
Linkage e passaggio dei Parametri su Stack

Nuovo Corso di Calcolatori Elettronici I

Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli "Federico II"

DIS - Dipartimento di Informatica e Sistemistica - Università di Napoli



Svantaggio del pass. dei par. su reg. e su mem.

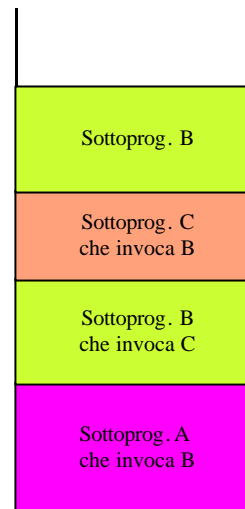
- Sia nel caso del passaggio dei parametri su registri che sulla memoria, uno svantaggio consiste nel non poter realizzare sottoprogrammi che possono essere invocati ricorsivamente.
- All'atto di una chiamata, infatti, le locazioni contenenti i dati vengono sovrascritte provocando la perdita irreversibile di eventuali dati presenti in esse.
- Ancora una volta, per risolvere questo problema, si può pensare a ricorrere ad una struttura a stack che contenga tutti i dati relativi a chiamate successive ad un sottoprogramma.
- La strutturazione dei programmi ci assicura che una struttura ad accesso LIFO sia sufficiente a tale scopo.

DIS - Dipartimento di Informatica e Sistemistica - Università di Napoli



III soluzione: Parametri in area a Stack

- È possibile passare i parametri di input/output in aree organizzate a stack.
- È una forma di allocazione dinamica della memoria.
- Lo stack tende a riempirsi all'ingresso di un sottoprogramma e torna a svuotarsi all'uscita di esso.
- Ogni sottoprogramma utilizzerà solo una serie di locazioni prossime alla testa dello stack, ignorando locazioni più in profondità appartenenti ad altri sottoprogrammi di livello inferiore.



DIS - Dipartimento di Informatica e Sistemistica - Università di Napoli



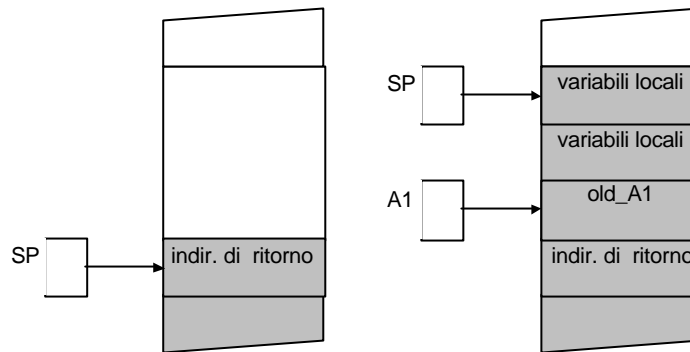
Parametri in area a Stack

- Perché un sottoprogramma possa invocare ricorsivamente se stesso, occorre che sia i parametri effettivi, sia le variabili locali siano allocate in un'area di memoria (detta *record di attivazione*) definita al momento della chiamata al sottoprogramma.
- Per facilitare la allocazione sullo stack del record di attivazione di un sottoprogramma, i progettisti del 68000 hanno dotato il processore delle istruzioni LINK A, *im* e UNLK A.
- L'istruzione LINK salva il contenuto del registro A sullo stack, carica in A il valore aggiornato dello Stack Pointer, ed infine incrementa SP dell'offset *im*. Sommando il valore negativo *im* al contenuto di SP, l'istruzione LINK riserva un'area di memoria di *im* byte sulla cima dello stack. Quest'area è utilizzata per l'allocazione delle variabili locali del sottoprogramma.
- L'istruzione UNLK A ripristina il contenuto del registro A e lo stato dello stack.

