UNIVERSIDAD DE LIMA | 60 años

Introducción a la programación en C (I)

Sistemas Operativos
Ciclo 2023_2 – Semana_03

Programación

- **UNIDAD 1: Fundamentos de Sistemas Operativos**
  - Semana_01: Introducción a los Sistemas Operativos, Virtualización
  - Semana_02: Cloud y funciones del Sistema Operativo
  - **Semana_03: Introducción a la programación en C**
  - Semana_04: Programación en C

Agenda

1. Comandos relevantes en Linux
2. Estructura de un programa en C
3. Manipulación de bits: Operadores Lógicos, Bitwise. Bit Masking

COMANDOS RELEVANTES EN LINUX

- ifconfig
- ip address
- sudo adduser
- sudo deluser
- ping
- mkdir
- sort
- grep
- hostname

LENGUAJE DE PROGRAMACIÓN C

- El **lenguaje de programación C** fue desarrollado por Dennis Ritchie entre 1969 y 1972 en los Laboratorios Bell para ser usado en UNIX.
- Es un lenguaje de programación del tipo **estructurado**, similar a Pascal, Fortran, Basic.
- C utiliza instrucciones y sentencias como **if**, **else**, **for**, **do** y **while**.

```
#include <stdio.h>
main()
{
      printf("hello, world\n");
}
```

Dennis M. Ritchie (1941-2011)

ESTRUCTURA DE UN PROGRAMA EN C

ESTRUCTURA BÁSICA DE UN PROGRAMA EN C

Esta línea le dice al **compilador** que incluya el contenido del archivo de encabezado de la **biblioteca estándar stdio.h** en el programa. Los encabezados son archivos que contienen declaraciones de funciones, macros y tipos de datos, y debe incluir el archivo de encabezado antes de usarlos.

Esta línea comienza la definición de una función. Indica el nombre de la función **(main)**, el tipo y el número de argumentos que espera y el tipo de valor que devuelve esta función **(int)**. La ejecución del programa se inicia en la función **main ()** .

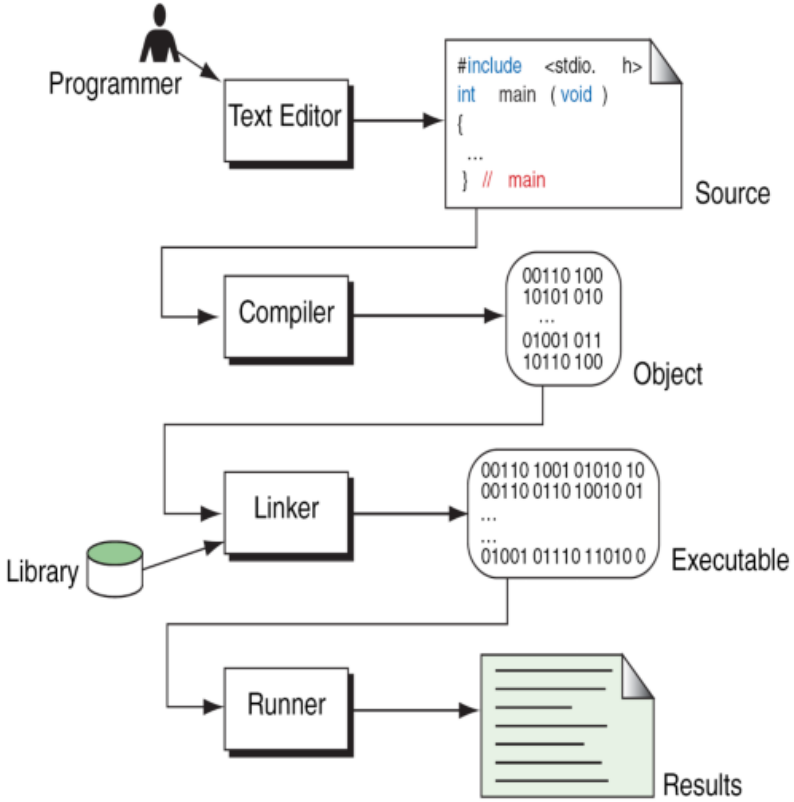
Esta línea llama a la función **puts()** para enviar el texto a la salida estándar (la pantalla, por defecto), seguida de una nueva línea. La cadena a emitir se incluye dentro de los paréntesis. **"ULIMA"** es la cadena que se escribirá en la pantalla. En C, cada valor literal de cadena debe estar dentro de las comillas dobles "…" .

Cuando definimos **main()** , lo declaramos como una función que devuelve un **int** , debe devolver un entero. En este programa, estamos devolviendo el valor entero **0**, que se utiliza para indicar que el programa salió correctamente. Después de **return 0;** el proceso de ejecución terminará. En los programas de C, cada declaración debe terminar con un punto y coma (**;**).

