



Programmi di
dump esadecimale

R. Medves

35

CNUCE

NOTA TECNICA

Copyright C ottobre 1973


by C. N. U. C. E. Centro Nazionale Universitario di Calcolo Elettronico
dell'Università di Pisa.

Introduzione

Queste note descrivono un gruppo di programmi scritti in linguaggio Assembler e inseriti nelle librerie del sistema OS che permettono di analizzare in formato esadecimale zone di memoria all'interno di un proprio programma o di stampare dump parziali o totali di files su nastro magnetico o su disco. L'uso di tali programmi è stato facilitato con l'introduzione di opportune procedure catalogate all'interno del sistema OS e di macro definizioni.

R M D U M P

E' una subroutine scritta in linguaggio Assembler, richiamabile da un altro programma tramite la macro CALL.

Questa subroutine riceve come argomenti due indirizzi di memoria; la zona di memoria compresa tra i due indirizzi viene esaminata 4 bits alla volta e il contenuto, convertito in cifra esadecimale, viene stampato in uscita.

La subroutine può venire chiamata con 2 o 4 argomenti.

I primi due argomenti che devono sempre essere forniti, indicano l'indirizzo di partenza e di arrivo tra i quali si vuole ottenere il dump, il terzo argomento facoltativo, può contenere una stringa alfanumerica di caratteri, che verrà stampata sull'uscita, all'inizio del dump desiderato, e che servirà come intestazione di riconoscimento, il quarto parametro sarà in tal caso una mezza voce che conterrà la lunghezza, in binario, della stringa alfanumerica. Esempi:

1) Codifica: CALL RMDUMP, (AREA1, AREA2)

Uscita: 0003A806 0009B4C5 F7F21A20 . . .

2) Codifica: CALL RMDUMP, (AREA, AREA+6, TITOLO, L)

```
TITOLO DC 'DUMP DI AREA'
      L  DC H'12'
```

Uscita: DUMP DI AREA

0003A806 009B

La subroutine è stata inoltre predisposta a saltare a pagina nuova solo la prima volta che essa viene chiamata, sia che debba o meno stampare l'intestazione, mentre le chiamate successive saltano solo due righe: ciò è ottenuto rendendo il programma non rientrante e non serialmente riúsabile a livello del singolo task; solo una nuova compilazione riporta il programma alle condizioni iniziali.

Inoltre, in uscita, l'area da stampare viene suddivisa in blocchi per una lettura facilitata: su di una riga vengono stampate 5 voci (blocchi di 4 bytes ciascuno) separate fra di loro da un carattere bianco, altre 5 voci vengono stampate sulla riga successiva e così via fino all'ultima.

Nelle pagine seguenti diamo la lista della routine.

```

      LA      8,1(0,8)
*   SEPARA CON UN BLANK COPPIE DI CIFRE ESADECIMALI
ZZZZZZ12 BC   15,ZZZZZZ11
      LA      8,1(0,8)
ZZZZZZ11 XI   ZZZZZZ12+1,X'FO'
      XI      ZZZZZZ14+1,X'FO'
ZZZZZZ14 BC   0,ZZZZZZ15
      BCT     12,ZZZZZZ15
      B       ZZZZZZ16
ZZZZZZ15 NOP  ZZZZZZ15
      BCT     10,ZZZZZZ10   CICLA FINCHE' NON HA STAMPATO UNA WORD
      LA      3,4(0,3)
      LA      8,5(0,8)       SEPARA DUE WORDS CON 5 BLANK
*   CICLA FINCHE' NON HA STAMPATO UNA RIGA (5 WORDS)
      BCT     11,ZZZZZZ09
ZZZZZZ16 PUT  ZZZZZZ07,ZZZZZZ06   STAMPA UNA RIGA
*   CICLA FINCHE' NON HA STAMPATO TUTTE LE RIGHE
      BCT     5,ZZZZZZ17
      MVI     ZZZZZZ06,C' '
      MVC     ZZZZZZ06+1(132),ZZZZZZ06
      MVI     ZZZZZZ06,C'0'
      OI      ZZZZZZ13+1,X'FO'
ZZZZZZ18 NOP  ZZZZZZ18
      CLOSE  (ZZZZZZ07)
      L       13,4(13)
      RETURN (14,12)
ZZZZZZ01 DC   18F'0'
ZZZZZZ02 DC   C'0123456789ABCDEF'
ZZZZZZ03 DC   F'3'
ZZZZZZ04 DC   F'4'
ZZZZZZ05 DC   F'5'
ZZZZZZ06 DS   0CL133
      DC     C'1'
      DC     132C' '
ZZZZZZ07 DCB  DSORG=PS,MACRF=(PM),DDNAME=ZZZZZZ00
      END    RMDUMP

