Passaggio dei parametri su stack

- È possibile passare i parametri di input/output in aree organizzate a stack
- È una forma di allocazione dinamica della memoria
- I parametri di input, quelli di output e l'indirizzo di ritorno vengono messi in un'area di memoria organizzata a stack

Passaggio dei parametri su stack

PROGRAMMA CHIAMANTE

1) Riserva spazio sullo stack per i parametri di output
2) Pone sullo stack i parametri di input
3) Salta alla subroutine (indirizzo di ritorno su stack)
4) Preleva i parametri di output dallo stack

SUBROUTINE

1) Preleva l'indirizzo di ritorno dallo stack (utilizzando l'offset relativo allo SP) e lo salva in un registro indirizzi
2) Preleva i parametri di input dallo stack (utilizzando gli offset relativi allo SP) e li copia nei registri dati/indirizzi
3) Effettua la propria elaborazione e calcola i parametri di output
4) Pone i parametri di output sullo stack nelle posizioni per essi riservate (utilizzando gli offset relativi allo SP)
5) Ripulisce lo stack
6) Salta all'indirizzo di ritorno

Passaggio dei parametri su stack - Esempio

Scrivere e assemblare un programma che calcola il prodotto scalare fra due vettori utilizzando una subroutine a cui vengono passati in ingresso i vettori V1 e V2 (indirizzo del vettore – 4 byte) e la loro dimensione DIM (2 byte).

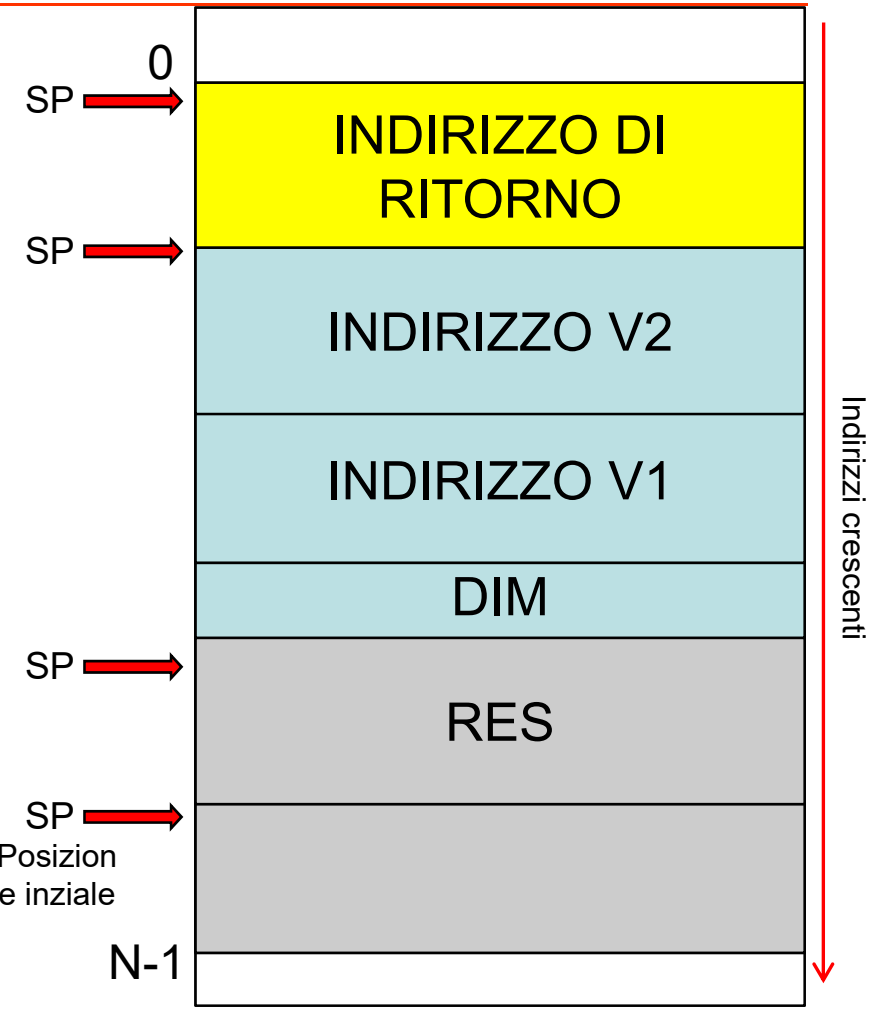
Si adotti il meccanismo di scambio dei parametri tramite stack.

