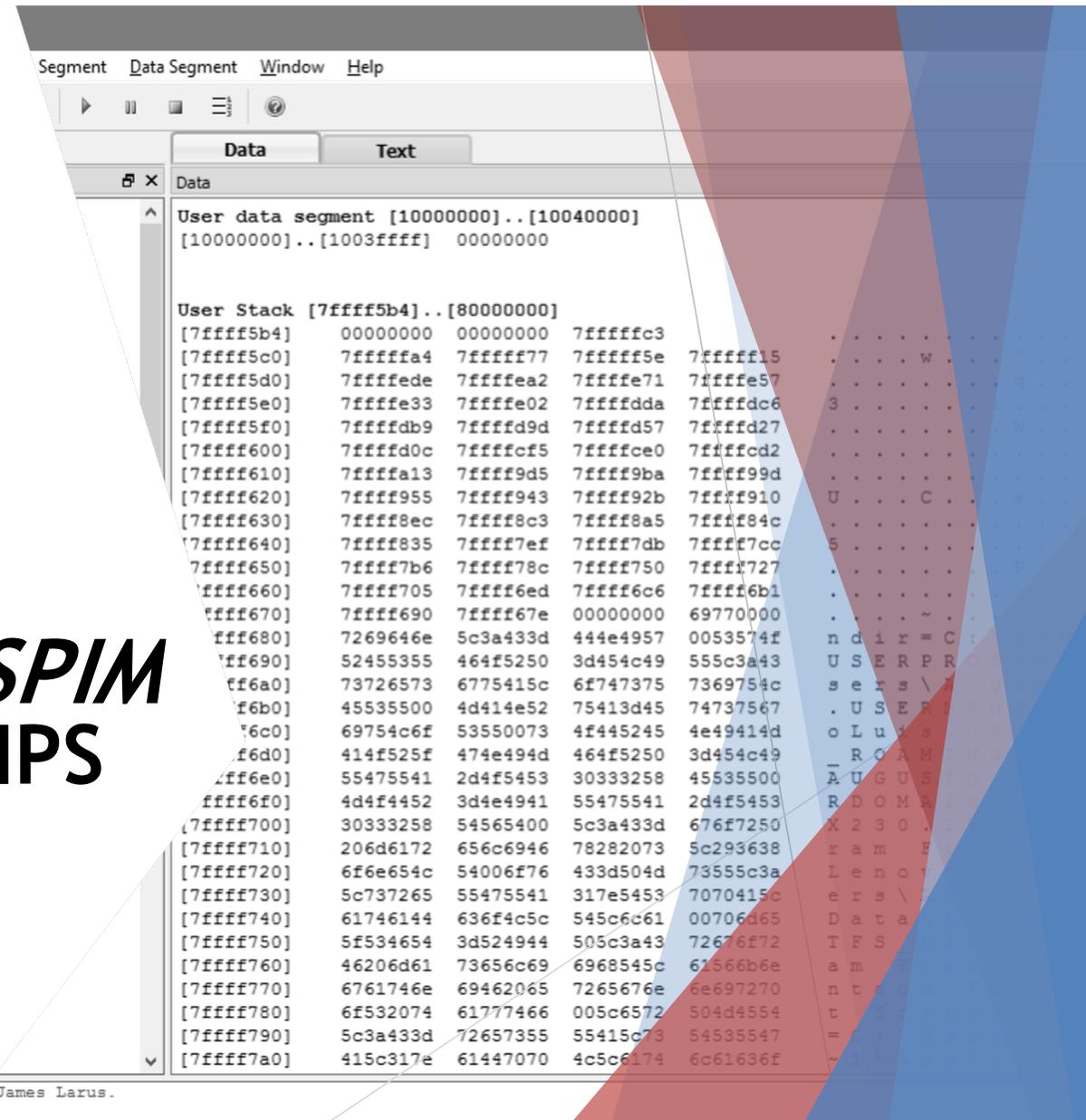


# Laboratorio 1

## Introduzione a QTSPIM

### Simulatore Arch. MIPS



# QTSPIM

