Unidad 3

Lenguaje estructurado

Principal material bibliográfico utilizado

- www.jorgesanchez.net
- Fundamentos de Programación C/C++ - Ernesto Peñaloza Romero.
- Lenguaje C – Adolfo Beltramo, Nélida Matas.

El lenguaje C
Introducción

- Dennis Ritchie fue el creador por los años 60 y 70.
- ALGOL -> CPL -> BCPL -> B -> C
- Esta muy ligado al sistema Unix.
- Se diseño pensando en programar sistemas operativos.
- Es un lenguaje creado por programadores para programadores, diferencia por ejemplo con COBOL, BASIC, etc.
- Proliferaron diferentes versiones por lo que por los años 80 el ANSI empezó a concebir un C estándar que luego fue aceptado por ISO.
- En educación se considera un buen lenguaje para empezar a programar.

El lenguaje C
Introducción

- Es un lenguaje de alto nivel pero posee características que le permiten trabajar a muy bajo nivel (manipulación de memoria, de registros del procesador y otras partes del hardware).

- C no cuenta con operaciones complejas para trabajar con los datos, C ofrece operaciones simples, secuenciales, de selección, de iteración, subprogramas, etc., que gracias a ello hace que sea un lenguaje de fácil aprendizaje.

- Programas como Windows y Office fueron manufacturados a través de este lenguaje.

- C tiene diversos herederos: C++, Java, C# se basan en la filosofía de C, por lo que su aprendizaje ayuda a la comprensión de ellos.

El lenguaje C
Introducción

- C y C++, cabe destacar que si bien la forma de programar de ambos es diferentes, C++ comparte de forma casi idéntica la sintaxis con C.

- Es un lenguaje compilado.

- Permite crear todo tipo de aplicaciones.

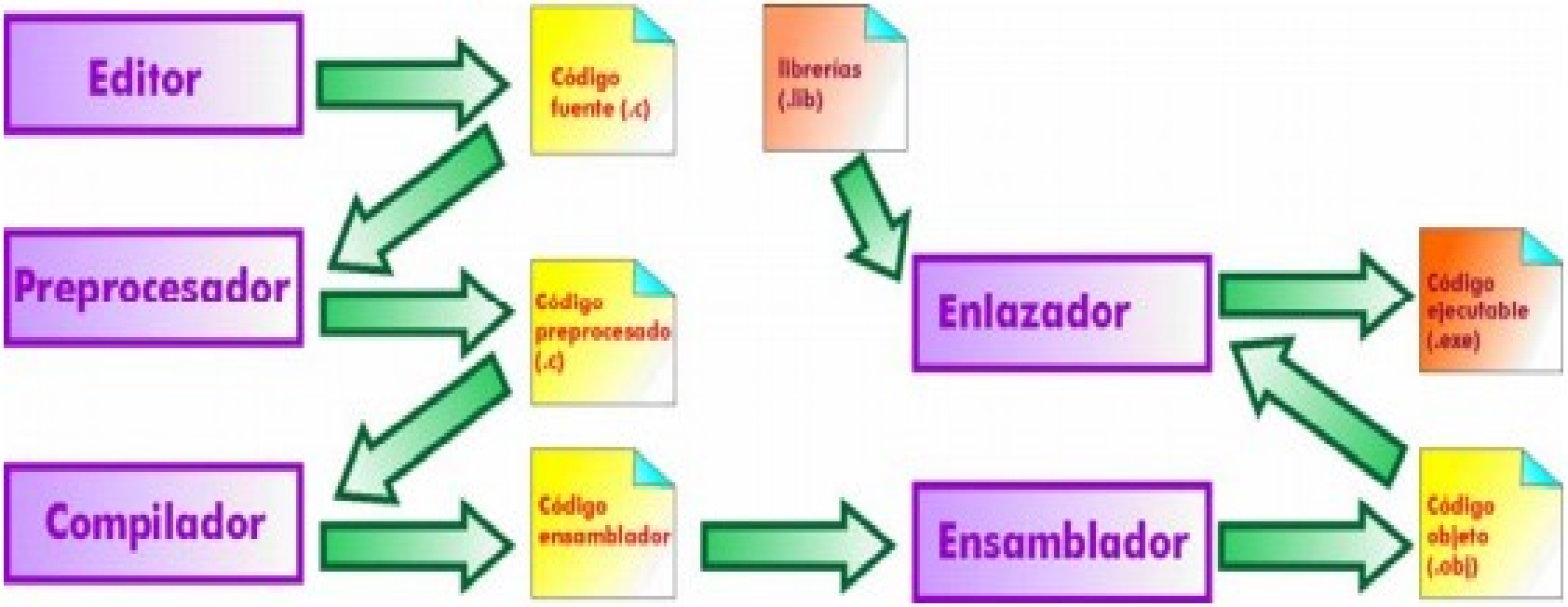
El lenguaje C
Introducción

- El lenguaje C es un lenguaje basado en el concepto de funciones.