DIS - Dipartimento di Informatica e Sistemistica - Università di Napoli



Esempio: istruzione LINK A1,#-8



DIS - Dipartimento di Informatica e Sistemistica - Università di Napoli



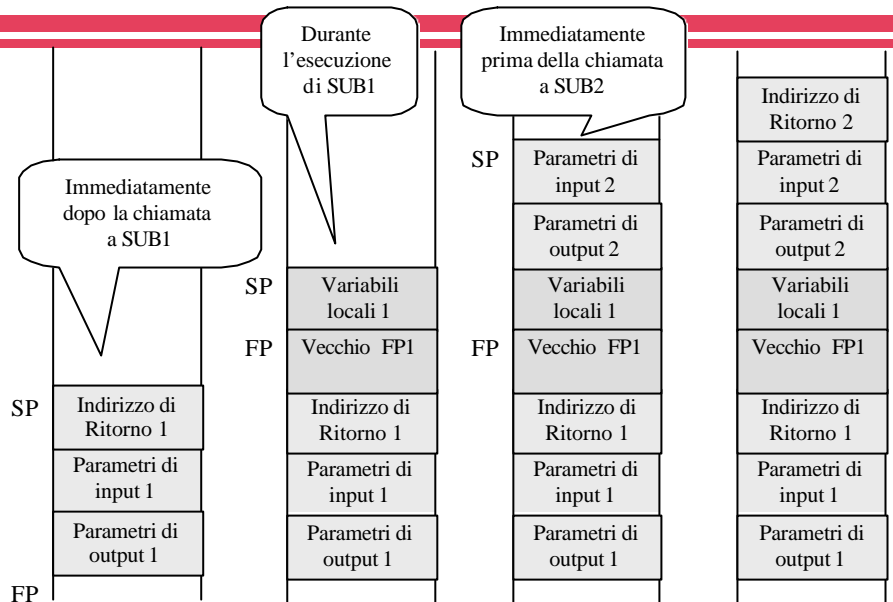
Parametri in stack

- I parametri di input, quelli di output e l'indirizzo di ritorno vengono messi in un'area di memoria organizzata a stack
- Il **programma chiamante**
 - » Riserva spazio sullo stack per i parametri di output
 - » Mette i parametri di input sullo stack
 - » Chiama la subroutine
- La **subroutine**
 - » Sceglie un registro da usare come Frame Pointer
 - » Inizializza FP a SP
 - » Esegue il proprio codice (utilizzando lo stack per memorizzare, eventualmente, il contenuto di registri che la procedura utilizza per portare a termine il proprio compito)
 - » Mette i parametri di output sullo stack
 - » Ripulisce lo stack
 - » Torna al programma chiamante

DIS - Dipartimento di Informatica e Sistemistica - Università di Napoli



Evoluzione dello stack



DIS - Dipartimento di Informatica e Sistemistica - Università di Napoli



Link and allocate

- Sintassi:
 - » LINK An,#<displacement>
- Funzionamento:
 1. Eseguo push su stack del contenuto del registro indirizzo specificato
 2. Il registro indirizzo specificato viene caricato con il nuovo valore dello stack pointer
 3. Il displacement viene esteso in segno e sommato a SP. Questo valore viene assegnato a SP.
 4. Durante l'esecuzione SP varia, mentre FP rimane costante

DIS - Dipartimento di Informatica e Sistemistica - Università di Napoli



LINK Link and allocate (1/2)

- **Operation:** $[SP] \leftarrow [SP] - 4; [M([SP])] \leftarrow [An];$
 $[An] \leftarrow [SP]; [SP] \leftarrow [SP] + d$
- **Syntax:** LINK An,#<displacement>
- **Sample syntax:** LINK A6,#-12
- **Attributes:** Size = word
- **Description:** The contents of the specified address register are first pushed onto the stack. Then, the address register is loaded with the updated stack pointer. Finally, the 16-bit sign-extended displacement is added to the stack pointer. The contents of the address register occupy two words on the stack. A negative displacement must be used to allocate stack area to a procedure. At the end of a LINK instruction, the old value of address register An has been pushed on the stack and the new An is pointing at the base of the stack frame. The stack pointer itself has been moved up by d bytes and is pointing at the top of the stack frame. Address register An is called the frame pointer because it is used to reference data on the stack frame. By convention, programmers often use A6 as a frame pointer.

