ESERCIZIO 1

MASTER COPY
Tel. 050 8312126
Cell. 388 9837745

COMANDI: -CD YEMU-CE
- ./INSTALL
- CREO CARTELLA DIR, SPOSTO DENTRO "2015_MII-CG", CREO ES1.S
- SPOSTO CONTENUTO PROVA1.CPP IN CC.R
- G++ -O ES1 -FNO-ELIDE-CONSTRUCTORS ES1.S PROVA1.CPP

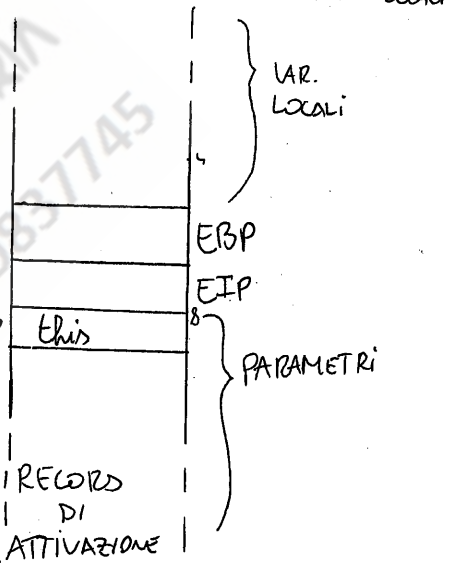
NOMI: COSTRUTTORE: _ZN mNomeClasseC1E TipoParametri
R (RE) P (PUNTATORE) + c (CHAR) i (INT)
mNome SE NON DEFINITO PER DEFAULT
(VETTORE VIENE VISTO COME PUNTATORE (QUINDI CHAR c[] → P_c)

FUNZ. MEMBRO: _ZN mNomeClasseE parametri

PROLOGO:
PUSHL %EBP
MOVL %ESP %EBP
SUBL $V, %ESP
V = NUMERO SPAZIO OCCUPATO DALLE VARIABILI LOCALI

EPILOGO:
LEAVE
RET

MASTER COPY
Tel. 050 8312126
Cell. 388 9837745

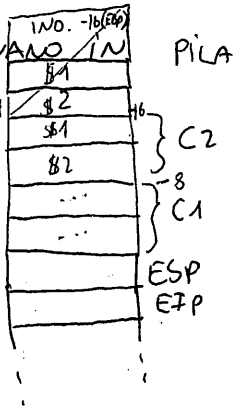


CHIAMATA: CHIAMANTE: - DOPO AVER INSERITO LE VAR. LOCALI CLASSICA CHE SPAZIO ANCHE PER IL RISULTATO DELLA FUNZIONE CHIAMATA, E. CL CLA (c, s1)) (SE FUNZ. DI UNA CLASSE)
- INSERISCE IN PILA I PARAMETRI ATTUALI
- INSERISCE COME ULTIMO ELEMENTO (IND. + NEGATIVO) IN PILA L'INDIRIZZO INIZIALE AL QUALE È ALLOCATO L'OGGETTO
- DOPO LA CHIAMATA EFFETTUA ADDL $X, %ESP, CON X = NUMERO LOCAZIONI OCCUPATE DA PARAM. PASSATI E DALL'INDIRIZZO INSERITO DELL'OGG. (RIPULISCE PILA)

PASSAGGIO PARAMETRI: SI SALVANO IN PILA AL CONTRARIO

E.: CLAI C2 (1, 2);

```
PUSHL $2
PUSHL $1
LEAL -16(%EBP), %EAX
PUSH %EAX
CALL FUNZIONE
ADDL $12, %ESP
```



MASTER COPY
Tel. 050 8312126
Cell. 388 9837745

- **DA B → L:** SI USA Es. `MOVSBL %DL, %EDX`

- **DA L → B:** Es. ~~`MOVB %EDX, %BL`~~ SE IL DATO É IN %ECX, BASTA USARE IL REG %CL

- **DIVISIONE x2:** SI USA `SAR %REG`

- **MOLTIPLICAZIONE x2:** SI USA `SAL %REG`

- **FOR:**
```
    MOVL $0, %ECX
    MOVL $4, -4(%EBP)
CICLO: CMP -4(%EBP), %ECX
    JE FINE
    ...
    INCL %ECX
    JMP CICLO
FINE: ...
```

- **IF:** Es. IF(X < Y) → `CMP X, Y` `JLE NOTIF` ... } X ≤ Y
  `NOTIF:` ... } X > Y oppure X ≤ Y dopo aver fatto tutto

FARE OPERAZIONE n VOLTE
```
LEAL 8(%EAX), %EDI   // COPIO n VOLTE
LEAL 12(%EBX), %ESI  // PARTENDO DA
CLD                  // 12(%EBX) (ESI)
MOVL $n, %ECX        // IN LOCAZIONI
REP MOVSL            // CONSECUTIVE SEMPRE
                     // MAGGIORI (CLD)
                     // IN 8(%EAX) (EDI)
                     // CIOÉ FINCHÉ ECX
                     // NON É 0, CONTINUO
                     // A FARE MOVSL %ESI, %EDI
                     // (REP)
```

- **FUNZ NON VOID:** NON SI SALVA (E SI RICARICA) %EAX IN PILA, IN QUANTO CONTIENE IL RISULTATO.