Passaggio dei parametri su stack - Esempio

PROGRAMMA CHIAMANTE

(**1**) Riserva spazio sullo stack per i parametri di output
(**2**) Pone sullo stack i parametri di input
(**3**) Salta alla subroutine (indirizzo di ritorno su stack)
(**4**) Preleva i parametri di output dallo stack

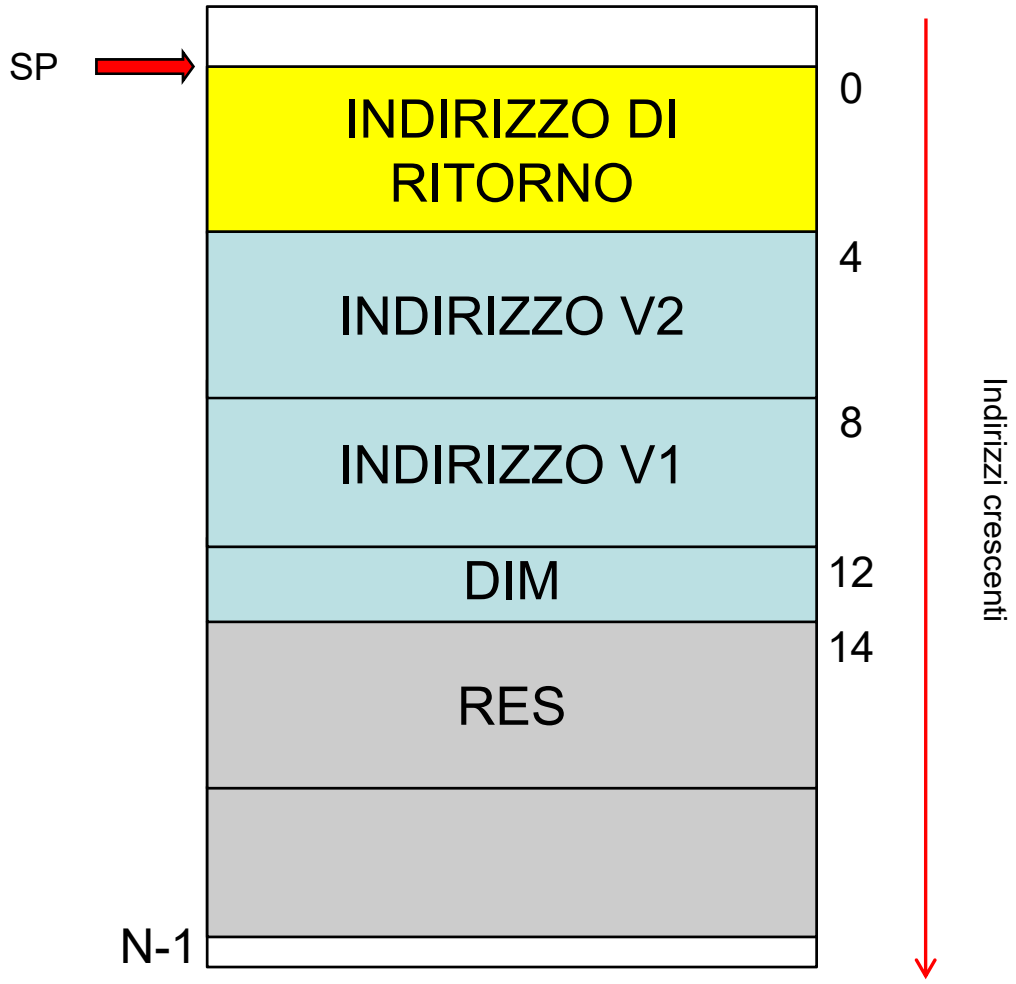
```
          ORG $8000
INIZIO    ADDA #-4,A7
          MOVE.W DIM,-(A7)
          MOVE.L #V1,-(A7)
          MOVE.L #V2,-(A7)
          JSR PRODSCAL
          MOVE.L (A7)+,RES

FINE      JMP FINE
```