Las llaves se utilizan en pares para indicar dónde comienza y termina un bloque de código.

```
#include
<stdio.h>


int main(void)
{

puts("ULIMA");


return 0;

}
```

Ejecución de un Programa en Lenguaje C

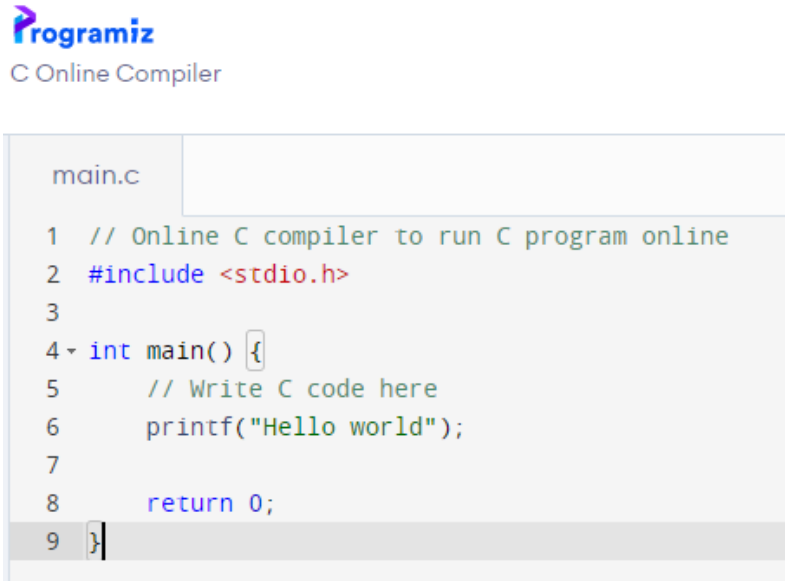


Extraído de: “C Programming – Department of Information Technmology- Vardhaman College of Engineering”

Actividad

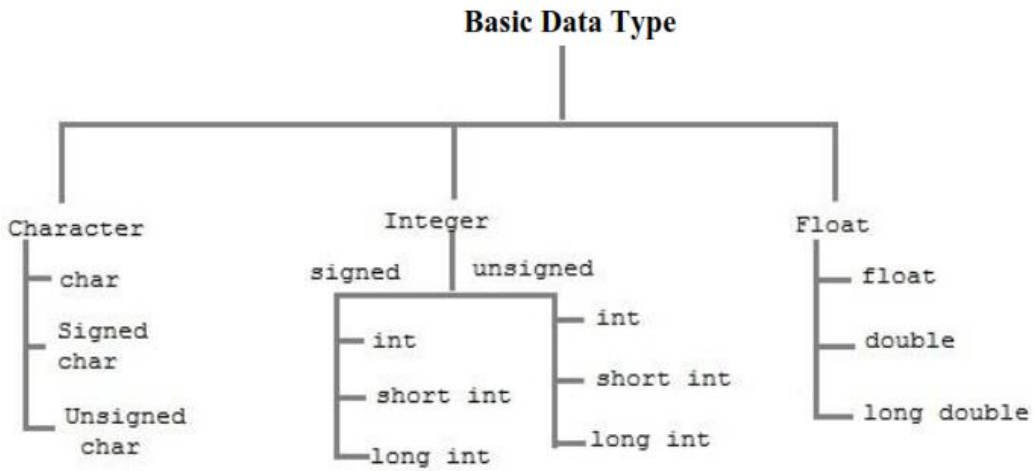
Verifique la estructura del siguiente programa en C

https://www.programiz.com/c-programming/online-compiler/



TIPOS DE DATOS

Tipos Básicos de Datos en Lenguaje C, Tamaño y Rangos



Type	Size (bytes)	Range	Control String
char or signed char	1	-128 to 127	%c
unsigned char	1	0 to 255	%c
int or signed int	2	-32768 to 32767	%d or %i
unsigned int	2	0 to 65535	%u
short int or signed short int	1	-128 to 127	%d or %i
unsigned short int	1	0 to 255	%d or %i
long int or signed long int	4	-2147483648 to 2147483647	%ld
unsigned long int	4	0 to 4294967295	%lu
float	4	3.4E-38 to 3.4E+38	%f or %g
double	8	1.7E-308 to 1.7E+308	%lf
long double	10	3.4E-4932 to 1.1E+4932	%Lf

Extraído de: “C Programming – Department of Information Technmology-Vardhaman College of Engineering”

Otros tipos de Datos Lenguaje C

Types	Data Types
Basic Data Type	int, char, float, double
Derived Data Type	array, pointer, structure, union
Enumeration Data Type	enum
Void Data Type	void

OPERADORES LÓGICOS

Operadores Lógicos

- Permiten evaluar expresiones

Operator	Meaning	Example	Return value
&&	Logical AND	(9>2)&&(17>2)	1
||	Logical OR	(9>2) || (17 = = 7)	1
!	Logical NOT	29!=29	0