▶ Simulatore dell'architettura MIPS32

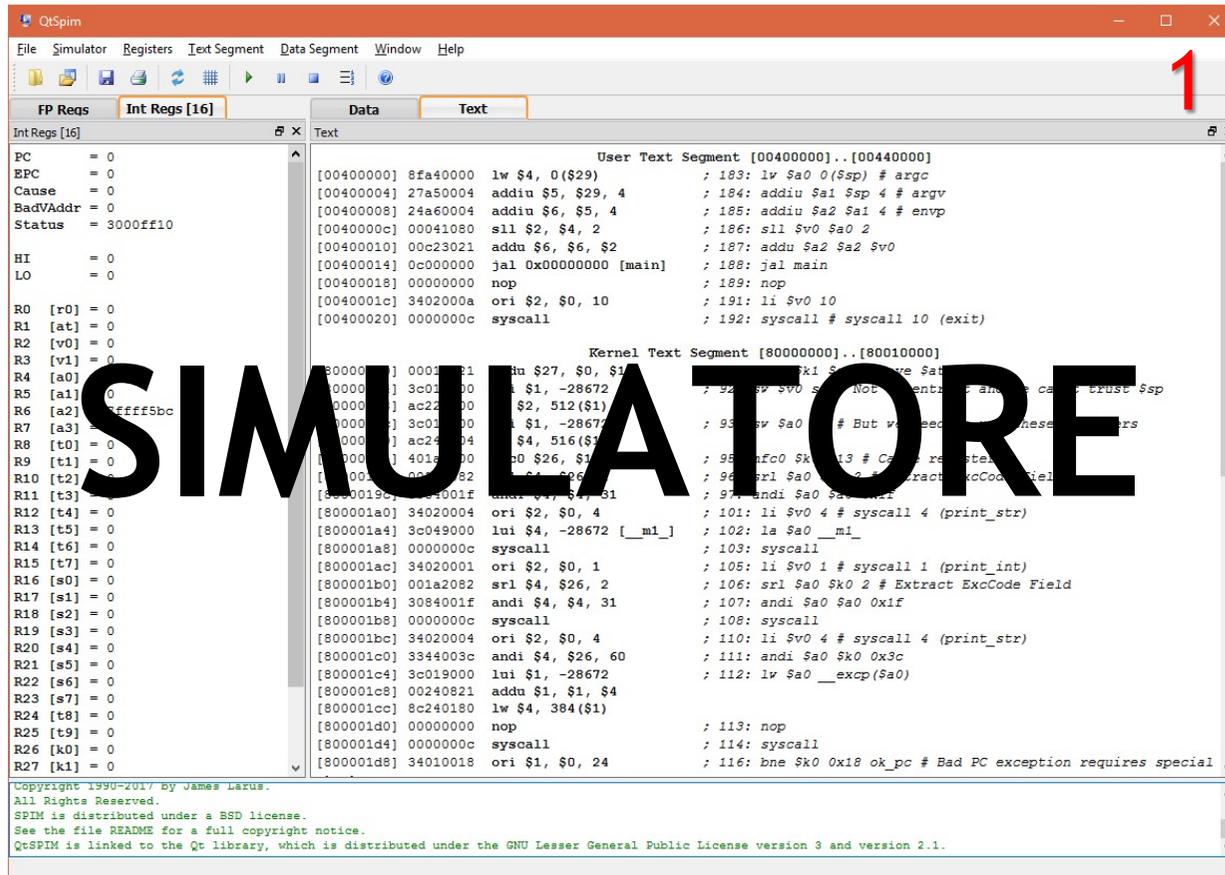
▶ Può essere scaricato all'indirizzo ...

