Glosario C

Programación / Laboratorio I

Autores: *Mg. Mauricio Dávila*

Revisores: *Esp. Ernesto Gigliotti*

Curso: 000
Versión : 0.3

Índice de contenido

Algoritmo

Método que describe cómo se resuelve un problema en término de las acciones que se ejecutan para lo cual especifica el orden en que se ejecutan estas acciones.

Ámbito / Alcance / Scope de una variable

Se extiende desde el punto donde se define hasta la primera llave que empareja con la llave de apertura antes de que la variable fuese definida. Eso quiere decir que un ámbito se define por su juego de llaves «más cercanas». Si hablamos del scope global es afuera de cualquier llave.

Argumento

Información pasada a una función. Los argumentos se suelen llamar también parámetros o parámetros actuales.

Array, vector, lista

Variable que almacena una secuencia indexada de los mismos tipos de datos. Normalmente los elementos individuales se referencian por el valor de un índice. El índice es un valor entero que , suele comenzar, en 0 para el primer elementos, 1 para el segundo y así sucesivamente.

Asignación

Almacenamiento de un valor en una variable.

Biblioteca

En informática, una biblioteca es una colección o conjunto de subprogramas usados para desarrollar software. La mayoría de los lenguajes compilados tienen una biblioteca estándar, aunque los programadores también pueden crear sus propias bibliotecas personalizadas. En español, biblioteca también suele referirse como "librería", una mala traducción de library (que es en realidad biblioteca).

<ctype.h>
Contiene funciones para clasificar caracteres según sus tipos o para convertir entre mayúsculas y minúsculas independientemente del conjunto de caracteres (típicamente ASCII o alguna de sus extensiones).

<math.h>
Contiene las funciones matemáticas comunes.

<stdio.h>
Proporciona el núcleo de las capacidades de entrada/salida del lenguaje C (incluye la venerable función printf).

<stdlib.h>
Para realizar ciertas operaciones como conversión de tipos, generación de números pseudo-aleatorios y gestión de memoria dinámica.

<string.h>
Para manipulación de cadenas de caracteres.

<time.h>
Para tratamiento y conversión entre formatos de fecha y hora.

Binario

Representación numérica en base 2. En esta base sólo se utilizan los números 0 y 1. Las posiciones de los dígitos representan potencias sucesivas de 2.

Bit

Dígito binario que puede tomar dos valores posibles: 0 y 1. Los bits son elementos básicos de construcción de programas y dato. Su nombre deriva de **b**inary dig**it**.

Bloque

Sentencias y declaraciones encerradas entre una pareja de llaves (apertura y cierre), ´{´ y ´}´.

Código fuente

Texto de un programa antes de ser compilado.

Comentario

Trozos de texto que tienen como objetivo documentar el programa y mostrar como se ha construido. Los comentarios no son sentencias de programación y son ignorados por el compilador. En C los comentarios están precedidos por dos barras (//) en una línea o encerrados entre /* y */ en múltiples líneas. Es muy común utilizarlos para deshabilitar un bloque de código.

```c
#include <stdio.h>
#include <stdlib.h>

int main() {
	//Comentario en linea independiente.
	int edad;
	edad=32; //Comentario en linea junto a codigo.
	//edad++;  Linea de codigo comentada
	printf ("La edad es %d\n", edad);
	/*
	comentario de un bloque de codigo
	printf ("La edad es %d\n", edad);
	printf ("La edad es %d\n", edad);
	*/
	edad=32;
	/* Finalizamos el programa, comentario en bloque en
	multiples lineas */
	return 0;
	/*Comentario en bloque en una sola linea*/
}
```

Comparación

Se realiza mediante los operadores (> , < , >= , <= , == y !=) . Se comparan valores de variables por ejemplo para saber si una variable contiene el mismo valor que otra variable o si una es mayor a otra. Estas operaciones dan como resultado 1 (verdadero) y 0 (falso).

Compilación

Proceso de traducción de un lenguaje de programación. Normalmente este proceso implica la traducción de un lenguaje de programación de alto nivel a lenguaje de programación de bajo nivel, o el formato binario de un conjunto de instrucciones específicas.

Constante

En programación, una constante es un valor que no puede ser alterado durante la ejecución de un programa. En C para declarar una variable como constante se utiliza la palabra reservada const (const int constante = 9;). Es importante no confundir al **define** con el **const**.

```
const int cantidad = 5;
#define CANTIDAD 5
```

En la primera declaración se indica que **cantidad** es una constante de tipo int mientras que en la segunda se indica mediante una directiva de preprocesador que donde aparezca en el código la palabra **CANTIDAD** deberá ser reemplazada por 5 directamente.

