

IST. EL. INF.
BIBLIOTECA
Posiz. ARCHIVIO

B4-26

Consiglio Nazionale delle Ricerche

**ISTITUTO DI ELABORAZIONE
DELLA INFORMAZIONE**

PISA

**PIP: Una procedura per la conversione dei formati
dati BIS-386/IPS² GEPIM**

Progetto IEI-AERITALIA 1989

Enrico Fantini

Nota Tecnica B4-26

GIUGNO 1990

PIP : Una procedura per la conversione dei formati dati BIS-386/IPS² GEPIM

Enrico Fantini

La procedura denominata PIP, è nata dalla necessità di scambiare file immagini tra i sistemi IPS², BIS-386, operanti in ambiente MS-DOS da un lato e il sistema GEPIM, operante in ambiente GOULD MPX-32, dall'altro.

Mentre per i primi due sistemi, IPS² e BIS-386, il formato con cui vengono archiviate le immagini su disco è lo stesso, per il terzo GEPIM è stata effettuata una scelta diversa, dettata dalle possibilità offerte al momento del progetto e dalle necessità dettate dagli algoritmi che si andavano ad implementare.

Il formato riconosciuto dai sistemi IPS² e BIS-386 (fig.1) è caratterizzato dall'assenza, all'interno dei dati, di caratteri di controllo ed il file si presenta come un insieme di byte ognuno dei quali rappresenta un pixel dell'immagine, che è organizzata una riga dietro l'altra. All'interno del file i primi due bytes riportano il numero delle righe dell'immagine e i secondi due le colonne.

L'archiviazione di tutti gli altri parametri caratteristici dell'immagine, quali i valori minimo e massimo presenti nei dati, i valori dell'istogramma, le eventuali note esplicative etc., viene effettuata su una tabella separata legata al file dei dati da un riferimento univoco.

Il file immagine archiviato nel formato GEPIM (fig.2) contiene al suo interno e più precisamente nei suoi primi 256 bytes tutte le notizie riguardanti l'immagine. Inoltre la memorizzazione dei dati avviene intercalando ad ogni riga tutti quei caratteri di controllo, tipicamente LF e CR, necessari per il loro corretto reperimento. Ogni pixel è rappresentato su 16 bit anche se non sarebbe necessario visto che i livelli di grigio rappresentabili dal sistema di restituzione variano dal valore 0 a 255.

La procedura è organizzata come un file di comandi TSM che attraverso delle scelte effettuate dall'operatore, attiva quattro moduli preposti alla trasformazione dei dati.

I quattro moduli, che verranno illustrati più diffusamente nelle schede riportate di seguito, sono:

PIPFIL - per la trasformazione di un file dal formato BIS-386/IPS² al formato GEPIM

FILEPIP - per la trasformazione di un file dal formato GEPIM al formato BIS-386/IPS²

PIPSVP - per visualizzare sulla stazione grafico-pittorica SVP2000 una immagine archiviata con il formato BIS-386/IPS²

SVPPIP - per l'archiviazione nel formato BIS-386/IPS² di una immagine presente nella memoria di quadro della stazione SVP2000.

Si ricorda che allo stato attuale, in mancanza di una unità a floppy-disk installata in ambiente MPX-32, l'unico sistema possibile per il trasferimento dei file dati tra il sistema operante in MS-DOS e quello in MPX-32 è rappresentato dal collegamento attraverso la rete Ethernet dove questa sia disponibile, pertanto per l'utilizzo della rete si rimanda ai manuali relativi al collegamento dei due ambienti operativi.

La procedura richiede che siano rispettate le seguenti condizioni, salvo modifiche dei pathname inseriti all'interno dei comandi PIP.

- il file di comandi MPX denominato PIP deve necessariamente risiedere nella directory SYSTEM del disco di sistema

- sul disco di sistema deve essere presente una directory di nome UTILITY dove devono essere copiati i file in formato eseguibile denominati:

CLEVI (routine di servizio per il clear dello schermo dal terminale)

PIPMASK (è la maschera che viene presentata sul terminale all'avvio della procedura)

PIPFIL, FILEPIP, PIPSVP, SVPPIP (programmi per la trasformazione dei formati).