<http://spimsimulator.sourceforge.net>

- MIPS (e quindi QtSPIM) usa 32 registri, ciascuno di 32 bit
- MIPS indirizza il **singolo byte** (8 bit)
- Gli indirizzi delle parole (i.e., **word**) sono **multipli di 4** (1 word = 4 byte)
- Gli indirizzi di **due word consecutive** differiscono di **4 unità**
- In MIPS le word iniziano sempre ad **indirizzi multipli di 4**, cioè 0, 4, 8, 12, ...

11	11100000	}	Parola
10	11111111		
9	10100011		
8	01010101		
7	11110000	}	Parola
6	00000000		
5	00101010		
4	11111111		
3	11000000	}	Parola
2	00101000		
1	11100000		
0	10010001		

L'interfaccia del simulatore QtSpim è composta da due parti principali



**SIMULATORE**

**FINESTRA  
DI INPUT  
OUTPUT**

# INTERFACCIA DEL SIMULATORE QtSpim

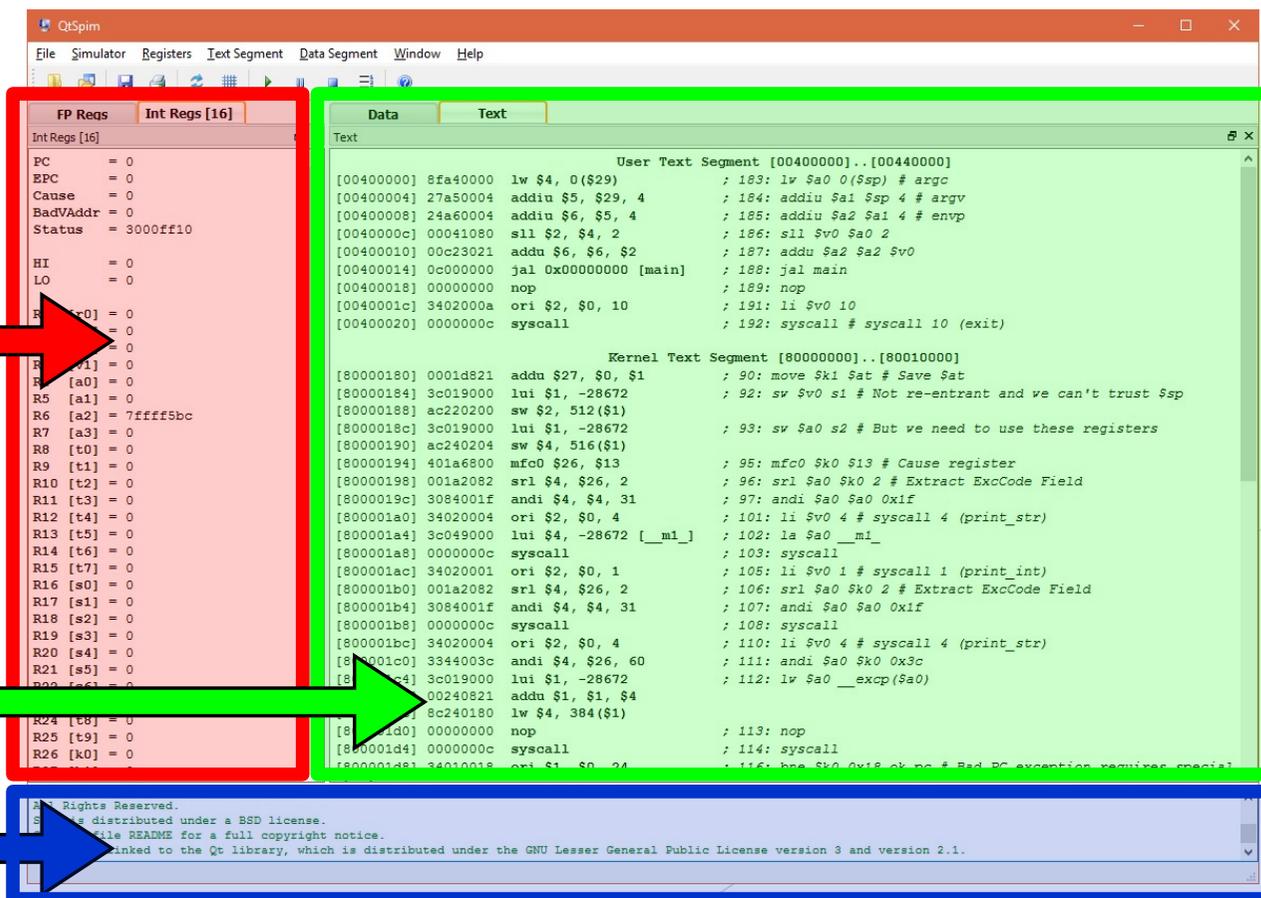
LA FINESTRA PRINCIPALE È SUDDIVISA IN TRE PARTI

IL PANNELLO SULLA SINISTRA VISUALIZZA I REGISTRI FLOATING POINT

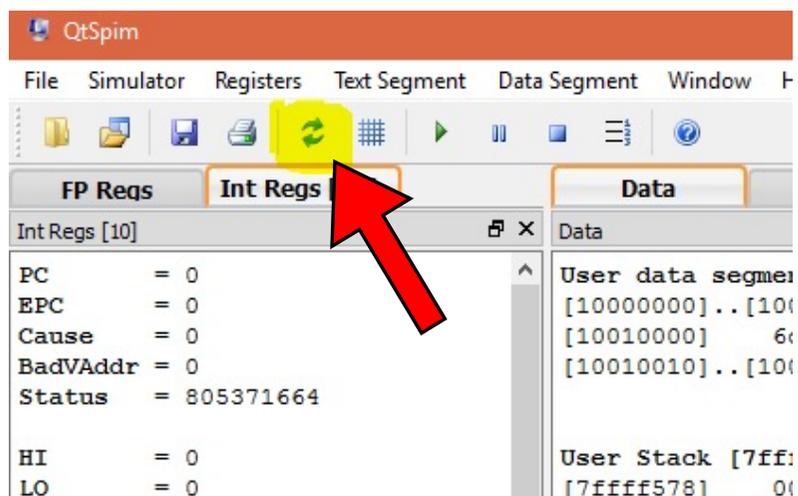
oppure I REGISTRI INTERI, A SECONDA DELLA SCHEDA IN PRIMO PIANO

IL PANNELLO SULLA DESTRA VISUALIZZA IL TEXT SEGMENT (CONTIENE LE ISTRUZIONI MACCHINA) ED IL DATA SEGMENT (CONTIENE I DATI IN MEMORIA), A SECONDA DELLA SCHEDA IN PRIMO PIANO

