 \\ \hline \end{array} = 1101000_2$$

001 101 000

$$36_8 = \begin{array}{|c|} \hline 3 \\ \hline \end{array} \begin{array}{|c|} \hline 6 \\ \hline \end{array} = 11110_2$$

011 110

Introducción

Sistema Hexadecimal

- El sistema hexadecimal, considera la utilización de 16 dígitos.
- Podemos convertir un número binario en hexadecimal:
 - Agrupe en grupos de a 4 iniciando por el lado derecho. Si al terminar de agrupar no completa 4 dígitos, entonces agregue ceros a la izquierda.
 - Posteriormente vea el valor que corresponde.
 - La cantidad correspondiente en hexadecimal se agrupa de izquierda a derecha.

Bin	0000	0001	0010	0011	0100	0101	0110	0111
Hex	0	1	2	3	4	5	6	7

Bin	1000	1001	1010	1011	1100	1101	1110	1111
Hex	8	9	A	B	C	D	E	F

Introducción

Sistema Hexadecimal

Ejemplo 5

$$110111010_2 = \begin{array}{|c|c|c|c|} \hline 0 & 0 & 0 & 1 \\ \hline \end{array} \begin{array}{|c|c|c|c|} \hline 1 & 0 & 1 & 1 \\ \hline \end{array} \begin{array}{|c|c|c|c|} \hline 1 & 0 & 1 & 0 \\ \hline \end{array} = 1BA_{16}$$

1 B A

$$11011110101_2 = \begin{array}{|c|c|c|c|} \hline 0 & 1 & 1 & 0 \\ \hline \end{array} \begin{array}{|c|c|c|c|} \hline 1 & 1 & 1 & 1 \\ \hline \end{array} \begin{array}{|c|c|c|c|} \hline 0 & 1 & 0 & 1 \\ \hline \end{array} = 6F5_{16}$$

6 F 5

$$100001101100_2 = \begin{array}{|c|c|c|c|} \hline 1 & 0 & 0 & 0 \\ \hline \end{array} \begin{array}{|c|c|c|c|} \hline 0 & 1 & 1 & 0 \\ \hline \end{array} \begin{array}{|c|c|c|c|} \hline 1 & 1 & 0 & 0 \\ \hline \end{array} = 86C_{16}$$

8 6 C

Para convertir un número hexadecimal a binario, simplemente cada dígito hexadecimal se convierte a su equivalente de 4 dígitos binarios.

Ejemplo 6

$$ABF_{16} = \boxed{A} \quad \boxed{B} \quad \boxed{F} = 101010111111_2$$

1010 1011 1111

$$47D_{16} = \boxed{4} \quad \boxed{7} \quad \boxed{D} = 10001111101_2$$

0100 0111 1101

$$1BE9_{16} = \boxed{1} \quad \boxed{B} \quad \boxed{E} \quad \boxed{9} = 1101111101001_2$$

0001 1011 1110 1001