

Uso del lenguaje c y c++

- C programación estructurada
- C++ programación orientada a objetos

C UNIX (Programación estructurada)

-Tipos de datos

int, float, char, bool, ...

-Control de flujo (define el comportamiento de los datos)

- if-else,
- for(n),
- while(<>!=),
- Do (<> != ==),
- break,
- switch.

-Lenguaje de máquina

- 1001010010001

-Lenguaje natural

- Idioma en el que hablamos

Ejemplo:

```
int a=10;
char nombre="";

while(a < 10)
{
    a =a +1;
}
```

"ANSI C" es un estándar para indicar la utilización de las cosas

-Colección de encabezados (headers) Nos ayuda para usar librerías

- Import < librerías>
- Funciones() < return> procedimientos()

Algoritmia

1. Problema

2. Solución

- Algoritmos(pseudocódigo)
- Diagrama de flujo
- Código (debugging)
- Traza (trace) - prueba de escritorio

Diagrama de flujo



Ejemplo:

1. Problema : ¿Determinar el area de un rectángulo?

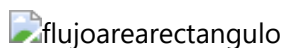
2. Solución

- AreaRect = largo x ancho (metros)



Algoritmia: pseudocódigo, diagrama de flujo, trazabilidad

- imprimir "introduzca una longitud en metros:"
- leer longitud
- imprimir "introduzca una anchura en metros:"
- leer ancho
- asignar arearectangulo = largo * ancho
- imprimir "El área es de " arearectangulo "metros cuadrados"



Workshop : main

Comandos para compilar

```
1 ls
2 g++ src/main.c -o output/main.exe
3 ./output/main.exe
4 ./output/main.exe
5 history
```

Crear un directorio

- resource


Agregar extensiones

- Draw.io Integration
- Drawio Preview

Algoritmia

- Problema : mayor de dos números.
- Solución
 - Pseudocodigo
 - Diagrama de flujo (DFD, Dreaw.io)

Diagrama de flujo

Maximo2numeros

Codigo

```
void maxnumero()  
{  
    int a=0, b=0;  
    printf("Ingrese 1er valor:");  
    scanf("%d" , &a);  
    printf("Ingrese 1er valor:");  
    scanf("%d" , &b);  
  
    if(a>b)  
    {  
        printf("El mayor es: %d",a);  
    }  
    else  
    {  
        printf("El mayor es: %d",b);  
    }  
}
```

30/05/2022

Introducción general

Programación ensamblador

C++ es la evolución de C

paracompilar

int -> la variable de la lista son enteros

float -> punto flotante(números que puede tener una parte fraccionaria)

Datos básicos

- char -> caracter -un solo breve
- short -> entero corto
- long -> entero largo
- double -> punto flotante de doble precisión



Declaración de variables e inicialización

Estructura de control

- Procedimiento

```
void NombreProcedimiento() {...}
```

- Procedimiento con parametros

```
void NombreProcedimiento(para1, para2, ..) {...}
```

```
para1 = <TipoDato> NombreParametro
```

- Función

```
<TipoDato> NombreProcedimiento()  
{... return <TipoDato>}
```

- Función con parametros

```
<TipoDato> NombreProcedimiento(para1, para2 ...)  
{... return <TipoDato> }
```

Constantes simbólicas



Flujo de control y diagrama de flujo



Arreglo de caracteres

1.- Arrays-Bufer-Vector-String: cadena de caracteres.

getchar-obtener un caracter, capturar EOF- final de archivo, genera un desbordamiento si me dice si entonces salta el caracter almacenado entonces paso al siguiente nivel y si esta capturado paso al siguiente, si uno de los tres esta mal hasta ahí, i es menor que cinco en la segunda parte c es menor , hay que tener cuidado con en el orden, tener cuidado en los pararentesis por que puede generar error, tener en cuenta la precedencia de condición y tambien la precedencia de los símbolos Las funciones pueden tener un nombre y parametro.

carateres + cadena de caracteres : string / array /vector

```
char letra = 'a';
char caracter = '1';
char palabra[10] = "hola";
char palabra[] = "como estas Angel";
```

Entrada desde consola

```
char c = getchar(); //--> EOF
while (c! = EOF)
//{....}
```

TIPOS, OPERADORES Y EXPRESIONES

En las constantes se sugiere que esten en las librerias.

```
#define VTAB '\013'
#define BELL '\007'
#define ENTER '\n'
#define ALERT '\a'
```

"\a" caracter de alarma

"\b" retroceso

"\f" avance de hoja

"\n" nueva linea

"\r" regreso de carro

"\t" tabulador horizontal

"\v" tabulador vertical

"\" diagonal invertida

"\" interrogación

"\' " apóstrofo

"\" \" comillas

"\ooo" número octal

"\xhh" número hexadecimal

"\0" fin de cadena

```
char s[] = "hola";           // longitud de cadena
int i = strlen(s);

char s[] = "125";           // de cadena a entero
int numero = atoi(s)
isdigit(c)
```

Enumeradores + operadores

enum escapes{

```
BELL          = '\a',

RETROCESO     = '\b',

TAB           = '\t',

NVALAN        = '\n',

VTAB          = '\t',

RETURN        = '\r'
```

};

- & AND de bits
- | OR inclusivo de bits
- ^ OR exclusivo de bits
- << corrimiento a la izquierda
- >> corrimiento a la derecha
- -- complemento a uno (unario)

Funciones

getline = ayuda a obtener los archivos de todos los caracteres

- `char línea[300] getline(línea, 300);`

stindex = nos dice en la posición en la que se encuentra

atoi atof

- sirve la conversión de streams a números

`f=i; // conversión implícita = cast ó casting // f = 123.00000`

`i=(int)f; // conversión explícita = cast ó casting`

Archivos de texto

- Apertura y cierre de fichros

Cuando alguien invoca un archivo ya sea para leer o escribir

archivo binario: nos ayuda a guardar los tipos de datos en forma binaria.

- b = binario
- r = lectura
- w = escritura
- a = añade datos al final del mismo

Enviar a un txt

```
archivo("files\\texto.txt");
```