```

Per rendere più affidabile e più semplice l'uso della routine RMDUMP, è stata inoltre creata una macro istruzione che porta lo stesso nome. Tale macro istruzione può essere codificata in due maniere distinte:

a) Con passaggio di parametri: RMDUMP (AREA1,AREA2).

In tal caso la macro provvede a chiamare la routine di dump, passando, oltre agli indirizzi AREA1 e AREA2 tra i quali si vuole ottenere il dump, anche una stringa alfanumerica del tipo '(AREA1,AREA2)' con la rispettiva lunghezza in bytes, stringa che verrà stampata in uscita come intestazione.

b) Senza passaggio di parametri: RMDUMP.

In tal caso la macro provvede a memorizzare in una zona di memoria temporanea i 15 registri generali e successivamente a richiamare la routine di dump per l'analisi di tale area.

La stringa alfanumerica di intestazione sarà in tal caso '(REGS 0-15)'.

In entrambi i casi, la macro provvede preventivamente al salvataggio dei 15 registri generali in un'area di comodo e al caricamento nel registro 13 di un'area di salvataggio temporanea: ciò permette la massima affidabilità sull'uso della routine e l'inserimento della macro in un punto qualsiasi di un programma, a scopo di debug, senza il pericolo di danneggiare i registri generali o il contenuto della eventuale "save area" del programma chiamante.

```

MACRO
&NAME RMDUMP &PARM
      LCLA &A
      AIF (N°&PARM NE 0).ERO
&A    SETA 11
      CNOP 0,4
      B R&SYSNDX.M0
R&SYSNDX.M DS 34F
R&SYSNDX.M2 DC H°&A°
R&SYSNDX.M1 DC C°(REGS 0-15)°
R&SYSNDX.M0 STM 0,15,R&SYSNDX.M
      LA 13,R&SYSNDX.M+64
      CALL RMDUMP,(R&SYSNDX.M,R&SYSNDX.M+64,
R&SYSNDX.M1,R&SYSNDX.M2),VL X
      LM 0,15,R&SYSNDX.M
MEXIT
°ERO  AIF (N°&PARM NE 2).ER1
&A    SETA K°&PARM
      CNOP 0,4
      B R&SYSNDX.M0
R&SYSNDX.M DS 34F
R&SYSNDX.M2 DC H°&A°
R&SYSNDX.M1 DC C°&PARM°
R&SYSNDX.M0 STM 0,15,R&SYSNDX.M
      LA 13,R&SYSNDX.M+64
      CALL RMDUMP,(&PARM(1),&PARM(2),
R&SYSNDX.M1,R&SYSNDX.M2),VL X
      LM 0,15,R&SYSNDX.M
MEXIT
°ER1  MNOTE 'ERRATO NUMERO DI PARAMETRI'
      MEXIT
      MEND

```

R M T A P E

Questa routine che permette la lettura e la stampa in formato esadecimale del contenuto di un nastro o di un data set su disco.

Alla subroutine viene passato come parametro il numero di blocchi che devono essere esaminati su nastro: se tale numero è negativo, ciò sta a significare che deve essere listato tutto il file su nastro.

Al momento della codifica del programma che richiama tale subroutine, si dovrà provvedere a fornire nel parametro DCB, un valore per il BLKSIZE superiore o al più uguale alla massima lunghezza del blocco che si pensa possa trovarsi su tale nastro.