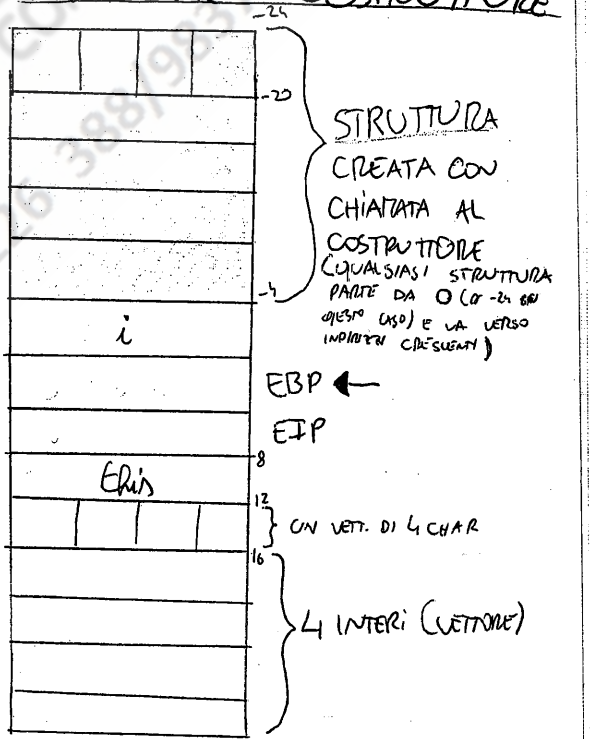
CREAZIONE OGGETTO CLASSE CON CHIAMATA AL COSTRUTTORE

Memory Records
ATTIVAZIONE CHIAMANTE

-24(%EBP) → É L'IND. DELLA STRUTTURA CREATA

CIOÉ PRIMA DELLA CHIAMATA DEL COSTRUTTORE, COME ULTIMO PARAMETRO SI METTE IN PILA L'INDIRIZZO DELL'OGGETTO CREATO

```
LEAL -24(%EBP), %EDX
PUSHL %EDX
```



⇒ QUANDO CI SI RIFERISCE AD UN OGGETTO CLASSE (Es. con EBis) O CON -24(%EBP), SI AGGIUNGONO POSIZIONI, CIOÉ L'OGGETTO PARTE DALL'INDIRIZZO E VA VERSO LOCAZIONI MAGGIORI
(Es. 4(%ECX) con %ECX = 8(%EBP) cioé EBis)

CIOÉ SI PRENDE L'INDIRIZZO DI QUELLA LOCAZIONE, UNA COSA C'É DENTRO A -24(%EBP)

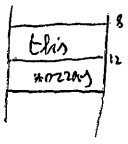
- **PASSARE VETTORE NATL VEL) AL CHIAMANTE:**
```
LEAL 16(%EBP), %ESI
MOVL %ESP, %EDI   (SUBL $m, %ESP)
MOVL $4, %ECX
REP MOVSL
```
(m = QUANTI BYTE DEVO METTERE IN PILA)

COSTRUTTORE:

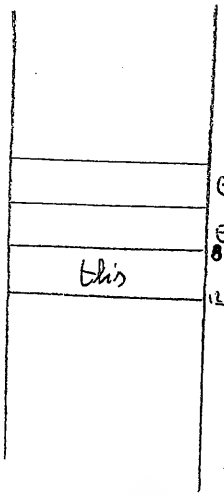
IL NOME AVRÀ CIE, SE HA SOLO PARAMETRI CLASSE, IL NOME SARÀ

-ZN ..... CIE ..... S-
KARAPEN

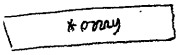
- SE NON HA PARAMETRI cl::cl()
IL NOME SARÀ -ZN2CLCIEV



es.
MOVL 12(%EBP), %EBX
ADDL $4, (%EBX, %ECX, 4)



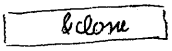
- int array[] → È UN PUNTATORE, OCCUPA 1 BYTE



- cl clone → OCCUPA QUANTO L'INTERA CLASSE



- cl &clone → RIFERIMENTO, OCCUPA 1 BYTE:

