

**Architettura  
Elaboratori  
Elettronici  
ESERCITAZIONI  
CHIAMATE A SISTEMA**

**Franco Liberati**  
**[liberati@di.uniroma1.it](mailto:liberati@di.uniroma1.it)**

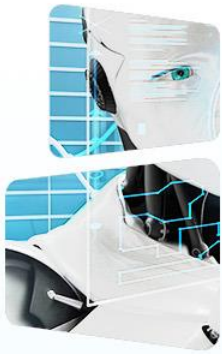


# Argomenti

- ❑ Chiamate a sistema

- ❑ Istruzioni macchina





# Chiamate a Sistema

## (Generalità)

Una **chiamata a sistema** è una istruzione che rimanda all'esecuzione di un programma realizzato da terzi (routine di gestione della memoria virtuale, interazione con i dispositivi di I/O, ...)

```
READ (a)
READ (b)
Massimo=MAX(a,b)
PRINT (Massimo)
```

```
OPEN CONNECTION TO PRINTER
IS_READY?
READ_DATA
PUT_DATA_IN_MEMORY
RETURN FROM INTERRUPT
```

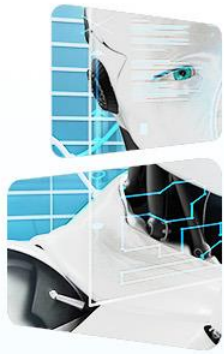


# Chiamate a Sistema

Interruzione software

Le istruzioni di **SYSCALL** nascondono delle **TRAP** (delle **interruzioni software**) che possono essere considerate come operazioni sincrone di salto

La differenza principale con una operazione di salto è che la TRAP prevede il **salvataggio dello stato della macchina** (almeno PC e SR)



# Chiamate a Sistema

## Interruzione software: salvataggio stato macchina

In accordo alle istruzioni ci può essere un alterazione dello stato della macchina e una variazione dei flag presenti nel Registro di Stato

ES:

li \$t0, -1

li \$t1, 1

add \$t2, \$t0, \$t1



Flags:

N (negative): 0

Z (zero) : 1

C (carry) : 1

W (overflow): 0



# Chiamate a Sistema

## Interruzione software: salvataggio stato macchina

Se si verifica una interruzione prima di un salto condizionato si potrebbe creare una situazione anomala (qualora non venisse salvato il Registro di Stato)

```
li    $t0, 10  
  
loop_label:  
sub   $t0, $t0, 1  
BNEZ loop_label
```

Testa il valore del registro di stato  
Se il valore di \$t0 non è uguale a zero  
(quindi Z=0)  
allora si effettua il salto **B**

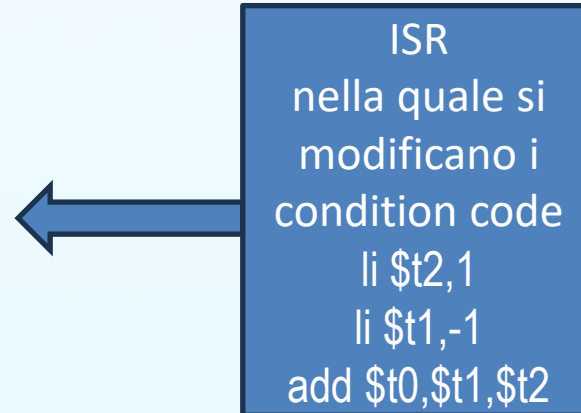


# Chiamate a Sistema

## Interruzione software: salvataggio stato macchina

Se si verifica una interruzione prima di un salto condizionato si potrebbe creare una situazione anomala (qualora non venisse salvato il Registro di Stato)

```
li    $t0, 10  
loop_label:  
sub   $t0, $t0, 1  
BNEZ loop_label
```

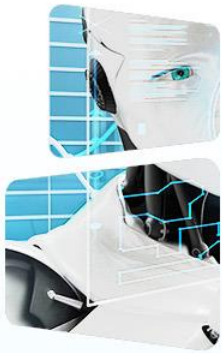




# Chiamate a Sistema MARS

SERVIZI	CODICE in \$V0	ARGOMENTI	RISULTATO
<code>print_int</code>	1	\$a0 = integer (da stampare)	
<code>print_float</code>	2	\$f12 = float (da stampare)	
<code>print_double</code>	3	\$f12 = double (da stampare) <b>attinge anche a \$f13</b>	
<code>print_string</code>	4	\$a0 = string (da stampare)	Richiede l'indirizzo della prima locazione di memoria dove è definita stringa (istruzione classe: <b>la</b> )
<code>read_int</code>	5		<b>integer in \$v0</b>
<code>read_float</code>	6		<b>float in \$f0</b>
<code>read_double</code>	7		<b>double in \$f0 (e in \$f1)</b>
<code>read_string</code>	8	\$a0=buffer, \$a1=length	Richiede l'indirizzo della prima locazione di memoria dove è definita stringa (istruzione classe: <b>la</b> )
<code>random_int</code>	42	\$a0= integer (seme della casualità) \$a1 = limite superiore intero (escluso)	Genera un numero casuale tra 0 e il valore immesso in \$a1 [0,\$a1). <b>Numero casuale intero in \$a0</b>
<code>exit</code>	10		Terminazione del programma





