# PROGRAMMI DI SVILUPPO

# Programmi di sviluppo considerati (1)

- **Programmi di sviluppo:**
  - **Traduttori (Assemblatore, Compilatore C, Compilatore C++), Collegatore, Caricatore.**
- **Compilatori C/C++:**
  - **traducono i singoli file in linguaggio Assembler;**
  - **per il sistema di sviluppo GNU/GCC, sono stati esaminati alcuni standard di traduzione del Compilatore C++, soprattutto per quanto riguarda le tecniche di aggancio e la traduzione di identificatori di funzioni/operatori, globali e membro.**
- **Assemblatore, Collegatore, Caricatore:**
  - **esame del funzionamento interno, con una trattazione non strettamente vincolata al sistema di sviluppo GNU/GCC, ma funzionalmente equivalente.**
- **Traduzione mediante l'Assemblatore:**
  - **se richiesto, produzione anche un *file listato,* contenente, per ogni comando, la traduzione in linguaggio macchina e il comando Assembler stesso;**
  - **si ottiene con il comando (già visto):**

    **as id_file.s -o id_file.o –a > id_file.l**

# Programmi di sviluppo considerati (2)

- **Assemblatore:**
  - **traduce ogni singolo file (file *sorgente)*, comprensivo anche dei file inclusi (direttiva Assembler *.include*);**
  - **un file sorgente può contenente più sezioni dati e più sezioni testo:**
    - **per ogni sezione vengono tradotti in linguaggio macchina i *comandi* (istruzioni o pseudo-istruzioni), mentre le direttive vengono risolte durante la traduzione stessa;**
    - **la traduzione di un file sorgente dà luogo a un nuovo file, detto file *oggetto*, con un'unica sezione dati e un'unica sezione testo.**
- **Collegatore:**
  - **elabora tutti i file oggetto dando luogo a un unico file (file *eseguibile*).**
- **Caricatore:**
  - **trasferisce in memoria il file eseguibile (se del caso, rilocandolo preliminarmente).**
- **Modello di memoria utilizzato:**
  - **quello PIC (già visto).**

# Processo di traduzione

- **Processo di traduzione di un file:**
  - **utilizzo di due variabili interne di conteggio, dette *Contatori di Locazion*i CLd e CLt, uno per la sezione dati e uno per la sezione testo, rispettivamente.**
- **Ogni contatore CL:**
  - **azzerato quando viene incontrata per la prima volta la corrispondente direttiva *.data* o *.text;***
  - **incrementato man mano che vengono esaminati i vari comandi in una sezione *.data* o *.text*, rispettivamente;**
  - **ogni incremento è pari al numero di byte richiesti per la traduzione del comando in esame.**
- **Valore di ogni contatore CL:**
  - **rappresenta l'indirizzo numerico di un comando, riferito all'inizio della sezione *.data* o *.text* in cui il comando compare;**
    - **i valori di ogni CL vengono scritti in esadecimale (con 4 cifre).**
- **Scorrimento del file (due volte):**
  - **una prima volta, per costruire alcune tabelle:**
    - **tabella degli identificatori simbolici (utilizzata durante il secondo scorrimento);**
    - **tabella delle sezioni (utilizzata dal collegatore);**
    - **tabella degli identificatori globali (utilizzata dal collegatore);**
    - **tabella degli identificatori esterni (utilizzata dal collegatore);**
  - **una seconda volta, per:**
    - **effettuare la traduzione vera e propria;**
    - **costruire la tabella di collegamento (utilizzata dal collegatore).**

# Primo scorrimento

- **Tabella degli identificatori (o simboli):**

  **Identificatore Sezione Valore del pertinente CL**

  **… … …**

  – **associa ad ogni identificatore simbolico definito nel file il valore del pertinente CL in corrispondenza del comando che ha quell'identificatore (indirizzo numerico relativo all'inizio della sezione).**

- **Tabella delle sezioni:**

  **Nome_file Sezione Lunghezza**

  **… ... ...** …

  – **specifica la lunghezza di ogni sezione (valore del pertinente CL successivo all'ultimo comando, allineato a multipli di 16 (ultima cifra esadecimale uguale a 0)).**

- **Tabella degli identificatori globali:**

  **Nome_file Sezione Identificatore globale Valore del pertinente CL**

  **… ... ... …**

  – **è un sottoinsieme della tabella dei simboli, contenente solo le righe relative agli identificatori dichiarati globali.**

- **Tabella degli identificatori esterni:**

  **Nome file Identificatore utilizzato e non definito**

  **… ...**

  – **contiene gli identificatori utilizzati e non definiti in quel file, a prescindere dal fatto che siano stati o meno dichiarati esterni.**

# Secondo scorrimento

- **Traduzione del file in modo sequenziale:**
  - **il file tradotto è costituito da due sezioni, *.data* e *.text*;**
  - **in ogni sezione, ogni byte del comando tradotto ha associato un *indice:***
    - **per il primo byte, l'indice coincide col valore del pertinente CL (indirizzo relativo del comando);**
    - **per i successivi byte, l'indice si ottiene incrementando il valore dell'indice riferito al byte precedente;**
- **Campo DISP:**
  - **può essere lungo un byte, due byte (parola) oppure 4 byte (parola lunga);**
    - **in genere, è lungo uno o quattro byte.**
- **Campo IMM:**
  - **ha le stesse caratteristiche di lunghezza del campo DISP (non viene utilizzata l'istruzione MOVABSQ);**
  - **nelle istruzioni macchina, segue il campo DISP, se questo è presente.**
- **Nel comando tradotto;**
  - **ciascun byte viene scritto con due cifre esadecimali;**
  - **i byte dei campi DISP e IMM, quando sono più di uno, compaiono in ordine inverso (sono invece nell'ordine posizionale giusto le due cifre esadecimali che rappresentano il valore di ogni byte).**

