



**MANUALE CP/M**

*Rapporto Interno C88-13*

Enrico Fisoni  
Stefano Groppioni

*30 Marzo 1988*

MANUALE CP/M

# IL SISTEMA OPERATIVO CP/M

## Introduzione:

E' fondamentale avere una copia di ciascun dischetto come backup onde prevenire una perdita accidentale dei dati ivi contenuti.

## Il sistema SDX CP/M con un dischetto

L' implementazione di questo CP/M riconosce il primo dischetto con indirizzo B; i successivi drives hanno indirizzo C, D etc.

Nel manuale vi sono dei punti in cui viene specificato l' indirizzo C del drive; in certe situazioni tali riferimenti possono essere ignorati, ma in altre situazioni il CP/M assume che vi siano per lo meno due dischetti connessi al sistema.

Qualora si abbia la configurazione con un solo drive i programmi di utility SYSCOPY e FORMAT non possono essere usati correttamente ed inoltre la procedura PIP non puo' essere usata per copiare files da un dischetto ad un altro. Sono invece permessi gli altri usi della PIP.

Per aggirare tali ostacoli questo sistema ha delle versioni modificate di tali utilities idonee a lavorare su di un solo drive. Le istruzioni relative all' uso di tali utilities sono descritte nella sezione "SDX CP/M Single Drive Utilities Programs".

La configurazione minima per girare questa versione del sistema CP/M deve avere per lo meno 64Kbytes di RAM ed il relativo programma Monitor.

Dopo aver preso confidenza con il sistema CP/M saremo in grado di scrivervi dei programmi di utility.

## L' Hardware

Tale sistema consta di tre unita':

- Un dischetto da 5.25" ed una unita' di alimentazione connessa alla piastra disc-Controller.
- La scheda "disc-Controller" che viene connessa alla sinistra della tastiera.
- Una piastra di gestione di un display ad 80 colonne con inoltre una porta seriale RS232.

## SDX CP/M Single Drive Utilities Programs

### Disco sistema

Il disco sistema deve contenere per lo meno i seguenti programmi di utility:

- STAT.COM Per il controllo dello spazio libero e l' impostazione dell' I/O.
- FRMSDX.COM Per il formattamento di dischetti vergini.
- SYS.COM Per la copia del sistema operativo all' interno dei dischetti formattati.
- COPY.COM Per copiare files da un dischetto ad un altro.
- PIP.COM Per trasferire i files da/a i devices di I/O.
- ENTER.COM
- SUB.COM Per l' esecuzione di procedure in ambiente batch.
- FDXB.COM Dischetto BASIC.

Il sistema basic FDXB.COM richiede come minimo la presenza dei seguenti files sul dischetto di sistema. Possono essere inoltre usati altri files quali STAT.COM.

- FDXB.COM E' l' interprete del BASIC.
- MTX.COM Quando eseguito passa il controllo al basic presente sulla ROM (basic a 40 colonne).
- NCPM.COM Richiesto per il sistema NODE e per una partenza a caldo con il comando DISC QUIT.

### NewWord Working Disc

E' possibile creare un dischetto NewWord da un dischetto formattato (usando la procedura FRMSDX) con un sistema operativo (copiato usando la procedura SYS) ed i seguenti files (copiati usando la procedura COPY).

- NM.COM Gestisce terminali monocromatici.
- N.COM Gestisce terminali a colori.
- NWMSG.S.OVR Contiene i messaggi del sistema.
- NW.OVR Contiene la procedura eseguibile.
- NWPRINT.OVR Gestisce la stampante.

## Dischetto SUPERCALC

Invece del sistema NewWord si puo' usare il sistema SUPERCALC. I files necessari per tale sistema sono i seguenti:

SC.COM    SC.OVR    SC.HLP

Nel seguito sono riportati i programmi di utility per la gestione di un sistema monodischetto. Essi rimpiazzano i programmi FORMAT, SYSCOPY e PIP che richiedono piu' di un dischetto per la loro operativita'.

## FRMSDX.COM rimpiazza il comando di CP/M FORMAT.COM

Questo programma viene usato per formattare un dischetto. Esso puo' essere richiamato scrivendo:

FRMSDX B:

Sullo schermo apparira' il seguente messaggio:

```
SDX Single disc FORMAT
06.06.85
Insert disc to be formatted
Type CTRL-C to abort or any other key to continue.
```

Se viene premuto un tasto diverso da CTRL-C il computer evidenziera' il seguente messaggio:

```
WAIT .. FORMATTING
```

Una volta che sara' terminato il formattamento del dischetto il computer rispondera' con il seguente messaggio:

```
FINISHED
Insert system disc and press any key
```

A questo punto il controllo tornera' al CP/M.

Nota: Il vostro disco sistema contiene anche il file FORMAT.COM che potra' essere usato una volta che abbiate espanso il vostro sistema con un altro dischetto.

## SYS.COM rimpiazza il comando CP/M SYSCOPY.COM

Questo programma viene usato per copiare le tracce del sistema CP/M da un dischetto sorgente (o dischetto sistema) ad un dischetto destinatario. Il programma viene richiamato lanciando il comando SYS. Il computer scrivera' i seguenti messaggi:

```
Insert source disc and press any key
Insert destination disc and press any key
Press RET to repeat or any other key to quit
```

## COPY.COM rimpiazza il comando PIP.COM nella copia dei files

Questo programma viene usato per copiare un file da un dischetto ad un' altro. Il programma viene richiamato lanciando il comando:

```
COPY DEST.EXT SOURCE.EXT
```

Dove i campi DEST.EXT e SOURCE.EXT sono i filenames CP/M. Il programma copia il file SOURCE.EXT dal dischetto sorgente al nuovo file DEST.EXT nel dischetto destinatario.

Il computer evidenziera' i seguenti messaggi:

Insert source disc and press any key  
Insert destination disc and press any key  
Finished  
Insert system disc and press any key

Se il file e' piu' grande di 16Kbytes, il programma vuole che il dischetto sorgente venga reinserito una volta che siano stati trasferiti i primi 16K sul dischetto destinatario e tale procedura andra' ripetuta finche' non sia stato copiato l' intero file.

### Come si fa' una copia di backup del proprio disco sistema

Il dischetto puo' essere protetto per cio' che concerne la scrittura tappando la finestrella appositamente predisposta.

Per avere una copia del dischetto sistema e' necessario eseguire i seguenti passi:

- Formattare un nuovo dischetto facendo uso del comando FRMSDX.COM
- Ricopiare le tracce del sistema CP/M all' interno del nuovo dischetto facendo uso del comando SYS.COM
- Copiare tutti i files dal proprio disco sistema (source) al nuovo dischetto (destination) usando il comando COPY.COM

Il nuovo dischetto diviene ora il disco sistema.

### Che cosa e' il CP/M?

Il CP/M (Control Program for Microcomputers) e' un sistema operativo che si occupa dell' accettazione di comandi dalla tastiera, della scrittura di informazioni sul video e della gestione di periferiche quali ad esempio stampanti.

Il CP/M permette inoltre la gestione dei dischetti magnetici (Disc Operating System o DOS) e dei dischi fissi (Hard-disk).

Tale sistema operativo rende inoltre compatibili i vari computers che fanno uso dello stesso sistema operativo. Programmi scritti per altri sistemi possono cosi' girare anche nel nostro sistema a patto che tali sistemi avessero come sistema operativo il CP/M.

Il CP/M e' contenuto totalmente su dischetto e permette la stesura di procedure scritte dall' utente ed il loro impiego come se fossero comandi del sistema operativo.

Inoltre il CP/M permette di richiamare alcune procedure di utility che risiedono sul dischetto stesso.

### Procedure di partenza

Se il vostro sistema e' stato assemblato correttamente, all' atto dell' accensione apparira' la seguente scritta:

```
MTX BOOTSTRAP PROM #@SDX01-V03  
RAM TEST OK  
BOOT B03
```

Se cio' non accade ripetere la procedura di POWER.

Se cio' ancora non avviene inserite il vostro dischetto sistema nel drive B.  
A questo punto sullo schermo apparira' la scritta:

```
MTX 54K MEMOTECH BIOS: 03 APR 84
A> CONFIG B: 03,C: 03
DISC CONFIGURATION STATUS
SYSTEM BOOTED FROM ... B:
DRIVE A: MAPPED TO ... B:
FREE SPACE POINTER ... D588
TOP AVAILABLE RAM ... F000
DRIVE B: IS TYPE 03-FL5 D/D/D/S 48T
DRIVE C: IS TYPE 03-FL5 D/D/D/S 48T
```

A questo punto il CP/M e' gia' stato caricato all' interno del vostro computer.  
Il carattere di prompt e':

A >

Tale carattere indica che il computer e' pronto per accettare le nostre istruzioni.

La stringa "A >" puo' essere sostituita con "B >" o "C >" se sono stati acceduti tali drives. Cio' non implica che voi dobbiate avere tre drive dischette.

Il sistema-dischi ha due drives fisici che vengono chiamati B e C. Il drive A e' un drive logico che puo' essere assegnato al drive B o C oppure ad un qualsiasi altro drive esterno.

Ulteriori spiegazioni sulla configurazione dei dischetti possono essere trovate nella sezione "MTX special CP/M utilities".

L' inserimento del dischetto di sistema nel drive B provoca automaticamente il caricamento del nucleo del CP/M in memoria. I programmi di utilita' vengono caricati solo su richiesta esplicita dell' utente.

Il campo "FREE SPACE POINTER" specifica l' indirizzo piu' alto che e' stato riservato al CP/M. Il campo "TOP OF AVAILABLE RAM" indica lo spazio di alta memoria che e' stato riservato per girare alcune routines essenziali del CP/M che verranno caricate all' occorrenza.



## Regole fondamentali sull' uso dei floppy-discs:

- 1- Il floppy disc e' un supporto conveniente e rimane attendibile a lungo se esso viene trattato con cura.  
Le regole fondamentali per un suo corretto uso sono le seguenti:
  - Non toccate la superficie magnetica
  - Riponeteli sempre nelle loro custodie dopo l' uso
  - Non piegateli
  - Non scrivete sull' etichetta con un lapis od una penna biro ma usate unicamente pennarelli.
  - Non spingete il drive con il dischetto ancora inserito.
- 2- Inserimento: Inserite il dischetto con la faccia girata all' insu' ed il foro di protezione scrittura sulla sinistra.
- 3- Finestrella Write-Protect: Se la finestrella e' lasciata libera il dischetto puo' essere sia letto che scritto, se invece viene tappata il disco puo' essere solamente letto (Read Only).
- 4- Come gia' detto, onde evitare la possibilita' di perdita di ore di lavoro, e' indispensabile mantenere sempre una copia di backup del proprio disco di lavoro. Tale copia deve essere aggiornata frequentemente e verra' usata solo nel caso che l' originale venga distrutto.

Il CP/M interpreta ogni riga che gli venga presentata dalla tastiera come comando.

Qualora sia errata la sintassi del comando e' possibile averne la spiegazione dal CP/M stesso facendo seguire alla parola chiave il carattere "?".

Il CP/M esegue automaticamente la conversione dei caratteri da minuscoli in maiuscoli. Il carattere digitato viene subito mostrato sul display. Per correggere un carattere si puo' usare il tasto "back-space" oppure "delete". Se si vuole cancellare l' intera riga di input premere "CTRL-X".

Il tasto "CTRL" in aggiunta ad altri tasti permette di ottenere dal CP/M particolari funzioni.

CTRL-S	Stop/Start Display
CTRL-E	Break line
CTRL-R	Re-type present line
CTRL-U	Remove line
CTRL-P	Printer ON/OFF
CTRL-Z	End of text
CTRL-X	Remove line
CTRL-C	Reset CP/M
CTRL-H	Remove character

Esistono inoltre i due seguenti tasti:

RETURN    Enter line  
RUB/DEL    Remove character

Una volta che e' stata richiesta al CP/M l' esecuzione di un comando digitandolo sulla tastiera e premendo RETURN il CP/M passa alla sua esecuzione.

## Comandi di sistema:

Quando il CP/M e' stato caricato in macchina esso e' pronto per eseguire uno qualsiasi dei 6 seguenti comandi:

COMANDO	RISULTATO
DIR filename (optional)	directory del disco
ERA filename	cancella un file
TYPE filename	stampa il contenuto del file
REN new = old	cambia nome ad un file
USER number	cambia lo "user number"
SAVE b filename	salva un nuovo file

## DIR (directory)

Formato: DIR  
DIR NAME.TYP

Ciascun gruppo di dati o programma ha nel disco un suo nome. Il CP/M puo' richiamarlo direttamente, oppure dall' interno di un altro file, specificando tale nome. Il comando DIR viene usato per avere la lista (directory) dei files presenti sul dischetto. La riga di input sara' pertanto:

A > DIR Seguito da RETURN

Il CP/M ci rispondera' con una riga del tipo:

```
A: FF DAT: LAST LNK: SUBMIT COM: XSUB COM
A: CONTACT DOC: ASM COM: SMG COM: LOAD COM
A: STAT COM: DUMP COM: DUMP ASM: BIOS ASM
```

Tale directory e' riferito al disco "A" e cio' viene evidenziato facendo precedere ciascun rigo della risposta dai caratteri "A:". Le parole che seguono sono i nomi di ciascun file contenuto nel disco. Non e' possibile che due files abbiano la stessa identificazione all' interno dello stesso dischetto.

