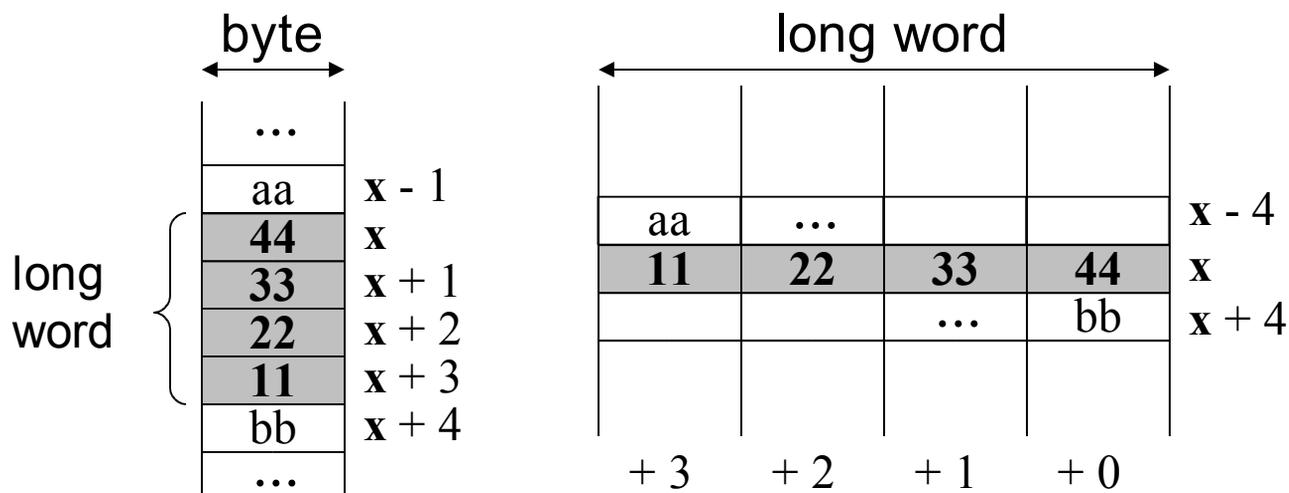


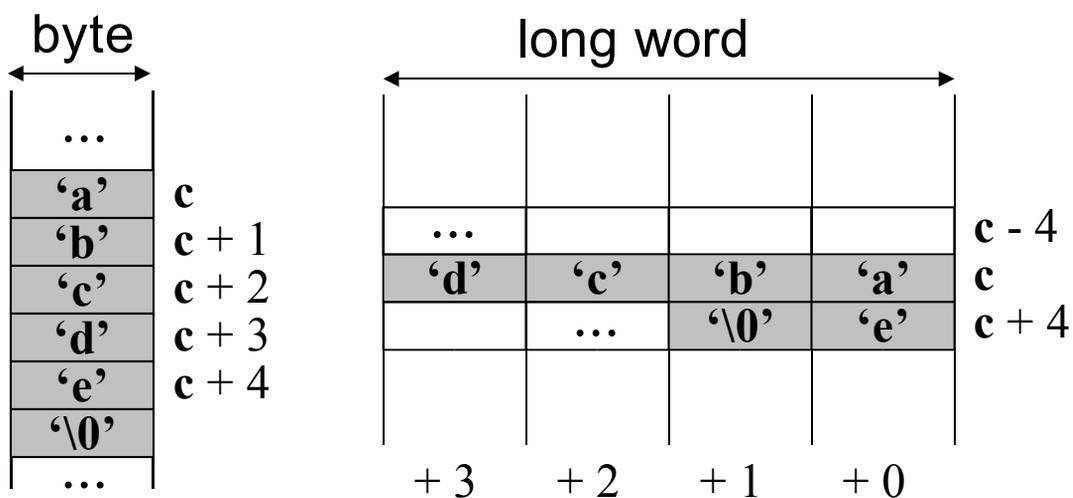
Rappresentazione Little-Endian

(little-end-first)

Es.: long word **0x11223344** all'indirizzo **x**



Attenzione: `char c[] = "abcde";`



MOV vs LEAL

```
movl -10(%ebp, %ecx, 4), %eax  
leal -10(%ebp, %ecx, 4), %eax
```