Passaggio dei parametri su stack - Esempio

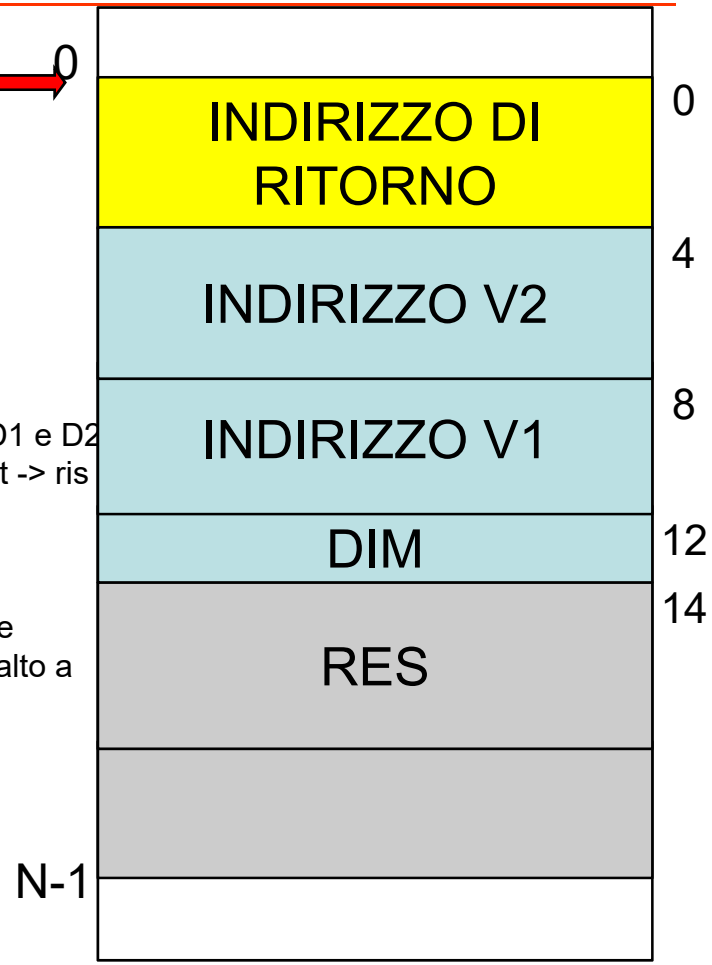
DEFINIZIONE DEGLI OFFSET

```
RITOFF EQU 0
V2OFF EQU 4
V1OFF EQU 8
DIMOFF EQU 12
RESOFF EQU 14
```



Passaggio dei parametri su stack - Esempio

```
PRODSCAL  MOVEA.L RIToff(A7),A2

          MOVEA.L V2off(A7),A1
          MOVEA.L V1off(A7),A0
          MOVE.W DIMoff(A7),D0

          CLR.L     D7      accumulatore delle somme parziali

SOMMA     MOVE.B (A0)+,D1
          MOVE.B (A1)+,D2
MULT      MULU D1,D2  effettua la moltiplicazione (unsigned) fra D1 e D2
                      *e pone il risutato in D2 (operandi di 16 bit -> ris
32bit)
          ADD.L D2,D7

CTRL      SUB #1,D0    controlla se ho ancora elementi da scorrere
          BNE SOMMA  ..se ho ancora elementi da considerare salto a
SOMMA

ESCI      MOVE.W D7,RESoff(A7) ..altrimenti esco
          ADDA.L #RESoff,A7
          JMP (A2)

               ORG $8100
          V1        DC.B  0,1,3,2
          V2        DC.B  3,2,2,1
          DIM       DC.W  4
          RES       DS.L  1
          END INIZIO
```



Passaggio dei parametri su stack con LINK/UNLK

PROGRAMMA CHIAMANTE

1) Riserva spazio sullo stack per i parametri di output
2) Pone sullo stack i parametri di input
3) Salta alla subroutine (indirizzo di ritorno su stack)
4) Preleva i parametri di output dallo stack