Ciascun identificatore di file e' composto da un massimo di due parole. La prima parola e' il "filename" e puo' essere lunga fino ad 8 caratteri, la seconda parola e' il "filetype" (opzionale) e puo' essere lunga al massimo 3 caratteri.

Mentre il filename puo' avere una qualsiasi configurazione il filetype indica generalmente il tipo dei dati contenuti all' interno del file. Il filetype "COM" e' il piu' comune ed indica che il file e' un programma che puo' essere girato sul computer.

Qualora si voglia lanciare il comando selettivamente su un file e' sufficiente specificarne il filename:

Es.: A > DIR DUMP.COM

Con tale comando si richiede il directory del file avente filename "DUMP" e filetype "COM". Il punto "." serve per separare i campi "filename" e "filetype".

Il sistema ci rispondera' con il rigo:

A: DUMP COM

se il file esiste, oppure con:

NO FILE

se il file non esiste.

### Selezione di un gruppo di files

E' possibile selezionare, tramite il comando DIR, un gruppo di files aventi le caratteristiche "filename, filetype" che sono specificate nel comando stesso. Per esempio tutti i filenames che contengono un gruppo di lettere specificate oppure che contengono dati numerici.

Cio' puo' essere fatto specificando il carattere "\*" o "?" all' interno del filename. L' asterisco puo' essere usato al posto di un intero "filename" o "filetype". Il punto interrogativo puo' prendere il posto di un qualsiasi altro carattere nel filename.

Per esempio il comando:

```
A> DIR *.VAL
```

specifica che ci interessa il directory di tutti i files aventi filetype VAL indipendentemente dal loro filename.

Il risultato dell' esecuzione di tale comando sara' per esempio:

```
A: JAN VAL: FEB VAL: MAR VAL: APR VAL
A: MAY VAL: JUN VAL: JUL VAL: AUG VAL
A>
```

L' asterisco puo' sostituire anche il filetype. Un esempio di comando con tale sintassi potrebbe essere:

```
A> DIR JAN.*
```

In tale comando si richiede il directory di tutti i files aventi filename "JAN" indipendentemente dal loro filetype. A questo comando il sistema potrebbe rispondere ad esempio nel seguente modo:

```
A: JAN VAL: JAN TXT: JAN MEM
A>
```

Cio' significa che il nostro dischetto contiene solo i files JAN.VAL, JAN.TXT e JAN.MEM che soddisfino tale comando.

Il punto interrogativo specifica invece che la corrispondente posizione del filename non deve essere testata. E' possibile usare tale carattere in qualsiasi posizione del filename e quante volte ci pare.

Es.: A> DIR J?N.VAL

Richiede il directory di tutti i filenames composti di tre lettere di cui la prima deve essere una "J", la terza deve essere una "N", mentre la seconda puo' essere una qualsiasi lettera; il filetype di tali files deve essere VAL.

Un esempio di risposta a tale comando potrebbe essere:

```
A: JAN VAL: JUN VAL
A>
```

E' inoltre possibile usare contemporaneamente i caratteri "\*" e "?".

Es.: A> DIR J?N.\*

Richiede il directory di tutti i files aventi un filename di tre caratteri di cui il primo sia il "J", il terzo l' "N" ed il filetype puo' essere qualsiasi.

Es.: A > DIR J??.\*

Richiede il directory di tutti i files aventi un filename di un massimo di 3 caratteri ed il primo dei quali sia "J"; anche in questo caso il filetype puo' essere qualsiasi.

Il programma CP/M non viene mai presentato nel directory del dischetto dato che esso risiede in una particolare area riservata al sistema operativo.

### ERA - Cancella un file dal dischetto

Formato: ERA FILENAME.TYP

Tale comando serve per cancellare un file dal dischetto al fine di liberare lo spazio da esso occupato per renderlo disponibile ad altri files.

In tale comando e' necessario specificare un filename.

Es.: A > ERA MOTHER.LET

Il CP/M rendera' disponibile l' area occupata da tale file per la scrittura di altri files.

Qualora il file non venga trovato riceveremo in risposta il messaggio:

NO FILE

A >

E' possibile cancellare piu' di un file per volta specificando i caratteri "\*" e "?" nel filename ed il carattere "\*" nel filetype.

Es.: A > ERA JUNK.\*

Richiede al CP/M la cancellazione di tutti i files che si chiamano con filename JUNK, qualsiasi sia il loro filetype.

Es.: A > ERA J??K.TXT

Richiede la cancellazione di tutti i files aventi un filename lungo 4 caratteri di cui il primo e l' ultimo siano rispettivamente "J" e "K" e che abbiano come filetype TXT.

Una volta che un file e' stato cancellato non ne e' piu' possibile il suo recupero. Per questo motivo se viene lanciato il comando:

A > ERA \*.\*

Il CP/M ci chiede conferma sulle nostre intenzioni per evitare errori operativi. In risposta a tale comando il CP/M dara' la seguente riga:

ALL (Y/N)?

Rispondendo "Y" verranno cancellati tutti i files, rispondendo "N" non verra' cancellato nessun file.

E' possibile proteggere dei files da una loro accidentale cancellazione facendo uso del programma di utilita' STAT.

## TYPE

Formato: TYPE FILENAME.TYP

Tale comando permette di evidenziare sul display il contenuto di un file. Per l'uso di tale comando e' necessario specificare il filename completo, non sono cioe' operativi i caratteri "\*" e "?"; non e' inoltre possibile displayare files il cui contenuto sia un programma in linguaggio macchina.

Es.: A > TYPE ABE.LET

Se il file "ABE.LET" esiste nel dischetto "A >" ne verra' evidenziato il suo contenuto.

E' possibile interrompere il display del contenuto del file sulla console premendo i tasti "CTRL-S", premendolo nuovamente verra' ripreso il display del suo contenuto. Premendo un qualsiasi altro tasto sulla tastiera si ha invece la cessazione di tale attivita' di display.

E' inoltre possibile indirizzare il display sulla stampante premendo i tasti "CTRL-P" prima del RETURN che chiude il comando TYPE. Premendo nuovamente i tasti "CTRL-P" abbiamo la cessazione della stampa del file sulla printer.

## REN - Cambia il nome ad un file

Formato: REN NEW.TYP=OLD.TYP

Il formato del comando RENAME e' il seguente:

A > REN NEW.TXT=OLD.LET

Il file avente identificazione uguale a quella specificata dopo il segno "=" (OLD.LET) cambia la sua identificazione in quella che viene specificata prima del segno "=" (NEW.TXT). All'interno della specificazione del file non e' possibile usare i seguenti caratteri:

blank < > . , ; : = ? \* ( )

Se il file di cui si vuole modificare il nome non esiste sul dischetto il CP/M ci dara' la seguente risposta:

NO FILE

A >

Se si tenta di dare come nuovo nome un nome di file che esiste gia' sul dischetto il sistema ci rispondera' nel seguente modo:

FILE EXISTS

A >

## USER

Formato: USER number

Il CP/M riconosce vari utenti; il primo utente che esso definisce (valore default dell'utente che lavora con il sistema) e' l'utente numero 0.

Un altro utente puo' lavorare sempre nelle stesse aree di lavoro di tale utente, oppure puo' decidere di definirsi delle proprie aree di lavoro all'interno dello stesso dischetto. Per fare cio' esso deve cambiare il numero dell'utente.

Se tale utente lancia ad esempio il comando:

```
A> USER 1
```

Esso definirebbe all'interno del dischetto di lavoro una sua area di lavoro riconosciuta come area di lavoro dell'utente 1 e non avrebbe alcuna possibilita' di interagire con i files dell'utente 0.

Lo stesso discorso puo' essere espanso fino ad un massimo di 16 utenti.

Tutti gli utenti hanno a disposizione le facilities del CP/M, ma ognuno di loro ha delle sue aree di lavoro sul dischetto senza poter agire sui files che sono stati creati da altri utenti.

Una volta che si faccia un RESET del sistema riprendera' il controllo del sistema l'utente 0.

## SAVE

Formato: SAVE b FILENAME.TYP

Normalmente i files vengono creati in ambienti diversi dal CP/M (ambiente editor od altri ambienti programma). E' pero' disponibile un comando CP/M che permette la creazione di files.

Il comando viene introdotto con una dimensione del file (b) ed una identificazione per il file che verra' creato. Un esempio di impiego di tale comando e' il seguente:

```
A> SAVE 4 HELP.ME
```

In seguito a tale comando vengono salvati 4 blocchi di dati, attualmente presenti in memoria, su di un file che si chiama HELP.ME. Il CP/M impiega 4 blocchi per ogni kbyte che deve essere salvato. Cosi', ad esempio, per salvare un file residente in memoria e che occupa 20 kbytes saranno necessari 80 blocchi.

```
A> SAVE 80 FILENAME.TYP
```

Per cio' che concerne l'identificazione del file valgono le stesse limitazioni definite nel comando RENAME.

E' indispensabile verificare che sul dischetto non vi sia gia' un file avente quella identificazione altrimenti tale file verrebbe distrutto. Se si verifica tale eventualita' il CP/M inviera' il messaggio:

IT WILL REPLACE THE EXISTING FILE WITH THE NEW ONE JUST SAVED

E quindi procedera' con il salvataggio del file.

E' possibile salvare un massimo di 255 blocchi. Vi e' inoltre un limite imposto dallo spazio rimasto libero sul dischetto e dal numero di identificazioni di files che possono essere tenute nel directory del dischetto.

Qualora l'operazione di salvataggio richieda piu' spazio disco di quello disponibile oppure il directory non abbia piu' spazio per altre identificazioni di files, il sistema inviera' a console il seguente messaggio:

NO SPACE

```
A>
```

E verra' salvata solo una parte del file. A questo punto e' possibile premere i tasti "CTRL-C" e cambiare il dischetto con uno vuoto per poi ritentare il salvataggio. Se viene specificato un numero di blocchi mag-

giore di 255 l'operazione di salvataggio non viene nemmeno iniziata ed il CP/M risponde con il messaggio "nnn?", dove "nnn" specifica il numero dei blocchi che erano specificati nel comando stesso.

### Come si richiama l'esecuzione di un programma

Ogni input da tastiera viene prima controllato come comando di sistema. Se il CP/M riconosce in tale input uno dei suoi comandi di sistema lo esegue. Se invece non viene riconosciuto come comando di sistema allora viene assunto come nome di programma e viene ricercato all'interno del dischetto un programma con tale filename.

Mentre non c'è nessun vincolo sul filename, il filetype deve essere "COM" che sta per "COMMAND". Così, ad esempio, in seguito all'aver ricevuto da tastiera la seguente stringa di input:

A > OTHELLO

Il CP/M, non riconoscendo nella stringa OTHELLO nessun comando di sistema, andrà a cercare sul dischetto un file avente identificazione "OTHELLO.COM". Se tale file viene trovato si passerà alla sua esecuzione, se tale file non viene trovato avremo in risposta la riga:

OTHELLO ?

E' possibile che il programma che si richiama richieda una identificazione di file come parametro. In tal caso il parametro seguirà il nome del programma separato da uno spazio. Così ad esempio il programma di editing "EDITOR" richiede l'identificazione del file che si intende editare. Il formato della riga di input sarà pertanto il seguente:

A > EDITOR BUDGET.TXT

Con tale rigo si richiede l'esecuzione del programma "EDITOR" e l'applicazione di tale programma al file che si identifica come "BUDGET.TEXT".

Nel caso che il programma risieda su di un altro dischetto e' ugualmente possibile andarlo a prendere su tale dischetto. Per la procedura da seguire in questo caso vedere il capitolo "Come si cambia dischetto e drive".

### SAVE - Per il salvataggio dei dati

Formato: SAVE b FILENAME.TYP

Normalmente i files vengono salvati su dischetto da altri programmi che vengono richiamati dal CP/M quando richiesto dall'utente. E' però possibile salvare dei files direttamente da ambiente CP/M. Cio' viene fatto lanciando il comando di CP/M "SAVE" che ha come parametri la lunghezza del file da salvare e l'identificazione del file.

Per ulteriori dettagli su tale comando fare riferimento alla sezione "Comandi di sistema" in cui tale comando e' stato già presentato.

## Come si cambia dischetto o drive

Ogni volta che si accende il computer il CP/M viene caricato dal drive "A" e viene inoltre caricata in memoria una mappa di tale dischetto.

Per mappa si intende una tabella in cui il CP/M ha ricopiato le informazioni presenti nel directory del dischetto.

Qualora si renda necessario cambiare il dischetto sul drive "A" sara' pertanto necessario istruire il CP/M affinche' esso si carichi la mappa del nuovo dischetto.

Cio' viene fatto premendo i tasti "CTRL-C".

A > CTRL-C

Tale operazione viene chiamata "warm boot" (partenza a caldo).

Dato che in seguito a tale comando viene ricaricato anche il sistema operativo e' indispensabile che anche tale dischetto contenga una copia del CP/M (vedi comando SYSCOPY).

Se non viene usata tale procedura il CP/M andra' a cercare ugualmente all' interno del dischetto i files che vengono richiesti, ma nel caso che si intenda scrivere su tale dischetto si puo' avere la distruzione del contenuto di alcuni files. In tal caso il CP/M ci informera' inviando a console il messaggio:

GOOD FILES CAN BE ERASED OR HAVE THEIR CONTENTS CHANGED

Per questo motivo, a meno che non sia dichiarato diversamente dal programma che stiamo usando (ad esempio i programmi di copia), ogni qualvolta viene cambiato il dischetto sul drive e' necessario premere i tasti "CTRL-C".