Declaración

Es una instrucción que se ejecutará en el programa, a efectos de reservar espacio de memoria según el tipo de variable para poder escribir y leer valores.

```
int edad = 35;
```

Depuración

Proceso de encontrar y eliminar errores en un programa. En inglés se conoce como debugging, porque se asemeja a la eliminación de bichos (bugs).

Diagrama

Una representación gráfica construida utilizando una notación formal para visualizar y documentar las relaciones entre las entidades de una aplicación.

Diseño

Actividad de definir cómo se debe estructurar e implementar un programa.

Documentar

En la cursada utilizaremos Doxygen, es el programa que nos va a servir para documentar el código muy rápido, de manera simple y sencilla.

```
/ **
  * <Una descripción breve de una línea>
  * *
  * <Descripción más larga>
  * <Puede abarcar varias líneas o párrafos según sea necesario>
  * *
  * @param nombreDelParametro Descripción del parámetro de la función
  * @param ...
  * @return Descripción del valor de retorno
  * /
```

Entidad

Una entidad es un objeto concreto o abstracto que presenta interés para el sistema y sobre el que se recoge información la cual va a ser representada en un sistema. En C la forma de representarlas son las estructuras.

Ejecutable

Es el producto obtenido luego de compilar el código, por lo general en lenguaje de máquina. Es el conjunto de instrucciones nativas que la computadora ejecutará en el hardware.

Estructuras

Una entidad es un objeto concreto o abstracto que presenta interés para el sistema y sobre el que se recoge información la cual va a ser representada en un sistema. Los campos de la estructura son las características de la entidad relevante para el sistema.

```
struct sPersona
{
	char nombre[30];
	char apellido[40];
	int edad;
};
```

Entero

Un entero en C hace referencia a un número sin parte decimal definidos con la palabra reservada int.

Función

Es un bloque de código dentro del programa que se encarga de realizar una tarea determinada.

Función Static

Cuando una función se declara como estática, tan sólo puede ser invocada desde el archivo en el que está definida.

Git

Es una herramienta que realiza una función del control de versiones de código de forma distribuida, permite guardar una versión actual de un proyecto a la cual se podrá regresar en el futuro.

Hardcodear

Una mala práctica en el desarrollo de software que consiste en incrustar datos directamente en el código fuente del programa, en lugar de obtener esos datos de una fuente externa como un archivo de configuración o parámetros de la línea de comandos, o un #define.

Heap

Es una estructura dinámica utilizada para almacenar datos en ejecución. A diferencia de la **pila** (stack) que solamente almacena las variables declaradas en los bloques previos a su ejecución, el heap permite reservar memoria dinámicamente. Las variables globales y estáticas también son almacenadas en él.

Hexadecimal

El sistema hexadecimal es el sistema de numeración de base 16. , dado que el sistema usual de numeración es de base decimal y, por ello, sólo se dispone de diez dígitos, se adoptó la convención de usar las seis primeras letras del alfabeto latino para suplir los dígitos que nos faltan.

Índice de un array

A los datos almacenados en un array se les denomina elementos; al número de elementos de un array se les denomina tamaño o rango del vector. Para acceder a los elementos individuales de un array se emplea un índice que será un número entero no negativo que indicará la posición del elemento dentro del array.

IDE

Software para ayudar a los programadores a escribir código de manera eficiente.

Identificador

Nombre de una variable o función.

Implementación

La actividad de escribir, compilar, probar y depurar el código de un programa.

Inicializar

Para declarar una variable usaremos una instrucción compuesta del nombre del tipo de datos de la variable, el nombre de la variable y opcionalmente un operador de asignación y un valor inicial. Inicializar una variable es asignar por primera vez un valor dentro del programa.

```
int edad = 44;
```

Lista

Es otra manera de denominar a los Arrays.

Lista enlazada

Una lista enlazada (linked list) es una estructura de datos dinámica en la que cada elemento (llamado nodo) está formado por dos elementos: los datos y una referencia (o puntero) que apunta al siguiente nodo. Una lista enlazada es una colección de nodos donde cada nodo está conectado al siguiente nodo a través de un puntero.