IL PANNELLO INFERIORE CONTIENE I MESSAGGI DEL SIMULATORE

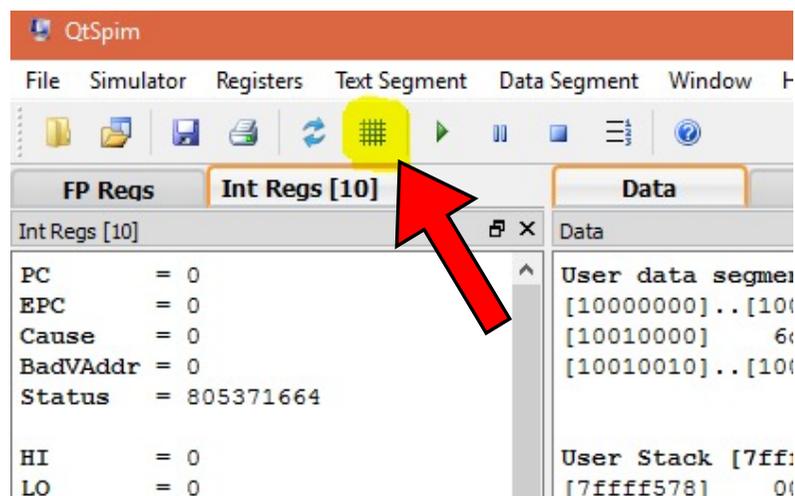


# PULSANTI UTILI PER QUESTA ESERCITAZIONE



**PULSANTE  
«CLEAR REGISTER»**

**CANCELLA IL CONTENUTO DEI  
REGISTRI, MA NON  
LA MEMORIA (AREA DATI)**

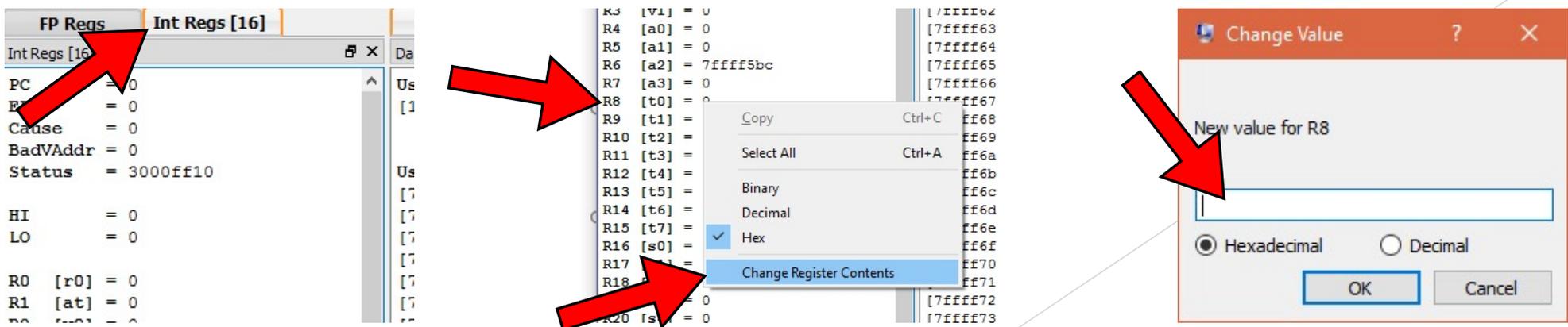


**PULSANTE  
«REINITIALIZE SIMULATOR»**

**REIMPOSTA LA MEMORIA  
(AREA DATI), E  
I REGISTRI**

# COME CAMBIARE IL VALORE DI UN REGISTRO (e.g. R8)?

- SELEZIONARE LA SCHEDA DEI REGISTRI DEGLI INTERI «INT REGS [16]» ED INDIVIDUARE IL REGISTRO R8 (Register8)
- FARE CLICK CON IL PULSANTE DESTRO IN CORRISPONDENZA DEL REGISTRO R8
- SELEZIONARE «CHANGE REGISTER CONTENTS»
- INSERIRE IL VALORE (IN FORMATO HEX oppure DEC)



# ESEMPIO: IMPOSTARE R8 CON IL VALORE 0x3

1  
INDIVIDUO R8

```
R4 [a0] = 0  
R5 [a1] = 0  
R6 [a2] = 7ffff5bc  
R7 [a3] = 0  
R8 [t0] = 0  
R9 [t1] = 0  
R10 [t2] = 0  
R11 [t3] = 0
```

2  
CLICK DESTRO +  
«CHANGE REGISTER...»

```
R7 [a3] = 0  
R8 [t0] = 0  
R9 [t1] = 0  
R10 [t2] = 0  
R11 [t3] = 0  
R12 [t4] = 0  
R13 [t5] = 0  
R14 [t6] = 0  
R15 [t7] = 0  
R16 [s0] = 0  
R17 [s1] = 0  
R18 [s2] = 0
```