# Espressioni numeriche e espressioni indirizzo

- **Espressioni numeriche:**
  - **composte solo da operandi numerici;**
  - **per semplicità, supponiamo che i campi IMM delle istruzioni e i valori iniziali delle variabili possano essere costituiti solo da espressioni numeriche.**
- **Espressioni indirizzo:**
  - **contengono almeno un identificatore simbolico;**
    - **nel caso più semplice, coincidono con un identificatore simbolico;**
  - **supponiamo che solo il campo DISP delle istruzioni (nel modello PIC prevedono solo indirizzamento relativo), sia operative che di salto, possa essere determinato anche con espressioni indirizzo.**

# Espressioni indirizzo

- **Espressione indirizzo in una istruzione di salto:**
  - **se l'espressione indirizzo è costituita da un identificatore ID definito nella sezione *.text* del file, essa viene calcolata utilizzando la tabella dei simboli, come CLt(ID) (valore di CLt corrispondente all'identificatore ID), e il campo DISP assume il valore:**

    **DISP = CLt(ID)-CLt* (DISP è sempre di 32 bit per l'istruzione *call*);**

    **dove CLt* è il valore di CLt corrispondente all'istruzione immediatamente successiva a quella da tradurre;**
  - **se l'espressione indirizzo è costituita da un identificatore esterno, essa non viene calcolata e il campo DISP (sempre di 32 bit) assume provvisoriamente il valore 0.**
- **Espressione indirizzo in una istruzione operativa:**
  - **l'espressione indirizzo può essere costituita da un identificatore definito nella sezione *.data* del file (presente nella tabella dei simboli) o da un identificatore esterno;**
  - **in entrambi i casi non viene calcolata, e il campo DISP (sempre di 32 bit) assume provvisoriamente il valore 0.**
- **Esempio di lavoro:**
  - **programma *codifica1*, costituito da due file, *codifica1a.s* e *codifica1b.s,* il primo dei quali include il file *ser.s* .**

# Esempio di primo scorrimento: file *codifica1a.s* (1)

```
# Programma codifica1: file principale codifica1a.s
.include   "ser.s"
.data
0000    buff:       .byte 0

.text
0000    tastiera:   pushq    %rbx
0001                pushq    %rcx
0002                pushq     %rdx
0003                movq     $3, %rax
000a                movq     $0, %rbx
0011                leaq      buff(%rip),%rcx
0018                movq     $1, %rdx
001f                int       $0x80
0021                movb     buff(%rip), %al
0027                popq     %rdx
0028                popq     %rcx
0029                popq      %rbx
002a                ret

002b    video:      pushq    %rax
002c                pushq     %rbx
002d                pushq     %rcx
002e                pushq      %rdx
002f                movb      %bl, buff(%rip)
0035                movq     $4, %rax
003c                movq      $1, %rbx
0043                leaq       buff(%rip), %rcx
004a                movq      $1, %rdx
0051                int        $0x80
0053                popq      %rdx
0054                popq      %rcx
0055                popq      %rbx
0056                popq       %rax
0057                ret
0058    uscita:     movl      $0, %ebx
005d                movl     $1, %eax
0062                int       $0x80
```

# Esempio di primo scorrimento: file *codifica1a.s (continua)* (2)

```
# Programma codifica1: file principale codifica1a.s
# continua …
.extern     alfa, beta, esamina
.data
0001            kappa:    .fill      8, 1
0009

.text
.global     _start
                _start:
0064            ancora:     call    tastiera
0069                        cmpb  $'\n', %al
006b                        je      fine
006d                        movb  %al, %bl
006f                       call    video
0074                       movb   $' ', %bl
0076                       call    video
007b                       movb  %al, alfa(%rip)
0081                       leaq    kappa(%rip), %rax
0088                       movq  %rax, beta(%rip)
008f                       call     esamina
0094                       leaq  kappa(%rip), %rax
009b                       movq   $0, %rsi
00a2            ripeti:   movb  (%rax, %rsi), %bl
00a5                       call     video
00aa                       incq     %rsi
00ad                       cmpq   $8, %rsi
00b1                       jb       ripeti
00b3                       movb   $'\n', %bl
00b5                       call     video
00ba                       jmp     ancora
00bc              fine:    jmp     uscita
00be
```

