

1 Hola mundo

Introducción a la programación. Lenguajes de programación. Lenguaje máquina, ensamblador y de alto nivel. Código fuente. Compiladores e intérpretes. El lenguaje de programación C. Programa *hello world*. Compilar y ejecutar en Linux. Pasar argumentos a `main`. Paradigmas de programación: programación imperativa y estructurada.

2 Tipos de datos

Tipos de datos primitivos. Literales, expresiones y enunciados. Variables y constantes. Lenguajes de tipado estático y dinámico. Números enteros y de coma flotante. Caracteres y *strings*. Operadores aritméticos, lógicos y relacionales. Operaciones lógicas *bitwise*. Operador ternario. Operador de asignación. Operaciones de asignación compuesta. Conversiones entre tipos y *casts*.

3 Funciones, lo básico

Declarar y llamar funciones. Parámetros y argumentos. Valores de retorno. Alcance (*scope*) de una variable. Pasaje de valores por copia. Funciones de la librería estándar de C. Números aleatorios.

4 Condicionales y loops

Estructuras de control: secuencial, de selección e iterativas. Diagramas de flujo: proceso, E/S y decisión. Enunciados condicionales: *if ... else*. Selección múltiple: *switch ... case*. Condicionales anidados y encadenados. Enunciados de repetición: *while, do ... while* y *for*. Variable de control del *loop*, contadores y acumuladores. *Loops* anidados. *Loops* infinitos y *break*. Bloques de código y alcance (*scope*) de variables.

5 Arrays

Arrays unidimensionales. Índices y acceso aleatorio. Contiguidad de los elementos de un *array* en memoria. *Arrays* de caracteres. *Strings* en C. Funciones de `string.h`. Matrices o *arrays* bidimensionales. Tipos de datos de referencia, punteros.

6 Entrada y salida

Standard streams: *stdin, stdout* y *stderr*. Redirección y *pipes*. Librería estándar de C: `stdio.h`. Archivos de texto: lectura y escritura. Procesamiento de datos usando archivos de texto y binarios.

7 Más sobre funciones

Programación modular. Pasaje de argumentos por valor y referencia. Recursión. Equivalencia entre iteración y recursión. Programación dinámica.

8 Búsqueda y ordenamiento

Análisis de complejidad de un algoritmo. Búsqueda exhaustiva. Búsqueda lineal. Búsqueda binaria. Ordenamiento por inserción. Otros algoritmos de ordenamiento cuadráticos. Algoritmos lineales de tipo *divide and conquer*: *mergesort* y *quicksort*. Ordenamiento de enteros. Aplicaciones para optimizar problemas de búsqueda.

9 Tipos de datos abstractos

Listas enlazadas. *Arrays* redimensionables. Pilas y colas (*stacks* y *queues*). Aplicaciones de pilas y colas. Tabla de símbolos (*map*) y conjunto (*set*). Implementaciones basadas en una *hash table* y en un BST (*binary search tree*).