SUBROUTINE

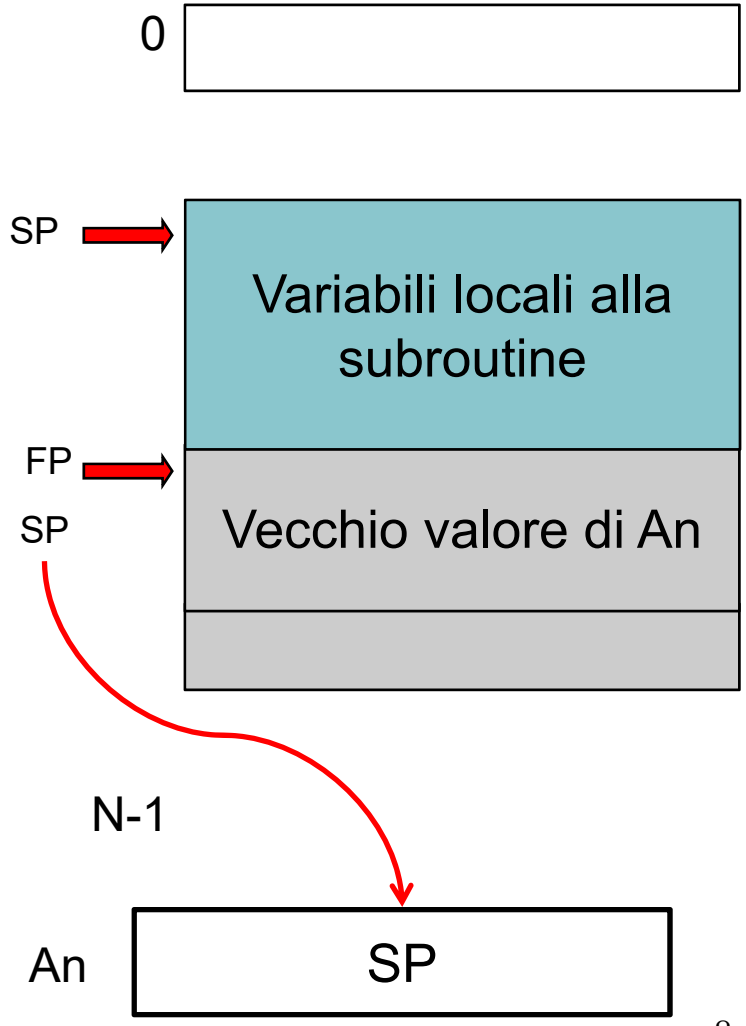
1) Effettua il LINK
2) Preleva l'indirizzo di ritorno dallo stack (utilizzando l'offset relativo a FP) e lo salva in un registro indirizzi
3) Preleva i parametri di input dallo stack (utilizzando gli offset relativi a FP) e li copia nei registri dati/indirizzi
4) Effettua la propria elaborazione (utilizzando eventualmente variabili locali allocate nello stack frame della subroutine) e calcola i parametri di output
5) Pone i parametri di output sullo stack nelle posizioni per essi riservate (utilizzando gli offset relativi al FP)
6) Effettua l'UNLK
7) Ripulisce lo stack
8) Salta all'indirizzo di ritorno

Passaggio dei parametri su stack con LINK/UNLK

OPERAZIONE DI LINK

LINK An,#<displacement>

1) Esegue il 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 (negativo per allocare spazio a una procedura) viene esteso in segno e sommato a SP. Questo valore viene assegnato a SP.
4) Durante l'esecuzione SP varia, mentre FP rimane costante