# Esempio di primo scorrimento: tabelle per il file *codifica1a.s*

- **Tabella degli identificatori (o simboli) (valori di CLd o di CLt):**

| | | | |
|---|---|---|---|
| **buff** | **.data** | **0000000000000000** | **(CLd)** |
| **kappa** | **.data** | **0000000000000001** | **(CLd)** |
| **tastiera** | **.text** | **0000000000000000** | **(CLt)** |
| **video** | **.text** | **000000000000002b** | **(CLt)** |
| **uscita** | **.text** | **0000000000000058** | **(CLt)** |
| **_start** | **.text** | **0000000000000064** | **(CLt)** |
| **ancora** | **.text** | **0000000000000064** | **(CLt)** |
| **ripeti** | **.text** | **00000000000000a2** | **(CLt)** |
| **fine** | **.text** | **00000000000000bc** | **(CLt)** |

- **Tabella delle sezioni (lunghezza allineata a multipli di 16):**

| | | |
|---|---|---|
| **codifica1a.s** | **.data** | **0009 allineata a 0010** |
| **codifica1a.s** | **.text** | **00be allineata a 00C0** |

- **Tabella degli identificatori globali (valori di CLd o di CLt):**

| | | | | |
|---|---|---|---|---|
| **codifica1a.s** | **.text** | **_start** | **0000000000000064** | **(CLt)** |

- **Tabella degli identificatori esterni**

| | |
|---|---|
| **codifica1a.s** | **alfa** |
| **codifica1a.s** | **beta** |
| **codifica1a.s** | **esamina** |

# Esempio di secondo scorrimento: generazione del file *codifica1a.o* (1)

```
# Programma codifica1: file principale codifica1a.o
.data
0000 00       buff:          .byte 0

.text
0000 53       tastiera:      pushq   %rbx
0001 51                      pushq   %rcx
0002 52                      pushq    %rdx
0003 48C7C0 03000000         movq    $3, %rax
000a 48C7C3 00000000         movq    $0, %rbx
0011 488D0D 00000000         leaq     buff(%rip),%rcx
0018 48C7C2 01000000         movq    $1, %rdx
001f CD80                    int     $0x80
0021 8A05 00000000           movb    buff(%rip), %al
0027 5A                      popq    %rdx
0028 59                      popq    %rcx
0029 5B                      popq     %rbx
002a C3                      ret

002b 50       video:         pushq   %rax

002c 53                      pushq    %rbx
002d 51                      pushq    %rcx
002e 52                      pushq     %rdx
002f 881D 00000000           movb     %bl, buff(%rip)
0035 48C7C0 04000000         movq     $4, %rax
003c 48C7C3 01000000         movq     $1, %rbx
0043 488D0D 00000000         leaq     buff(%rip), %rcx
004a 48C7C2 01000000          movq     $1, %rdx
0051 CD80                     int      $0x80
0053 5A                       popq     %rdx
0054 59                       popq     %rcx
0055 5B                       popq     %rbx
0056 58                       popq     %rax
0057 C3                       ret

0058 BB 00000000 uscita: movl   $0, %ebx
005d B8 01000000          movl    $1, %eax
0062 CD80                 int     $0x80
```

# Esempio di secondo scorrimento: generazione del file *codifica1a.o (continua)* (2)

```
# Programma codifica1: file principale codifica1a.o
# continua …
.extern         alfa, beta, esamina
.data
0001 00000000       kappa:      .fill       8, 1
       00000000
0009

.text
.global         _start
                            _start:
0064 E8 97FFFFFF  ancora:  call     tastiera
0069 3C 0A                 cmpb   $'\n', %al
006b 74 4F                 je         fine
006d 88C3                  movb   %al, %bl
006f E8 B7FFFFFF           call     video
0074 B3 20                 movb   $' ', %bl
0076 E8 B0FFFFFF           call     video
007b 8805 00000000         movb   %al, alfa(%rip)
0081 488D05 00000000       leaq    kappa(%rip), %rax
0088 488905 00000000       movq   %rax, beta(%rip)
008f E8 00000000           call      esamina
0094 488D05 00000000   leaq   kappa(%rip), %rax
009b 48C7C6 00000000   movq   $0, %rsi
00a2 8A1C30       ripeti: movb  (%rax, %rsi), %bl
00a5 E8 81FFFFFF         call       video
00aa 48FFC6              incq       %rsi
00ad 4883FE 08           cmpq   $8, %rsi
00b1 72 EF               jb         ripeti
00b3 B3 0A               movb   $'\n', %bl
00b5 E8 71FFFFFF         call       video
00ba EB A8               jmp     ancora
00bc EB 9A         fine:   jmp     uscita
00be
```

# Generazione del file *codifica1a.o*: esempi di calcolo di DISP

- **Istruzione *call tastiera*, CLt 0064:**
  - **tastiera: 00000000 (tabella dei simboli, sezione testo)**
  - **CLt*: 00000069**
  - **DISP = 00000000–00000069 = FFFFFF97**
- **Istruzione *call video*, CLt 006f:**
  - **video: 0000002B (tabella dei simboli, sezione testo)**
  - **CLt*: 00000074**
  - **DISP = 0000002B–00000074 = FFFFFFB7**
- **Istruzione *je fine*, CLt 006b:**
  - **fine: 000000BC (tabella dei simboli, sezione testo)**
  - **CLt*: 0000006D**
  - **DISP (positivo rappresentato con 8 bit) = BC-6D = 4F**
- **Istruzione *jp ancora*, CLt 00ba:**
  - **ancora: 00000064 (tabella dei simboli, sezione testo)**
  - **CLt*: 000000BC**
  - **DISP (negativo rappresentato con 8 bit) = 64 –BC = A8**

