

# **Subrutinas**

---

**Conceptos y necesidad**

# Definición

---

- Los programas de alto nivel hacen uso de estos bloques de código a los que se les pasan unas variables (parámetros) con las que ejecuta una operación y calcula unos resultados (funciones, procedimientos o métodos).
- **Subrutina:** porción de código que realiza una operación en base a unos valores dados como parámetros y que puede ser invocado desde cualquier parte del código, incluso desde sí misma (contexto de lenguaje ensamblador)

## **Subrutina - ventajas**

---

- División del problema en tareas más fáciles de escribir y depurar
- Evita código redundante
- Encapsulado del código: las tareas se comunican a través del paso de parámetros y resultados, por lo que un cambio en una de las tareas no implica cambios en el resto del programa
- Permite su reutilización en más de un programa (bibliotecas)

# Subrutina - Generalidades

---

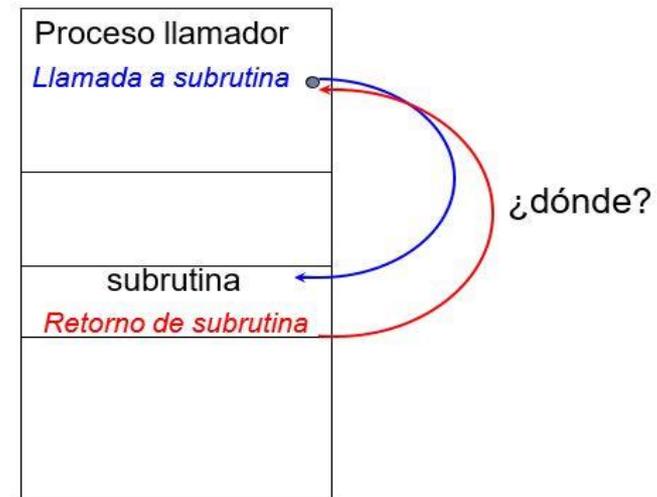
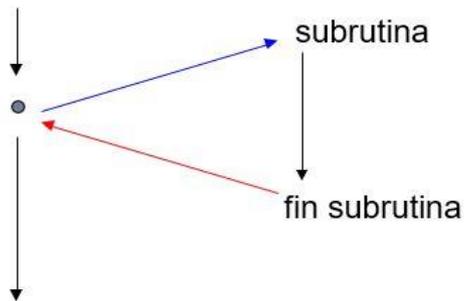
- Resolver un mismo problema ante diferentes datos (parámetros) dotarla de generalidad:
  - Llamada y retorno de subrutina
  - Paso de parámetros y resultados

# Subrutina: llamada y retorno

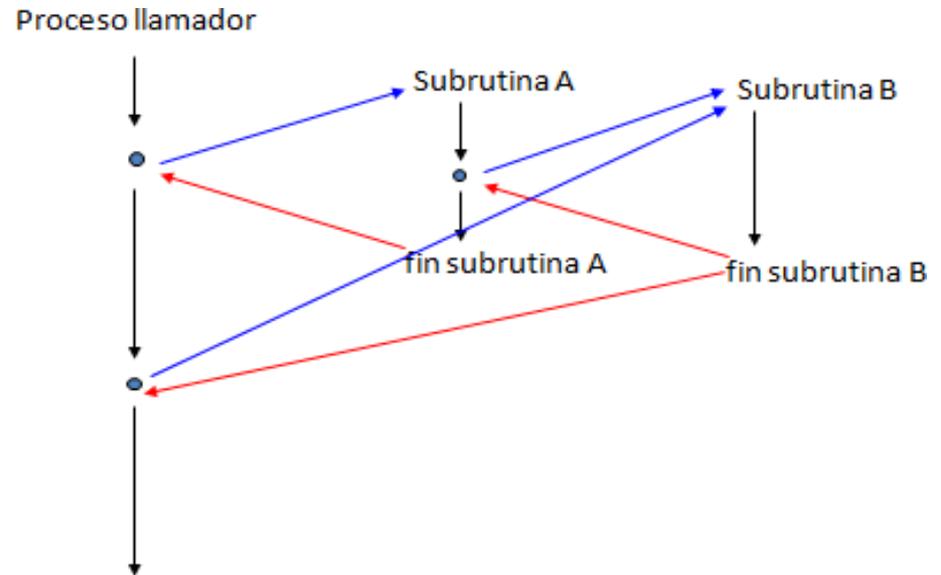
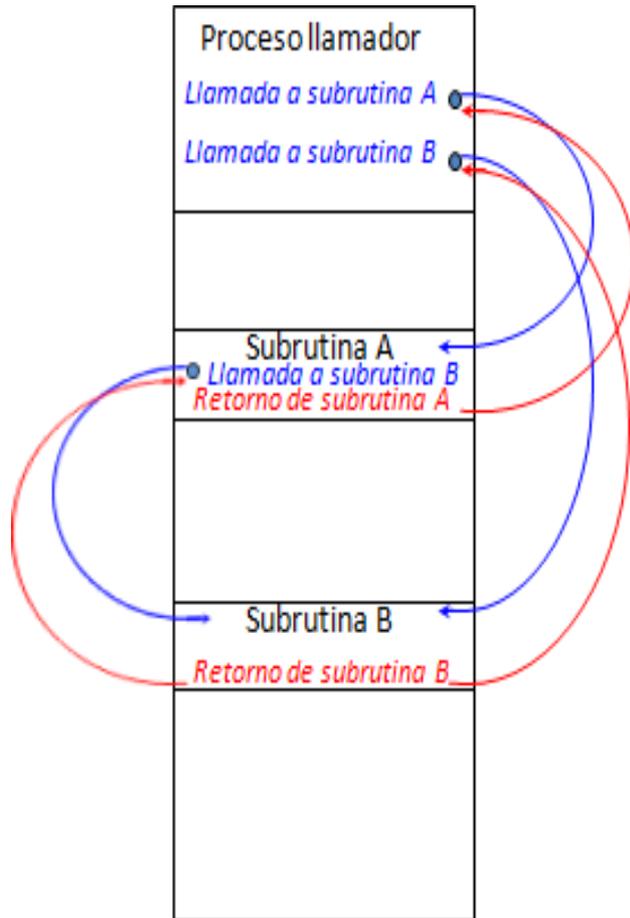
---