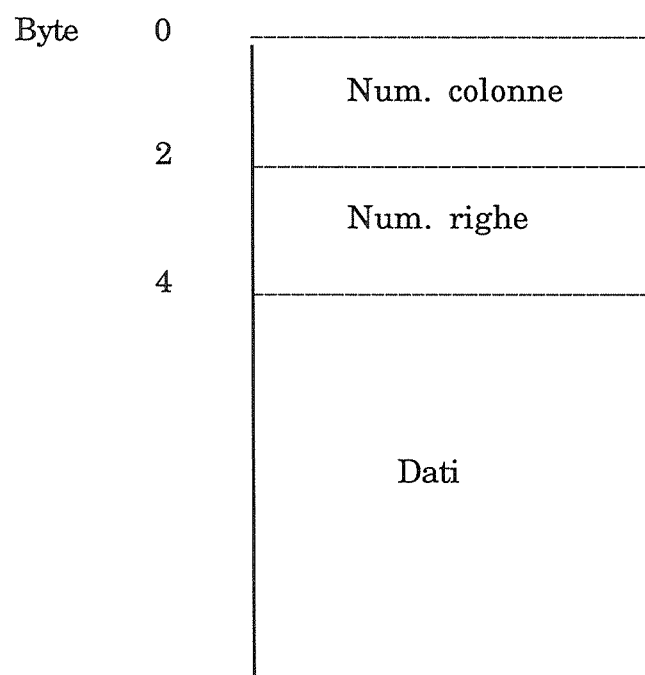


Fig. 1 - Rappresentazione del formato dati in ambiente BIS-386/IPS²

Word	0	Data creazione FILE <hr/> Formato dati (I,F,D) <hr/> Num. righe <hr/> Num. colonne <hr/> * SC * <hr/> Area USER (MIN,MAX,etc.) <hr/> Commento <hr/> Dati
(2 byte)	3	
	4	
	5	
	6	
	8	
	98	
	128	

Fig. 2 - Rappresentazione del formato dati in ambiente GEPIM

Modulo: PIPFILE

Funzione: Leggendo un file immagine archiviato nel formato BIS-386/IPS² restituisce in uscita un file dati contenente la stessa immagine ma archiviata nel formato GEPIM.

Input:

- Nome del file di ingresso (quello nel formato BIS-386/IPS²).
- Nome del file di uscita (quello nel formato GEPIM).

Output: File archiviato su disco.

Gestione errori: Vengono generati vari messaggi di errore visualizzati sul terminale che si sta utilizzando per il dialogo. Alle segnalazioni di errore bisogna sempre rispondere con un RETURN per poter continuare nel lavoro.

Librerie usate: GEPIM.LIB

Note: L'immagine da convertire non può avere dimensioni maggiori di 2048 righe o di 2048 colonne. Nell'impostare i nomi dei file seguire le regole dettate dal S.O. MPX-32 e cioè [disco] [(directory)] nomefile con i parametri chiusi tra parentesi quadre di carattere opzionale. Al termine della conversione viene data all'operatore la possibilità di effettuarne una successiva trasformazione senza uscire dal modulo.

Modulo: FILEPIP

Funzione: Legge un file immagine archiviato nel formato GEPIM e restituisce in uscita un file dati contenente la stessa immagine ma archiviata nel formato BIS-386/IPS².

Input:

- Nome del file di ingresso (quello nel formato GEPIM).
- Nome del file di uscita (quello nel formato BIS-386/IPS²).

Output: File archiviato su disco nel formato BIS-386/IPS².

Gestione errori: Vengono generati vari messaggi di errore visualizzati sul terminale che si sta utilizzando per il dialogo.

Alle segnalazioni di errore bisogna sempre rispondere con un RETURN per poter continuare nel lavoro.

Librerie usate: GEPIM.LIB

Note: L'immagine da convertire non può avere dimensioni maggiori di 2048 righe o di 2048 colonne. Nell'impostare i nomi dei file bisogna seguire le regole dettate dal S.O. MPX-32 e cioè [disco] [(directory)] nomefile con i parametri chiusi tra parentesi quadre di carattere opzionale. Al termine della conversione viene data all'operatore la possibilità di effettuarne una successiva trasformazione senza uscire dal modulo.

Modulo: PIPSVP

Funzione: Visualizza su una memoria di quadro della stazione SVP2000 una immagine letta da un file di disco archiviato nel formato BIS-386/IPS².

Input:

- Nome del file di ingresso (quello nel formato BIS-386/IPS²).
- Numero della memoria di quadro dell'SVP2000 (1÷4).
- Coordinate x,y delle memorie di quadro da cui si deve iniziare la visualizzazione dell'immagine.

Output: Immagine visualizzata sulla memoria di quadro scelta dall'operatore.

Gestione errori: Vengono generati vari messaggi di errore visualizzati sul terminale che si sta utilizzando per il dialogo.

Alle segnalazioni di errore bisogna sempre rispondere con un RETURN per poter continuare nel lavoro.

Inoltre viene visualizzato un messaggio di errore se la scelta delle coordinate di inizio non lascia lo spazio sufficiente per la visualizzazione dell'immagine cioè se $(\text{inizio} + \text{lunghezza} \text{ (o larghezza)} - 1) > 512$.

Librerie usate: GEPIM.LIB - SVP.LIB

Note: L'immagine da visualizzare non può essere più grande di 512x512. Nell'impostare il nome dei file di ingresso bisogna seguire le regole dettate dal S.O. MPX-32 e cioè [disco] [(directory)] nomefile con i parametri chiusi tra parentesi quadre di carattere opzionale. Al termine della conversione viene data all'operatore la possibilità di effettuarne una successiva trasformazione senza uscire dal modulo.

Modulo: SVPPIP

Funzione: Legge i dati da una memoria di quadro della stazione SVP2000 e restituisce in uscita un file dati contenente l'immagine archiviata nel formato BIS-386/IPS².