# Regole per il collegamento (1)

- **Parole lunghe da collegare (campi DISP):**
  - **il loro valore è stato calcolato provvisoriamente (in molti casi con 0);**
  - **il traduttore deve specificare al Collegatore, per ognuna di esse 1) l'indice del primo byte, e 2) l'espressione (simbolica) da utilizzare come operando.**
- **Istruzioni di controllo, con identificatore definito nella sezione *.text* del file:**
  - **in fase di traduzione, DISP è stato tradotto come CLt(ID) – CLt*, dove CL(ID) e CLt* sono valori ottenuti presupponendo che la sezione testo inizi da 0;**
  - **indicando con *Itext* l'indirizzo effettivo della sezione testo, il collegamento comporterebbe il seguente calcolo:**

    **DISP =CL(ID) + *Itext* –(CLt*+ *Itext*) = CL(ID) – CLt***

    **Tale formula risulta uguale a quella utilizzata in fase di traduzione, per cui il collegamento non è richiesto.**

# Regole per il collegamento (2)

- **Istruzioni operative, con identificatore definito nella sezione *.data* del file:**
  - **dopo il collegamento, il campo DISP, provvisoriamente tradotto con 0, deve divenire:**

    **DISP = CLd(ID)+*Idata* –*(CLt*+Itext)*;**
- **Istruzioni operative e istruzioni di controllo, con identificatore esterno (sia *Iidentif* il suo valore):**
  - **dopo il collegamento, il campo DISP, provvisoriamente tradotto con 0, deve essere calcolato come:**

    **DISP = Iidentif -(CLt*+Itext)**

# Esempio di secondo scorrimento: tabella di collegamento per il file codifica1a.o

- **Tabella di collegamento per un file oggetto:**
  - **specifica il Nome del file, la Sezione e l'indice della parola lunga da collegare e l'espressione (simbolica) da utilizzare (secondo le regole precedenti):**

| Nome-file | Sezione | Indice | espressione simbolica |
|---|---|---|---|
| … | … | … | … |

**Tabella di collegamento per il file codifica1a.o**

| Nome-file | Sezione | Indice | espressione simbolica CLd(ID)/Iidentif | CLt* | # identificatore coinvolto |
|---|---|---|---|---|---|
| codifica1a.o | .text | 0014 | 00000000+Idata- | (00000018+Itext) | buf |
| codifica1a.o | .text | 0023 | 00000000+Idata- | (00000027+Itext) | buf |
| codifica1a.o | .text | 0031 | 00000000+Idata- | (00000035+Itext) | buf |
| codifica1a.o | .text | 0046 | 00000000+Idata- | (0000004A+Itext) | buf |
| codifica1a.o | .text | 007D | *Ialfa* | -(00000081+Itext) | *alfa* |
| codifica1a.o | .text | 0084 | 00000001+Idata- | (00000088+Itext) | kappa |
| codifica1a.o | .text | 008B | *Ibeta* | -(0000008F+Itext) | *beta* |
| codifica1a.o | .text | 0090 | *Iesamina* | -(00000094+Itext) | *esamin*a |
| codifica1a.o | .text | 0097 | 00000001+Idata- | (0000009B+Itext) | kappa |

# Esempio per il file *codifica1b.s*: primo scorrimento con avanzamento dei contatori (1)

```
# Programma codifica1: file secondario codifica1b.s
 .data
 .global        alfa, beta
 0000           alfa:       .byte    0
 0001           beta:       .quad    0
 0009

.text
 .global        esamina
 0000           esamina:     pushq   %rax
 0001                        pushq   %rbx
 0002                        pushq   %rsi
 0003                         movb   alfa(%rip), %al
 0009                         movq   beta(%rip), %rbx
 0010                         movq   $0, %rsi
 0017             ciclo:      testb  $0x80, %al
 0019                         jz     zero
 001b                         movb   $'1', (%rbx, %rsi)
 001f                         jmp    avanti
 0021             zero:       movb   $'0', (%rbx, %rsi)
 0025             avanti:      shlb  $1, %al
 0027                         incq   %rsi
 002a                         cmpq   $8, %rsi
 002e                         jb     ciclo
 0030                         popq    %rsi
 0031                         popq     %rbx
 0032                         popq    %rax
 0033                         ret
 0034
```

# Esempio per il file *codifica1b.s*: primo scorrimento con costruzione delle tabelle (2)

- **Tabella dei simboli**

| | | | |
|---|---|---|---|
| **alfa** | **.data** | **0000000000000000** | **(valore di CLd)** |
| **beta** | **.data** | **0000000000000001** | **(valore di CLd)** |
| **esamina** | **.text** | **0000000000000000** | **(valore di CLt)** |
| **ciclo** | **.text** | **0000000000000017** | **(valore di CLt)** |
| **zero** | **.text** | **0000000000000021** | **(valore di CLt)** |
| **avanti** | **.text** | **0000000000000025** | **(valore di CLt)** |

- **Tabella delle sezioni (lunghezza allineata a multipli di 16)**

| | | |
|---|---|---|
| **codifica1b.s** | **.data** | **0009 allineato a 0010** |
| **codifica1b.s** | **.text** | **0034 allineato a 0040** |

