

Funciones en C

Pasaje por parámetros
Referencia

AMBITO DE LA VARIABLE

Todas las variables tienen un ambito donde “viven”

Este lugar donde viven se conoce como ámbito.

Las variables solo pueden ser invocadas desde el Lugar donde “viven”.

Aquí solo veremos el ámbito de las funciones.

Construcción de la función

```
tipo_del_resultado NOMBRE(tipo_param1 param1, tipo_param2 param2, ... )  
{  
    /* Cuerpo de la función */  
}
```

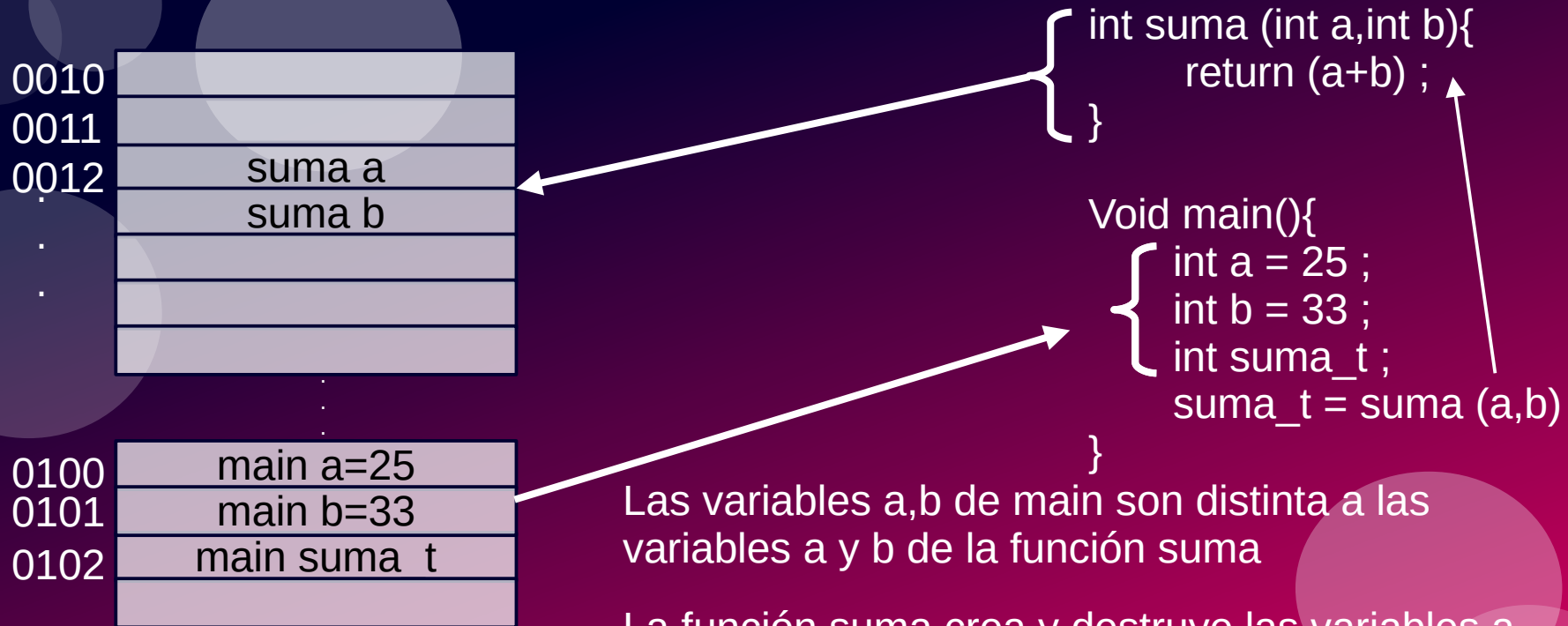
Ejemplos de funciones

```
int suma(int a, int b)  
{  
    return (a+b);  
}
```

```
int mul(int a, int b)  
{  
    return (a*b);  
}
```

```
int mayor(int a, int b)  
{  
    int mayor ;  
    if (a>b){  
        mayor = a ;  
    }else {  
        mayor = b  
    }  
    return mayor;  
}
```