Macro

Una macro es un nombre dado a un bloque de sentencias C como directiva de preprocesador, esto quiere decir que son cosas que van a pasar antes de compilar. Una macro se define con la directiva de preprocesador, **#define**. La ventaja de usar una macro radica en no estar escribiendo lo mismo en múltiples lugares, esto ayudará a evitar el error conocido como “hardcodeo”.

Memoria dinámica

En C, la memoria dinámica se asigna desde la zona de memoria conocida como HEAP utilizando algunas funciones de biblioteca estándar. Las dos funciones clave de memoria dinámica son malloc () y free ().

La función malloc () toma un solo parámetro, que es el tamaño del área de memoria solicitada en bytes. Devuelve un puntero a la memoria asignada. Si la asignación falla, devuelve NULL. El prototipo para la función de biblioteca estándar es así:

```
void * malloc (size_t cantidadDeBytes);
```

La función free () toma el puntero devuelto por malloc () y desasigna la memoria. No se devuelve ninguna indicación de éxito o fracaso. El prototipo de la función es así:

```
void free  (void* puntero);
```

Memoria estática

La zona estática de memoria permite almacenar variables globales y de tamaño estático. Si se encuentra una variable definida fuera de las funciones (incluyendo main) o en otro archivo fuente (.c), se la considera global, el compilador les asigna un espacio determinado y genera las referencias para accesarlas en la zona estática.
El tamaño de las variables no puede cambiarse durante la ejecución del programa, es asignado en forma estática. El tiempo de vida de las variables de la zona estática es la duración del programa. Estas variables son visibles para todas las funciones que estén definidas después de ellas en el archivo en que se definan.

Modificador

Una palabra reservada en C que especifica las propiedades de los datos o funciones y cómo se pueden utilizar (Ej. static)

Nodo

Es una estructura de datos dinámica en la que cada elemento está formado por dos elementos: los datos y una referencia (o puntero) a otro elemento de igual tipo.

Operador

Permite realizar operaciones para valores de tipos primitivos de datos. Ejemplos de operadores son +,-,*,/ y %

Operadores Lógicos

El operador lógico AND (&&) genera el valor 1 si ambos operandos tienen valores distintos de cero. Si alguno de los operandos es igual a 0, el resultado es 0. Si el primer operando de una operación AND lógica es igual a 0, el segundo operando no se evalúa.

El operador lógico OR (||) realiza una operación OR inclusivo en sus operandos. El resultado es 0 si ambos operandos tienen valores 0. Si cualquiera de los operandos tiene un valor distinto de cero, el resultado es 1. Si el primer operando de una operación OR lógica tiene un valor distinto de cero, el segundo operando no se evalúa.

Los operandos de las expresiones AND y OR lógicas se evalúan de izquierda a derecha. Si el valor del primer operando es suficiente para determinar el resultado de la operación, el segundo operando no se evalúa.

Operador Relacional

Se llaman operadores relacionales o de comparación a aquellos que permiten comparar dos valores evaluando si se relacionan cumpliendo el ser menor uno que otro, mayor uno que otro, igual uno que otro, etc.

Pasaje por valor

La función recibe sólo una copia del valor que tiene la variable, o sea que no la puede modificar.(las variables escalares de pasan de esta manera)

Pasaje por referencia

Se pasa la posición de memoria donde está guardada la variable, por lo que la función sabe físicamente donde se encuentra y puede acceder a ella ya sea para escribirla o leerla.

Palabra reservada

Es una palabra definida como parte del lenguaje de programación, Un nombre de palabra reservada no se puede utilizar para ningún otro propósito.

Parámetro formal

Parámetros definidos en la firma o prototipo de la función.

Parámetro actual

Valor que se pasa a una función cuando se invoca. Los parámetros reales (actuales) deben concordar en tipo, orden y número con los parámetros formales. Cuando se invoca a una función, los valores de los argumentos actuales se copian en los correspondientes argumentos formales.

Preprocesador

El preprocesador es una parte del compilador que se ejecuta en primer lugar, cuando se compila un fuente C, y que realiza unas determinadas operaciones, independientes del propio lenguaje C. Estas operaciones se realizan a nivel léxico y consisten en:

1. La inclusión de otros textos en un punto del archivo fuente
2. La realización de sustituciones
3. La eliminación de ciertas partes del código fuente

Debemos tener en cuenta que el preprocesador trabaja únicamente con el texto del fuente y no tiene en cuenta ningún aspecto sintáctico ni semántico del lenguaje.