- Un programa C es una colección de una o más funciones.

- Por lo general se puede utilizar cualquier nombre de función, excepto “main” que es la función principal de C, que se reserva para el inicio de la ejecución de cualquier programa.

El lenguaje C
Componentes del sistema



El lenguaje C
Mi primer programa

Programa que escribe “!Hola Mundo!!!!!”

```
/*programa saludo*/
#include <stdio.h>
void main(void)
{
printf("\n !Hola Mundo!!!!!");
}
```

•/* */ símbolos que encierran comentarios.
•main nombre de la función principal de C, que no requiere ningún parámetro de entrada ni de salida (hecho mencionado con las palabras “void”)|.
•{} delimitan los bloques, en este caso donde empieza y donde termina la función principal.
•printf, es una función, llamada desde el main que muestra en pantalla la frase !Hola Mundo!!!!! luego de dejar una línea “\n”.
•La directiva “#include” indica que se hará uso de la lista de encabezados que se encuentra en el archivo descrito entre “<>”, el compilador los requiere para reconocer las funciones utilizadas en el programa.

El lenguaje C
El concepto inicial de funciones

Dijimos que C se basaba en el concepto de funciones, a continuación se presenta el modelo general de declaración de una función:

```
Tipo-devuelto  nombre-de-funcion(lista de parámetros)
{
      Secuencia de sentencias;
}
```

Las funciones se llaman desde el cuerpo principal del programa a traves de su nombre, pasándole los parámetros que necesita.
Muchas veces el código de la función se encuentra en archivos externos, por lo que se hace necesario pasarle a nuestro programa el nombre del archivo externo en donde se encuentra : #include<nombre-archivo>.
Por lo general en C estos archivos tienen extension .h.
Una de las más conocidas y que contiene las funciones que permiten leer y escribir por consola es <stdio.h>

Ejemplo: printf(“hola mundo”);

El lenguaje C
Sentencias, identificadores, palabras reservadas, variables.

- Toda sentencia termina con el símbolo ;

- Los bloques se delimitan con {}

- C distingue mayúsculas de minúsculas, todas las palabras reservadas están en minúscula.

- Las palabras reservadas no pueden ser utilizadas en nombres, a modo de ejemplo algunas de las palabras reservadas son: char, int, const, do, else, void, for, if, return ....

- Los comentarios se ponen entre /* */, lo que aparezca entre dichos símbolos será ignorado por el compilador, es fundamental el uso de comentarios para la documentación del programa que estamos efectuando.

El lenguaje C
Sentencias, identificadores, palabras reservadas, variables.

- Las variables son elementos que identifican los valores que utilizamos en el programa.

- Las variables permiten almacenar valores que pueden variar durante la ejecución del programa.

- En C las variables se almacenan (casi siempre) en la memoria del ordenador.

- Las variables ocupan y reservan posiciones de memoria. Estan formados por espacios medidos en bytes de la memoria, en los cuales se almacenarán los valores que utilicemos.

- Las variables se identifican con nombres, a esto se le llama identificadores de variables. Por ejemplo si dinero es un identificador de variable numérica, en un momento guardara un valor, en otro momento guardara otro valor y así dependiendo de lo que requiera el programa.

El lenguaje C
Sentencias, identificadores, palabras reservadas, variables.

- Los identificadores se usan en los nombres que les damos a las variables y a las funciones.

- El C es sensible al tamaño, cuidado con las mayúsculas y minúsculas, no es lo mismo Apynom, que apynom que APYNOM.

- El límite de tamaño de los identificadores es de 32 caracteres (por lo general). El compilador puede no validarlo, pero no funcionará correctamente.

- Los identificadores deben comenzar por una letra o por el signo subrayado.