Pasaje de parámetros por valor



Las variables a,b de main son distinta a las variables a y b de la función suma

La función suma crea y destruye las variables a y b en otra zona de la memoria del dispositivo
Esto se conoce como pasaje por parámetros

PASAJE POR REFERENCIA

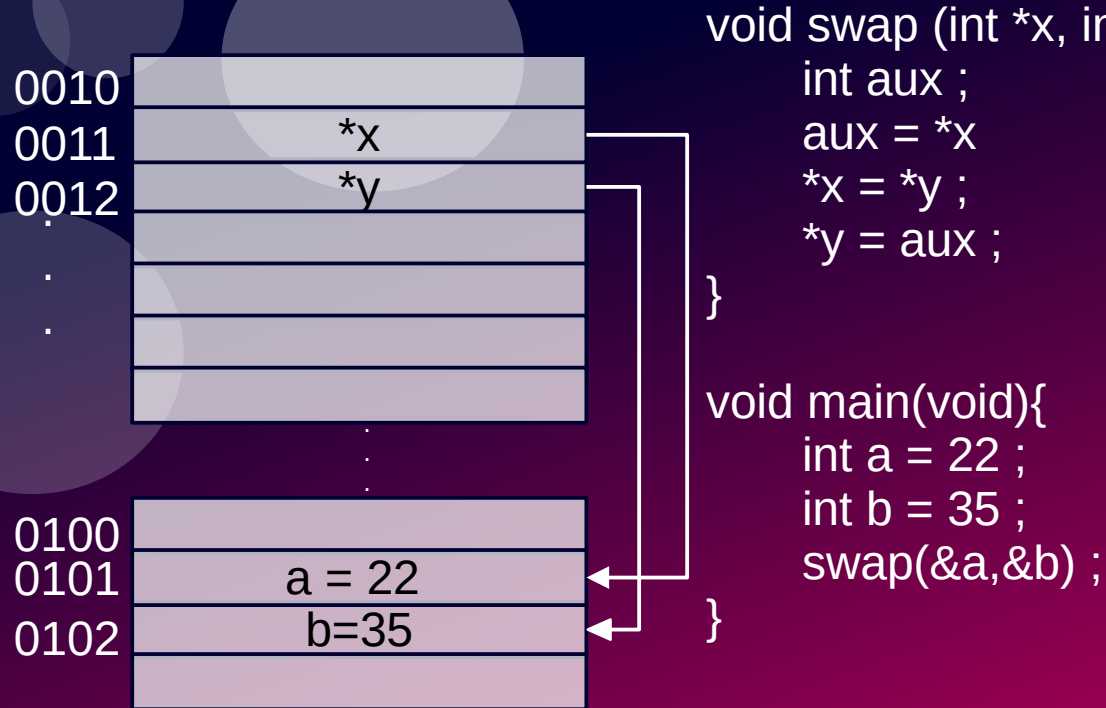
Supongamos que quiero intercambiar dos valores: a y b. El contenido de a lo quiero volcar en b y viceversa

Quiero realizar esta operación muchas veces, entonces debemos construir una función que realice este paso.

En clases pasadas se vieron punteros: en lugar de pasarle una copia, le pasamos la direcciones de memoria que queremos cambiar. Entonces deja de ser una copia, y se trabaja con la variable.

```
void swap (int *x, int *y){  
    int aux ;  
    aux = *x  
    *x = *y ;  
    *y = aux ;  
}
```

Como funciona la función que acabamos de crear



OTROS USOS DE PUNTEROS:

Retornar un vector
Obtener configuraciones de algún periférico

Tarea de casa

- Investigue la diferencia entre “void funcion(void) y “void funcion()
- Ejemplifique una función de este tipo que conozca