Input:

- Numero della memoria di quadro della stazione SVP2000 da cui si vuole prelevare l'immagine.
- Coordinate x,y di inizio, di fine e il passo di scansione di riga e colonna all'interno dell'immagine.
- Nome del file su cui si vuole archiviare l'immagine.

Output: File di disco archiviato nel formato BIS-386/IPS².

Gestione errori: Vengono generati vari messaggi di errore visualizzati sul terminale che si sta utilizzando per il dialogo.
Alle segnalazioni di errore bisogna sempre rispondere con un RETURN per poter continuare nel lavoro.

Librerie usate: GEPIM.LIB - SVP.LIB

Note: L'immagine da archiviare non deve superare le dimensioni di 512 righe o 512 colonne. Nell'impostare il nome del file di uscita bisogna seguire le regole dettate dal S.O. MPX-32 e cioè [disco] [(directory)] nomefile con i parametri chiusi tra parentesi quadre di carattere opzionale. Al termine della conversione viene data all'operatore la possibilità di effettuare una successiva trasformazione senza uscire dal modulo.

```

*--I.E.I.-C.N.R.--Pisa-----E. Fantini-1989-*
!
!          P P P P P P P P P P          I I I I I I I I I I          P P P P P P P P P P          !
!          P P P P P P P P P P          I I I I I I I I I I          P P P P P P P P P P          !
!          P P P P   P P P P          I I I I          P P P P   P P P P          !
!          P P P P   P P P P          I I I I          P P P P   P P P P          !
!          P P P P   P P P P          I I I I          P P P P   P P P P          !
!          P P P P P P P P P P          I I I I          P P P P P P P P P P          !
!          P P P P P P P P P P          I I I I          P P P P P P P P P P          !
!          P P P P          I I I I          P P P P          !
!          P P P P          I I I I          P P P P          !
!          P P P P          I I I I I I I I I I          P P P P          !
!          P P P P          I I I I I I I I I I          P P P P          !
!
!          Trasformazione di un file in formato PIP-1024 in un file          !
!          in formato DGEFI o in una immagine su frame memory dell'SVP.          !
!          E viceversa.          !
!
*-----*

```

```

DEFM TIPO
OPTION NOCOM
PAGE 0
@SYSTEM(UTILITY)CLEVI
IFA %TIPO LMASK
IFT %TIPO EQ C SCELTA
GOTO CONTIP
%LMASK
LIST @SYSTEM(UTILITY)PIPMASK
SET %1
@SYSTEM(UTILITY)CLEVI
%SCELTA
NOTE
NOTE
NOTE          #####   EFFETTUARE   SCELTA   #####
NOTE
NOTE
NOTE
NOTE
NOTE
NOTE   (F) = File      PIP1024  -----> File      DGEFI
NOTE   (D) = File      DGEFI    -----> File      PIP1024
NOTE   (I) = Immagine  SVP2000  -----> File      PIP1024
NOTE   (S) = File      PIP1024  -----> Immagine  SVP2000
NOTE
NOTE   (X) = FINE
NOTE
NOTE
NOTE   (F,D,I,S,X)  =====>
SET %TIPO
@SYSTEM(UTILITY)CLEVI
%CONTIP
PAGE
IFT %TIPO EQ F PIPDGEFI
IFT %TIPO EQ D DGEFIPIP
IFT %TIPO EQ I SVPPPIP
IFT %TIPO EQ S PIPSV
IFT %TIPO EQ X ENDE
GOTO ERRO
%PIPDGEFI
@SYSTEM(UTILITY)PIPFIL
GOTO FINE
%DGEFIPIP
@SYSTEM(UTILITY)FILEPIP
GOTO FINE
%SVPPPIP
@SYSTEM(UTILITY)SVPPPIP
GOTO FINE
%PIPSVP
@SYSTEM(UTILITY)PIPSVP
GOTO FINE
%ERRO
@SYSTEM(UTILITY)CLEVI
NOTE
NOTE   *-----*
NOTE   !   ERRORE NEL PARAMETRO.   !
NOTE   !   VALORI POSSIBILI :   !
NOTE   !   !   !   !   !
NOTE   !   VALORE           INPUT           OUTPUT           !
NOTE   !   !   !   !   !   !
NOTE   !   F             PIP1024           DGEFI           !
NOTE   !   D             DGEFI           PIP1024         !
NOTE   !   I             SVP2000          PIP1024         !
NOTE   !   S             PIP1024          SVP2000         !
NOTE   !   !   !   !   !   !
NOTE   !   X   =       FINE PROCEDURA   !
NOTE   !   !   !   !   !   !
NOTE   !   Premi RETURN per continuare   !
NOTE   !   !   !   !   !   !
NOTE   *-----*
NOTE
NOTE
SET %TIPO
%FINE