Tale procedura non e' necessaria se si hanno a disposizione piu' drives. In tal caso la procedura da seguire e' invece la seguente.

## Come si cambia drive

Ogni sistema puo' avere a disposizione anche piu' di un drive. Ad ognuno di essi e' associata una lettera. Il drive dal quale viene fatta la partenza iniziale e' il "B" (drive logico "A"). Questo e' il drive che e' sempre attivo all' atto della partenza del sistema. Il rimanente drive e' chiamato "C".

E' possibile andare a prendere dati dal dischetto caricato sul drive "C". Cio' puo' essere fatto in due diversi modi.

Il primo di tali modi e' quello di rendere attivo il drive "C". Cio' puo' essere fatto lanciando il comando:

A > C:

I due punti che seguono il "C" indicano al CP/M che si tratta di un indirizzo di drive. Se essi venissero omessi il CP/M andrebbe alla ricerca di un programma di nome "C" sul dischetto "A".

Al precedente comando il CP/M rispondera' nel seguente modo:

C >

A questo punto il CP/M vedra' come disco sistema il disco "C" e non piu' l' "A". E' possibile selezionare con tale metodo un qualsivoglia drive.

Un altro modo per indirizzare la ricerca del CP/M su un disco che non e' quello attualmente attivo e' quello di fare precedere il nome del programma dall' indirizzo del disco.

Così, ad esempio, la riga:



A > C: EDITOR

Ordina al CP/M di andare a cercare un programma di nome EDITOR all' interno del suo disco "C". In questo caso il disco attivo rimane l' "A" per cui tutte le operazioni di I/O eseguite dal programma richiamato verranno fatte su tale drive. E' possibile usare l' indirizzo del drive interessato dal comando anche dopo un comando residente del CP/M.

Es.: A > DIR C:

Richiede il directory del dischetto presente sul drive "C".

A > DIR C: \*.TXT

Richiede il directory di tutti i files presenti sul disco "C" ed aventi filetype "TXT".

A > TYPE C: LETTER.TXT

Richiede la stampa del contenuto del file "LETTER.TXT" presente sul drive C (il disco attivo e' l' "A").

C > ERA B: \*.LET

Richiede la cancellazione dei files presenti sul disco "B" e che hanno filetype "LET" (il disco attivo e' il "C").

C > REN B: NEW.TYP=OLD.TYP

Va' a cercare sul disco "B" il file avente identificazione "OLD.TYP" e se ve lo trova cambia la sua identificazione in "NEW.TYP" (il disco attivo e' il "C").

C > SAVE 10 B: PROGRAM

Salva 10 blocchi della memoria centrale su di un file di nome PROGRAM che verra' creato sul dischetto presente sul drive "B" (il disco attivo e' il "C").

### MTX special CP/M utilities

I programmi di utilita' sono programmi che vengono forniti assieme al sistema al fine di aumentarne l' efficienza. Una breve descrizione di tali programmi e' riportata qui sotto; successivamente essi verranno ripresi in esame e descritti uno per uno.

FORMAT Serve per formattare un dischetto e renderlo adatto ad essere usato nel nostro sistema.

SYSCOPY Serve per copiare il CP/M su di un altro dischetto.

PIP Viene usato per copiare files da un dischetto ad un' altro. Esso permette anche di scambiare dati fra le varie periferiche attaccate al sistema.

STAT Permette di avere lo stato dei dischetti presenti nei drives e delle altre periferiche. Puo' fornire la dimensione dei files oppure dare lo spazio rimasto libero in un dischetto. Esso e' inoltre in grado di modificare lo stato dei files e delle periferiche nel sistema.

MOVCPM Permette di spostare il CP/M da un' area di memoria ad un' altra.

WRTCPM Scrive la sezione CCP e BDOS del CP/M 2.2 nell' area di bootstrap del dischetto specificato.

WRTBIOS Scrive la sezione BIOS e "COLD-START loader" del CP/M 2.2 nell' area di bootstrap del dischetto specificato.

Nel seguito sono riportati alcuni esempi di come possono essere usati tali programmi di utility al fine di svolgere tre particolari funzioni.

### Come si fa' una copia di backup del disco CP/M

- 1- Si usi il programma FORMAT per preparare il dischetto ad accettare dati.
- 2- Si usi il comando PIP \*.\* per copiare l' intero dischetto CP/M originario sul nuovo dischetto.

### Come si copia un nuovo disco programma

- 1- Usare il programma FORMAT per preparare il nuovo dischetto a ricevere dati.
- 2- Usare il comando PIP \*.\* per copiare l' intero disco programma sul dischetto vuoto.
- 3- Usare il programma SYSCOPY per copiare il CP/M all' interno del nuovo dischetto cosicche' esso possa poi essere usato sul drive "A" per inizializzare il sistema.

### Come si crea un disco per uso generale

- 1- Formattare il dischetto tramite la procedura FORMAT.
- 2- Usare la procedura SYSCOPY al fine di copiare il CP/M nel nuovo dischetto cosicche' esso possa poi essere usato come disco sistema.
- 3- Usare la procedura PIP per copiare i files che ci interessano dal dischetto originario al nuovo dischetto.

## FORMAT

Un dischetto, cosi' come esce di fabbrica, non e' in grado di ricevere dati; per renderlo atto a ricevere dati e' necessario formattarlo con questo programma. Il formattamento consiste in un riempimento del dischetto con dei caratteri che sono riconosciuti dal CP/M come carattere "vuoto" (configurazione esadecimale E5).

Affinche' il dischetto possa essere formattato e' necessario che esso non sia protetto per cio' che concerne la scrittura. La protezione sull' operazione di scrittura la si ottiene coprendo la finestrella a sinistra del dischetto. Il carattere di riempimento viene a costituire dei piccoli gruppi (settori) che sono concentrici e che vanno a costituire le tracce.

Il formato del comando e' il seguente:

FORMAT d:

Dove "d:" e' l' indirizzo del disco attuale ("B" o "C").

## SYSCOPY

Tutti i dischetti che poi verranno usati nel nostro sistema debbono contenere una copia del CP/M. Per copiare tale sistema sui vari dischetti che si intende utilizzare abbiamo a disposizione la procedura di utility SYSCOPY.

Tale copia del sistema viene messa nelle tracce sistema (che sono le prime due tracce create dal programma FORMAT). Tale area non puo' essere usata per quelli che sono i files di utente.

La procedura viene richiamata lanciando il seguente comando:

SYSCOPY dd: < = ds: >

Tale comando produce come effetto la copia dell' area di bootstrap dal disco specificato nel campo "ds" del comando, oppure dal disco attualmente allocato se tale campo non viene specificato, all' area di bootstrap del disco destinatario, specificato nel campo "dd" del comando. Qualora i due indirizzi "dd e ds" coincidano viene generata una pausa per dar modo all' operatore di sostituire il dischetto sorgente con quello destinatario. La procedura riprendera' la sua esecuzione solo dopo che e' stato premuto RETURN.

## PIP - Che cosa e'?

E' questo uno dei piu' potenti programmi di utilita'. Tale sigla significa "Peripheral Interchange Program". Tale programma permette di svolgere le seguenti funzioni:

- Copia di specifici files, uno per volta oppure diversi per volta. Cio' permette di mettere diversi programmi, aventi differenti sorgenti, all' interno dello stesso dischetto.
- Esso permette inoltre lo scambio di dati fra le periferiche oppure la modifica di dati durante le procedure di copia.

Tale programma puo' essere usato in due diversi modi. Il primo e' quello di digitare il suo filename seguito dalla funzione che si intende svolgere.

A > PIP funzione

In questo modo il PIP svolgera' la funzione richiesta e quindi ripassera' il controllo al CP/M.

Tale forma del comando permette di svolgere una singola funzione per volta. Se si intende far svolgere piu' funzioni a tale utility e' senz' altro conveniente mantenere la copia del programma in memoria. Cio' viene fatto lanciando il comando:

A > PIP

A tal punto il programma ci rispondera' con il suo prompt "\*".

A >

A > PIP

\*

A questo punto siamo in ambiente PIP e possiamo sottomettere al programma i nostri comandi. Una volta terminata l' esecuzione del comando il programma ci inviera' nuovamente il suo segnale di prompt "\*".

Per uscire da tale ambiente programma e' sufficiente inviare come comando una riga vuota. Il controllo tornera' quindi al CP/M.

## Come si usa il PIP per copiare files

La funzione principale di tale programma e' quella di eseguire copie di files. E' possibile creare files di backup degli originali sullo stesso disco in cui sono contenuti gli originali oppure su un altro dischetto.

Il file che si intende creare e' sempre specificato per primo, seguito dal segno "=" e quindi dal file che si intende copiare. Il formato base del comando e' pertanto:

A > PIP NEW.TYP=OLD.TYP

Se non e' specificato nessun indirizzo di drive viene preso in considerazione solamente il drive attivo. Tale comando prendera' percio' il file presente sul disco "A" ed avente identificazione "OLD.TYP" e lo ricopiera' in un nuovo file avente identificazione "NEW.TYP". Se il file "NEW.TYP" esiste gia' sul disco "A" il suo contenuto verra' rimpiazzato dal contenuto del file "OLD.TYP".

E' anche possibile copiare un file presente su un dischetto montato su un drive all' interno di un file su di un dischetto montato su di un altro drive. Cio' viene fatto specificando l' indirizzo del drive interessato dall' operazione.

A > PIP C: NEW.TYP = OLD.TYP

In seguito a tale comando viene creata una copia del file "OLD.TYP" presente sul disco attivo (attualmente il disco "A") su di un file di identificazione "NEW.TYP" che verra' creato sul disco "C".

E' possibile creare tale copia prendendo il file sorgente da un qualsiasi disco e portandolo su di un qualsiasi altro disco semplicemente specificando gli indirizzi di tali drives.

A > PIP C: NEW.TYP = B: OLD.TYP

In questo caso il file da copiare viene preso dal disco "B" e viene ricopiato nel disco "C". Il disco attivo rimane l' "A".

Qualora nel comando sia specificato l' indirizzo del drive ma non sia specificata l' identificazione del file ricevente il PIP assume come default lo stesso nome del file originario. Cosi', ad esempio, il comando:

A > PIP C: = OLD.TYP

Significa: prendi il file "OLD.TYP" presente sul disco attivo (attualmente il disco "A") e ricopialo sul disco "C" su di un file avente la stessa identificazione del file sorgente.

Tale programma accetta come identificazione dei files anche i caratteri "\*" e "?" il cui impiego e' gia' stato descritto a proposito del comando DIR. Cio' permette di selezionare, per l' operazione di copia, un gruppo di files al posto del singolo file. Per esempio il comando:

A > PIP C: = \*.TXT

dice al PIP di prelevare dal disco attualmente attivo (disco "A") tutti i files aventi filetype "TXT" e ricopiarli all' interno del disco "C" mantenendo anche la loro identificazione. La procedura evidenziera' sulla console l' identificazione dei vari files man mano che essi vengono copiati.

Per avere una copia di tutti i files presenti su un disco all' interno di un altro disco e' sufficiente specificare il carattere "\*" sia come filename che come filetype. Cosi' il comando:

A > PIP C: = \*.\*

Verra' interpretato nel seguente modo. Prendi tutti i files presenti sul disco attivo (attualmente l' "A") e ricopiali nel dischetto "C" mantenendo la stessa identificazione del file di partenza.

E' possibile eseguire tale copia a partire da un qualsiasi drive verso un qualsiasi altro drive semplicemente specificando l' indirizzo del drive nel comando stesso. Per esempio il comando:

A > PIP B: = C: \*.\*

copiera' tutti i files presenti sul disco "C" all' interno del disco "B" mantenendo anche la loro identificazione. Tale operazione di copia procede in due passi.

- Il primo passo e' quello di copiare i dati dal file sorgente su di un file di servizio avente il filename specificato nel comando e filetype "\$\$\$".
- Solo se la copia termina senza nessun errore viene fatto il rename del file dandogli l' identificazione specificata nel comando stesso.  
In questo modo viene sempre salvaguardata la copia originale del file.

## Come si usa PIP per copie fra periferiche

La procedura PIP puo' essere usata, oltre che per copiare files, anche per lo scambio di dati fra tutte le periferiche connesse al sistema (dischi, stampanti, modems etc.) e fra diversi tasks.

Tramite tale procedura e' possibile trasferire files da dischetto a stampante, da tastiera a dischetto, da dischetto a console, trasmettere dati via modem oppure riceverne dal modem e dirottarli sulla console o sulla stampante. Si ha inoltre la possibilita' di trasferire i dati direttamente dalla console alla stampante.

Per fare cio' e' necessario specificare il tipo di device che si intende usare nel comando stesso. Il PIP riconosce quattro tipi di devices; essi sono LST:, CON:, RDR: e PUN:.

Il significato di tali devices per il PIP e' il seguente:

CON: Console- Si intende usare la tastiera per introdurre dati ed il display per vedere i risultati.

LST: List- Si intende usare la stampante per listare i dati.

RDR: Reader- E' un qualsiasi device ausiliario che legge dati verso il computer. Puo' essere un qualsiasi device attaccato al sistema (per esempio un modem) che legge dati e li trasferisce verso il computer.