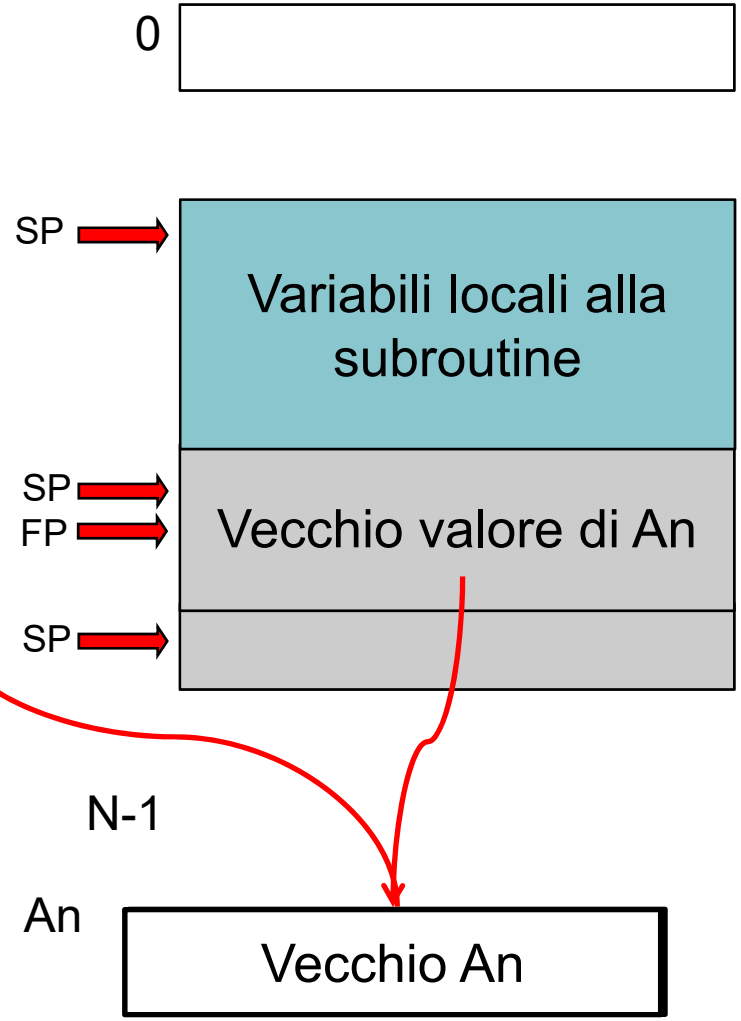


Passaggio dei parametri su stack con LINK/UNLK

OPERAZIONE DI UNLINK

UNLK An

1) Carica il vecchio valore dello stack pointer dal registro An
2) Il registro indirizzo specificato viene caricato con il vecchio valore che era stato salvato sullo stack



Link/Unlk - Esempio

Scrivere e assemblare un programma che calcola il prodotto scalare fra due vettori utilizzando una subroutine a cui vengono passati in ingresso i vettori V1 e V2 (indirizzo del vettore – 4 byte) e la loro dimensione DIM (2 byte).

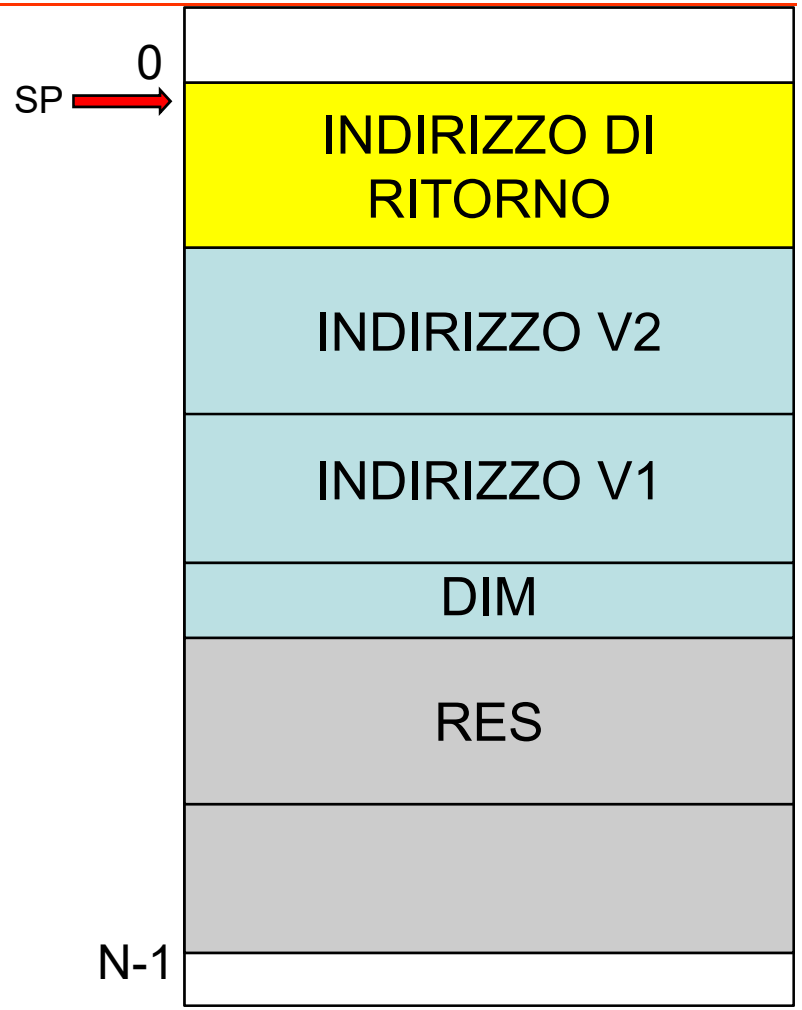
Si adotti il meccanismo di scambio dei parametri tramite stack con allocazione di uno stack frame per le variabili locali alla procedura (accumulatore delle somme parziali).