Stack

Es una estructura dinámica de datos que almacena la información sobre las subrutinas activas de un programa en ejecución. Cuando hacemos una llamada a una función, un bloque en el tope de la pila es reservado para guardar las variables locales junto con algunos datos necesarios para garantizar el funcionamiento adecuado de la estructura (como la dirección a la que tendrá que retornar el hilo de ejecución cuando termine la función). Después de retornar, el bloque de la pila que ocupaba el llamado, se libera para poder utilizarse más adelante de ser necesario.

Se llama pila de ejecución porque es una estructura de datos que funciona bajo el concepto de LIFO (last in first out). Esto hace que sea más sencillo mantener el control de los bloques que deben ser liberados, puesto que será aquel que esté en el tope de la pila.

Tipo de dato

Son palabras reservadas que se utilizan para definir variables, como int, float, char, etc...

Variable Local

Variable definida en el interior de una función.

Variable Global

Son declaradas en la cabecera del programa, antes de la declaración de funciones y antes de la función main. Decimos que son variables globales porque estas variables son conocidas por todas las funciones en el programa, incluida la función main.

Variables Static

En el lenguaje de programación C se usa static con variables globales y para restringir su ámbito al archivo donde se definen.

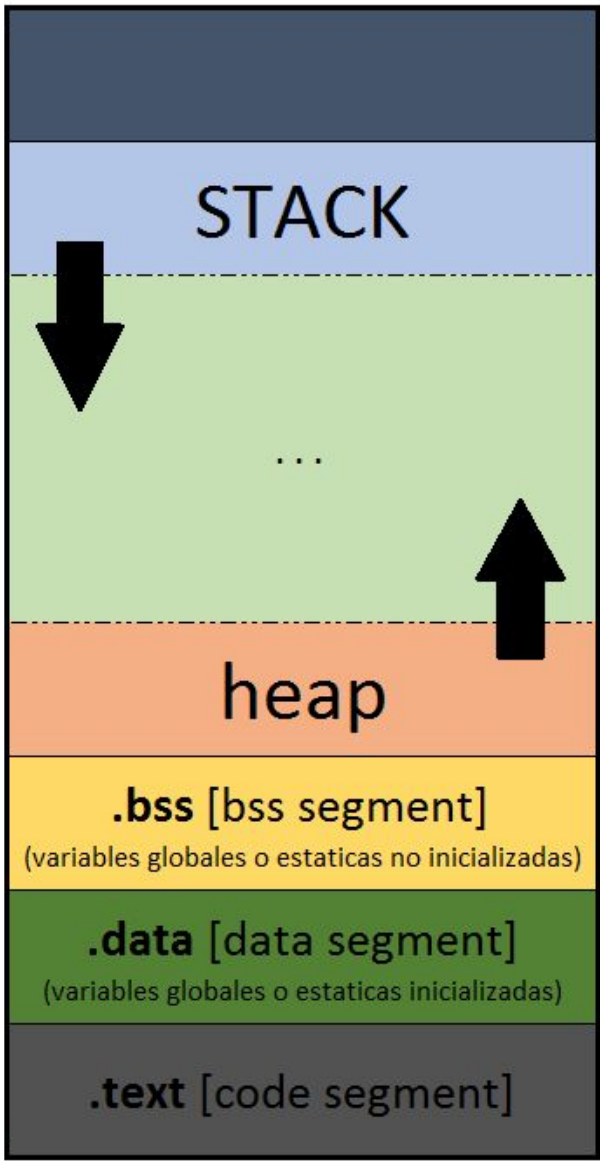
Validar

Es el proceso de asegurar que los datos sean correctos y útiles.

Zonas de memoria

La segmentación de memoria de un programa se divide en varias zonas que pasamos a detallar a continuación.

- **Segmento de código:** Es de tamaño fijo y de solo lectura. En esta parte se almacenan todas y cada una de las instrucciones en código máquina que componen el programa que se está ejecutando.
- **Segmento de datos**: Se almacenan las variables globales inicializadas del programa. De tamaño fijo y permite la escritura.
- **Segmento BSS** : Se almacenan las variables globales sin inicializar. De tamaño fijo y permite la escritura.
- **Heap** : Segmento de memoria reservado para la memoria dinámica del programa.
- **Stack** Es una estructura dinámica de datos que almacena la información sobre las subrutinas activas de un programa en ejecución.


STACK
...
heap
.bss [bss segment]
(variables globales o estaticas no inicializadas)
.data [data segment]
(variables globales o estaticas inicializadas)
.text [code segment]