PUN: Punch- E' un qualsiasi device ausiliario che e' in grado di ricevere dati dal computer (per esempio un modem).

Tali nomi di periferiche possono essere usati all' interno del comando per trasmettere o ricevere dati. Per esempio il comando:

```
A > PIP LST: = FILENAME.TYP
```

indica al PIP che si intende trasferire il contenuto del file "FILENAME.TYP" verso la stampante.

Il nome del device viene sempre specificato per primo seguito dal segno "=" e dalla sorgente dei dati.

All' interno del comando e' possibile specificare anche un indirizzo di device. Per esempio il comando:

```
A > PIP LST: = C: DOGS.DAT
```

viene interpretato nel seguente modo. Prendi dal disco "C" il file avente identificazione "DOGS.DAT" e trasferiscilo verso la stampante.

In modo similare e' possibile trasferire un file sulla console. Il comando da lanciare in tal caso sara' il seguente:

```
A > PIP CON: = FILENAME.TYP
```

In seguito a tale comando viene preso dal disco attivo (attualmente il disco "A") il file di identificazione "FILENAME.TYP" e trasferito sulla console.

Tale comando produce lo stesso risultato del comando di sistema "TYPE", e' pero' possibile specificare all' interno del "PIP" alcuni comandi di elaborazione del testo, cosa che non e' invece possibile nel comando TYPE.

Il seguente comando:

```
A > PIP FILENAME.TYP = CON:
```

ci permette invece di creare sul disco attivo (attualmente il disco "A") un file avente identificazione "FILENAME.TYP" che verra' riempito con i caratteri che verranno digitati via via sulla tastiera.

Il file viene chiuso premendo i due tasti "CTRL-Z" (ETX). A questo punto saremo di nuovo in ambiente CP/M.

Il comando:

```
A > PIP LST: = CON:
```

crea invece un canale diretto fra la tastiera e la stampante per cui tutti i caratteri digitati sulla tastiera verranno evidenziati immediatamente sulla stampante senza passare attraverso il computer. Per chiudere tale canale e' sufficiente premere i tasti "CTRL-Z".

I devices "RDR:" e "PUN:" fanno riferimento ad una porta ausiliaria. L'impiego che puo' essere fatto di tale porta e' funzione del modello del computer.

Se nel nostro sistema tale porta e' connessa ad un modem, il comando

A > PIP CON: = RDR:

avra' l'effetto di prelevare i dati in arrivo dal modem e di evidenziarli sulla console di sistema.

Nel caso che si vogliano indirizzare i dati in arrivo dalla linea sulla stampante sara' invece necessario lanciare il comando:

A > PIP LST: = RDR:

Se intendiamo invece trasmettere dati dalla tastiera verso la linea di trasmissione il comando da lanciare sara' il seguente:

A > PIP PUN: = CON:

Se intendiamo invece trasmettere sulla linea il contenuto di un file avente identificazione "SCRAM.TXT" e residente sul disco attivo (il disco "A") dovremo lanciare il seguente comando:

A > PIP PUN: = SCRAM.TXT

Ogniqualevolta si stanno scambiando dati fra le periferiche (purche' non si tratti di trasferimenti disco-disco o tastiera-disco) il trasferimento viene interrotto definitivamente non appena si preme un qualsiasi tasto della tastiera.

Il controllo delle operazioni ritorna al PIP ed il comando che era in corso di esecuzione viene abortito.

E' possibile modificare l'associazione fra i devices conosciuti dal PIP ed i devices fisici semplicemente facendo uso del comando STAT.

### Opzioni per la modifica dei files

E' possibile modificare i dati che vengono copiati facendo uso della procedura "PIP" usando una o piu' opzioni speciali.

Ogni opzione e' costituita da una lettera racchiusa fra parentesi e che segue il file sorgente oppure il device sorgente. E' possibile specificare piu' opzioni per volta racchiudendole tutte in una coppia di parentesi. E' possibile, ma non necessario, usare fra le varie lettere delle opzioni degli spazi bianchi.

## Opzioni per la modifica del testo

Le opzioni che ci permettono di modificare il contenuto di un file sono le seguenti:

D	Delete characters
E	Echo characters
F	Form-feeds removed
L	Lowercase only
N	Number lines
P	Page form-feeds
Q	Quit copying
S	Start copying
T	Tab space
U	Uppercase only
Z	Zero parity bit

Vediamo ora come operano tali opzioni.

### Dn - Delete characters

Il file interessato dall' operazione viene troncato alla colonna "n". I caratteri che superano tale colonna vengono persi.

### E - Echo characters

Viene creato un' eco dei caratteri in corso di trasferimento, tali caratteri vengono cioè evidenziati anche sulla console.

### F - Forms-feeds removed

Tale opzione permette di eliminare dal file destinatario tutti i caratteri "form-feed" prima che il file sorgente venga ricopiato in esso. Il carattere "form-feed", quando incontrato all' interno di un testo, provoca un salto pagina.

### L - Lowercase only

Tale opzione permette la conversione di tutti i caratteri maiuscoli del file sorgente in caratteri minuscoli nel file destinatario.

### N - Number lines

#### N2

Nella forma "(N)" tale opzione permette di numerare le linee che vengono via via trasferite a partire da 1 con incrementi unitari. Gli zero di riempimento non vengono usati ed il numero è seguito da un carattere " ".

Nella forma "(N2)" vengono usati gli zero di riempimento ed il numero viene fatto seguire da un carattere di tabulazione.

### P - Page form-feeds added

#### Pn

Nella forma "(P)" l' opzione permette di inserire un carattere form-feed ogni 60 righe di testo. Il carattere form-feed, se riconosciuto dalla stampante, provoca un salto pagina.



Nella forma "(Pn)" il numero "n" specifica il numero di righe che debbono essere contenute in ogni pagina. Dopo che tale numero "n" di linee sono state trasferite il programma introduce un carattere form-feed.

#### Qphrase|Z - Quit copying

Con tale opzione il trasferimento dati si fermerà automaticamente non appena verrà incontrata la stringa "phrase" durante il trasferimento dati.

Cio' permette di trasferire il file dall' inizio fino al punto in cui viene incontrata la stringa "phrase". La stringa "|Z" indica il fatto che sono stati premuti i tasti "CTRL" e "Z".

ES.: A > PIP EDIT.TXT = TOTAL.TXT(QBE?|Z)

Il file avente identificazione "TOTAL.TXT" (contenuto nel disco "A") viene copiato all' interno del file "EDIT.TXT" (sempre sul disco "A"). Tale operazione di copia cesserà non appena verrà incontrato nel file sorgente la stringa "BE?". Dato che il CP/M converte in caratteri maiuscoli quelli che sono i comandi a lui diretti, con tale forma del comando non sarà possibile controllare stringhe contenenti lettere minuscole. Cio' può essere invece fatto usando la seconda forma del comando "PIP".

La sequenza dei comandi sarà in questo caso la seguente:

```
A > PIP
* EDIT.TXT = TOTAL.TXT(Qbe?|Z)
*
A >
```

#### Sphrase|Z - Start copying

Questa opzione permette di fare un trasferimento dati da un file partendo dal punto in cui viene incontrata la stringa specificata come "phrase". Fintanto che non viene incontrata tale stringa il file viene letto ma non viene indirizzato nessun dato verso l' unita' di output. Tale trasferimento dati termina solo con la fine del file oppure se ne può delimitare il punto di fine facendo uso dell' opzione "(Q)".

Così, ad esempio il comando:

A > PIP EDIT.TXT = TOTAL.TXT(STHAT|Z)

avrà il seguente effetto.

Viene letto il file avente identificazione "TOTAL.TXT" e contenuto nel disco "A". Il contenuto di tale file viene scartato fino a che non viene incontrata la stringa "THAT". Una volta che è stata trovata tale stringa viene aperto (sempre sul disco "A") un file di nome "EDIT.TXT" all' interno del quale vengono copiati tutti i dati letti dal file sorgente ivi inclusa la stringa "THAT" usata nel comando.

Tale copia terminerà con la fine del file.

Se la stringa da ricercare contiene caratteri minuscoli è necessario usare la seconda forma del comando PIP.

```
A > PIP
*EDIT.TXT = TOTAL.TXT(Sthat|Z)
*
```

A >

È possibile combinare assieme l' uso delle due opzioni S e Q al fine di avere un estratto del file. Così, ad esempio, il comando:

A > PIP EDIT.TXT = TOTAL.TXT(STHAT|ZQTHE|Z)

Verrà interpretato nel seguente modo.

Copia all' interno del file "EDIT.TXT" il contenuto del file "TOTAL.TXT" a partire dal punto in cui trovi la stringa "THAT" e chiudendo la copia allorché trovi la stringa "THE".

### Tn - Tab space

Tale opzione permette di definire il numero "n" di caratteri blank che debbono essere generati ogniqualvolta viene incontrato un carattere di tabulazione.

Così, ad esempio, il comando

```
A > PIP FINAL = ORIGINAL(T5)
```

produrrà il seguente effetto.

Viene copiato il file "ORIGINAL" all'interno del file "FINAL" espandendo a 5 caratteri blank ogni carattere di tabulazione che venga trovato nel file sorgente.

### U - Uppercase only

Tale opzione converte tutti i caratteri minuscoli contenuti nel file sorgente in caratteri maiuscoli.

Così, ad esempio, il comando:

```
A > PIP CON: = C: LOW(U)
```

Provocherà un trasferimento di dati dal file "LOW" contenuto sul drive "C:" alla console facendo inoltre la conversione dei caratteri minuscoli contenuti all'interno del file in caratteri maiuscoli.

### Z - Zero parity bit

Tale opzione permette di convertire il bit di parità di ciascun carattere da trasferire da 1 a 0.

L'informazione viene rappresentata all'interno del computer con 7 bits dati più un bit di parità.

Tale bit deve essere normalmente a 0 se si fa' uso dei programmi standard del CP/M, e' però possibile che esso venga impostato ad 1 da altri programmi (editors etc.). Tale conversione del bit di parità può essere fatta usando questa opzione. Così, ad esempio, il comando

```
A > PIP STANDARD = UNIQUE(Z)
```

produrrà il seguente risultato. Il file "STANDARD" contiene gli stessi dati del file "UNIQUE" eccezion fatta per il bit di parità che è stato impostato a 0. In tal modo il file "STANDARD" è CP/M compatibile, mentre il file "UNIQUE" non lo è.

## Opzioni per copie generiche

Oltre alle opzioni designate alla copia dei files di tipo "testo" il PIP ne ha altre di uso più generale. Anche queste ultime sono racchiuse fra parentesi.

Questo ultimo tipo di opzioni raggruppa le seguenti:

B	Block mode copy
G	Get from user
R	Read system file
V	Verify copy
W	Write over

Vediamo ora come operano tali opzioni.

## B - Block mode copy

Quando il PIP trasferisce i dati esso opera generalmente a piccoli blocchi. Il device trasmittente deve perciò attendere che sia stato trasferito il primo blocco prima di inviare il secondo, e così via. Ma alcune unità, una volta che sono state avviate, non possono essere fermate (per esempio il registratore). Tale tipo di unità vuole trasmettere tutti i dati dall'inizio alla fine.

Questa opzione informa il PIP che deve trasferire un intero blocco di dati. Tali dati verranno immagazzinati nella memoria centrale. Solo quando è terminata la ricezione del blocco dati il PIP si preoccuperà di trasferirlo al device destinatario.

Così, ad esempio, il comando

```
A > PIP ALL = RDR:(B)
```

verrà interpretato nel seguente modo.

Prendi tutti i dati in arrivo dal reader (RDR:) e immagazzinali in memoria.

Una volta che il file è completo trasferiscilo all'interno del file "ALL" sul dischetto attualmente attivo (il disco "A"). La dimensione di tale file sarà limitata dalla memoria disponibile nel nostro sistema. Se tale memoria non risulta sufficiente il PIP ci informerà con un messaggio ed il trasferimento verrà abortito.

## Gn - Get file from another user area

Questa opzione abilita il PIP a prelevare un file sorgente da un'altra "user-area". Il numero "n" specifica la "user-area" all'interno della quale il PIP deve andare a cercare il file sorgente (a tal proposito si veda il comando "USER" nella sezione "comandi di sistema").

Così, ad esempio, il comando

```
A > PIP A: = NEEDIT(G2)
```

verrà interpretato nel seguente modo.

Preleva il file di nome "NEEDIT" all'interno della "user-area 2" e copialo, mantenendo la stessa identificazione, sul disco "A" nella "user-area" che è attualmente attiva.

Il PIP, per trasferire files ad una "user-area", necessita di risiedere in tale "user-area". Vediamo allora come si può fare per trasferire il PIP su di una nuova "user-area"?

Si pone il PIP in memoria quando siamo nella "user area 0", ci si sposta nella "user-area" che ci interessa e si salva il PIP in tale area facendo uso del comando di CP/M "SAVE". Nel caso che si voglia salvare il PIP nella "user-area 3" la procedura da seguire sarà la seguente:

Quando siamo nella user-area 0 scrivere:

```
A > PIP
```

In seguito a tale comando il PIP viene caricato in memoria e ci fornirà il suo carattere prompt "\*\*".

```
A > PIP  
*
```

Adesso premere RETURN per tornare in ambiente CP/M.

```
A > PIP  
*  
"RETURN"  
A >
```

A questo punto possiamo selezionare la user-area che ci interessa (nell' esempio la 3).

```
A > PIP
*
"RETURN"
A > USER 3
```

Ed il CP/M ci inviera' il suo carattere prompt ("A >").

```
A > PIP
*
"RETURN"
A > USER 3
A >
```

A questo punto procedere al salvataggio del PIP (occupa 8 Kbytes per cui gli necessitano 32 blocchi) tramite il comando di CP/M "SAVE".

```
A > PIP
*
"RETURN"
A > USER 3
A > SAVE 32 PIP.COM
```

Adesso avremo una copia del comando PIP nella "user-area 3" e potremo servircene anche quando siamo in tale "user-area".