3  
SCRIVO IL NUMERO

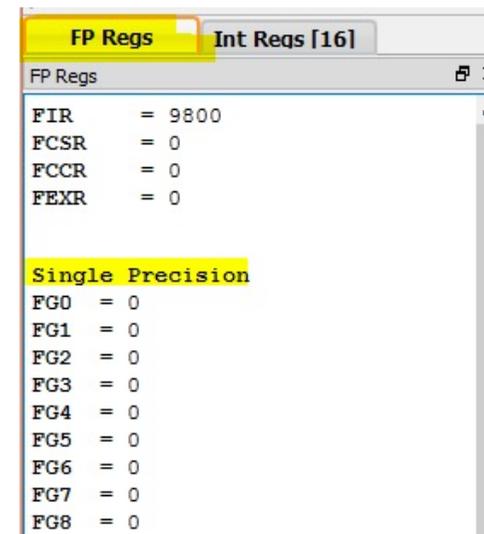
New value for R8  
3  
 Hexadecimal  Decimal  
OK Cancel

4  
RISULTATO

```
R4 [a0] = 0  
R5 [a1] = 0  
R6 [a2] = 7ffff5bc  
R7 [a3] = 0  
R8 [t0] = 3  
R9 [t1] = 0  
R10 [t2] = 0  
R11 [t3] = 0
```

# REGISTRI FLOATING POINT SINGLE PRECISION (VIRGOLA MOBILE, SINGOLA PRECISIONE)

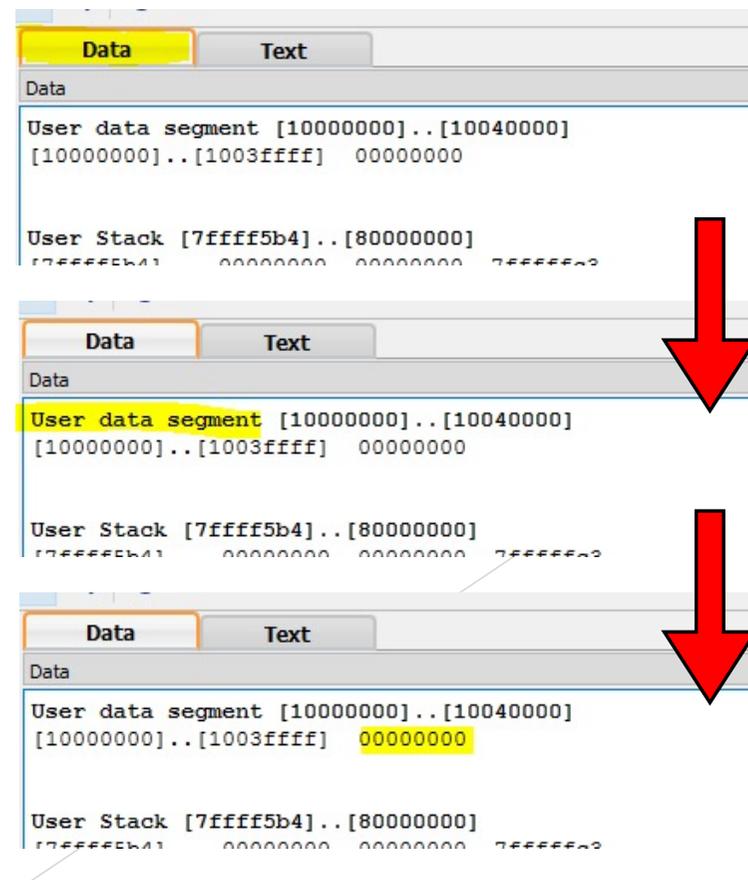
- SI VISUALIZZANO ABILITANDO LA SCHEDA «FP Regs»
- SI MODIFICANO ALLO STESSO MODO DEI REGISTRI INTERI
- LA RAPPRESENTAZIONE DEI NUMERI SEGUE LO **STANDARD IEEE754**



N.B. il contenuto del registro è visualizzato a partire dalla prima cifra diversa da 0

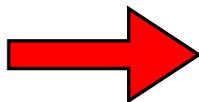
# ***OLTRE AI REGISTRI SI POSSONO CAMBIARE ANCHE I VALORI CONTENUTI IN MEMORIA...***

- SELEZIONARE LA SCHEDA «DATA»
- INDIVIDUARE IL «DATA SEGMENT»
- FARE CLICK-DESTRO SULL'INDIRIZZO DELL'AREA DA MODIFICARE