Ciò permette l'acquisizione dinamica di un buffer di lunghezza sufficiente a contenere qualsiasi record letto e ad effettuare così il successivo esame.

Esempio di chiamata, per la lettura di 12 blocchi da nastro:

```
[ CALL  RMTAPE, (N)
  N    DC    F'12'
```

E' da notare che tale routine non fa alcuna distinzione tra dati, parole di controllo all'inizio dei records o blocchi e label di nastro, e che pertanto permette la stampa e il controllo di tutti questi campi e in particolare della label di volume e delle informazioni di controllo su nastro.

```

RMTAPE    OBJECT
SAVE (14,12),,*
BALR 2,0
USING *,2
LA 12,SAVE1
ST 12,8(13)
ST 13,4(12)
LR 13,12
L        3,0(1)
L        3,0(3)
LTR      3,3
BC       11,CONT
NI       TORNA,X'00'
NI       TORNA+1,X'00'
OI       TORNA,X'47'
OI       TORNA+1,X'F0'
CONT     NOP      CONT
OPEN     (DCBIN)
USING    IHADCB,4
LA       4,DCBIN
LH       0,DCBBLKSI
GETMAIN R,LV=(0)
LR       5,1
LEGGE    LR       6,5
GET      DCBIN,(5)
LH       7,DCBLRECL
AR       6,7
CALL     RMDUMP,((5),(6))
TORNA    BCT      3,LEGGE
FINE     CLOSE   (DCBIN)
L        13,4(13)
RETURN  (14,12)
SAVE1    DS      18F
DCBIN    DCB     DSORG=PS,MACRF=(GM),DDNAME=DDIN,EODAD=FINE
         DCBD    DSORG=PS
         END     RMTAPE

```

TPDUMP

Allo scopo di rendere più flessibile il programma RMTAPE precedentemente descritto, è stato creato il programma TPDUMP, richiamabile tramite l'uso di una procedura catalogata con lo stesso nome.

L'uso di tale programma e di tale procedura permettono di eseguire tutte le analisi precedentemente descritte tramite un programma assembler ma direttamente, richiamando in esecuzione il programma TPDUMP tramite una scheda EXEC.

Ciò permette una estrema facilità di uso anche da parte di chi non conosce il linguaggio Assembler.

Il parametro NREC, nella procedura catalogata e nel programma, è posto uguale a F per avere un dump completo di tutto un file di un nastro: in tal caso infatti il programma TPDUMP provvede a richiamare la subroutine RMTAPE passando come parametro un numero negativo.

Se invece il parametro NREC è posto uguale a P, si ha un dump parziale di 5 blocchi del nastro, che sono sufficienti, nella maggior parte dei casi, a individuare il contenuto del nastro.

```
TPDUMP  RMSTART (12,SAVE)
L        3,0(1)
CLI     2(3),C'F'
BNE     PARZ
L        2,=F'-1'
ST      2,N
B       DUMP
PARZ    CLI     2(3),C'P'
        BNE     FINE
        L        2,=F'5'
        ST      2,N
DUMP    NOP     DUMP
        CALL    RMTAPE,(N)
FINE    RMRETURN
N       DC      F'0'
        END     TPDUMP
```

```
MEMBER NAME TPDUMP
//GO PROC JNIT=TAPE9, BLKSIZE=3000, SER=$DUMMY, NREC=F, DEN=2 0000
//GO EXEC PGM=TPDUMP, PARM='&NREC' 0000
//STEPLIB DD DSN=SYS1.LINK2, DISP=SHR 0000
//ZZZZZZ00 DD SYSOUT=A, DCB=(RECFM=FA, BLKSIZE=133) 0000
//DDIN DD JNIT=&UNIT, VOL=SER=&SER, LABEL=(, BLP), DISP=OLD, 0000
// DCB=(RECFM=U, BLKSIZE=&BLKSIZE, DEN=&DEN) 0000
```