### R - Read system file

Un qualsiasi file sul dischetto puo' essere dichiarato, tramite la procedura STAT, come file di sistema. Il file potra' pertanto essere usato ma esso non apparira' quando viene lanciato il comando DIR. Inoltre il file non puo' essere trattato dalla procedura PIP a meno che non sia specificata l' opzione "(R)".

Cosi', ad esempio, il comando

```
A > PIP C: = HIDDEN(R)
```

verra' interpretato nel seguente modo.

Prendi il file di sistema "HIDDEN" e ricopialo nel dischetto presente sul drive "C".

### V - Verify copy

Tale comando informa il PIP che si vuole avere una verifica dei dati che sono stati copiati. Una volta che i dati sono stati trasferiti sul dischetto destinatario il PIP va' a leggerli e li confronta con quelli letti sul file sorgente.

Se tali dati sono eguali il PIP procede nel suo lavoro, se essi non lo sono esso ce ne informera'. Cio' puo' accadere per un cattivo funzionamento del dischetto. Se viene specificata tale opzione il destinatario deve essere per forza un file su disco.

Cosi', ad esempio, il comando

```
A > PIP C: = *.*(V)
```

verrà interpretato nel seguente modo.

Prendi tutti i files dal disco attivo (attualmente l' "A") e ricopiali nel disco "C" eseguendo anche la verifica sul buon andamento della copia. Tale opzione provoca un allungarsi del tempo di elaborazione per ciò che concerne le operazioni di copia, ma in compenso ci garantisce del buon andamento di tali operazioni. Se ne raccomanda pertanto l' uso ogniqualvolta si debba eseguire una operazione di copia.

### W - Write over protected files

Qualsiasi file del dischetto può essere protetto contro una qualsiasi cancellazione accidentale (cio' viene fatto tramite la procedura STAT). Tale protezione rimane valida anche per il PIP. Quando la procedura PIP trova un file protetto essa si arresta e chiede conferma per la modifica del suo contenuto (per fare procedere tale modifica sarà sufficiente rispondere "Y" oppure "N" se non si vuole modificare il file).

Se vogliamo che il PIP non si arresti di fronte a tale evento è sufficiente specificare nel comando tale opzione.

Così, ad esempio, il comando

```
A > PIP C: = SAFE(W)
```

verrà interpretato nel seguente modo.

Copia il file "SAFE" dal disco attivo (attualmente il disco "A") su di un file, avente la stessa identificazione, sul disco "C" anche se su tale disco esiste già un file avente tale identificazione e tale file è protetto in scrittura. Il contenuto del file destinatario verrà perciò rimpiazzato dal contenuto del file sorgente.

Se si intendono copiare più files all' interno di un unico file che è protetto in scrittura, l' opzione "(W)" deve essere specificata solo dopo l' identificazione dell' ultimo file.

La procedura PIP è anche in grado di concatenare diversi files.

## Impieghi speciali della procedura PIP

### Come si concatenano i files

La procedura PIP permette anche di ricopiare più files all' interno di un unico file che sarà pertanto costituito dal concatenamento di tutti i files sorgenti.

Un esempio di comando che svolge tale funzione è il seguente:

```
A > PIP BOOK = CHAPTER1,CHAPTER2,CHAPTER3,CHAPTER4
```

In seguito a tale comando verrà creato un file sul disco "A" di identificazione "BOOK" ed all' interno del quale saranno ricopiati i files "CHAPTER1", "CHAPTER2", "CHAPTER3" e "CHAPTER4".

È possibile specificare all' interno del comando che i files non debbono essere cercati all' interno del disco attivo ma all' interno di altri dischi. Ciò viene fatto specificando, nell' identificazione del singolo file, anche l' indirizzo del disco all' interno del quale cercare il file.

Es.:

```
A > PIP C: BOOK = B: CHAPTER1,B: CHAPTER2,C: CHAPTER3,C: CHAPTER4
```

Tale comando verrà interpretato nel seguente modo.

Prendi i files "CHAPTER1" e "CHAPTER2" dal disco "B", i files "CHAPTER3" e "CHAPTER4" dal disco "C" concatenali nell' ordine e crea un unico file, avente identificazione "BOOK", nel disco "C".

Dato che per l' operazione di copia il PIP crea un file temporaneo (con filetype \$\$\$), e' possibile usare come file sorgente anche il file destinatario.

Es.: A > PIP BOOK = BOOK,CHAPTER5,CHAPTER6

Tale comando verra' interpretato nel seguente modo.

Prendi il contenuto del file di identificazione "BOOK", concatenalo nell' ordine ai contenuti dei files di nome "CHAPTER5" e "CHAPTER6" e con il file risultante crea un nuovo file "BOOK" che va' a rimpiazzare il vecchio.

### Concatenamento di files che non sono di tipo testo

Le operazioni di concatenamento finora prese in esame sono di assai semplice esecuzione da parte del PIP in quanto la fine di tutti i files di tipo "text" e' contrassegnata dal carattere ETX (CTRL-Z).

I problemi cominciano a nascere allorché si rende necessario copiare files di tipo diverso (es. HEX filetype).

I problemi sorgono dato che il carattere ETX, su tali tipi di files, puo' far parte del testo da trasferire. Per informare il PIP di tale problema esiste l' opzione "(O)" che dice al PIP di copiare il file fino alla sua fine senza controllare se viene trovato il carattere ETX all' interno del testo da trasferire. Per esempio, il comando

A > PIP NEW = DATA1(O),DATA2(O),DATA3(O)

verra' interpretato nel seguente modo.

Prendi i files di tipo oggetto (i moduli oggetto possono avere al loro interno tutte le configurazioni esadecimali) "DATA1", "DATA2" e "DATA3" e concatenali copiandoli poi all' interno del file "NEW".

La copia si fermara' solo dopo che e' stato completato "DATA3".

I files di tipo HEX vengono usati (dalla utility LOAD) per creare un file eseguibile (di tipo COM). In questi tipi di files la fine del file non e' piu' contraddistinta dal carattere ETX ma viene invece usata la stringa ":00". Se si concatenano due o piu' files di tipo HEX all' interno del file destinatario verranno copiati anche tali caratteri di fine file. L' utility LOAD, quando applicata al file risultante, si fermara' quando incontrera' la prima di tali stringhe di fine file. I rimanenti files che erano stati concatenati all' interno del file principale non verranno percio' presi in considerazione da tale utility. Esiste un' altra opzione, la "(I)" che sta' per "ignore", che dice al PIP di ignorare i caratteri di fine programma.

Così, ad esempio, il comando:

A > PIP NEW.HEX = FILE1.HEX(I),FILE2.HEX(I),FILE3.HEX

agira' nel seguente modo.

Quando viene creato il file "NEW.HEX" vengono ignorati i caratteri di fine programma dei files "FILE1" e "FILE2" e viene copiata solamente la fine programma del file "FILE3". Inoltre quando viene usata tale opzione il PIP controlla i dati che vengono trasferiti. Se durante il trasferimento vengono trovate codifiche esadecimali non valide verra' lanciato un messaggio di avvertimento nella console.

Se necessitate del controllo da parte del computer dei suoi dati potete usare l' opzione "H" per il controllo dei dati esadecimali.

I dati nel formato esadecimale vengono salvati a gruppi. Per il controllo dell' accuratezza, tutti i numeri, in ogni gruppo di dati, vengono sommati tra loro e debbono dare come risultato un numero uguale all' ultimo del gruppo. Questo tipo di controllo viene chiamato "Intel Checksum Format".

Se si e' avuto un errore di copiatura la suddetta condizione non sara' verificata ed il PIP ci segnalera' tale errore.

Così, per esempio, il comando

```
A > PIP FILE.HEX = RDR:(H)
```

produrrà il seguente risultato.

Viene copiato un file dal device reader ad un file su disco. Tale file si chiama "FILE.HEX" ed è allocato sul disco montato sul drive "A". Viene inoltre eseguito il controllo sulla correttezza dell'operazione di trasferimento dati (Intel Checksum Control).

## Devices Speciali

Alcune periferiche richiedono particolari codici per rendere usabili i dati ricevuti.

Esistono per il PIP cinque particolari devices che sono dedicati a svolgere tale funzione. I primi tre sono "EOF:", "NULL:", e "PRN:" e sono dedicati a svolgere particolari funzioni. I rimanenti due sono "INP:" ed "OUT:" ed informano il PIP che esso deve prendere dati da un programma che sta girando in macchina.

Vediamo ora alcuni esempi di impiego di tali devices.

**EOF:** End-Of-File - Trasmette un carattere EOF al file od al device ricevente. In condizioni normali il PIP trasmette automaticamente tale carattere se sta trasferendo files di tipo TXT. Nel caso però che si stiano trasferendo files di tipo particolare è necessario forzare manualmente tale carattere specificando questo device.

```
Es.: A > PIP PUN: = NEW,EOF:
```

Il file "NEW" viene trasferito dal disco "A" al device punch. Il trasferimento verrà chiuso con il carattere End Of File.

**NULL:** NULL - Questo device trasmette 40 caratteri Nulls (codici esadecimali 00) al device ricevente. Tale device viene generalmente usato per lasciare degli spazi vuoti all'inizio od alla fine di un file se esso deve essere trasferito su nastro magnetico o su nastro di carta. Tale area vuota successivamente ne renderà più facile il caricamento.

```
Es.: A > PIP PUN: = NULL: C: XMAS.LST,NULL:
```

Il device punch riceverà inizialmente una testa di 40 caratteri nulls successivamente riceverà il file XMAS.LST residente sul disco "C" e quindi riceverà una coda di altri 40 caratteri nulls.

**PRN:** Print - I dati vengono trasmessi al device "print" così come se avessimo usato il device "LST:" con l'eccezione che i caratteri di tabulazione vengono espansi ad 8 spazi, tutte le linee vengono numerate, e la stampante esegue automaticamente un salto pagina dopo 60 linee di testo (tale device non può pertanto essere usato su nastri perforati).

```
A > PIP PRN: = DATA.ASM
```

In seguito a tale comando verrà preso il file "DATA.ASM" dal disco "A" ed indirizzato alla printer. La stampa di ciascuna riga verrà preceduta dal suo numero all'interno del file, ciò può essere utile se si intende trattare tale file con qualche programma che permetta l'indirizzamento della singola riga (ad esempio l'Editor).

L'impiego del device "PRN:" è analogo all'impiego del device "LST:" specificando però in quest'ultimo caso l'opzione (T8NP).

I due devices speciali "INP:" ed "OUT:" dicono al PIP di usare un programma esterno per processare l'informazione. Tale programma deve essere un programma utente. L'impiego di tali devices e' pertanto riservato ai programmatori.

### **INP: Input data**

Es.: A > PIP FIXED=INP:

Viene creato un file di identificazione "FIXED" e riempito con i dati forniti dal nostro programma.

### **OUT: Output data**

Es. A > PIP OUT:=CON:

I caratteri che vengono digitati sulla console vengono trasferiti al nostro programma.



## STAT - Controllo dello stato del sistema

Questo programma e' in grado di svolgere le seguenti 5 funzioni base:

- Evidenzia lo spazio disco rimasto libero.
- Fornisce la dimensione di un file.
- Imposta lo stato dei files e dei dischi.
- Controlla l' assegnamento delle periferiche.
- Imposta l' assegnamento delle periferiche.

Tale procedura esegue il lavoro richiesto dal comando che la richiama e quindi cede nuovamente il controllo al CP/M.

### Controllo dello spazio disco rimasto libero

Per avere la situazione dello spazio rimasto libero su tutti i dischi acceduti in quell' istante e' sufficiente lanciare il comando:

```
A > STAT
```

Una possibile risposta a tale comando e' la seguente:

```
A: R/W,Space: 124K  
A >
```

Tale risposta significa che l' unico disco attualmente acceduto e' il disco "A" e che al suo interno rimangono 124 Kbytes di spazio disco libero. Inoltre il disco si trova, per cio' che riguarda la procedura STAT, in condizione R/W.

La risposta sarebbe stata invece R/O se il disco fosse stato protetto in scrittura tramite il comando STAT.

E' possibile selezionare l' indirizzo del drive che ci interessa specificandolo subito dopo la parola chiave STAT.

Es.: A > STAT C:

In seguito a tale comando la procedura STAT ci fornira' lo spazio rimasto libero sul disco "C", non ce ne fornira' invece il suo stato (cioe' se esso e' R/O o R/W).

### Controllo della dimensione di un file

Per avere la dimensione di un file e' sufficiente specificare il nome del file dopo la parola chiave "STAT". In tal caso il comando diverra' il seguente:

```
A > STAT PIP.COM
```

Con tale comando si richiede alla procedura STAT la dimensione del file "PIP.COM". Il risultato che potremmo avere da tale comando potrebbe essere, ad esempio, il seguente:

```

Recs  Bytes  Ext  Acc
64    8K    1   R/O A: PIP.COM
A>

```

In risposta al comando viene fornito lo spazio disco occupato dal file (8K), il numero di records che compongono il file (64), il numero di Extents (1) usate nel disco, il suo modo di accesso (R/O), il disco acceduto ed il nome del file interessato dal comando, ed infine lo spazio rimasto libero.