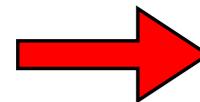


# ESEMPIO: IMPOSTARE IL VALORE 0x3 ALL'INDIRIZZO DI MEMORIA 0x10000000

1. INDIVIDUO IL DATA SEGMENT E L'INDIRIZZO 0x10000000



2. CLICK DESTRO + «CHANGE MEMORY...»



3. INSERISCO VALORE

```
Data
User data segment [10000000]..[10040000]
[10000000]..[1003ffff] 00000000

User Stack [7ffff5b4]..[80000000]
[7ffff5b4] 00000000 00000000 7ffff5b3
```

```
Data
User data segment [10000000]..[10040000]
[10000000]..[1003ffff] 00000000

User Stack [7ffff5b4]..[80000000]
[7ffff5b4] 00000000 00000000
[7ffff5c0] 7fffffa4 7fffff7
[7ffff5d0] 7ffffede 7ffffea
[7ffff5e0] 7ffffe33 7ffffe0
[7ffff5f0] 7ffffdb9 7ffffd9
[7ffff600] 7ffffd0c 7ffffcf
[7ffff610] 7ffffa13 7ffff9d
```

- Copy Ctrl+C
- Select All Ctrl+A
- Binary
- Decimal
- Hex
- Change Memory Contents

Change Value ? X

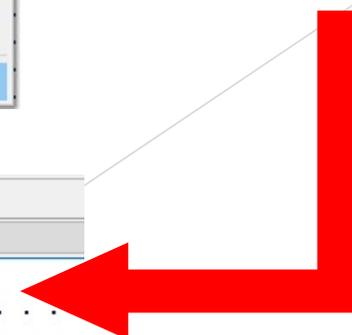
New value for 10000000

Hexadecimal  Decimal

OK Cancel

4. RISULTATO

```
Data
User data segment [10000000]..[10040000]
[10000000] 00000003 00000000 00000000 ...
[10000010]..[1003ffff] 00000000
```



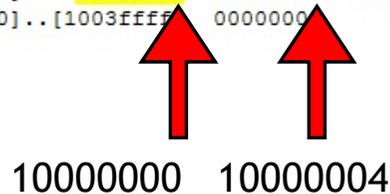
# COSA È CAMBIATO?

*PRIMA...*

```
Data
User data segment [10000000]..[10040000]
[10000000]..[1003ffff] 00000000
```

*...DOPO*

```
Data
User data segment [10000000]..[10040000]
[10000000] 00000003 00000000 00000000 00000000 .....
[10000010]..[1003ffff] 00000000
```



N.B. gli indirizzi sono in esadecimale! quindi ad ogni riga è visualizzato il contenuto di celle di memoria che si trovano 16 indirizzi dopo rispetto alla riga precedente. Infatti il primo indirizzo della seconda riga è 10000010, ovvero  $10000000_{16} + 16_{10}$  ( $10_{16}$ )

- QTSPIM VISUALIZZA «4 WORD», LA RIGA COMPLETA
- CON QUESTO METODO, NON E' POSSIBILE MODIFICARE L'AREA DI MEMORIA AD UN INDIRIZZO «NON VISUALIZZATO»

# VISUALIZZAZIONE DELLA MEMORIA

**INDIRIZZI IN  
FORMATO  
ESADECIMALE  
A BLOCCHI DI 4**

10000000      10000004      10000008      1000000C

**OGNI WORD MEMORIZZA 4 BYTE ORDINANDOLI «DA DESTRA A SINISTRA»  
OVVERO PARTENDO DAL BYTE MENO SIGNIFICATIVO. AD ESEMPIO...**

Indirizzo: **10000000**

Indirizzo: **10000001**

Indirizzo: **10000002**

Indirizzo: **10000003**

**MENO SIGNIFICATIVO** -----> **PIU' SIGNIFICATIVO**