EBP

1000

memoria

ECX

10

1026

($1000 + 10 * 4 - 10 =$) 1030

20

1034

movl: EAX

20

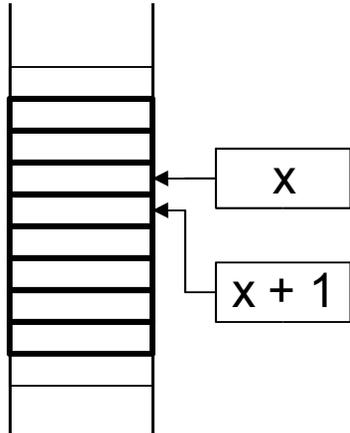
leal: EAX

1030

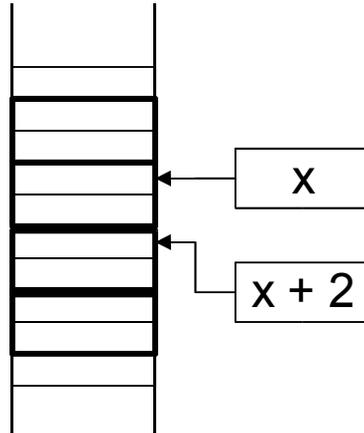
Stringhe

Sequenze di dati dello stesso tipo (byte/word/long word) allocati in locazioni contigue di memoria

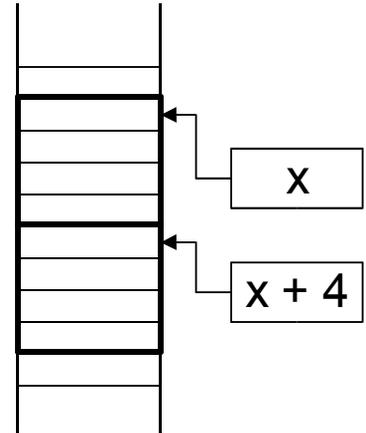
Stringa di byte



Stringa di word



Stringa di long word



Operazioni tipiche:

- copiare due stringhe
- confrontare due stringhe
- cercare un valore in una stringa
- inizializzare una stringa
- operazioni di I/O su sequenze di dati

Tipicamente realizzate con cicli e registri puntatore:

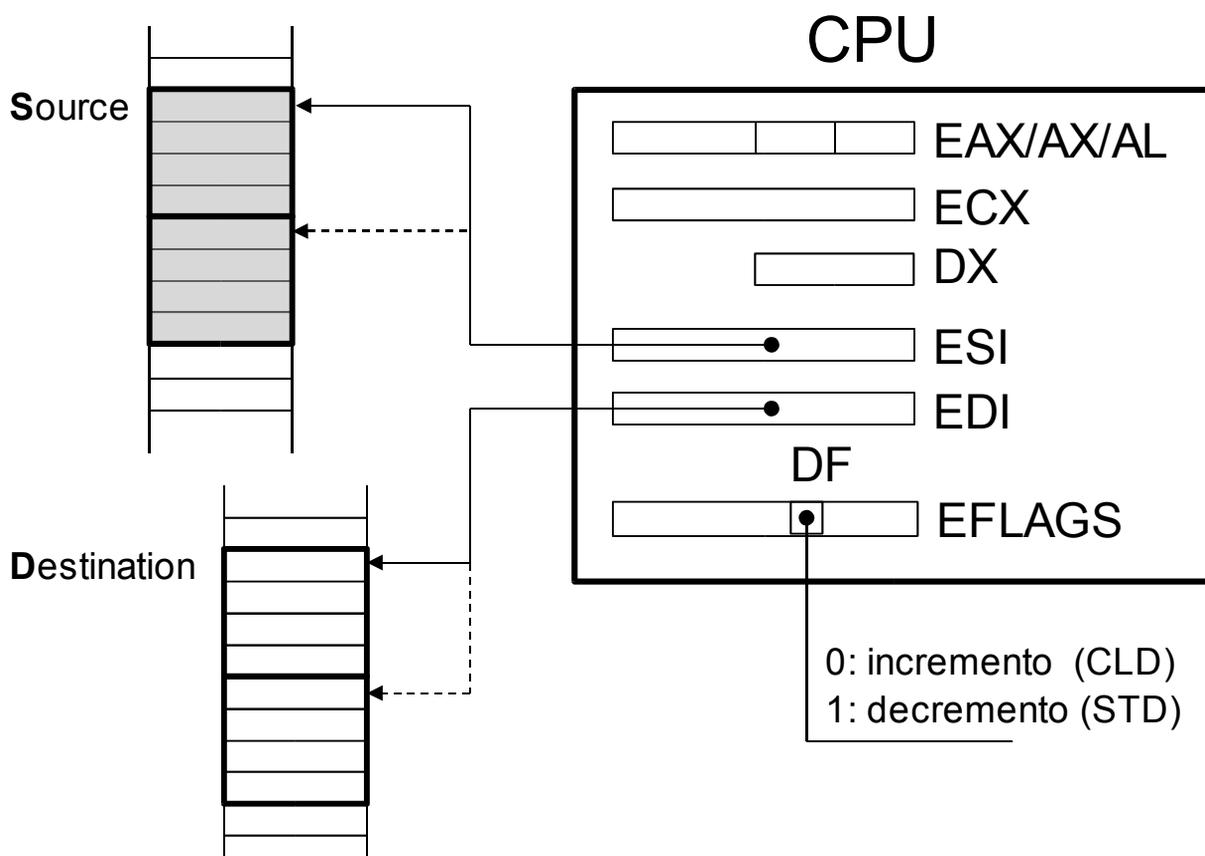
```
movl $indirizzo_sorgente,%esi
movl $indirizzo_destinazione,%edi
ciclo: ...
movl (%esi),%eax
movl %eax,(%edi)
addl $4,%esi #si incrementano i registri puntatore
addl $4,%edi #per indirizzare l'elemento successivo
...
jmp ciclo
```