Nel formato CP/M un record contiene 128 bytes del file ed inoltre un gruppo di 128 records viene raggruppato e costituisce un "Extent".

Se, in fase di salvataggio di un file, il CP/M non trova uno spazio libero contiguo grande a sufficienza, esso salvera' il file in piu' aree di dimensione minore.

Verra' cosi' sfruttato tutto lo spazio disco a disposizione ed inoltre il CP/M sara' in grado di ricostruire il file in quanto all' interno del directory sono riportate le informazioni necessarie per la ricostruzione del file.

Il file e' impostato in modo R/O al fine di proteggerne la scrittura o la cancellazione. Tale modo di accesso e' impostato settando un bit tramite il comando "STAT".

E' inoltre possibile selezionare il drive sul quale andare a cercare il file. Cio' viene fatto specificando, prima dell' identificazione del file, il drive address.

Es.: A> STAT C: PIP.COM

E' permesso anche l' uso dei caratteri "\*" e "?" nel campo identificazione del file. Cio' ci permette di controllare piu' files con un solo comando. Così, ad esempio, il comando

```
A > STAT *.TXT
```

potrebbe fornire il seguente risultato.

```

Recs  Bytes  Ext  Acc
16    2K    1   R/W A: CHAP-A.TXT
8     1K    1   R/W A: CHAP-B.TXT
64    8K    1   R/W A: CHAP-C.TXT
32    4K    1   R/W A: CHAP-D.TXT
64    8K    1   R/W A: CHAP-E.TXT

```

Si noti che tale lista viene fornita ordinando le identificazioni dei files per ordine alfabetico.

La scrittura sullo schermo procede facendo lo scroll-up, se si desidera arrestarla e' necessario premere i tasti "CTRL-S".

Nel caso che il file interessato dal comando "STAT" sia un file ad accesso random, il risultato ottenuto puo' essere falso. Per informare la procedura che sta' operando su un file ad accesso random e' necessario specificare l' opzione "\$S". Il comando diverra' pertanto:

```
A > STAT PIP.COM $S
```

## Impostazione dello stato dei files o dei dischi

E' possibile usare la procedura "STAT" anche per svolgere funzioni che non sono esclusivamente informative (come quelle viste finora). Per esempio, tale procedura permette di proteggere dischi o files da una loro cancellatura accidentale.

Per avere una lista di tutti i comandi disponibili e' sufficiente usare l' opzione "VAL:". Il comando relativo diverra' percio' il seguente:

A> STAT VAL:

Il risultato di tale comando sara' quello di evidenziare a console la seguente schermata:

```
STAT VAL:
Temp R/O Disk: d: = R/O
Set Indicator: d: filename.typ $R/O $R/W $SYS $DIR
Disk Status: DSK: d: DSK
User Status: USR
Iobyte Assign:
CON: = TTY: CRT: BAT: UC1:
RDR: = TTY: PTR: UR1: UR2:
PUN: = TTY: PTP: UP1: UP2:
LST: = TTY: CRT: LPT: UL1:
```

Vediamo ora il significato di tale risposta.

A linea 1 troviamo:

```
Temp R/O Disc: d: = R/O
```

Tale comando serve per proteggere in scrittura il dischetto. Con tale comando il dischetto specificato nel comando stesso viene dichiarato R/O, e' cioe' possibile leggere i files contenuti al suo interno ma non e' invece possibile scrivere sul dischetto o cancellare files.

Tale comando agisce a livello di drive e' percio' necessario specificare il drive interessato dall' operazione.

Es.: STAT A: = R/O

Il disco presente sul drive "A" viene dichiarato R/O. Lo stato R/O e' temporaneo e permane solo fino a che non venga ricaricato il CP/M oppure fino a che non vengano premuti i tasti "CTRL-C".

Alla linea 2 troviamo la scritta:

```
Set Indicator: d: filename.typ $R/O $R/W $DIR $SYS
```

Questa forma del comando interessa il singolo file specificato nel comando stesso e serve per proteggerlo da una cancellazione accidentale oppure per renderlo un file di sistema.

Nell' identificazione del file e' possibile specificare il carattere "\*" e "?" se vogliamo che i files interessati dal comando siano piu' di uno.

E' possibile specificare come identificazione del file anche l' indirizzo del drive sul quale andare a cercare tale file.

L' identificazione del file e' sempre seguita da uno spazio, un dollaro e quindi uno degli indicatori R/O, R/W, SYS o DIR.

Vediamo ora a che cosa servono tali indicatori.

## Protezione di un file dalla cancellazione

A > STAT B: EXAMPLE.TXT \$R/O

Il file avente identificazione "EXAMPLE.TXT" e presente sul disco "B" viene dichiarato R/O. Tale file puo' cioe' essere letto ma non e' invece possibile modificarlo o cancellarlo.

Per annullare tale protezione e rendere il file nuovamente modificabile e' necessario annullare l' effetto di tale comando lanciando il comando:

A > STAT B: EXAMPLE.TXT \$R/W

Per far divenire un file generico un file di sistema e' invece necessario usare l' opzione "\$SYS".

Il comando assumerà pertanto la seguente forma:

A > STAT \*.COM \$SYS

L' indicatore "\$SYS" imposta i files che hanno filetype "COM" come files di sistema. Essi potranno pertanto essere usati ma non appariranno nel directory del dischetto e quindi non verranno listati dal comando "DIR".

Per fare il passaggio inverso e cioe' portare un file che era stato dichiarato come file di sistema a file standard e' necessario usare l' indicatore "\$DIR".

Questa seconda forma del comando diverrà pertanto:

A > STAT FORMAT.COM \$DIR

L' opzione "\$DIR" riporta il file "FORMAT.COM" al suo normale uso. Esso verrà pertanto reinserito nel directory e quindi verrà trovato dal comando "DIR".

Alla linea 3 troviamo invece:

Disc Status: DSK: d: DSK:

Tale parametro ci permette di controllare il tipo di disco.

Il comando assumerà pertanto la seguente forma:

A > STAT DSK:

In risposta avremo una schermata che ci descriverà le caratteristiche del dischetto presente sul drive "A".

Se vogliamo controllare un altro drive diverso dall' "A" sarà sufficiente specificarne l' indirizzo prima del parametro "DSK:". Così, ad esempio, se vogliamo controllare le caratteristiche del dischetto montato sul drive "C:" il comando da lanciare sarà il seguente:

A > STAT C: DSK:

Alla linea 4 abbiamo trovato la scritta:

User Status: USR:

Tale parametro permette di controllare gli users che hanno allocato delle aree su questo dischetto.

Così, ad esempio, il comando:

A > STAT USR:

richiede una informazione riguardo agli utenti che hanno allocato files sul dischetto (si ricorda che ciascun user ha un suo numero di identificazione compreso fra 0 e 15).

Una possibile risposta a tale comando e' la seguente:

```
Active User: 0  
Active Files: 0 1 2
```

Tale risposta deve essere interpretata nel seguente modo. L'utente attivo e' quello avente "user number" 0, ma sul dischetto sono presenti anche files degli utenti aventi "user-number" 1 e 2.

### Controllo degli assegnamenti delle periferiche

Il computer deve comunicare con dei dispositivi periferici esterni. Cosi' esso dovra' comunicare con la tastiera al fine di prelevare i comandi a lui diretti, con il display per fornire le risposte all'operatore e con gli altri dispositivi per elaborare dati.

Le ultime cinque righe della precedente risposta fanno appunto riferimento ai devices che sono attaccati al sistema.

Tali righe della risposta erano le seguenti:

```
Iobyte Assign:  
CON:= TTY: CRT: BAT: UC1:  
RDR:= TTY: PTR: UR1: UR2:  
PUN:= TTY: PTP: UP1: UP2:  
LST:= TTY: CRT: LPT: UL1:
```

Le quattro categorie principali che il CP/M usa per l'I/O sono la console (CON:), il reader (RDR:), il punch (PUN:) e la lista (LST:). Il CP/M selezionera' la categoria che gli serve in funzione del tipo di lavoro che deve svolgere.

Cosi', se esso deve ricevere comandi e fornire risposte, esso usera' la categoria console (CON:). Per comunicare con i vari dispositivi periferici esso usera' la categoria reader (RDR:) per ricevere dati e la categoria punch (PUN:) per trasmettere dati. Se esso invece deve stampare dati selezionera' la categoria list (LST:).

Per una maggiore flessibilita' del sistema e' pero' possibile assegnare ciascuna categoria ad uno qualsiasi dei devices connessi al sistema. Tali devices sono associati ad un codice di tre lettere che si trova alla destra del segno uguale.

Vediamo ora che cosa stanno ad indicare tali codici.

BAT:	Batch (input da RDR, output su LST)
CRT:	Cathode Ray Tube (video console/tastiera)
LPT:	Line printer
PTP:	Paper Tape Punch
PTR:	Paper Tape Reader
TTY:	Teletype
UC1	User Console 1
UL1:	User Line Printer 1
UP1:	User Punch 1
UP2:	User Punch 2
UR1:	User Reader 1
UR2:	User Reader 2

Tali nomi rappresentano in realta' una particolare connessione al computer. Il CP/M indirizza prima una categoria principale (CON:, RDR:, PUN:, LST:) e solo successivamente la connessione per il singolo device (le connessioni possibili sono in tutto quattro). Ogni connessione con il device fisico e' chiamata "porta".

Supponiamo per esempio di volere la stampa di una lettera. Inizialmente il CP/M inviera' ciascun carattere alla categoria "LST:". Ma da tale categoria esso puo' andare sia alla connessione "TTY:", sia alla "CRT:", sia alla "LPT:" che alla "UL1:". Quale connessione deve essere usata?

Per avere la lista dei devices associati ad una categoria e' sufficiente lanciare il comando "STAT" seguito dalla parola chiave "DEV:".

A > STAT DEV:

La risposta che otterremo dal CP/M sara' del tipo:

```
CON: is  TTY:
RDR: is  UR2:
PUN: is  UP2:
LST: is  LPT:
A >
```

Tale risposta ci mostra che tutti i comandi di console (CON:) arrivano dal CRT (Display video + tastiera). Il lettore (RDR:) ed il punch (PUN:) sono assegnati al connettore TTY. La lettera che si intende stampare verra' girata dalla categoria "LST:" alla connessione "LPT:". Se al connettore "LPT:" e' attaccata la stampante essa iniziera' a stampare ogniqualvolta vengono indirizzati dati alla categoria "LST:".

### Come si modifica l' assegnamento delle periferiche

Il CP/M esegue automaticamente l' assegnamento delle periferiche ogni volta che viene caricato in memoria. E' pero' possibile modificare tale assegnamento in ogni momento semplicemente ponendo la categoria uguale al device.

Alcuni esempi di comando che modificano la configurazione dell' I/O sono i seguenti.

A > STAT LST: = CRT:

Tale comando ruota le informazioni destinate alla categoria "LST:" sul proprio video. Cio' puo' essere fatto per verificare la correttezza di un documento prima di passarlo alla stampa. Per tornare all' associazione iniziale dei devices e' necessario lanciare il comando:

A > STAT LST: = LPT:

Da questo momento in poi i files destinati alla categoria "LST:" verranno indirizzati sulla porta "LPT:" (che e' quella della stampante) e quindi stampati.

Nel caso che si abbiano a disposizione due stampanti, una piu' veloce ed una con migliore qualita' di stampa, si puo' operare nel seguente modo.

Si attacchi la piu' veloce sulla porta "LPT" e quella piu' lenta sulla porta "UL1".

Dopo aver lanciato il comando

A > STAT LST: = LPT:

i files destinati alla categoria "LST" verranno inviati sulla stampante veloce. Se invece vogliamo una stampa di qualita' sara' necessario assegnare la categoria "LST:" alla porta "UL1:" dove e' stata attaccata la stampante di qualita'. Per fare cio' e' sufficiente lanciare il comando:

A > STAT LST: = UL1:

Se invece si intende usare un MODEM per trasmettere dati ad un utente remoto e' necessario connettere tale modem al connettore TTY. Successivamente, lanciando il comando:

A > STAT LST: = TTY:

avremo che i files destinati alla categoria "LST" vengano inviati all' utente remoto.

Oppure, lanciando il comando:

A > STAT CON: = TTY:

tutti i comandi ed i messaggi del CP/M verranno trasferiti al modem, ed inoltre il remoto potra' controllare il nostro sistema.

**N.B. Gli stessi devices definiti per la procedura "STAT" possono essere usati anche dalla procedura "PIP". Cosi', ad esempio, il comando:**

A > PIP UL1: = FILENAME.TYP

trasferira' il file avente identificazione "FILENAME.TYP" alla porta "UL1" dove e' stata connessa la stampante di qualita'. Tale file verra' quindi stampato su tale stampante.

## Automazione del sistema con STAT

Esiste un "device-name" che non rappresenta una particolare porta del computer, tale "device-name" e' "BAT:" e rende disponibile l' uso della console.

Quando e' selezionato tale assegnamento, il CP/M prende i suoi comandi dal device assegnato alla categoria "RDR:" ed il risultato della sua elaborazione viene trasferito al device associato alla categoria "LST:".