Link/Unlk- Esempio

PROGRAMMA CHIAMANTE

(**1**) Riserva spazio sullo stack per i parametri di output
(**2**) Pone sullo stack i parametri di input
(**3**) Salta alla subroutine (indirizzo di ritorno su stack)
(**4**) Preleva i parametri di output dallo stack

```
          ORG $8000
INIZIO    ADDA #-4,A7
          MOVE.W DIM,-(A7)
          MOVE.L #V1,-(A7)
          MOVE.L #V2,-(A7)
          JSR PRODSCAL
          MOVE.L  (A7)+,RES

FINE      JMP FINE
```

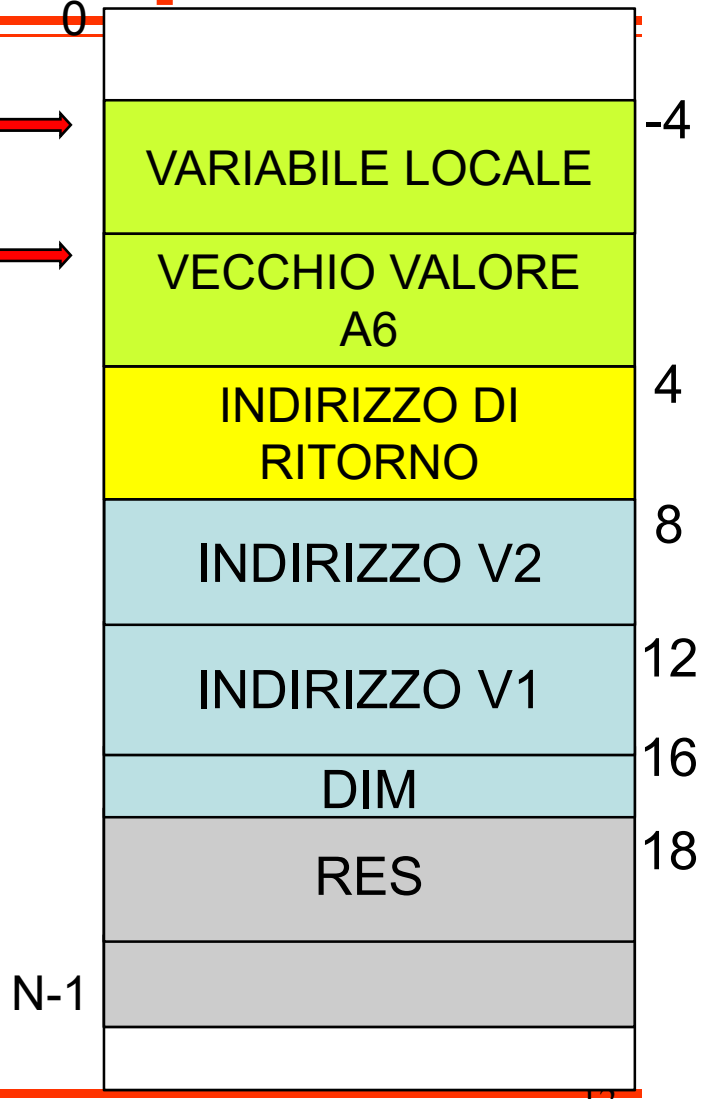


Link/Unlk- Esempio

DEFINIZIONE DEGLI OFFSET

Dopo l'operazione di LINK lo stack assume la configurazione in figura. Gli offset vengono definiti relativamente al LR