- Solo se admiten letras del abecedario ingles y el caracter de subrayado.

El lenguaje C
Declaración de variables

- Todas las variables en C deben ser declaradas antes de ser usadas. El objetivo es informar al compilador el tipo de representación que deberán tener los datos y de reservar de antemano (por ahora) el espacio en memoria para almacenar su contenido.

- Una declaración de tipo es la definición del tipo de datos que va a contener una variable. Forma general:

  tipo lista_de_variables;
  tipo: tipo de datos valido en C
  lista_de_variables: uno o mas identificadores separados por comas.

- En C una variable se puede declarar en cualquier lugar (antes de usarla), pero es muy buena practica declararlas al inicio del bloque ó función en el que se van a utilizar.

El lenguaje C
Declaración de variables

- Los tipos de datos básicos del C son:

tipo de datos	rango de valores posibles	tamaño en bytes
char	0 a 255 (o caracteres simples del código ASCII)	1
int	-32768 a 32767 (algunos compiladores consideran este otro rango: -2.147.483.648 a -2.147.483.647	2 (algunos 4)
float	3.4E-38 a 3.3E+38	4
double	1.7E-308 a 1.7E+308	8
void	sin valor	0

Esos rangos y tamaños son clásicos, pero el rango que pueden almacenar y el tamaño dependerá de la máquina en la que se este trabajando.

Los números enteros se representan con int y los números reales con float o double (en este último se pueden representar números más altos y con mayor precisión)

Los char permiten guardar un carácter, pero para C, un número puede representar también un carácter (representa el código en la tabla ASCII)

El lenguaje C
Declaración de variables

Ejemplos de declaración:

int i, z;

char inicial_nombre;

- Modificadores de tipo: existen formas de hacer variar el funcionamiento de los tipos de datos. Algunos de los más utilizados:

- unsigned: permite utilizar el rango de números negativos para incrementar el de positivos.

- signed: permite que el tipo admita valores negativos.

- long: aumenta el rango del tipo.

- short: achica el rango.

Algunos de los mas usados:

tipo de datos	rango de valores posibles	tamaño en bytes
signed char	-128 a 127	1
unsigned int	0 a 65335	2
long int (o **long** a secas)	-2147483648 a 2147483647	4
long long	-9.223.372.036.854.775.809 a 9.223.372.036.854.775.808 (casi ningún compilador lo utiliza, casi todos usan el mismo que el **long**)	8
long double	3.37E-4932 a 3,37E+4932	10

El lenguaje C
Asignación de variables

- Forma general de asignación:

  variable = expresión;

  variable: es un nombre de variable.
  expresión: es una constante o cualquier operación (simple o compleja).

  Ejemplos: x=3;
  x=x+1;
  inicial_nombre=‘a’;
  a=b;
  b=aux;

  También se puede declarar e inicializar a la vez una variable:
  Ejemplo: int a=8;

El lenguaje C
Asignación de variables

- El separador de decimales es el punto.
- Si un número entero se escribe anteponiendo un 0, se entiende que es un numero octal.
- Si un número entero se escribe anteponiendo 0x se entiende que es un numero hexadecimal.
- Las variables carácter simple (char) se escriben entre comillas simples ‘j’.
- Las variables texto, (char[longitud máxima de la cadena]), se escriben entre comillas dobles ejemplo “Hola”.
- Existen formas de almacenar caracteres especiales, por ejemplo como almacenamos en una variable char una comilla simple?

  char barra=‘’’; genera error.

Para poder almacenarlos, existe lo que se llama secuencias de escape:

\’ : para ‘
\” : para “
\\: para \
\n: nueva línea
\t: tabulación
\r: retorno de carro

El lenguaje C
Operadores principales

- Aritméticos:

operador	significado
+	Suma
-	Resta
*	Producto
/	División
%	resto (módulo)
++	Incremento
--	Decremento

- Relacionales (los utilizaremos cuando veamos sentencias de control):

operador	significado
>	Mayor que
>=	Mayor o igual que
<	Menor que
<=	Menor o igual que
==	Igual que
!=	Distinto de

- Lógicos: (los utilizaremos cuando veamos sentencias de control):

operador	significado
&&	Y (AND)
||	O (OR)
!	NO (NOT)

El lenguaje C
Operadores principales

- De asignación: el principal es el =.