AND

Op1	Op2	Op1 && Op2
true	true	true
true	false	false
false	true	false
false	false	false

OR

Op1	Op2	Op1 // Op2
true	true	true
true	false	true
false	true	true
false	false	false

NOT

Op1	Op1 !
true	false
false	true

Extraído de: “C Programming – Department of Information Technmology- Vardhaman College of Engineering”

MANIPULACIÓN DE BITS (BITWISE)

SENTENCIAS BIT A BIT EN C

- Permite manipular datos a nivel de bits.
- Solo operan sobre enteros

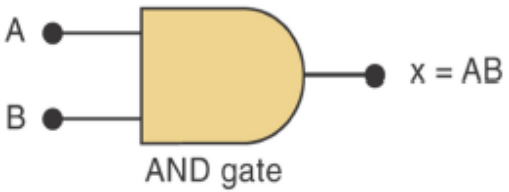
Operator	Meaning
&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR
<<	Shift left
>>	Shift right
~	One's complement.

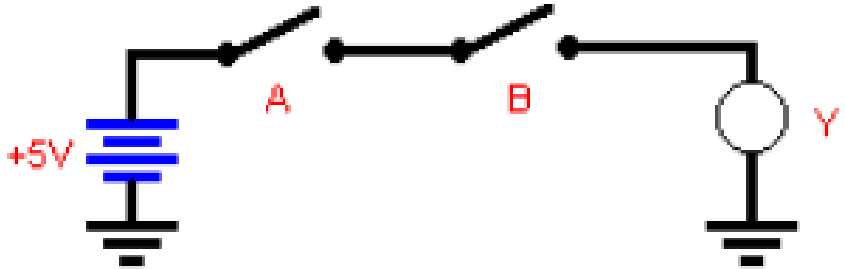
OPERACIÓN AND (&) / BITWISE-AND

AND (Multiplicación Booleana): **La salida es verdadera si todas las entradas son verdaderas**

AND

A	B	x = A • B
0	0	0
0	1	0
1	0	0
1	1	1





OPERACIÓN OR (|) / BITWISE-INCLUSIVE-OR

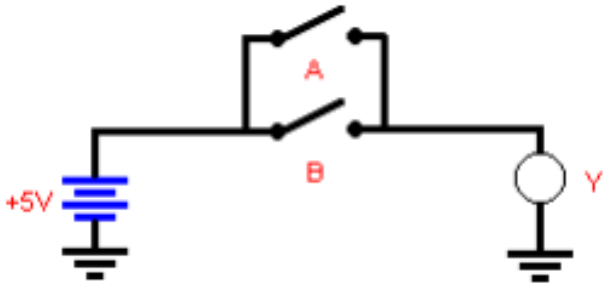
Existen tres operaciones básicas (operaciones lógicas): OR, AND Y NOT.

OR (Suma Booleana): **La salida es verdadera si al menos una de las entradas es verdadera**

OR

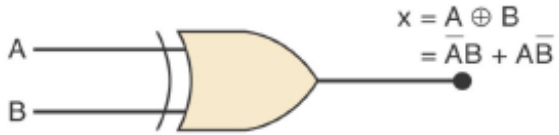
A	B	x = A + B
0	0	0
0	1	1
1	0	1
1	1	1

OR Gate



OPERACIÓN XOR (^) / BITWISE-EXCLUSIVE-OR

Devuelve un “1” en cada posición en la que los bits correspondientes tienen un “0” o un “1” en alguno de los operandos, pero no en ambos.



A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

OPERACIÓN COMPLEMENTO A LA BASE (~) / BITWISE NOT

Invierte los bits del operando

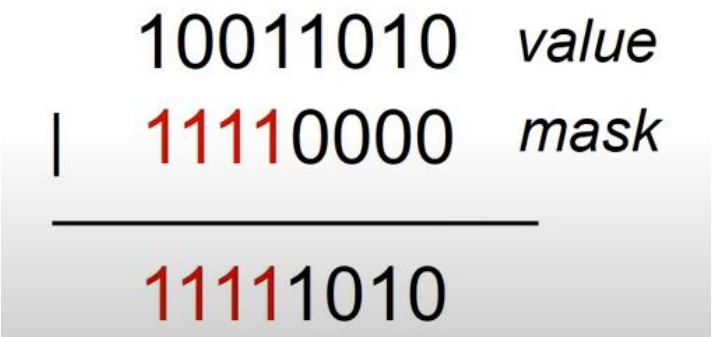
Por ejemplo

~ 00010111 = 11101000

BITMASKING

- Bitmasking se utiliza para cambiar solo un bit o una porción deseada de bits de un valor.

Ejemplo 1 : Colocar a 1 los cuatro bits de la izquierda del valor 154



Ejemplo 2 : Invertir los cuatro dígitos de la derecha del valor 154