DIS - Dipartimento di Informatica e Sistemistica - Università di Napoli



LINK Link and allocate (2/2)

- **Application:** The LINK and UNLK pair are used to create local workspace on the top of a procedure's stack. Consider the code:

```
Subrtn      LINK A6,#-12      Create a 12-byte workspace
            .
            MOVE D3,(-8,A6)  Access the stack frame via A6
            .
            UNLK A6          Collapse the workspace
            RTS              Return from subroutine
```

- **Condition codes:** X N Z V C

- - - - -

The LINK instruction does not affect the CCR.

DIS - Dipartimento di Informatica e Sistemistica - Università di Napoli



UNLK Unlink

- **Operation:** $[SP] \leftarrow [An]; [An] \leftarrow [M([SP])]; [SP] \leftarrow [SP] + 4$
- **Syntax:** UNLK An
- **Attributes:** Unsized
- **Description:** The stack pointer is loaded from the specified address register and the old contents of the stack pointer are lost (this has the effect of collapsing the stack frame). The address register is then loaded with the longword pulled off the stack.
- **Application:** The UNLK instruction is used in conjunction with the LINK instruction. The LINK creates a stack frame at the start of a procedure, and the UNLK collapses the stack frame prior to a return from the procedure.
- **Condition codes:** X N Z V C
- - - - -

DIS - Dipartimento di Informatica e Sistemistica - Università di Napoli



Gestione dello stack: conclusioni e osservazioni

- L'utilizzo dello stack per il passaggio dei parametri effettivi ad un sottoprogramma è una forma di allocazione dinamica usata sia da programmatori assembler che dai compilatori di linguaggi di alto livello come il Pascal ed il C.
- E' possibile pensare ad una convenzione per il passaggio dei parametri: tale convenzione assegna al chiamante la responsabilità di allocare sullo stack lo spazio richiesto dai parametri di output e nel caricare successivamente sullo stack i valori dei parametri di input.
- Il sottoprogramma crea uno *stack frame* mediante l'istruzione LINK, che inizializza un *frame pointer* ed alloca sullo stack spazio per le variabili locali. Il sottoprogramma può accedere ai parametri di scambio ed alle variabili locali mediante indirizzamento indiretto attraverso il *frame pointer*.
- Per l'accesso ai parametri di scambio di ingresso/uscita si usano valori di spiazzamento positivi. Eventuali variabili locali del sottoprogramma dovrebbero essere allocate nell'area al di sopra del *frame pointer*, e quindi sarebbero accedute mediante displacement a valori negativi.

DIS - Dipartimento di Informatica e Sistemistica - Università di Napoli



Esercizi

- Scrivere un programma che esegua la somma tra due interi. L'elaborazione principale avvenga nel corpo di una subroutine e i parametri tra il chiamante ed il chiamato vengano scambiati attraverso lo stack.
- Scrivere un programma per la ricerca di un elemento in un vettore di interi. Si utilizzi una subroutine per l'elaborazione. Alla subroutine si passi l'indirizzo di partenza del vettore, il suo numero di elementi ed il valore che si intende ricercare all'interno di esso. La subroutine restituisca l'indirizzo dell'elemento se esistente, 0 altrimenti. Il passaggio dei parametri avvenga attraverso lo stack.

DIS - Dipartimento di Informatica e Sistemistica - Università di Napoli



Parametri in stack – Vantaggi e svantaggi

- ↑ È possibile usare la stessa area di memoria sia per salvare gli indirizzi di ritorno che per passare i parametri di I/O
- ↑ È possibile usare lo stack per variabili temporanee
- ↑ Permette di realizzare la ricorsione

@ @ @ - L'allocazione della memoria è dinamica

- ↓ Per essere veloce, richiede risorse hardware extra

DIS - Dipartimento di Informatica e Sistemistica - Università di Napoli