Adicionalmente se permiten las siguientes abreviaciones:

operador	significado
+=	Suma y asigna
-=	Resta y asigna
*=	Multiplica y asigna
/=	Divide y asigna
%=	Calcula el resto y asigna

- Operador ?

Permite escribir expresiones condicionales

expresion_a_evaluar?si_verdadero:si_falso

Ejemplo:

z=(x>10?'V':'F');

- Operador sizeof

Permite conocer el tamaño en bytes que ocupa una variable en memoria

Ejemplo:

x=sizeof(y);

El lenguaje C
Precedencia de los operadores

- La precedencia de una operación es la prioridad de esa operación dentro de otra operación mayor.
- La precedencia va a depender de las operaciones que se estén realizando.
- La precedencia de los operadores comunmente más utilizados es:

1	() paréntesis
2	++ -- - (unario, el menos se utiliza para cambiar de signo)
3	* / %
4	+ -
5	=

Nota: aquí no estamos teniendo en cuenta operadores lógicos y relacionales.

- En general las expresiones se evalúan de izquierda a derecha, pero cuidado porque algunos compiladores pueden optimizar código y romper dicha regla.

Ejemplo: cual es el valor resultante de:

```
x=10;
y=3;
a=4;
z=-x*y+(10/y--)*5-a++
```

PROBAR EN PC

El lenguaje C
Conversión de tipos en expresiones

- En C, cuando tenemos una expresión, todos los tipos se convierten al mismo tipo correspondiente al del operando de mayor precisión.
Por ejemplo si el operando mayor es double, se convierte toda la expresión a double.

- Pero cuando se asigna el valor resultante, se convierte al tipo de la variable donde se almacena el resultado.
Por ejemplo:

```
int x=9.5*2
```

El 9.5 es double, mientras que el 2 es entero, por lo que el resultado será double, sin embargo x es entero, por lo que el resultado se convertira a entero.

El lenguaje C
Constantes

- Las variables declaradas con “const”, son constantes, lo que significa que son de solo lectura.

- El valor que se le asigna cuando se la declara, no varia durante el programa.

- Es obligatorio asignar un valor en la declaración.

- Ejemplo:

```
const  double pi=3.141592;
```

El lenguaje C
Concepto de alcance/ámbito – declaraciones globales y locales

- Habíamos mencionado que en C, las variables se pueden declarar en cualquier lugar del programa.

Sin embargo las variables tienen un ciclo de vida.
Dependiendo del lugar (bloque) en que se declaren, estas pueden tener un ámbito local o global. Por regla general, cuando se termina el bloque en que se declaró , la variable muere.
Variable global: en principio diremos que se declaran fuera del main.
conocida por todas las funciones.
Se puede utilizar en cualquier parte del programa.
Variable local: se declara dentro de un bloque en particular.
Conocida únicamente por dicho bloque.
Variable declarada en los parámetros formales de una función: idem local.

- Con los criterios anteriores nunca podremos tener dos variables con el mismo nombre dentro de un bloque, pero si en bloques diferentes. Sin embargo, declarar variables con el mismo nombre en diferentes bloques, puede generar problemas al momento de su uso

Nota: por bloque se entiende el código que está entre {}.

El lenguaje C
Concepto de alcance/ámbito – declaraciones globales y locales

- Ejemplos:

```c
#include <stdio.h>
int a=3; //La variable "a" es global
int main(){
    printf("%d",a);
  return 0;
}
```

```c
void func(){
   int a;
a=13;
   {
        int b;
        b=8;
   }/*Aquí muere b*/
   a=b; /*Error! b está muerta*/
}/*Aquí muere a*/
```

El lenguaje C
Entrada de datos por teclado y salida de datos por pantalla

- La entrada y salida de datos se realiza principalmente mediante dos funciones.

Ambas necesitan el archivo de biblioteca <stdio.h> (#include<stdio.h>)

- **Salida de datos: función printf()**

Si se desea sacar un texto literal :

Ejemplo: printf(“hola mundo”);

Si se necesita mostrar el contenido de variables :

printf(cadena de control, lista de argumentos)

cadena de control: es una cadena con los códigos que controlarán la forma como se desplegarán los resultados.

lista de argumentos: es la lista con las variables o las expresiones que serán desplegadas.