```

```
SELECT @SYSTEM(SYSTEM) PIP C  
ENDM
```

```
PROGRAM CLEVI  
INTEGER*1 CLR  
DATA CLR / 2Z1A /  
TYPE 1000,CLR  
1000 FORMAT(1H ,A1)  
STOP  
END
```

```
$JOB KK SLOF=FFF
$OPTION 2 3 4 5
$FORT77
```

```
      IMPLICIT INTEGER*2 (A-Z)
      CHARACTER NOME*64,TTY*6,RIT*1
      INTEGER*4 NULLO
      INTEGER*1 BUF(768)
      INTEGER*2 VET(2048),NOMEF(3),ICH(8),ICR(8)
      EQUIVALENCE (NULLO,BUF(1))

C
      LFCTTY=1
      CALL OPENTTY(LFCTTY,TTY)
89      IDCB=10
      LFC=777

C
11      CALL CLEAR
      WRITE(1,22)
22      FORMAT(/17X,'---> TRASFORMAZIONE FILE DGEFI <---',//,
-5X,'INGRESSO : File disco formato DGEFI (max 2048x2048)',//,
-5X,'USCITA : File disco formato PIP-1024 (stesse dimensioni)',
-//5X,'(Rev. 1.0 - Gennaio 1989)',//,
-5x,'(AUTORE : Fantini E. - I.E.I. C.N.R. Pisa)',////,
-17X,'##### SELEZIONE DEL FILE DI INGRESSO #####')
      CALL APRE(IDCB,NOMEF,ICH,ICR,1,IFO,VET,1,NREC,LUN,1)
      IF(LUN.LT.1.OR.LUN.GT.2048.OR.NREC.LT.1.OR.NREC.GT.2048) THEN
          CALL RING
          WRITE(1,334)
334      FORMAT(/' ##### ERRORE nelle dimensioni del file',
- ' #####',/, ' ---> Premere RETURN per continuare <---')
          READ(1,'(A)') RIT
          CALL CHIUD(IDCB,0)
          GOTO 11
      END IF
      WRITE(1,63)
63      FORMAT(/17X,'##### SELEZIONE DEL FILE DI USCITA #####')
      WRITE(1,1)
1      FORMAT(/' Nome del file trasformato (RETURN = FINE) : ')
      READ(1,'(A)') NOME
      IF(NOME.EQ.' ') GOTO 1000
      OPEN(LFC,FILE=NOME,STATUS='NEW',IOSTAT=IST)
      IF(IST.EQ.2) THEN
          CALL RING
          WRITE(1,222)
222      FORMAT(/' ##### ERRORE. Il file specificato esiste',
- ' #####',/, ' ---> Premere RETURN per continuare <---')
          READ(1,'(A)') RIT
          GOTO 11
      END IF
      IF(IST.NE.0) THEN
          WRITE(1,333) IST
          CALL RING
333      FORMAT(/' ##### ERRORE nella apertura del file : ',I4,
- ' #####',/, ' ---> Premere RETURN per continuare <---')
          READ(1,'(A)') RIT
          GOTO 11
      END IF
      CALL SWAPBYTE(NREC,LUN,NULLO)
      WRITE(1,2) LUN,NREC
2      FORMAT(/' Lunghezza record : ',I4,/,
- ' Numero di record : ',I4)
      INDB=5
      DO 100 K=1,NREC
          CALL LESC(IDCB,0,IFO,VET,K,LUN)
          DO 200 J=1,LUN
              IF(INDB.GT.768) THEN
                  CALL BUFFEROUT(LFC,1,BUF,192,,)
                  CALL M:WAIT(LFC)
                  INDB=1
              END IF
              BUF(INDB)=VET(J)
              INDB=INDB+1
          200      CONTINUE
```

```

100 CONTINUE
    INDB=INDB-1
    INUM=(INDB+3)/4
    IF (INUM.GT.0) THEN
        CALL BUFFEROUT(LFC,1,BUF,INUM,,)
        CALL M:WAIT(LFC)
    END IF
    CALL CHIUD(IDC,0)
    CLOSE(LFC)
78 WRITE(1,90)
90 FORMAT(// ' Altro file (s/n) [n] ? : ' )
    READ(1,'(A)') RIT
    IF(RIT.EQ.' ') GOTO 1000
    IF(RIT.EQ.'S'.OR.RIT.EQ.'s') GOTO 89
    IF(RIT.NE.'N'.AND.RIT.NE.'n') GOTO 78
1000 CALL EXITTASK
    END

```

```

C
C
C

```

```

SUBROUTINE SWAPBYTE(NREC,LUN,PASS)
IMPLICIT INTEGER*2 (A-Z)
INTEGER*4 PASS
INTEGER*4 NULLO
INTEGER*2 LN,NR
INTEGER*1 COM01,LR(4)
EQUIVALENCE (LN,LR(1)),(NR,LR(3))
EQUIVALENCE (NULLO,LR(1))
LN=LUN
NR=NREC
COM01=LR(1)
LR(1)=LR(2)
LR(2)=COM01
COM01=LR(3)
LR(3)=LR(4)
LR(4)=COM01
PASS=NULLO
RETURN
END

