

Prácticas de Arquitectura y Organización de Computadores



Javier Plaza Miguel
Laboratorio de Computación Hiperespectral



Universidad de Extremadura



Prácticas de AOC. – Sesión 3.- Gestión de Subrutinas

- * **Subrutina:** conjunto separado de instrucciones separadas del main
- * **Ventajas:**
 - * Mejor estructuración
 - * Facilidad de escritura, depuración y prueba
 - * Evita replicación de código
 - * Creación de libraries (reutilización de código)
- * Para poder usarlas, el procesador debe proporcionar mecanismos para:
 - * Llamar a la subrutina
 - * Pasar parámetros a la subrutina
 - * Devolver resultados al programa invocador
 - * Devolver el control al programa invocador



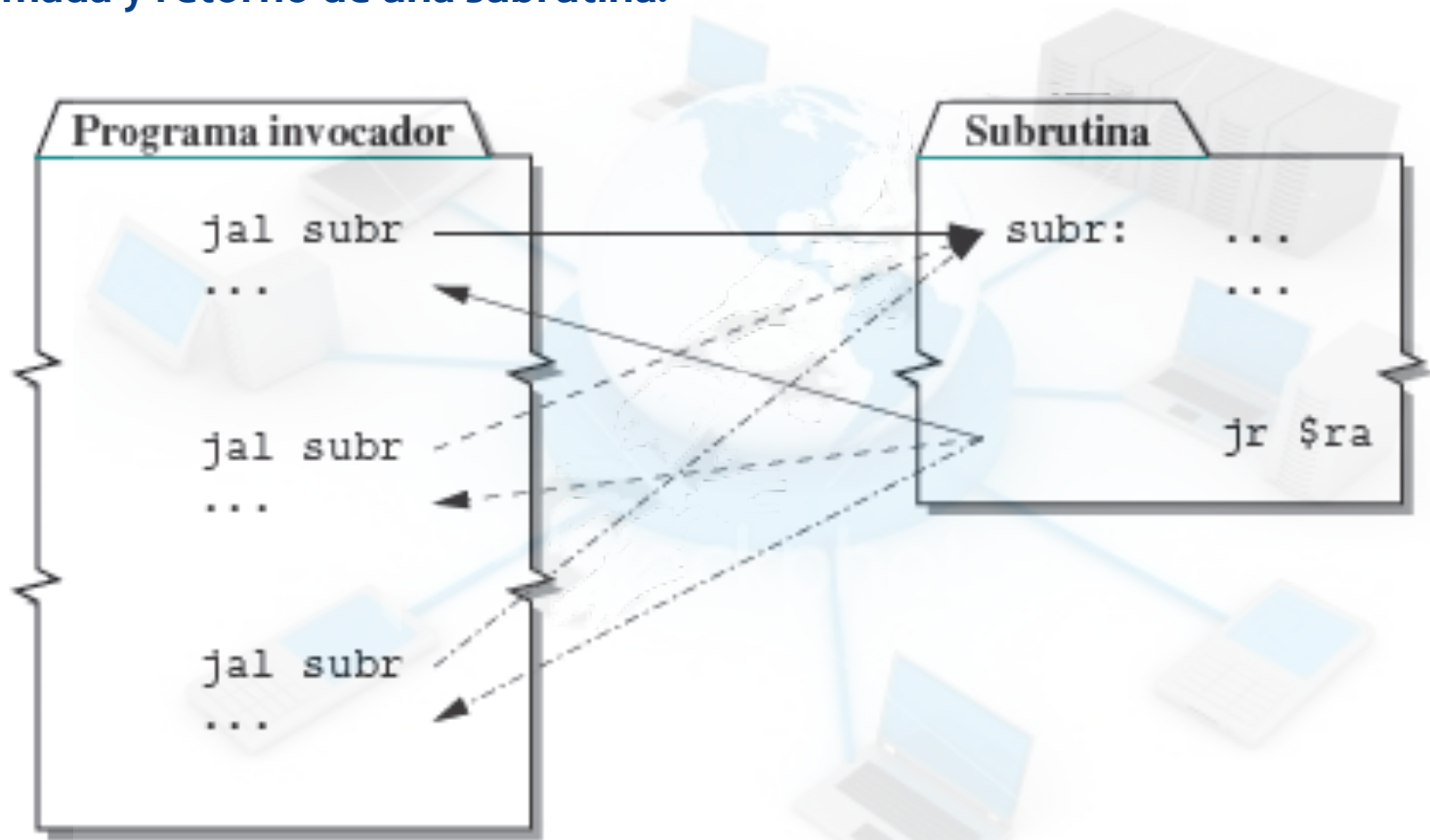
Prácticas de AOC. – Sesión 3.- Gestión de Subrutinas

- * **Llamada y retorno de una subrutina:**
- * Para **llamar** una subrutina, utilizaremos la instrucción de salto incondicional **jal**
- * **jal etiqueta** → almacena la dirección de memoria de la siguiente instrucción en registro return address (\$ra): $\$ra \leftarrow PC+4$ y justo después realiza un salto incondicional a la dirección especificada por etiqueta.
- * Para **retornar** de una subrutina, utilizaremos la instrucción de salto incondicional **jr**
- * **jr registro** → salta a la dirección de memoria almacenada en el registro que recibe como parámetro. Así pues, si queremos volver a la instrucción siguiente a la que realizó la llamada a la subrutina, debemos saltar a la dirección almacenada en \$ra.



Prácticas de AOC. – Sesión 3.- Gestión de Subrutinas

* Llamada y retorno de una subrutina:



Este sencillo mecanismo funcionará siempre que la subrutina no altere el contenido de `$ra`



Prácticas de AOC. – Sesión 3.- Gestión de Subrutinas

- * **Paso de parámetros:**
- * Los parámetros intercambiados pueden ser de 3 tipos:
 - * **Entrada:** proporcionan información del programa invocador a la subrutina
 - * **Salida:** devuelven información de la subrutina al programa invocador
 - * **Entrada/Salida:** proporcionan información del programa invocador a la subrutina y devuelven información de la subrutina al invocador.
- * **Convenio en MIPS32:** podemos pasar parámetros mediante los registros o mediante memoria (pila de los programas)
 - * **Registros:** por convenio, los parámetros de entrada se pasarán en los registros $\$a1$ y los parámetros de salida en los registros $\$v1$
 - * **Pila:** lo veremos en la siguiente sesión



Prácticas de AOC. – Sesión 3.- Gestión de Subrutinas

* Paso de parámetros:

* La transferencia de los parámetros puede producirse de una de estas dos formas:

* **Por valor:** cuando lo que transferimos es el dato en sí



* **Por referencia:** cuando transferimos la dirección de la posición de memoria que contiene el dato.