# Chiamate a Sistema

## Esempio I

Svolgere il quadro di un numero. Il numero è introdotto da tastiera con un messaggio testuale per guidare l'utente e il risultato è mostrato su videoterminale con un messaggio testuale a corredo



# Chiamate a Sistema

## Esempio I

.text

main:

.globl main

la \$a0,mex1

li \$v0,4

syscall

li \$v0,5

syscall

move \$t0,\$v0

mul \$t1,\$t0,\$t0

la \$a0,mex2

li \$v0,4

syscall

move \$a0,\$t1

li \$v0,1

syscall

#inserimento locazione dell'inizio del messaggio

#servizio di stampa di una stringa

#chiamata di sistema

#servizio di lettura di un intero da tastiera

#chiamata di sistema

#spostamento del valore letto da tastiera

#operazione ( $a^2$ )

#inserimento locazione dell'inizio del messaggio

#servizio di stampa di una stringa

#chiamata di sistema

#spostamento del valore intero da stampare

#servizio di stampa di un intero

#chiamata di sistema

.data

mex1:.asciiz "Inserire il numero:"

mex2:.asciiz "\nIl quadrato del numero è:"



# Chiamate a Sistema

## Esempio II

Realizzare un programma che chiede all'utente di inserire il nome e lo mostra su videoterminale con un saluto amichevole

Richiede la definizione preventiva di una stringa di byte (dimensione prestabilita 255)

Si utilizza la direttiva **.space**



# Chiamate a Sistema

## Esempio II

.text

main:

.globl main

la \$a0,mex1

li \$v0,4

syscall

#inserimento locazione dell'inizio del messaggio

#servizio di stampa di una stringa

#chiamata di sistema

la \$a0,mexinput

li \$a1,255

li \$v0,8

syscall

#zona in cui sarà conservata la stringa immessa da input

#dimensione della stringa da immettere

#servizio di lettura di una stringa

#chiamata di sistema

la \$a0,mex2

li \$v0,4

syscall

#inserimento locazione dell'inizio del messaggio

#servizio di stampa di una stringa

#chiamata di sistema

la \$a0,mexinput

li \$v0,4

syscall

#inserimento locazione dell'inizio del messaggio

#servizio di stampa di una stringa

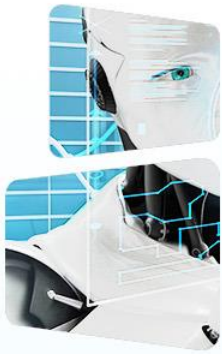
#chiamata di sistema

.data

mex1:.asciiz "Inserire il nome: "

mex2:.asciiz " \nCiao "

mexinput:.space 255



# Chiamate a Sistema

## Esempio III

### NUMERO MAGICO

Realizzare un programma in codice assembler che genera un numero casuale tra 0 e 10 e chieda all'utente di indovinare il suo valore (massimo tre tentativi) segnalando immediatamente la vittoria o, alla fine dei tre tentativi, la sconfitta con un messaggio amichevole



# Chiamate a Sistema

## Esempio III

```
.text
.globl main

main:
    li $t0,3      #contatore dei tentativi
    li $a0,4      #generazione del numero casuale
                  #e salvataggio nel registro t1

    li $a1,11
    li $v0,42
    syscall
    move $t1,$a0

riprova:
    li $v0,5      #lettura del valore intero immesso
                  #dall'utente e salvataggio in $t2

    syscall
    move $t2,$v0
    beq $t1,$t2, vinto #confronto per vedere se si è
#indovinato il numero e salto alla stampa della vittoria
    sub $t0,$t0,1 #decremento num. tentativi
    bgtz $t0,riprova #ripetizione inserimento valore
    j perso      #salto alla stampa della sconfitta
```

vinto:

```
la $a0,mex1 #messaggio vittoria
li $v0,4    #servizio di stampa di una stringa
syscall     #chiamata di sistema
j fine
```

perso:

```
la $a0,mex2 #messaggio sconfitta
li $v0,4    #servizio di stampa di una stringa
syscall     #chiamata di sistema
j fine
```

fine:

```
li $v0,10
syscall
```

.data

```
mex1:.asciiz "BRAVO!HAI INDOVINATO"
mex2:.asciiz "NON HAI INDOVINATO"
```

# ISTRUZIONI MACCHINA





# Istruzioni macchina

Le istruzioni di sistema sono istruzioni tipiche della macchina

<b>NOP</b>	<i>Nessuna operazione</i>
<b>ALT</b>	<i>Interruzione del sistema</i>
<b>BREAK</b>	<i>Interruzione del programma</i>



FINE