Istruzioni per la manipolazione delle stringhe

ESI: Extended **S**ource **I**ndex

EDI: Extended **D**estination **I**ndex

Incrementati (o decrementati) automaticamente dalle istruzioni che manipolano le stringhe



istruzione	op. sorg. (implicito)	op. dest. (implicito)
MOVSlun	(%ESI)	(%EDI)
CMPSlun	(%ESI)	(%EDI)
SCASlun	AL/AX/EAX	(%EDI)
LODSlun	(%ESI)	AL/AX/EAX
STOSlun	AL/AX/EAX	(%EDI)
INSlun	DX (spazio di I/O)	(%EDI)
OUTSlun	(%ESI)	DX (spazio di I/O)

Prefissi di ripetizione

REP *instr*

Nelle forme:

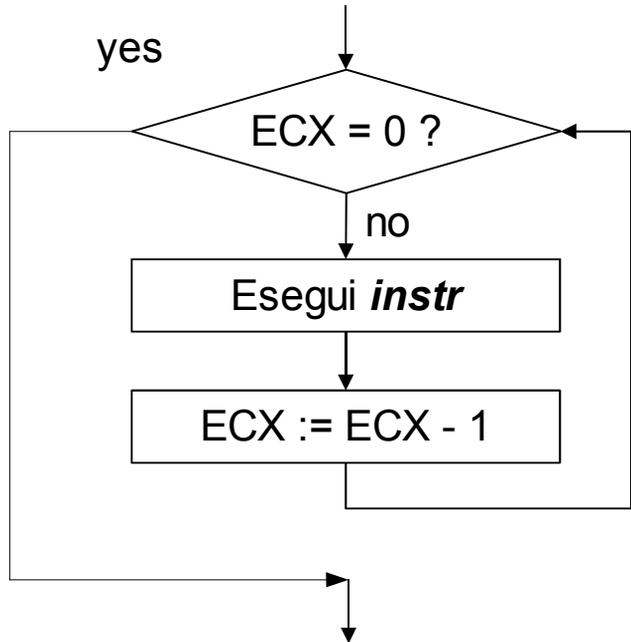
REP MOVSlun

REP LODSlun

REP STOSlun

REP INSlun

REP OUTSlun



REPE/REPZ *instr*

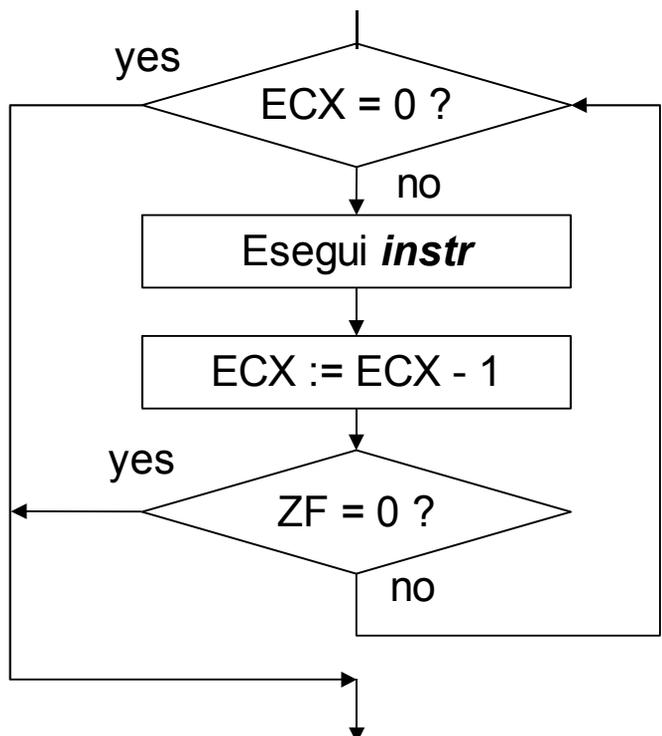