```

```

$IFT ABORT EXIT
$AS LIB TO @SYSTEM(LIBRERIE)GEPIM.LIB BLOC=N
$AS DIR TO @SYSTEM(LIBRERIE)GEPIM.DIR BLOC=N
$CATALOG
BUIL FILEPIP NOM
$EOJ
$$

```

```
$JOB KK SLOF=FFF
$OPTION 2 3 4 5
$FORT77
```

```
    IMPLICIT INTEGER*2 (A-Z)
    CHARACTER NOME*64,TTY*6,RIT*1
    INTEGER*1 BUF1(768)
    INTEGER*2 VET(2048),NOMEF(3),ICH(8),ICR(8),MINMAX(3)
    INTEGER*4 COMO4(2)
    EQUIVALENCE (COMO4(1),BUF1(1))

C
    LFCTTY=1
    CALL OPENTTY(LFCTTY,TTY)
89    IDCB=10
    LFC=777
    MIN=255
    MAX=0

C
11    CALL CLEAR
    WRITE(1,22)
22    FORMAT(/17X,'---> TRASFORMAZIONE FILE PIP-1024 <---',//,
-5X,'INGRESSO : File disco formato PIP-1024 (max 2048x2048)',//,
-5X,'USCITA : File disco formato DGEFI (stesse dimensioni)',
-//5X,'(Rev. 1.0 - Gennaio 1989)',//,
-5X,'(AUTORE : Fantini E. - I.E.I. C.N.R. Pisa)',////,
-17X,'##### SELEZIONE DEL FILE DI INGRESSO #####')
    WRITE(1,1)
1    FORMAT(// ' Nome del file da trasformare (RETURN = FINE) : ')
    READ(1,'(A)') NOME
    IF(NOME.EQ.' ') GOTO 1000
    OPEN(LFC,FILE=NOME,STATUS='OLD',IOSTAT=IST)
    IF(IST.EQ.1) THEN
        CALL RING
        WRITE(1,222)
222    FORMAT(/' ##### ERRORE. Il file specificato non esiste',
-' #####',/, ' ---> Premere RETURN per continuare <---')
        READ(1,'(A)') RIT
        GOTO 11
    END IF
    IF(IST.NE.0) THEN
        WRITE(1,333) IST
        CALL RING
333    FORMAT(/' ##### ERRORE nella apertura del file : ',I4,/,
-' #####',/, ' ---> Premere RETURN per continuare <---')
        READ(1,'(A)') RIT
        GOTO 11
    END IF
    CALL BREAD(LFC,BUF1)
    CALL LUNREC(BUF1,LUN,NREC)
    WRITE(1,2) LUN,NREC
2    FORMAT(/' Lunghezza record : ',I4,/,
-' ' Numero di record : ',I4)
    IF(LUN.LT.1.OR.LUN.GT.2048.OR.NREC.LT.1.OR.NREC.GT.2048) THEN
        CALL RING
        WRITE(1,334)
334    FORMAT(/' ##### ERRORE nelle dimensioni del file',
-' #####',/, ' ---> Premere RETURN per continuare <---')
        READ(1,'(A)') RIT
        GOTO 11
    END IF
    WRITE(1,63)
63    FORMAT(/17X,'##### SELEZIONE DEL FILE DI USCITA #####')
    CALL APRE(IDCB,NOMEF,ICH,ICR,-1,IFO,VET,1)
    INDV=1
    INDB=5
    RIGA=1
44    IF(INDB.GT.768) THEN
        CALL BREAD(LFC,BUF1)
        INDB=1
    END IF
    IF(INDV.GT.LUN) THEN
        CALL LESC(IDCB,1,IFO,VET,RIGA,LUN)
        INDV=1
```



```

        RIGA=RIGA+1
        IF (RIGA.GT.NREC) GOTO 999
    END IF
    CALL RIEMPI (INDB, INDV, BUF1, VET, LUN, MIN, MAX)
    GOTO 44
C
C
999  CLOSE (LFC)
      MINMAX(1)=2
      MINMAX(2)=MIN
      MINMAX(3)=MAX
      CALL CHIUD (IDCB, 1, IFO, VET, NREC, LUN, MINMAX)
78  WRITE (1, 90)
90  FORMAT (// ' Altro file (s/n) [n] ? : ' )
      READ (1, ' (A) ' ) RIT
      IF (RIT.EQ. ' ' ) GOTO 1000
      IF (RIT.EQ. ' S ' .OR. RIT.EQ. ' s ' ) GOTO 89
      IF (RIT.NE. ' N ' .AND. RIT.NE. ' n ' ) GOTO 78
1000 CALL EXITTASK
      END
C
C
C
      SUBROUTINE LUNREC (BUF, LUN, NREC)
      IMPLICIT INTEGER*2 (A-Z)
      INTEGER*2 LN, NR
      INTEGER*1 COM01, LR(4), BUF(1)
      EQUIVALENCE (LN, LR(1)), (NR, LR(3))
      DO 1 K=1, 4
1      LR(K)=BUF(K)
      COM01=LR(1)
      LR(1)=LR(2)
      LR(2)=COM01
      LUN=LN
      COM01=LR(3)
      LR(3)=LR(4)
      LR(4)=COM01
      NREC=NR
      RETURN
      END
C
C
C
      SUBROUTINE BREAD (LFC, BUF)
      IMPLICIT INTEGER*2 (A-Z)
      INTEGER*1 BUF(1)
      CALL BUFFERIN (LFC, 1, BUF, 192, , )
      CALL M:WAIT (LFC)
      RETURN
      END
C
C
C
      SUBROUTINE RIEMPI (INDB, INDV, BUF, VET, LUN, MIN, MAX)
      IMPLICIT INTEGER*2 (A-Z)
      INTEGER*1 BUF(1)
      INTEGER*2 VET(1)
C
      FINE=INDB+LUN-INDV
      IF (FINE.GT.768) FINE=768
      J=INDV
      DO 1 K=INDB, FINE
          VET(J)=BUF(K)
          IF (VET(J).LT.MIN) MIN=VET(J)
          IF (VET(J).GT.MAX) MAX=VET(J)
          J=J+1
1      CONTINUE
      INDB=FINE+1
      INDV=J
      RETURN
      END
$IFT ABORT EXIT

