

Introducción al Lenguaje C

LABORATORIO DE SISTEMAS DE OPERACIÓN I
(ci-3825)

Prof. Yudith Cardinale

Introducción al Lenguaje C

- Definición de estructuras con apuntadores

- Definición de tipos:

```
typedef struct {  
    int x;  
    int y;  
} PUNTO;
```

```
typedef struct {  
    PUNTO p1, p2;  
} RECTANGULO;
```

- Declaración de variable:

```
RECTANGULO *rectang;
```

- Abrir espacio en memoria:

```
rectang = (RECTANGULO *)  
    malloc(sizeof(RECTANGULO));
```

- Acceso

```
rectang->p1.x=1;  
rectang->p1.y=1;  
rectang->p2.x=-1;  
rectang->p2.y=-1;
```

- Liberar espacio de memoria:

```
free(rectang);
```

Introducción al Lenguaje C

- Definición de estructuras con apuntadores

- Definición de tipos:

```
typedef struct arbol{  
    int nodo;  
    struct arbol *izq;  
    struct arbol *der;  
} ARBOL;
```

- Declaración de variable:

```
ARBOL *miarbol;
```

- Abrir espacio en memoria, acceso y free:

```
miarbol = (ARBOL *) malloc(sizeof(ARBOL));  
miarbol->izq=NULL;  
miarbol->der=NULL;  
miarbol->nodo=0;  
...  
free(miarbol);
```

* se usa para declarar apuntadores y para devolver el contenido de la dirección de memoria
& devuelve la dirección de memoria de una variable o función
void * es un apuntador a cualquier cosa.

Introducción al Lenguaje C

- Definición de Arreglos

- Declaración y uso:

```
1. int arr[100], a;
```

```
...
```

```
for (a=0;a<100;a++)
```

```
arr[a]=a;
```

```
2. int num[]={1,2,3};
```

```
int matrix[2][2]={1,2,3,4};
```

```
...
```

```
printf("%d - %d",num[0], matrix[0][0]);
```

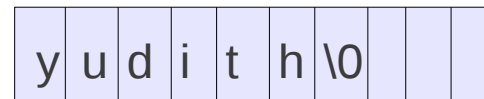
- string = arreglo de caracteres:

```
char nombre[10]= "yudith";
```

```
char *cadena;
```

```
....
```

```
cadena = (char *)malloc(sizeof(char)*MAXCAR);
```



El nombre de una variable declarada como arreglo, devuelve su dirección (no es necesario usar & si se pasa como parámetro a una función)

Introducción al Lenguaje C

- Librerías

De entrada/salida estándar:

- getchar();
- void putchar(int c)
- printf("formato con caracteres ascii", var1, var2, var3, ...)
- scanf("formato puro", var1, var2, var3, ...)

Formato:

`%d` : entero `%o`: octal `%x`: hexadecimal
`%s` : cadena de caracteres `%c`: char `%f`: reales (float,...)
`%n.mf`: n es el largo total, m nro. de decimales.

- Ejemplos:

```
printf("edad = %d, nombre = %s", edad, nombre);  
scanf("%d %d %s", &id, &edad, nombre);
```

Introducción al Lenguaje C

- Librerías

De entrada/salida en memoria:

- `sprintf(variable_salida, "formato puro", var1, var2, var3, ...)`
- `sscanf(variable_entrada, "formato puro", var1, var2, var3, ...)`

- Ejemplos:

1. `char salida[100], entrada[]="Hola mundo";`
`int i=1, j=2;`
`sprintf(salida, "%s %d %d", entrada, i, j);`

2. `char texto[]="Operativos_I 2 50";`
`char materia[20];`
`int seccion, t_alumnos;`
`sscanf(texto, "%s %d %d", materia, &seccion, &t_alumnos);`

Introducción al Lenguaje C

- Librerías

De entrada/salida en archivos:

- FILE *fp; // declaración de apuntador de archivo
- fp=fopen("nombre_archivo", "modo") //modo=r,w,a,r+,w+,a+...
- fprintf(apunt_archivo,"formato puro",var1,var2,var3,...)
- fscanf(apunt_archivo,"formato puro",var1,var2,var3,...)
- fclose(apunt_archivo);

- Ejemplo:

```
main () {
    FILE *fp; int altura; float duracion;
    if ((fp= fopen("melodia.txt","r")) == NULL)
        perror("fopen: ");
    while ((fscanf(fp,"%d %f", &altura,&duracion)) != EOF) {
        printf("Valores leidos: %d %4.2f\n",altura,duracion);
    }
    fclose(fp);
}
```

Introducción al Lenguaje C

- **Uso de perror():** función de C para el manejo de errores, muestra un mensaje por **stderr** que describe el error más reciente ante la llamada de una función de librería o una llamada al sistema.
- **Errno:** variable de ambiente que contiene el número del último error ocurrido en una llamada a función de librería o una llamada al sistema.

```
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
main() {
    FILE *fp;
    char filename[80];
    printf("Enter filename: ");
    gets(filename);
    if (( fp = fopen(filename, "r")) == NULL) {
        perror("You goofed!");
        printf("errno = %d.\n", errno);
        exit(1);
    } else {
        puts("File opened for reading.");
        fclose(fp);
    }
    return(0);
}
```

Posibles salidas:

```
Enter file name: list19_4.c
File opened for reading.
```

```
Enter file name: notafile.xxx
You goofed!: No such file or directory
errno = 2.
```