Nelle forme:

REPE CMPSlun

REPZ CMPSlun

REPE SCASlun

REPZ SCASlun



REPNE/REPNZ *instr*

Nelle forme:

REPNE/REPNZ CMPSlun

REPNE/REPNZ SCASlun

(il secondo test diventa "ZF = 1 ?")

Esempi

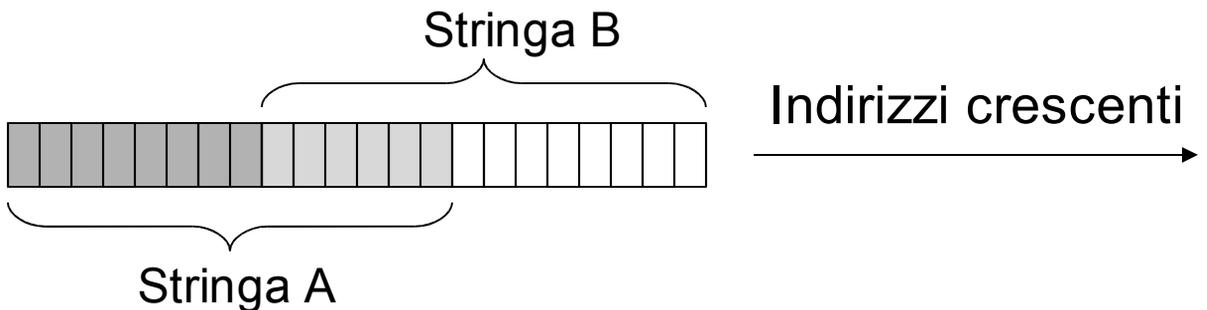
Copiare due stringhe:

```
movl $sorgente, %esi
movl $destinazione, %edi
movl $lunghezza, %ecx
cld
rep movsl
```

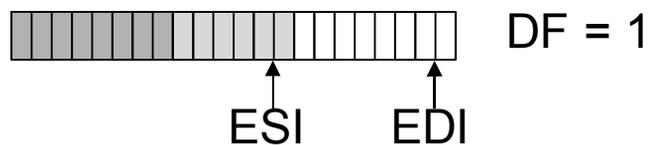
Ricerca di un valore in una stringa:

```
movl $valore, %eax
movl $stringa, %edi
movl $lunghezza, %ecx
cld
repne scasl
je trovato
```

DF: Direction Flag



$\xrightarrow{\text{MOVSlun}}$
Stringa A – Stringa B



$\xleftarrow{\text{MOVSlun}}$
Stringa A – Stringa B