- **Tabella degli identificatori globali**

| | | | |
|---|---|---|---|
| **codifica1b.s** | **.data** | **alfa** | **0000000000000000** | **(valore di CLd)** |
| **codifica1b.s** | **.data** | **beta** | **0000000000000001** | **(valore di CLd)** |
| **codifica1b.s** | **.text** | **esamina** | **0000000000000000** | **valore di CLt)** |

- **Tabella degli identificatori esterni**

  **vuota**

# Esempio per il file *codifica1b.s*: secondo scorrimento con produzione del file oggetto (3)

```
# Programma codifica1: file secondario codifica1b.o

 .data
0000 00                         alfa:   .byte           0
0001 0000000000000000   beta:   .quad           0
0009

.text
0000        50              esamina:    pushq  %rax
0001        53                          pushq  %rbx
0002        56                          pushq  %rsi
0003        8A05 00000000               movb   alfa(%rip), %al
0009        488B1D 00000000             movq   beta(%rip), %rbx
0010        48C7C6 00000000             movq   $0, %rsi
0017        A8 80           ciclo:      testb  $0x80, %al
0019        74 06                       jz     zero
001A        C60433 31                   movb   $'1', (%rbx, %rsi)
001F        EB 04                       jmp    avanti
0021        C60433 30       zero:       movb $'0', (%rbx, %rsi)
0025        D0E0            avanti:     shlb   $1, %al
0027        48FFC6                      incq   %rsi
002A        4883FE 08                   cmpq   $8, %rsi
002E        72 E7                       jb     ciclo
0030        5E                          popq   %rsi
0031        5B                          popq   %rbx
0032        58                          popq   %rax
0033        C3                          ret
0034
```

# Esempio per il file *codifica1b.s*: secondo scorrimento con calcolo di alcuni DISP (4)

- **Istruzione *jz zero*, CLt 0019:**
  - **zero: 00000021 (tabella dei simboli, sezione testo)**
  - **CLt*: 0000001B**
  - **DISP (positivo rappresentato con 8 bit) = 21-1B = 06**
- **Istruzione *jmp avanti*, CLt 001f:**
  - **avanti: 00000025 (tabella dei simboli, sezione testo)**
  - **CLt*: 00000021**
  - **DISP (positivo rappresentato con 8 bit) = 25–21 = 04**
- **Istruzione *jb ciclo*, CLt 002e:**
  - **ciclo: 00000017 (tabella dei simboli, sezione testo)**
  - **CLt*: 00000030**
  - **DISP (negativo rappresentato con 8 bit) = 17–30 = E7**

# Esempio per il file *codifica1b.s*: secondo scorrimento e tabella di collegamento (5)

- **Tabella di collegamento per il file codifica1b.o**

| Nome-file | Sezione | Indice | espressione simbolica CLd(ID) CLt* | # identificatore coinvolto |
|---|---|---|---|---|
| codifica1b.o | .text | 0005 | 00000000+Idata-(00000009+Itext) | alfa |
| codifica1b.o | .text | 000C | 00000001+Idata-(00000010+Itext) | beta |

# Collegatore (1)

- **Azioni preliminari:**
  - **prende in esame le tabelle degli identificatori globali e le tabelle degli identificatori esterni dei vari file:**
    - **verifica che l'identificatore *_start* sia dichiarato globale e che compaia in un solo file;**
    - **verifica che ciascun identificatore esterno compaia in una qualche tabella degli identificatori globali (in caso contrario viene segnalato errore di collegamento);**
- **Costruzione del file eseguibile, contenente due sezioni:**
  - **una sezione *Ctext* che inizia da un indirizzo specificato, tipicamente da indirizzo 0;**
  - **una sezione *Cdata* che inizia da un altro indirizzo specificato, per esempio dalla fine della sezione *Ctext*.**
  - **per far questo, costruisce preliminarmente tre tabelle, la *Tabella numerica delle sezioni,* la *Tabella numerica degli identificatori globali* e la *Tabella numerica di collegamento.***
- **Effettuazione le azioni di collegamento:**
  - **per far questo, costruisce la, che specifica, per ogni parola lunga da collegare, il valore numerico da utilizzare come operando.**

# Collegatore (2)

- **Tabella numerica delle sezioni:**

| Nome-File | Sezione | Lunghezza | Isezione |
|---|---|---|---|
| … | … | … | … |

  - **a partire dall'indirizzo della sezione testo del primo file, supponiamo 0, l'indirizzo di ogni sezione testo si ottiene da quello della sezione testo precedente sommandovi la lunghezza della sezione precedente stessa;**
  - **a partire dall'indirizzo della sezione dati del primo file (per esempio, quello successivo alla fine dell'ultima sezione testo), l'indirizzo di ogni sezione dati si ottiene da quello della sezione dati precedente sommandovi la lunghezza della sezione precedente stessa.**

- **Tabella numerica degli identificatori globali:**

| Nome-file | Sezione | Identificatore-globale | pertinente CL | Isezione | Iidentif |
|---|---|---|---|---|---|
| … | … | … | … | … | … |

  - **l'indirizzo di ogni identificatore globale si ottiene sommando il valore del pertinente CL (indirizzo numerico dell'identificatore relativo all'inizio della sezione) con l'indirizzo della sezione stessa, prelevato dalla tabella precedente.**