### MOVCPM nn \*

Questa e' una utility standard del CP/M. Il comando costruisce una versione del CP/M 2.2 di nn Kbytes completa dell' MDS BIOS e lascia una immagine del sistema operativo nelle locazioni che vanno dalla 980 in su'. Qualsiasi tentativo di modifica della sintassi di tale comando provochera' la caduta del sistema.

### WRTCPM <d: >

Questo comando scrive le sezioni CCP e BDOS del CP/M 2.2 all' interno dell' area di bootstrap del disco che si trova nel drive "d:". Se viene ommesso l' indirizzo del drive viene usato il drive attualmente attivo. I codici del CCP e del BDOS vengono prelevati dalla locazione di memoria 980 in poi; tale comando deve essere usato subito dopo un MOVCPM. Da questa operazione non vengono interessati i settori del BIOS e "Cold start loader". Il successo dell' operazione viene segnalato dalla procedura inviando a console un messaggio che informa l' operatore che una partenza "warm" puo' causare una caduta di sistema.

### WRTBIOS <d: >

Questa utility viene usata per scrivere le sezioni "BIOS" e "COLD START LOADER" del CP/M 2.2 all' interno dell' area di bootstrap dello specificato disco "d:". Se viene ommesso l' indirizzo del drive viene preso in considerazione il drive attualmente attivo. Il codice oggetto per tali aree viene prelevato dal file "CBIOS.HEX" che deve essere residente sul disco attualmente attivo.

Il file "CBIOS.HEX" viene creato assemblando il file "CBIOS.ASM". Si noti che l' area del "cold start loader" e' contenuta all' interno del source del CBIOS alla locazione BIOS + 380.



## Generazione di una nuova dimensione del sistema CP/M

Tale procedura mostra l'uso delle utilities WRTCPM e WRTBIOS descritte qui sopra. E' necessario rilocare il corpo del CP/M e le sezioni del BIOS indipendentemente e quindi riunirle assieme all'interno del disco. Per fare cio' si seguano i seguenti passi.

1- >ED CBIOS.ASM

All'interno di tale file e' necessario cambiare il rigo "SYSsize" che definisce l'area riservata al sistema. Tale parametro deve essere cambiato con il nuovo valore (espresso in Kbytes) della dimensione dell'area di memoria occupata dalla nuova versione del CP/M.

2- >ASM CBIOS

3- >WRTBIOS <d:>

Viene scritto il codice oggetto del BIOS nel disco "d:" specificato nel comando stesso.

4- >MOVCPM nn \*

5- >WRTCPM <d:>

Viene scritto il nuovo sistema ed inoltre esso e' mantenuto in memoria.

Si noti che una partenza "warm" da questo drive puo' provocare la caduta del sistema.

Tale procedura puo' essere automatizzata facendo uso della procedura "CPMGEN.SUB". E' necessario usare la sintassi corretta quando si richiama tale procedura.

Es.: SUB CPMGEN nn d:

Tale procedura assume che il file "CBIOS.ASM" sia residente sul disco attuale, se cio' non fosse la procedura non andrebbe a buon fine.

## Funzioni speciali del CP/M 2.2 SDX

Questa implementazione del CP/M prevede, senza nessuna modifica, il supporto di 2 drives (acceduti come B e C).

### Logical and physical drives

Logical	Physical	Controller	DRV-Select
A	Mapped to one of the following at cold boot		
B	B(0)	primary	1
C	C(1)	primary	2

Possono essere impiegati due tipi fondamentali di drives da 5.25", quelli S/S, S/D, 40 tracce a quelli 80 tracce D/S o D/D.

#### Config Codes

TYPE	Description
0	5 inch S/S S/D 40 Tk
1	5 inch D/S S/D 40 Tk
2	5 inch S/S D/D 40 Tk
3	5 inch D/S D/D 40 Tk
4	5 inch S/S S/D 80 Tk
5	5 inch D/S S/D 80 Tk
6	5 inch S/S D/D 80 Tk
7	5 inch D/S D/D 80 Tk

La configurazione di tali drives puo' essere modificata dinamicamente tramite un comando CP/M.

Inoltre, rendendo il drive "A" un drive logico che e' mappato all' interno di uno dei due drives fisici e' possibile liberare il drive "A" dalle costrizioni di qualsiasi particolare drive o tipo di drive.

Il firmware di bootstrap e' in grado di caricare il sistema da uno qualsiasi dei 2 drives fisici.

La procedura di bootstrap viene eseguita automaticamente all' atto dell' accensione. In alternativa puo' essere usata la procedura manuale interrompendo quella automatica durante la fase di "RAM check" con un carriage return. Cio' pone la procedura di bootstrap in un ambiente in cui essa accetta un gruppo limitato di comandi dalla console (si veda a tal proposito la sezione "The bootstrap PROM" nel manuale tecnico).

Una volta che il CP/M e' stato caricato esso mappa il drive logico "A" all' interno del drive fisico da cui e' stato caricato. Tale drive fisico restera' accessibile anche nel modo in cui esso era stato definito. Cosi', ad esempio, se il CP/M viene caricato dal drive "C", tale drive verra' mappato come drive logico "A" ma rimarra' anche accessibile come drive "C". Verra' assegnato un byte di I/O al valore default trovato nell' area di bootstrap del disco corrente e verra' eseguita automaticamente una stringa di comandi predefinita (a meno che non venga abortita dal carattere DEL introdotto da console). A tal punto il CP/M non e' consapevole dell' esistenza degli altri devices che non siano il drive "A" e del drive dal quale e' stato caricato che sono fisicamente lo stesso drive. Per installare un nuovo drive all' interno del sistema operativo e' necessario usare il comando CONFIG (tale comando fa' normalmente parte della stringa di comandi di partenza). Tale comando viene usato per personalizzare il CP/M sulla particolare installazione. Una volta che il CONFIG ha terminato la personalizzazione esso evidenzia a console la situazione dei drives attualmente installati.

La procedura CONFIG ha anche un altro modo di operare.

In tal modo i parametri vengono passati ad essa dalla riga comando. Tali parametri permettono di modificare la vecchia configurazione per crearne una nuova. E' possibile installare la configurazione cosi' modificata come configurazione default tramite il comando STARTUP. Tale procedura "STARTUP" permette anche di definire la stringa di comandi che deve essere generata all'atto della partenza del sistema (cold boot) ("STARTUP CONFIG - BASIC DEMO"; dove il carattere "-" viene usato per rappresentare la barra rovesciata che verra' poi tradotta in un "carriage return").

Vediamo ora quali sono le utilities fornite con il CP/M SDX.

## CONFIG

CONFIG < d: t,d: t, ... d: t >

Tale comando e' usato per configurare e riconfigurare il CP/M per varie combinazioni dei tipi di drives. E' responsabilita' dell'utente informare il CP/M del tipo di drive corrispondente ad un certo indirizzo. Il comando ha 2 diversi modi di operare.

- 1- Il modo default e' quello di non usare nessuna stringa comando, per inizializzare il CP/M. Cio' viene fatto usualmente al momento di una partenza a freddo (cold boot) con le configurazioni default memorizzate nelle tracce sistema del disco corrente. Tali configurazioni vengono create usando inizialmente la procedura CONFIG, nel modo 2, e successivamente la procedura STARTUP. Si noti che in questo modo di operare CONFIG non riconfigurerà un drive già configurato. Per avere una lista dello stato corrente della configurazione e' sufficiente specificare il comando CONFIG seguito da uno spazio bianco.
- 2- Quando viene introdotta una stringa comando valida essa verra' analizzata ed usata per generare le nuove configurazioni e riconfigurazioni. Nella stringa comando "d" rappresenta l'indirizzo del drive e "t" rappresenta il tipo che deve essere allocato a tale drive. "t" e' un digit od una coppia di digit edadecimali, si veda a tal proposito la descrizione dei numeri di "drive-type".

Quando si riconfigura un drive già configurato e' necessario osservare che sono permessi solamente cambiamenti fra drives omogenei. Se si intende riconfigurare il drive di sistema viene generata una pausa per permettere all'operatore di sostituire il supporto. E' necessario che il nuovo disco abbia la stessa dimensione del CP/M.

Una volta che la procedura CONFIG ha completato la sua operazione viene fornito il sommario della configurazione attuale. Vengono forniti frallaltro i valori dei puntatori della "FREE SPACE" della "TOP OF AVAILABLE MEMORY". Il valore del puntatore "FREE SPACE" viene incrementato quando vengono installati drives dato che vengono allocati "CHECK" ed "ALLOCATION VECTORS".

Viene rivelata una condizione di errore se viene incrementato il valore del puntatore "FREE SPACE" oltre il valore corrente del puntatore TOAM (TOP OF AVAILABLE MEMORY). Il TOAM viene impostato al momento della partenza del sistema (cold boot), esso punta al primo byte o buffer caricato dal disco di IPL. Se richiesta la installazione di tale codice in alta memoria, deve essere inserito un sufficiente spazio fra il TOAM e l' FS (FREE SPACE). Il TOAM deve essere aggiornato al fine di proteggere il nuovo codice. Si noti inoltre che se e' attiva una stringa "ENTER" essa puo' venire terminata prima della sua esecuzione.

### Config Codes

TYPE	Description
0	5 inch S/S S/D 40 Tk
1	5 inch D/S S/D 40 Tk
2	5 inch S/S D/D 40 Tk
3	5 inch D/S D/D 40 Tk
4	5 inch S/S S/D 80 Tk
5	5 inch D/S S/D 80 Tk
6	5 inch S/S D/D 80 Tk
7	5 inch D/S D/D 80 Tk

Logical	Physical	Controller	DRV-Select
A	Mapped to one of the following at cold boot		
B	B(0)	primary	1
C	C(1)	primary	2

## STARTUP

STARTUP < msg1 ↵ msg2 ↵ ... >

Questa utility viene usata per indirizzare la stringa comando di partenza registrata nel disco sistema. Essa ha tre diverse forme di sintassi.

- 1- Se non viene specificata nessuna stringa (viene usata solamente la parola chiave STARTUP) l'effetto e' quello di memorizzare il corrente byte di I/O e configurazione dischi nel disco sistema. Da questo momento la configurazione default di partenza sara' quella che e' stata attualmente salvata.
- 2- Con un carattere blank - L'effetto e' quello di cancellare la stringa comando di partenza all'interno del disco sistema.
- 3- Se viene specificata una stringa essa verra' scritta sul disco sistema ed usata come stringa comando di partenza nei prossimi IPL di sistema. Si usi il carattere "↵" per generare il carriage return. Per abortire l'esecuzione della stringa comando e' necessario premere a console il tasto "DEL" (Si noti che la stringa "↵ ↵" viene interpretata come un carattere "↵" e che ogni carattere preceduto da "¢" genera il corrispondente codice di controllo. La stringa "¢¢" genera un singolo "¢").

In tali considerazioni il carattere "↵" va' a sostituire la barra rovesciata; il carattere "¢" va' invece a sostituire l'accento circonflesso.

## COLDBOOT <d:>

Questo comando viene usato per generare una partenza a freddo (cold-start) dal drive specificato (d). Questo drive diviene il disco di sistema per tutte le successive partenze a caldo (warm start) ed a tale drive viene associato l'indirizzo logico "A". Se il disco "d" viene specificato come "\$" viene usato, per la partenza a freddo, il corrente disco sistema.

## BAUD <channel> <rate>

Il campo "channel" puo' essere specificato sia come "A" che come "B" e rappresenta il canale seriale che verra' influenzato dal comando (A = SIOA, B = SIOB).

Il campo "rate" puo' assumere uno dei seguenti valori: 9600, 4800, 2400, 1200, 600, 300, 110 e specifica la velocita' richiesta per tale canale.

## RCHECK <d:>

Questo comando controlla il drive specificato o, se e' omesso il campo "device-address", il disco sistema. Il carattere ":" viene evidenziato sulla console per ogni settore buono letto. Se si verifica un errore viene evidenziato un numero compreso fra 0 e 9 (0 = hardware-error, 9 = software-error) al posto del carattere ":". La stampa di un qualsiasi carattere durante l'esecuzione della procedura abortira' la procedura stessa. Il numero di settori letti e' funzione di come e' stato configurato il drive. Si noti che, in questo caso, si tratta di settori logici (128 Bytes) indipendentemente dalla dimensione attuale del settore per il disco analizzato.

## BATCH

Questa utility esegue una funzione simile alla SUBMIT con la differenza che, una volta che e' stata chiamata tale procedura, essa accetta linee comando del CP/M dalla console che verranno eseguite immediatamente dopo che tale utility e' stata terminata con il comando "@END".

## ENTER <msg1 ^ msg2 ^ ... ^ msgn >

Questa utility, quando viene usata all'interno di files SUBMIT od in blocchi comando della BATCH, ha l'effetto della simulazione dell'introduzione delle stringhe "msg1 ... msgn" da console. Il carattere "^" viene interpretato come carriage-return e viene assunto che alla fine della riga vi sia un CR. Per abortire l'esecuzione della stringa e' sufficiente premere il tasto "DEL". Questa procedura permette a programmi interattivi come "DDT" ed "ED" di essere elaborati come files di tipo SUBMIT o BATCH. Si noti che "^ ^" viene interpretata come un singolo carattere "^" e che qualsiasi carattere preceduto da "¢" genera il corrispondente codice di controllo. La stringa "¢¢" genera un singolo carattere "¢". Si noti che le stringhe

ENTER vengono accumulate ed e' percio' possibile usare contemporaneamente piu' righe di questo tipo. Le stringhe ENTER fanno uso dell' area "FREE SPACE" in alta memoria. Se viene introdotta una stringa cosi' lunga da uscire da tale area il comando viene abortito.