```

```
$AS LIB TO @SYSTEM(LIBRERIE)GEPIM.LIB BLOC=N
$AS DIR TO @SYSTEM(LIBRERIE)GEPIM.DIR BLOC=N
$CATALOG
BUIL PIPFILE NOM
$EOJ
$$
```

```

$JOB KK SLOF=FFF
$OPTION 2 3 4 5
$FORT77
    IMPLICIT INTEGER*2 (A-Z)
    CHARACTER NOME*64,TTY*6,RIT*1
    INTEGER*1 BUF1(1024)
    INTEGER*2 VET(6)
    INTEGER*4 COMO4(2)
    EQUIVALENCE (COMO4(1),BUF1(1))
C
    LFCTTY=1
    CALL OPENTTY(LFCTTY,TTY)
    LFC=777
11    CALL CLEAR
    WRITE(1,22)
22    FORMAT(/21X,'---> VISUALIZZAZIONE IMMAGINI <---',//,
-5X,'INGRESSO : File disco formato PIP-1024 (max 512x512)',//,
-5X,'USCITA : Memoria di quadro SVP-2000',//,
-5X,'(Rev. 1.0 - Gennaio 1989)',//,
-5X,'(AUTORE : Fantini E. - I.E.I. C.N.R. Pisa)',////,
-17X,'##### SELEZIONE DEL FILE DI INGRESSO #####')
    WRITE(1,1)
1    FORMAT(//' Nome del file da visualizzare (RETURN = FINE) : ')
    READ(1,'(A)') NOME
    IF(NOME.EQ.' ') GOTO 1000
    OPEN(LFC,FILE=NOME,STATUS='OLD',IOSTAT=IST)
    IF(IST.EQ.1) THEN
        CALL RING
        WRITE(1,222)
222    FORMAT(//' ##### ERRORE. Il file specificato non esiste',
- ' #####',/, ' ---> Premere RETURN per continuare <---')
        READ(1,'(A)') RIT
        GOTO 11
    END IF
    IF(IST.NE.0) THEN
        WRITE(1,333) IST
        CALL RING
333    FORMAT(//' ##### ERRORE nella apertura del file : ',I4,
- ' #####',/, ' ---> Premere RETURN per continuare <---')
        READ(1,'(A)') RIT
        GOTO 11
    END IF
    CALL BREAD(LFC,BUF1)
    CALL LUNREC(BUF1,LUN,NREC)
    WRITE(1,2) LUN,NREC
2    FORMAT(//' Lunghezza record : ',I4,/,
- ' Numero di record : ',I4)
    IF(LUN.LT.1.OR.LUN.GT.512.OR.NREC.LT.1.OR.NREC.GT.512) THEN
        CALL RING
        WRITE(1,334)
334    FORMAT(//' ##### ERRORE nelle dimensioni del file',
- ' #####',/, ' ---> Premere RETURN per continuare <---')
        READ(1,'(A)') RIT
        GOTO 11
    END IF
    WRITE(1,63)
63    FORMAT(//,
-16X,'##### PARAMETRI MEMORIA QUADRO DI USCITA #####')
    CALL ASINT
    CALL MEBA(M,JB,999)
33    WRITE(1,3)
3    FORMAT(//' Coordinate X,Y iniziali su video (1,512) [1,1] :')
    READ(1,4) INX,INY
4    FORMAT(2I4)
    IF(INX.EQ.0) INX=1
    IF(INY.EQ.0) INY=1
    IF(INX.LT.1.OR.INX.GT.512.OR.INY.LT.1.OR.INY.GT.512) THEN
        WRITE(1,34)
34    FORMAT(//' ##### ERRORE nella scelta delle coordinate #####',
- ' /' -----> R I P E T I <-----')
        CALL RING
        GOTO 33