# Collegatore (3)

- **Indice delle parole lunghe da collegare:**
  - **le parole lunghe da collegare vengono individuate da un indice complessivo, che si ottiene, nell'ordine, da quello specificato nella tabella simbolica di rilocazione relativa a una specifica sezione di uno specifico file , aggiungendovi l'indirizzo numerico della sezione del file stesso, ricavato dalla tabella delle sezioni.**
- **Valore numerico delle espressioni di collegamento:**
  - **calcolo dell'espressione simbolica, noti per ogni sezione *Itext* o *Idata* (Tabella numerica delle sezioni) e per ogni identificatore globale *Iidentif* (Tabella numerica degli identificatori globali).**
- **Tabella di numerica di collegamento:**

| Sezione | Indice | operando numerico |
|---|---|---|
| … | … | … |

- **Esecuzione sul file eseguibile delle azioni specificate nella tabella precedente.**

# Esempio: Tabelle numeriche per il file codifica1 (1)

- **Ipotesi relative all'esempio di lavoro:**
  - **supponiamo che la sezione *Ctext* cominci da indirizzo 0 e la sezione *Cdata* da indirizzo 100.**
- **Tabella numerica delle sezioni:**

| Nome-File | Sezione | Lunghezza | Isezione | |
|---|---|---|---|---|
| codifica1a.o | .text | 00C0 | 0000000000000000 | (Indirizzo iniziale) |
| codifica1b.o | .text | 0040 | 00000000000000C0 | |
| codifica1a.o | .data | 0010 | 0000000000000100 | (Indirizzo iniziale) |
| codifica1b.o | .data | 0010 | 0000000000000110 | |

- **Tabella numerica degli identificatori globali:**

| Nome-file | Sezione | Identif | CLt/CLd | Isezione | Iidentif |
|---|---|---|---|---|---|
| codifica1a.o | .text | _start | 0064 | 0000000000000000 | 0000000000000064 |
| codifica1b.o | .text | esamina | 0000 | 00000000000000C0 | 00000000000000C0 |
| codifica1b.o | .data | alfa | 0000 | 0000000000000110 | 0000000000000110 |
| codifica1b.o | .data | beta | 0001 | 0000000000000110 | 0000000000000111 |
| … | … | | | | |

# Esempio: Tabelle numeriche (2)

- **Tabella numerica di collegamento**

  **Riepilogo dei dati utilizzati:**
  **codifica1a.o: Idata=0100, Itext = 0000, Ialfa =0110, Ibeta=0111, Iesamina = 00C0**
  **codifica1b.o: Idata=0110, Itext=00C0**

| Sezione | Indice | CLd/Iidentif | Idata | CLt* | Itext | valore numerico |
|---|---|---|---|---|---|---|
| .ctext | 0014 | 00000000+ | 00000100- | (00000018+ | 00000000) = | 000000E8 |
| .ctext | 0023 | 00000000+ | 00000100- | (00000027+ | 00000000) = | 000000D9 |
| .ctext | 0031 | 00000000+ | 00000100- | (00000035+ | 00000000) = | 000000CB |
| .ctext | 0046 | 00000000+ | 00000100- | (0000004A+ | 00000000) = | 000000B6 |
| .ctext | 007D | *00000110* | | -(00000081+ | 00000000) = | 0000008F |
| .ctext | 0084 | 00000001+ | 00000100- | (00000088+ | 00000000) = | 00000079 |
| .ctext | 008B | *00000111* | | -(0000008F+ | 00000000) = | 00000082 |
| .ctext | 0090 | *000000C0* | | -(00000094+ | 00000000) = | 0000002C |
| .ctext | 0097 | 00000001+ | 00000100- | (0000009B+ | 00000000) = | 00000066 |
| .ctext | 01C5 | 00000000+ | 00000110- | (00000009+ | 000000C0) = | 00000047 |
| .ctext | 01CC | 00000001+ | 00000110- | (00000010+ | 000000C0) = | 00000041 |

# Esempio: File eseguibile *codifica1*(1)

**Ctext, indirizzo iniziale 0000000000000000, lunghezza 0100**

Indice da 0000 (x è un byte di valore non specificato)

```
0000 53                 tastiera:   pushq   %rbx
...
0011 488D0D E8000000                leaq    buff(%rip),%rcx
0018 …
0021 8A05 D9000000                  movb    buff(%rip), %al
0027 …
002F 881D CB000000                  movb    %bl, buff(%rip)
0035 …
0043 488D0D B6000000                leaq    buff(%rip), %rcx
004A …
0064 …          _start:
007B 8805 8F000000                  movb    %al, alfa(%rip)
0081 488D05 79000000                leaq    kappa(%rip), %rax
0088 488905 82000000                movq    %rax, beta(%rip)
008F E8 2C000000                    call    esamina
0094 488D05 66000000                leaq    kappa(%rip), %rax
009B …
00BC EB 9A              fine:       jmp     uscita
00BE x x
```

# Esempio: File eseguibile *codifica1* (2)

```
00C0 50          esamina:    pushq   %rax
...
00C3 8A05 47000000           movb    alfa(%rip), %al
00C9 488B1D 41000000         movq    beta(%rip), %rbx
00D0 48C7C6 00000000         movq    $0, %rsi
...
00F4  x x x x x x x x x x x x
0100
```

