

Subrutinas

Subrutinas de Lenguajes de Alto Nivel

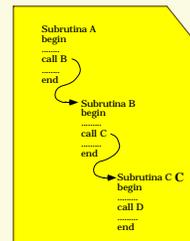
- ✓ Se promueve el uso de subrutinas para modularizar los programas.
- ✓ La modularización hace que los programas sean más fáciles de entender y de mantener.
- ✓ Las subrutinas son utilizadas frecuentemente
- ✓ Las actividades asociadas con las subrutinas toman un tiempo considerable del tiempo total de ejecución de un programa.

Actividades Asociadas con Subrutinas

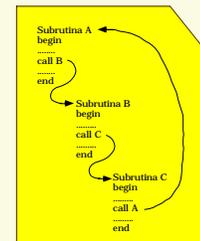
- ✓ Transferir parámetros
- ✓ Salvar información del estado (registros, PSW, etc.)
- ✓ Transferir control a la subrutina y preservar la localización de retorno.
- ✓ Tener acceso a las variables locales de la subrutina llamada y a los parámetros transferidos a éstas.

Nesting y Recursión de Subrutinas

Nesting



Recursión

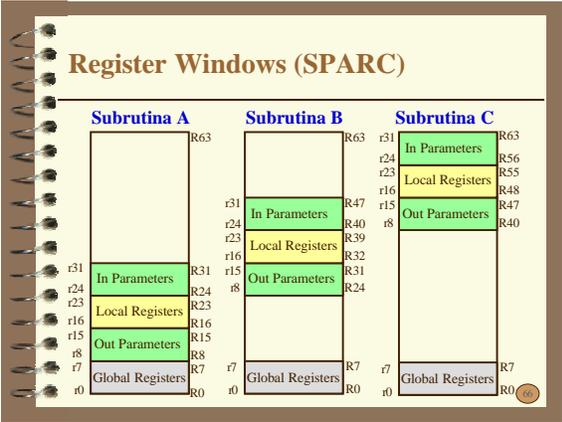


Mecanismos para Apoyar Subrutinas

- ✓ Register Windows
- ✓ Register Coloring
- ✓ In-Line Expansion
- ✓ Stack

Register Windows

- ✓ A cada subrutina se le asigna un grupo de registros (ventana) para sus variables locales, para recibir parámetros y para enviar parámetros a otra subrutina
- ✓ Una subrutina puede compartir registros con la subrutina que la llamó o con una que esta llame
- ✓ Todas las subrutinas comparten un grupo de registros globales
- ✓ La asignación de ventanas de registros se hace en demanda (cuando la subrutina es llamada durante la ejecución de un programa)



Overflow Trap

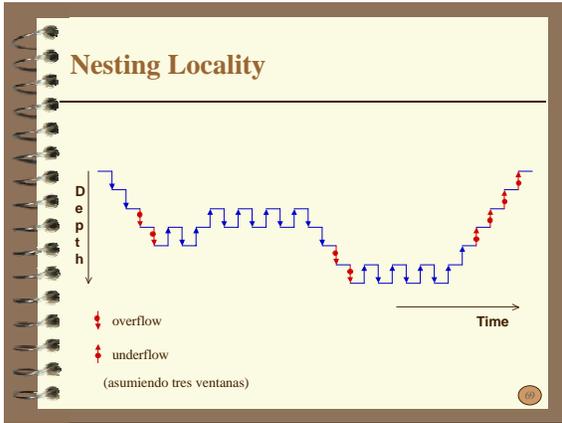
Cuando se llama a una subrutina y todas las ventanas han sido utilizadas

- **Genera un interrupt (trap) que brinca a una subrutina donde :**
 - se desaloja una de las ventanas enviándola a un stack en memoria
 - se le provee una ventana a la subrutina
- **El trap se limita a brincar**
- **El código del trap es responsabilidad de quien programa a nivel de lenguaje de máquina**

Underflow Trap

Cuando se retorna a una subrutina cuya ventana no se encuentra en los registros de register windows

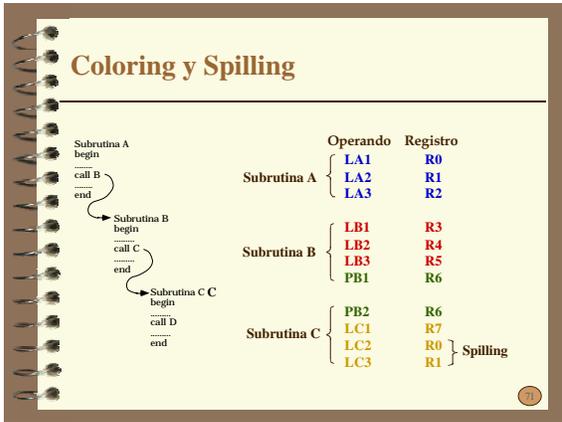
- **Genera un interrupt (trap) que brinca a una subrutina donde :**
 - se recuperara la ventana almacenada en el stack en memoria
 - se le provee una ventana a la subrutina
- **El trap se limita a brincar**
- **El código del trap es responsabilidad de quien programa a nivel de lenguaje de máquina**



Register Coloring

Es un algoritmo que aloja operandos locales y parámetros de subrutinas en registros para reducir el tráfico entre memoria y CPU.

- **La asignación de registros a operandos se hace durante la compilación**
- **Requiere de un número grande de registros**
- **La reducción en tráfico de memoria depende de cuan bueno es algoritmo prediciendo el comportamiento de un programa**

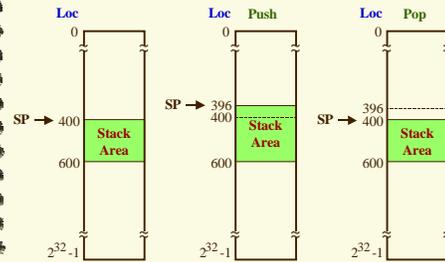


In-Line Expansion

Insertar el código que corresponde a una subrutina en lugar de la instrucción que se utilizaría para llamar la misma

- Se elimina la necesidad de:
 - brincar
 - guardar el estado
 - pasar parámetros
- El tamaño del programa puede incrementar

Stack



Manejo de Subrutinas Mediante Stack