Ejemplo:

```
int edad=18;
printf(“Tengo %d años”,edad);
```

El lenguaje C
Entrada de datos por teclado y salida de datos por pantalla

Ejemplo:

```
int valor=10;
char caracter='a';
printf("Imprimo un número %d  y una letra %c",valor,caracter);
```

- Mediante printf() se pueden especificar anchos para los textos, alineaciones, etc.

Ejemplo:

```
int a=127, b=8, c=76;
printf("%4d\n",a);
printf("%4d\n",b);
printf("%4d\n",c);
/* Sale:
 127
    8
   76
*/
```

El lenguaje C
Entrada de datos por teclado y salida de datos por pantalla

Código	Formato
%c	Carácter
%d	Enteros decimales con signo
%i	Enteros decimales con signo
%e	Notación científica (e minúscula)
%E	Notación científica (E mayúscula)
%f	Coma flotante
%g	Usar %e ó %f el que resulte más corto
%G	Usar %E ó %f el que resulte el más corto
%o	Octal sin signo
%s	Cadena de caracteres
%u	Enteros decimales sin signo
%x	Hexadecimales sin signo (letras minúsculas)
%X	Hexadecimales sin signo (letras mayúsculas)
%p	Mostrar puntero
%n	El argumento asociado es un puntero, a entero al que se asigna el número de caracteres escritos
%%	Imprimir el signo %

El lenguaje C
Entrada de datos por teclado y salida de datos por pantalla

- Para cada tipo de dato existe una cadena de control específica, así por ejemplo para representar datos long int se utilizará %ld ó %li.

- **Entrada de datos: función scanf()**

Para cargar variables :

scanf(cadena de control, lista de argumentos)

cadena de control: es una cadena con los códigos que controlarán la forma en la que se reciben los datos del dispositivo de entrada.

lista de argumentos: es la lista con las direcciones de las variables que serán leídas.

Ejemplo:

```
int edad;
scanf("%d",&edad);
```

El lenguaje C
Entrada de datos por teclado y salida de datos por pantalla

Código	Formato
%c	Carácter
%d	Enteros decimales con signo
%i	Enteros decimales con signo
%e	Notación científica
%f	Punto flotante
%o	Octal sin signo
%s	Cadena de caracteres
%x	Hexadecimales sin signo
%n	Recibe un valor entero igual al número de caracteres leídos
%u	Lee un entero sin signo
%[]	Muestra un conjunto de caracteres

El lenguaje C
Diferenciación variables carácter de variables cadena

- Variables carácter : almacenan un solo carácter.

  char nombre_variable;

  Para asignar valores:

  nombre_variable=nombre_otra_variable_carácter;

  nombre_variable='a';

  scanf("%c",&nombre_variable);

  nombre_variable=getchar();

  nombre_variable=getche();

  nombre_variable=getch();

- Variables cadena: almacenan un conjunto de caracteres. También son conocidas como arreglos de caracteres o string.

  char nombre_variable[longitud_maxima_de_cadena];

  Para asignar valores:

  char nombre_variable[n]="caracter1caracater2caracter3"; (en declaración)

  nombre_variable={'caracter1, 'caracter2', 'caracter3'};

  nombre_variable[0]='caracter1';

  nombre_variable[1]='caracter2´;

  nombre_variable[2]='caracter3;'

  nombre_variable[3]='\0'; (toda cadena termina con el carácter nulo, el que debe ser contemplado al calcular el tamaño de la cadena)

El lenguaje C
Diferenciación variables carácter de variables cadena

- Variables cadena:
  Para asignar valores: (continuación)
  scanf(“%s”,nombre_variable); no lleva el &.
  gets(nombre_variable);

- Algunas funciones comunes asociadas a variables cadena:
  strcpy(cadena1,cadena2);
  strcat(cadena1,cadena2);
  strlen(cadena);
  strcmp(cadena1,cadena2);

  Para utilizar dichas funciones hay que indicarse #include<string.h>

  NOTA: el manejo y uso de cadenas son fuente importante de errores.

Muy resumidamente

- Todos los programas en C tienen una función main(), que marca el punto en el cual comienza la ejecución del programa.

- Los programas deben declarar todas las variables antes de usarlas.

- El lenguaje C soporta una amplia variedad de tipos (char, int, float, etc.).

- La función printf vuelca información en la pantalla.

- La función scanf toma datos a partir del teclado.

- La ejecución del programa termina cuando encuentra la llave de cierre } de la función main.

FIN