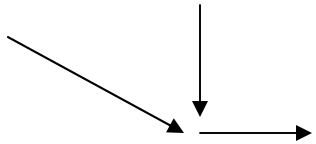
**Cdata, indirizzo iniziale 0000000000000100, lunghezza 0020**

```
Indice da 0100
0100 00                  buff:    .byte 0
0101 00000000            kappa:   .fill 8, 1
       00000000
0109 x x x x x x x
0110 00                   alfa:   .byte   0
0111 00000000             beta:   .quad  0
       00000000
0119 x x x x x x x
0120
```

# Esempio di esecuzione (1)

- Parola lunga collegata di indice 0014: **000000E8**

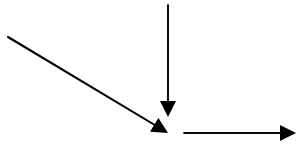
```
0011 488D0D E8000000          leaq    buff(%rip),%rcx
0018
...
0100                          buff:   .byte
0101                          kappa:  .fill 8, 1
```



In esecuzione, calcolo indirizzo identificatore buff: **000000E8+00000018=100** (vedi file eseguibile)

- Parola lunga collegata di indice 007D: **0000008F**

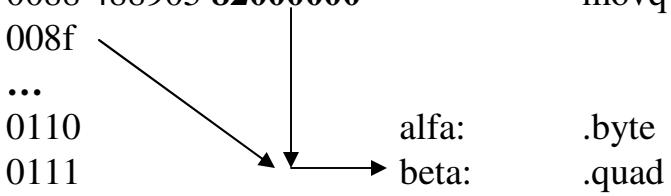
```
007b 8805 8F000000            movb  %al, alfa(%rip)
0081
...
0110                          alfa:   .byte
0111                          beta:   .quad
```



In esecuzione, calcolo indirizzo identificatore alfa: **0000008F+00000081=110** (vedi file eseguibile)

# Esempio di esecuzione (2)

**Parola lunga collegata di indice 008B: 00000082**




In esecuzione, calcolo indirizzo *beta*: **00000082+0000008F=00000111** (vedi file eseguibile)

**Parola lunga collegata di indice 0090: 0000002C**


In esecuzione, calcolo indirizzo *esamina*: **0000002C+00000094=000000C0** (vedi file eseguibile)

# Esempio di esecuzione (3)

- **Parola lunga collegata di indice 00C5: 0000047**

00C3 8A05 **47000000** movb alfa(%rip), %al
00C9
…
0110 alfa: .byte
0111 beta: .quad

In esecuzione, calcolo indirizzo *alfa*: **00000047+000000C9=0110** (vedi file eseguibile)

- **Parola lunga collegata di indice 00CC: 00000041**

00C9 488B1D **41000000** movq beta(%rip), %rbx
00D0
…
0110 alfa: .byte
0111 beta: .quad

In esecuzione, calcolo indirizzo *beta*: **00000041+000000D0=0111** (vedi file eseguibile)

# Tabella di caricamento

- **Tabella di caricamento:**
  - **contiene l'indirizzo e la lunghezza del sezioni *Ctext* e *Cdata*, e l'indirizzo dell'identificatore *_start*:**

| | | |
|---|---|---|
| **Ctext** | **Indirizzo** | **Lunghezza** |
| **Cdata** | **Indirizzo** | **Lunghezza** |
| **_start** | **Indirizzo** | |

- **Esempio di lavoro *codifica1*:**

| | | |
|---|---|---|
| **Ctext** | **00000000** | **100** |
| **Cdata** | **00000100** | **20** |
| **_start** | **00000064** | |

# Caricatore

- **Azioni effettuate:**
  - **trasferire il programma eseguibile in memoria;**
  - **inizializzare il contatore di programma RIP con il valore corrispondente a *_start*;**
  - **Inizializzare il registro CPL con una configurazione prefissata (vedi protezione);**
  - **inizializzare il puntatore di pila ESP con un valore standard;**
  - **inizializzare il registro EFLAGS con una configurazione prefissata;**
  - **tipicamente, le inizializzazioni vengono effettuate ponendo nella pila sistema 5 parole lunghe con i valori da utilizzare, ed eseguendo un'istruzione IRETQ (vedi protezione).**
- **Programma eseguibile:**
  - **può essere caricato in memoria nella sua globalità, a partire da un indirizzo qualsivoglia (rimane costante la differenza fra gli indirizzi di *Ctxt* e *Cdata*).**
    - **non devono essere effettuate modifiche (rilocazioni), in quanto tutti gli indirizzi di memoria sono relativi a RIP:**
    - **il programma è quindi indipendente dalla posizione (PIC).**