- El proceso llamador, llama a la subrutina, pasándole unos parámetros para que realice los cálculos y le devuelva un resultado, tras lo cual seguirá su ejecución

Proceso llamador



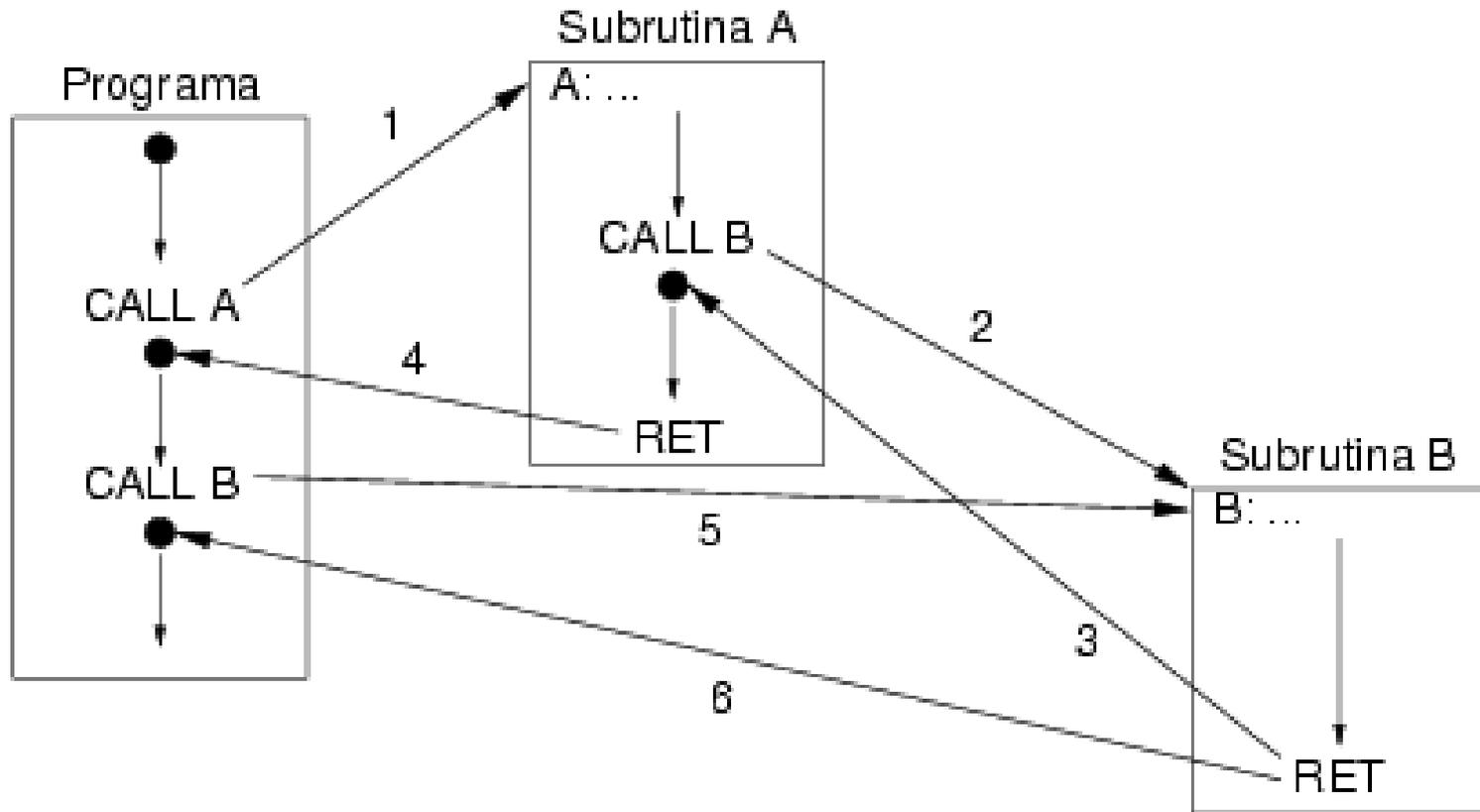
# Subrutina: llamada y retorno



Necesitamos un mecanismo que permita la invocación anidada de subrutinas.

# Subrutina: llamada y retorno

---



# Subrutina: llamada y retorno

---

- Llamada a subrutina
  - CALL etiqueta
- Retorno de subrutina
  - RET (no tiene parámetros)
- Etiqueta: en nuestro caso sería la dirección de la memoria donde empieza la subrutina.