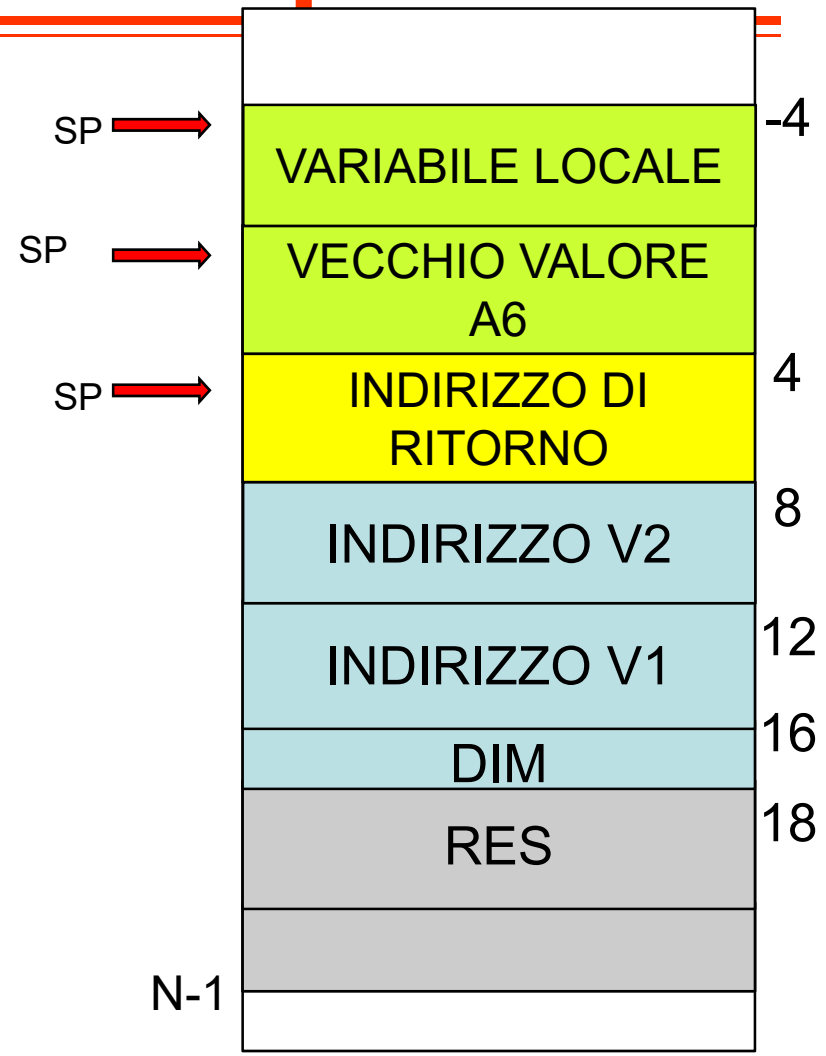
```
VAROFF EQU -4
RITOFF EQU 4
V2OFF EQU 8
V1OFF EQU 12
DIMOFF EQU 16
RESOFF  EQU  18
```



Link/Unlk- Esempio

SUBROUTINE

```
PRODSCAL        LINK A6,#-4
                MOVEA.L RIToff(A6),A2
                MOVEA.L V2off(A6),A1
                MOVEA.L V1off(A6),A0
                MOVE.W DIMoff(A6),D0

                MOVE 0,VAROFF(A6)

SOMMA   MOVE.B (A0)+,D1
                MOVE.B (A1)+,D2
MULT    MULU D1,D2
                ADD.L D2, VAROFF(A6)

CTRL    SUB #1,D0
                BNE
ESCI    MOVE.L VAROFF(A6),RESoff(A6)
                UNLK A6
                ADDA.L #RESoff-4,A7
                JMP (A2)
```



Link/Unlk - Esempio

Scrivere e assemblare un programma che conti i valori 0 di una matrice (memorizzata per righe) utilizzando una subroutine a cui vengono passati in ingresso la matrice (indirizzo del primo elemento – 4 byte), il numero di righe (2 byte) e di colonne (2 byte).

Scrivere e assemblare un programma che conti i valori 0 di una matrice (memorizzata per righe) utilizzando una subroutine a cui vengono passati in ingresso la matrice (indirizzo del primo elemento – 4 byte), il numero di righe (2 byte) e di colonne (2 byte) la quale a sua volta richiami una subroutine per il conteggio degli 0 in un vettore (parametri: indirizzo vettore, dimensione)