# Caricatore rilocante

- **Programma eseguibile:**
  - **Se non è disponibile un'unica zona di memoria, il Caricatore può trasferire il programma a partire da due indirizzi qualsivoglia, uno per la sezione testo e uno per la sezione dati;**
  - **in questo caso occorre *rilocare* alcune delle parole lunghe che sono state soggette a collegamento,:**
    - **in un'istruzione operativa (appartenente alla sezione Ctext), il campo DISP, sommato col valore di RIP, dà luogo a un indirizzo della sezione Cdata.**
      - **il campo DISP deve essere quindi modificato con la quantità $\Delta Cdata - \Delta Ctxt$;**
    - **in un'istruzione di controllo (appartenente alla sezione Ctext) il campo DISP, sommato col valore di RIP, dà luogo a un indirizzo della sezione Ctext;**
      - **il campo DISP non deve quindi essere modificato.**
- **Tabella di rilocazione:**
  - **il Collegatore deve predisporre per il caricatore una Tabella di rilocazione, con la specifica dell'indice delle parole lunghe da rilocare;**
  - **la Tabella di rilocazione viene ottenuta dalla Tabella numerica di collegamento, eliminando le righe relative a istruzioni di controllo.**
  - **la quantità (numerica) da utilizzare per la modifica è sempre $\Delta Cdata - \Delta Ctext$**

# Tabella di rilocazione

- **Tabella di rilocazione:**
  - **eliminazione, dalla tabella di collegamento, delle righe relative a istruzioni di controllo;**
  - **eliminazione dei valori per il collegamento, in quanto i valori per la rilocazione sono sempre dati da $\Delta Cdata - \Delta Ctext$.**
- **Per l'esempio di lavoro codifica1, la tabella di rilocazione è la seguente:**

**File codifica**

| **Sezione** | **Indice** |
|---|---|
| **.text** | **0014** |
| **.text** | **0023** |
| **.text** | **0031** |
| **.text** | **0046** |
| **.text** | **007D** |
| **.text** | **0084** |
| **.text** | **008B** |
| **.text** | **0097** |
| **.text** | **00C5** |
| **.text** | **00CC** |

# Programma *codifica1 rilocato:* *ΔCdata – ΔCtext* = 00001000 (1)

**Ctext parte da 001000, ΔCtext = 1000-0 = 1000, Cdata parte da 02100, ΔCdata = 2100-100 = 2000**

**Ctext:**

```
1000 53          tastiera:   pushq   %rbx
…
1011 488D0D E8100000         leaq    buff(%rip),%rcx
1018 …
1021 8A05 D9100000           movb    buff(%rip), %al
1027 …
102F 881D CB100000           movb    %bl, buff(%rip)
1035 …
1043 488D0D B6100000         leaq    buff(%rip), %rcx
104A …
1064 …           _start:
107B 8805 8F100000           movb  %al, alfa(%rip)
1081 488D05 79100000         leaq  kappa(%rip), %rax
1088 488905 82100000         movq  %rax, beta(%rip)
108F E8 2C000000             call  esamina # non rilocata:
1094 488D05 66100000         leaq  kappa(%rip), %rax
109B …
10BC EB 9A       fine:       jmp   uscita
10BE x x
```

# Programma *codifica1 rilocato:* *ΔCdata – ΔCtext* = 00001000 (2)

```
10C0 50          esamina:    pushq   %rax
...
10C3 8A05 47100000           movb    alfa(%rip), %al
10C9 488B1D 41100000         movq    beta(%rip), %rbx      # 1041+10D0=2111 (beta)
10D0 48C7C6 00000000         movq    $0, %rsi
...
10F4  x x x x x x x x x x x
1100
```

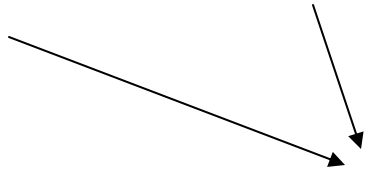
**Cdata: Idata parte da 002100,**

```
2100 00          buff:       .byte 0
2101 00000000    kappa:      .fill 8, 1
     00000000
2109 x x x x x x x
2110 00          alfa:       .byte  0
2111 00000000    beta:       .quad  0
     00000000
2119 x x x x x x x
2120
```

# Esempi di esecuzione

**Istruzione di indice 1021**

**Parola lunga *rilocata* di indice 1023 e valore 000010D9**

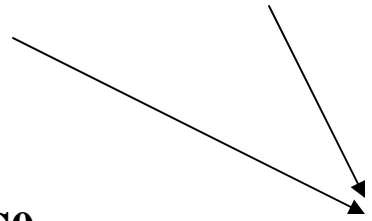


```
1021      8A05 D9100000                    movb    buff(%rip), %al
1027
…
2100                          buff:        .byte   0
```

**In esecuzione, calcolo dell'indirizzo *buff*: 00001027+000010D9=00002100**

**Istruzione di indice 108F**

**Parola lunga *non rilocata* di indice 1090 e valore 0000002C**



```
108F      E8 2C00000                       call    esamina
1094
…
010C0                         esamina:     pushq   %rax
```

**In esecuzione, calcolo dell'indirizzo *esamina*: 00001094+0000002C=000010C0**

# Nuova tabella di caricamento

- **Somma:**
  - **della quantità Δ*Ctext* (1000*)* all'indirizzo della sezione *Ctext* e al valore dell'identificatore _start;**
  - **della quantità Δ*Cdata* (2000) all'indirizzo della sezione *Cdata.***
- **Esempio per *codifica1*:**

| | | |
|---|---|---|
| **Ctext** | **00000000 + Δ*CText* = 1000** | **100** |
| **Cdata** | **00000100 + Δ*CData* = 2100** | **20** |
| **_start** | **00000064 + Δ*Ctext* = 1064** | |