```

```

END IF
IFX=LUN+INX-1
IFY=NREC+INY-1
IF (IFX.GT.512.OR.IFY.GT.512) THEN
35  WRITE(1,35)
    FORMAT(/' ## ERRORE area video < dimensioni immagine ##',
-      /'          ----->  R I P E T I  <-----')
      CALL RING
      GOTO 33
END IF
VET(1)=O'11406'
VET(2)=INY-1
VET(3)=IFY-1
VET(4)=INX-1
VET(5)=IFX-1
VET(6)=M-1
CALL TOUT(VET,6,IERR)
CALL SCAMBIA(BUF1,5,768)
CALL TOUT(COMO4(2),382,IERR)
NBL=(LUN*NREC+4)/768
RIM=(LUN*NREC+4)-(NBL*768)
DO 555 K=1,NBL-1
    CALL BREAD(LFC,BUF1)
    CALL SCAMBIA(BUF1,1,768)
    CALL TOUT(BUF1,384,IERR)
555  CONTINUE
    IF(RIM.GT.0) THEN
        CALL BREAD(LFC,BUF1)
        CALL SCAMBIA(BUF1,1,768)
        NN=RIM/2
        IF(MOD(RIM,2).NE.0) NN=NN+1
        CALL TOUT(BUF1,NN,IERR)
    END IF
999  CLOSE(LFC)
78   WRITE(1,79)
79   FORMAT(/// Altro file (s/n) [n] ? : ')
    READ(1,'(A)') RIT
    IF(RIT.EQ.' ') GOTO 1000
    IF(RIT.EQ.'S'.OR.RIT.EQ.'s') GOTO 11
    IF(RIT.NE.'N'.AND.RIT.NE.'n') GOTO 78
1000 CALL EXITTASK
    END
C
C
C
SUBROUTINE LUNREC(BUF,LUN,NREC)
IMPLICIT INTEGER*2 (A-Z)
INTEGER*2 LN,NR
INTEGER*1 COMO1,LR(4),BUF(1)
EQUIVALENCE (LN,LR(1)),(NR,LR(3))
DO 1 K=1,4
1   LR(K)=BUF(K)
    COMO1=LR(1)
    LR(1)=LR(2)
    LR(2)=COMO1
    LUN=LN
    COMO1=LR(3)
    LR(3)=LR(4)
    LR(4)=COMO1
    NREC=NR
    RETURN
END
C
C
C
SUBROUTINE BREAD(LFC,BUF)
IMPLICIT INTEGER*2 (A-Z)
INTEGER*1 BUF(1)
CALL BUFFERIN(LFC,1,BUF,192,,)
CALL M:WAIT(LFC)
RETURN
END