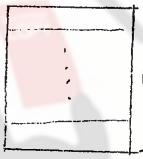


PER UTILIZZARLI BISOGNA PRIMA FARE MOVL IN UN REGISTRO CHE PRENDO L'INDIRIZZO DELL'OGGETTO PUNTATO (RIFERIM)

FUNZIONE MEMBRO: (NON VOID)

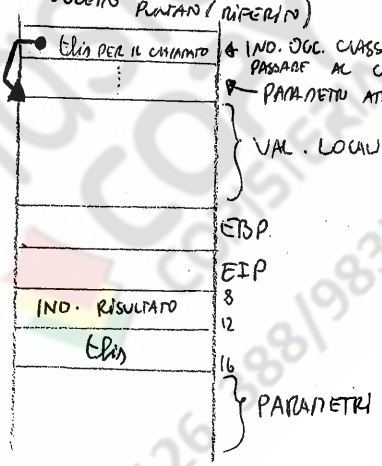
PARAMETRI DA PASSARE AL CHIAMATO:

SE cl cla → TUTTA LA CLASSE:



- SE *array o &cla → 1 BYTE (INDIRIZZO)





SE LA FUNZ. MEMBRO È VOID
LA PILA È UGUALE AL COSTRUTTORE

RITORNO DI VALORE: SE DA RITORNARE È UN OGGETTO CLASSE SERVE IL COSTRUTTORE DI COPIA:

- COSTRUTTORE DEFAULT (CIOÈ NON È PRESENTE):

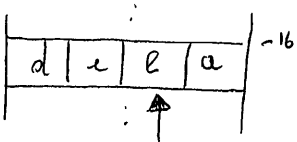
```
MOVL  8(%EBP), %EAX   ※IND. RISULTAT
MOVL  %EAX, %EDI
LEAL  -88(%EBP), %ESI  ※IND. OGGETTO DA
CLD                    ※RITORNARE
MOVL  $m_BYTE, %ECX
REP   MOVSL
```

- CON COSTRUTTORE DI COPIA:
(VA RICHIAMATO IL COSTR. DI COPIA)

```
LEAL  -152(%EBP), %EAX  ※INDIRIZZO OGGETTO DA RESTITUIRE
PUSHL %EAX
PUSHL %EBX  ※IND. RISULTATO (MOVL 8(%EBP), %EBX)
CALL  _ZN2CL.....
ADDL  $18, %ESP
MOVL  %EBX, %EAX
POPL  %
LEAVE
RET
```

MASTER COPY
Tel. 050 8312126
Cell. 388 9837745

RIFERIRE 1 BYTE:



⇒ MOVL -16(%.EBP), %.EBX

⇓

a = (%.EBX)

b = 1(%.EBX)

c = 2(%.EBX)

d = 3(%.EBX)

CMPL + SALTO CONDIZIONATO: (IF (cond))

CMPL source, dest] VIENE CONTROLLATO SE dest È cond DI source

Jcond

Esempio IF (a > c) ⇒

CMPL a, c

Jg TRUEIF

Jcc FALSEIF

POSSIBILI JMP:

JE : IF EQUAL

Jb, bE : IF Below / Below e Equal

JA, AE : IF Above / Above e Equal

JZ, : IF zero

i++ , ++i :

a = 2

b = 0 ⇒ b = a++ ⇒ a = 3, b = 2

a = 2

b = 0 ⇒ b = ++a ⇒ b = 3, a = 3

MASTER COPY
Tel. 050 8312126
Cell. 388 9837745

ESERCIZIO 2

SISTEMA.S: • CARICA_GATE TIPO_funztipo a_funz LIV_UTENTE

•
- .GLOBAL a_funz
- .EXTERN e_funz

a_funz:

- SALVA_REGISTRI // SE NON SI HA COMM. DI CONTESTO È VOID
  oppure
- SALVA_REG_RIT /// " " " ED NON È VOID.
  oppure
- CALL SALVA_STATO // SE CAMBIA CONTESTO

- COPIA_PARAM n_LONG n_REG
  - n_LONG: NUM. PARAMETRI PASSATI (num. PAROLE MACCHINE)
  - n_REG: 0 SE FACCIO SALVA STATO oppure 7 SE FACCIO SALVA_REGISTRI oppure 6 SE FACCIO SALVA_REG_RIT (NON C'È EAX)

- ADDL $n_LONG, %ESP // SE HO PASSAGGIO PARAM DA NON CAMBIO CONTESTO

- CALL e_funz

CARICA_REGISTRI
oppure
CARICA_REG_RIT
oppure
CALL CARICA_STATO

- IRET

MASTER COPY
Tel. 050 8312126
Cell. 388 9837745