```

C
C
C

```
SUBROUTINE SCAMBIA (BUF, IND, LUN)
IMPLICIT INTEGER*2 (A-Z)
INTEGER*1 BUF (1), COMO
```

C

```
DO 100 K=IND, LUN, 2
  COMO=BUF (K)
  BUF (K)=BUF (K+1)
  BUF (K+1)=COMO
```

```
100 CONTINUE
RETURN
END
```

\$IFT ABORT EXIT

\$AS LIB TO @SYSTEM (LIBRERIE) GEPIM. LIB BLOC=N

\$AS DIR TO @SYSTEM (LIBRERIE) GEPIM. DIR BLOC=N

\$AS L01 TO @SYSTEM (LIBRERIE) SVP. LIB BLOC=N

\$AS D01 TO @SYSTEM (LIBRERIE) SVP. DIR BLOC=N

\$CATALOG

AS TSC TO DEV=U0

AS TSR TO LFC=TSC

BUIL PIPSVP NOM

\$EOJ

\$\$

```

$JOB KK SLOF=FFF
$OPTION 2 3 4 5
$FORT77
    IMPLICIT INTEGER*2 (A-Z)
    CHARACTER NOME*64,TTY*6,RIT*1
    INTEGER*4 NULLO
    INTEGER*1 BUF(768)
    INTEGER*2 VET(512)
    EQUIVALENCE (NULLO,BUF(1))
C
    LFCTTY=1
    CALL OPENTTY(LFCTTY,TTY)
89    IDCB=10
    LFC=777
C
11    CALL CLEAR
    WRITE(1,22)
22    FORMAT(/23X,'---> REGISTRAZIONE IMMAGINI <---',/,
-5x,'INGRESSO : Memoria di quadro SVP-2000',/,
-5X,'USCITA : File disco formato PIP-1024 (max 512x512)',
-//5X,'(Rev. 1.0 - Gennaio 1989)',/,
-5x,'(AUTORE : Fantini E. - I.E.I. C.N.R. Pisa)',////,
-16X,'##### PARAMETRI MEMORIA QUADRO DI INGRESSO #####'//)
    CALL ASINT
    CALL MEBA(MEM,JB,-1)
33    WRITE(1,3)
3    FORMAT(//' Coordinate X inizio, fine, passo [1,512,1] :')
    READ(1,4) INX,FIX,PASX
4    FORMAT(3I4)
    IF(INX.EQ.0) INX=1
    IF(FIX.EQ.0) FIX=512
    IF(PASX.EQ.0) PASX=1
    LUN=(FIX-INX+1)/PASX
    IF(INX.LT.1.OR.INX.GT.512.OR.INX.GT.FIX.OR.
-LUN.LT.1.OR.LUN.GT.512) THEN
        WRITE(1,34)
34    FORMAT('/' ##### ERRORE nella scelta delle coordinate #####',
-        /'          <-----> R I P E T I <-----')
        CALL RING
        GOTO 33
    END IF
35    WRITE(1,36)
36    FORMAT(//' Coordinate Y inizio, fine, passo [1,512,1] :')
    READ(1,4) INY,FIY,PASY
    IF(INY.EQ.0) INY=1
    IF(FIY.EQ.0) FIY=512
    IF(PASY.EQ.0) PASY=1
    NREC=(FIY-INY+1)/PASY
    IF(INY.LT.1.OR.INY.GT.512.OR.INY.GT.FIY.OR.
-NREC.LT.1.OR.NREC.GT.512) THEN
        WRITE(1,34)
        CALL RING
        GOTO 33
    END IF
    WRITE(1,63)
63    FORMAT(//17X,'##### SELEZIONE DEL FILE DI USCITA #####'//)
    WRITE(1,1)
1    FORMAT(//' Nome del file trasformato (RETURN = FINE) : ')
    READ(1,'(A)') NOME
    IF(NOME.EQ.' ') GOTO 1000
    OPEN(LFC,FILE=NOME,STATUS='NEW',IOSTAT=IST)
    IF(IST.EQ.2) THEN
        CALL RING
        WRITE(1,222)
222    FORMAT('/' ##### ERRORE. Il file specificato esiste',
-        '#####',/, ' ---> Premere RETURN per continuare <---')
        READ(1,'(A)') RIT
        GOTO 11
    END IF
    IF(IST.NE.0) THEN
        WRITE(1,333) IST
        CALL RING

```

```

333  FORMAT(/' ##### ERRORE nella apertura del file : ',I4,
- ' #####',/, ' ---> Premere RETURN per continuare <---')
      READ(1,'(A)') RIT
      GOTO 11
      END IF
      CALL SWAPBYTE(NREC,LUN,NULLO)
      WRITE(1,2) LUN,NREC
2    FORMAT(/' Lunghezza record : ',I4,/,
- ' Numero di record : ',I4)
      INDB=5
      DO 100 K=INY,FIY,PASY
          CALL LRIG(MEM,K,VET)
          DO 200 J=INX,FIX,PASX
              IF(INDB.GT.768) THEN
                  CALL BUFFEROUT(LFC,1,BUF,192,,)
                  CALL M:WAIT(LFC)
                  INDB=1
              END IF
              BUF(INDB)=VET(J)
              INDB=INDB+1
200     CONTINUE
100     CONTINUE
          INDB=INDB-1
          INUM=(INDB+3)/4
          IF(INUM.GT.0) THEN
              CALL BUFFEROUT(LFC,1,BUF,INUM,,)
              CALL M:WAIT(LFC)
          END IF
          CLOSE(LFC)
78     WRITE(1,90)
90     FORMAT(//' Altro file (s/n) [n] ? : ')
          READ(1,'(A)') RIT
          IF(RIT.EQ.' ') GOTO 1000
          IF(RIT.EQ.'S'.OR.RIT.EQ.'s') GOTO 89
          IF(RIT.NE.'N'.AND.RIT.NE.'n') GOTO 78
1000    CALL EXITTASK
          END
C
C
C
      SUBROUTINE SWAPBYTE(NREC,LUN,PASS)
      IMPLICIT INTEGER*2 (A-Z)
      INTEGER*4 PASS
      INTEGER*4 NULLO
      INTEGER*2 LN,NR
      INTEGER*1 COMO1,LR(4)
      EQUIVALENCE (LN,LR(1)),(NR,LR(3))
      EQUIVALENCE (NULLO,LR(1))
      LN=LUN
      NR=NREC
      COMO1=LR(1)
      LR(1)=LR(2)
      LR(2)=COMO1
      COMO1=LR(3)
      LR(3)=LR(4)
      LR(4)=COMO1
      PASS=NULLO
      RETURN
      END
$IFT ABORT EXIT
$AS LIB TO @SYSTEM(LIBRERIE)GEPIM.LIB BLOC=N
$AS DIR TO @SYSTEM(LIBRERIE)GEPIM.DIR BLOC=N
$AS L01 TO @SYSTEM(LIBRERIE)SVP.LIB BLOC=N
$AS D01 TO @SYSTEM(LIBRERIE)SVP.DIR BLOC=N
$CATALOG
AS TSC TO DEV=U0
AS TSR TO LFC=TSC
BUIL SVPIP NOM
$EOJ
$$

```