### ERAQ filename

Questa utility fornisce una funzione di erase associata a quella di query. Il campo "filename" puo' essere un qualsiasi filename CP/M (inclusi i caratteri "\*" e "?"). Prima che venga eseguita la cancellazione del/i file(s) specificati l' operatore viene avvisato a console. Esso puo' rispondere con "Y" se intende cancellare il file, con "N" se intende conservarlo oppure CR per abortire il programma. In questo caso vengono presi in considerazione anche i files di sistema.

### Utilities di programmazione

Tali programmi di utility sono i seguenti:

ED  
DDT  
ASM  
SUBMIT  
LOAD  
XSUB  
DUMP

Il linguaggio Assembler e' uno dei diversi linguaggi usati per scrivere programmi (altri linguaggi sono il BASIC, il FORTRAN, il COBOL etc.). Il linguaggio Assembler e' pero' quello che sfrutta nel modo migliore le risorse del sistema. Vediamo ora come lavorano tali programmi.

### ED - Editor

Formato: A > ED FILENAME.ASM

Questo e' un programma di editing "line-oriented" che viene usato per creare il source del programma. Tutte le linee vengono numerate ed e' necessario dire all' editor quale e' la linea che intendiamo modificare e quali posizioni all' interno della linea dobbiamo cambiare.

Per svolgere queste due funzioni base l' editor ha due diversi modi di operare, il modo "insert" che permette di introdurre testi ed il modo "comandi" che ne permette la modifica. Per passare da un modo di operare all' altro e per dare all' editor le sue istruzioni vengono usati tasti speciali.

Una volta terminata la scrittura del source e' possibile salvarlo su disco con il filename specificato nel comando stesso. Il filetype deve soddisfare le richieste del programma assembler che viene usato. Se, per esempio, stiamo usando l' assembler standard del CP/M e' necessario usare come filetype "ASM".

## ASM - Assembler

Formato: A > ASM filename.123

Il programma "ASM" converte il codice mnemonico nel codice usato dal computer (codice macchina 8080). Il programma sorgente deve avere filetype "ASM". Usando questo file sorgente il programma "ASM" produrrà due altri files su disco.

Il primo file contiene il codice macchina ed ha filetype "HEX". Questo file può essere usato con l'utility "LOAD" per creare una versione finale, eseguibile, del programma.

Il secondo file è un listing del programma contenente, per ogni istruzione in codice mnemonico, il relativo sviluppo in codice macchina. Il filetype di tale file è "PRN". Tale file è utile solamente per la messa a punto del programma.

Le tre posizioni del filetype che sono state rappresentate con "123" specificano dove è allocato il file sorgente e dove debbono essere scritti i due files "HEX" e "PRN". Se tale campo viene lasciato vuoto viene usato il drive di default per tutti i tre files.

Specificando invece tale campo è possibile usare drives che non siano quello attualmente attivo sia per prelevare il sorgente che per andare a scrivere i files "HEX" e "PRN".

Vediamo ora quale configurazione possono assumere queste tre posizioni.

### Posizione 1

Tale posizione contiene l'indirizzo logico del drive (A, B, C ... etc) all'interno del quale il programma "ASM" deve prelevare il file sorgente da assemblare.

### Posizione 2

Se viene specificato un indirizzo logico di drive (A, B, C ...) il file "HEX" verrà scritto all'interno del disco presente su tale drive. Se viene specificato "Z" il file non viene creato, ciò permette di risparmiare tempo durante la fase di correzione degli errori di sintassi.

### Posizione 3

Un indirizzo logico di device in questa posizione viene interpretato come l'indirizzo del device sul quale deve essere scritto il file "PRN". Se viene specificato "Z" il file "PRN" non verrà creato. Se viene specificato "X" il file "PRN" verrà trasmesso alla console senza che venga creato il file su disco.

Una volta che l'assemblaggio del nostro programma sia terminato senza nessuna segnalazione di errore dell'assemblatore, il programma è pronto per essere caricato in memoria e fatto girare.

L'utility che permette il caricamento del file "HEX" in memoria è la "LOAD".

## LOAD - Loader

Formato: A > LOAD FILENAME

Questa utility converte il file "HEX" creato dal programma "ASM" in un file comando direttamente eseguibile e lo salva su disco. Il filetype di questo file sarà "COM".

Una volta che il programma è stato caricato ed il file "COM" è stato creato è possibile richiamare il programma semplicemente specificandone il filename come comando CP/M. Per la messa a punto di tale programma, qualora esso non svolga le funzioni desiderate, viene fornita un'altra utility che prende il nome di "DDT".

## DDT - Dynamic Debugging Tool

Formato: A > DDT Filename.Type

Questo programma e' specializzato per la messa a punto dei programmi utente. Esso puo' essere richiamato fornendo l' identificazione del file che si intende debuggare oppure non specificandolo. Se viene specificato esso verra' caricato in memoria, se esso non viene specificato esso verra' introdotto una volta richiamata l' utility.

Questa utility ci avvertira' che e' pronta inviando a console un prompt costituito dal carattere "-".

Tale programma ha un suo set di comandi che permette di displayare, introdurre, o modificare i contenuti del programma sotto test. I comandi permessi da tale utility, e la loro funzione, sono i seguenti.

### Command Result

- A Puo' essere introdotto un insieme di righe in codice mnemonico Assembler.
- D Vengono displayati i contenuti degli indirizzi di memoria.
- F Viene riempita la memoria con i dati.
- G Viene eseguito un salto all' indirizzo di memoria ed eseguita l' istruzione ivi contenuta.
- I Viene introdotto un file di parametri .
- L Viene listato il contenuto della memoria usando i codici assembler.
- M Vengono mossi dati in memoria.
- R Viene letto un file su disco.
- S Viene sostituito un valore in memoria con un altro.
- T Viene tracciato il programma cosi' come esso gira.
- U Viene annullata la traccia del programma.
- X Esamina, ed eventualmente modifica, lo stato del microprocessore.

Assieme a queste singole lettere, molti comandi richiedono anche informazioni aggiuntive cosi' come l' indirizzo di memoria che si intende modificare. Questo comando, per esempio, sara' S1000 per esaminare ed opzionalmente cambiare l' indirizzo 1000.

Una volta che sono stati fatti i cambiamenti voluti e' possibile provare immediatamente il programma oppure tornare al CP/M premendo i tasti "CTRL-C". E', a questo punto, possibile salvare il programma (con le sue modifiche) facendo uso del comando di CP/M "SAVE".

## DUMP - Display file

Formato: A > DUMP Filename.Type



L' utility DUMP displaya sulla console il contenuto del file specificato. Tutti i dati vengono mostrati in forma esadecimale.

Il prodotto di tale utility potrebbe essere, ad esempio, il seguente:

```
A > DUMP MESSAGE.TXT
0000 43 6F 6E 67 72 61 74 75 6C 61 74 69 6F 6E 73 2D
0010 79 6F 75 20 64 65 63 6F 64 65 64 20 74 68 69 73
A >
```

Le prime quattro cifre di ciascuna linea contraddistinguono la posizione del primo byte del rigo all' interno del file.

Premendo un qualsiasi tasto della tastiera viene abortito il display di tali dati. Premendo i tasti "CTRL-S" il display puo' essere interrotto e riprendera' allorché tali tasti saranno premuti nuovamente.

### SUBMIT - Automatic operation

Formato: SUBMIT Filename.SUB

La maggior parte dei lavori che si fanno con il CP/M sono lavori di routine (copie di utility ed assemblaggio di programmi etc.). Tali lavori possono essere fatti manualmente, oppure si puo' istruire il CP/M a farli usando questa utility.

E' sufficiente creare un file contenente i comandi che e' necessario dare, quindi scrivere "SUBMIT" ed il nome di tale file. Ciascun rigo del file verra' passato nella "command-line" per essere eseguito dal CP/M. Il filetype di tale file deve essere "SUB".

Supponiamo, per esempio, di avere creato un file che si chiama "DEMO.SUB" e che contiene le seguenti linee comando:

```
PIP B: = STAT.COM
PIP B: = PIP.COM
PIP B: = EDITOR.COM
DIR B:
```

L' esecuzione di tale gruppo di comandi puo' essere richiamata lanciando il comando:

```
A > SUBMIT DEMO
```

In seguito a tale comando vedremo che il CP/M va' a prendere il file sul disco e porta ciascuna riga del file nella linea comando eseguendo poi tale linea come comando CP/M.

### Impiego delle variabili con "SUBMIT"

La routine SUBMIT e' in grado di accettare anche delle variabili.

Per esempio, e' possibile usare tali variabili al posto di specifici filenames. Tali nomi di variabile sono composti da un numero preceduto dal carattere "\$".

Si supponga ad esempio di creare il seguente file avente nome "DEMO.SUB".

```
PIP C: = $1
PIP C: = $2
```

PIP C: = \$3  
DIR C:

I files che verranno copiati in seguito all' esecuzione di tale file verranno specificati nella riga comando che richiama il file stesso.

Si supponga, ad esempio, di lanciare il comando:

```
A > SUBMIT DEMO STAT.COM PIP.COM EDITOR.COM
```

Le variabili "\$1, \$2 e \$3" verranno sostituite con i nomi di files che sono stati forniti nella riga comando "SUBMIT". L' ordine di nomi determina il numero. La forma generale del comando SUBMIT diverra' pertanto:

```
File Variabile Variabile  
A > SUBMIT name $1 $2 ... Etc.
```

Nell' esempio precedente alle variabili "\$1, \$2 e \$3" verranno assegnati rispettivamente i valori "STAT.COM, PIP.COM e EDITOR.COM".

I comandi che verranno eseguiti dal CP/M saranno pertanto i seguenti:

```
A > PIP C: = STAT.COM  
A > PIP C: = PIP.COM  
A > PIP C: = EDITOR.COM  
A > DIR C:
```

Come variabile puo' essere usata una qualsiasi frase, i vincoli da soddisfare sono solo i seguenti:

- Il file SUBMIT deve risiedere sul disco A.
- Il disco non deve essere protetto in scrittura.

### XSUB - Extended SUBMIT

Normalmente la procedura SUBMIT permette di lanciare solamente comandi di CP/M. Una volta che un programma viene introdotto dovete fornirgli i comandi di cui esso necessita. Ma se nella prima riga del file SUB e' contenuto il comando XSUB allora anche i comandi all' interno di un programma verranno portati fuori.

Quando SUBMIT e XSUB vengono usate per programmi diversi dalle utilities esse sono limitate ad eseguire operazioni standard.

### SUB - Modified Submit

Formato: SUB filename

E' questa una versione modificata della SUBMIT che permette di operare da un qualsiasi drive (non siamo vincolati al drive "A"). Tale utility permette inoltre l' uso del carattere di controllo "¢". Tale procedura e' percio' da preferire alla SUBMIT.

## Caratteri di controllo del CP/M

### Tasti            Azione

- CTRL-C    Abortisce il programma corrente ed esegue una partenza a caldo.
- CTRL-E    Forza un carriage return, non passa però il comando al CP/M.
- CTRL-H    Esegue la funzione di back-space.
- CTRL-J    Esegue un line-feed.
- CTRL-M    Esegue la funzione di carriage-return.
- CTRL-P    Esegue una funzione di echo diretta dalla console alla stampante. Premendo nuovamente tali tasti si pone fine a tale funzione di echo.
- CTRL-R    Riscrive la linea comando corrente.
- CTRL-S    Interrompe il listing sulla console; premendo nuovamente tali tasti si ha la ripresa del listing.
- CTRL-U    Cancella la linea corrente; viene displayato il carattere "#" ed il cursore si muove di una linea al di sotto e si pone in attesa di un nuovo comando.
- CTRL-X    Cancella tutti i caratteri presenti nella riga comando.
- CTRL-Z    Separatore di stringa o di campo.
- BACKSPACE Muove il cursore all' indietro di una posizione cancellando il carattere precedente.
- DEL        Cancella i caratteri alla sinistra del cursore.
- RETURN    Produce un carriage-return.

## Filetypes CP/M

I filetypes comunemente usati in ambiente CP/M sono i seguenti.

### Filetype      Indicazione

- ASM - Contraddistingue i files scritti in linguaggio sorgente Assembler.
- BAK - E' un file di back-up creato dall' editor; la procedura editor fa' un "rename" del file che viene processato attualmente. Il file rimane quindi disponibile sul disco con tale filetype.
- BAS - File contenente un programma in linguaggio CBASIC.
- COM - File eseguibile dall' 8080.
- HEX - File contenente un programma in formato esadecimale.
- INT - File contenente un programma in linguaggio "CBASIC intermediate".
- IRL - File di tipo "indexed REL" prodotto dalla utility "LIB".
- LIB - Usato dalle procedure MAC ed RMAC per la creazione di librerie di macro. Il comando "ED R" legge i files di tipo "LIB". Il comando "ED X" scrive files di tipo LIB. Il file e' di tipo stampabile e viene displayato sulla console o sulla stampante.
- OVL - File di tipo overlay di un programma. Il compilatore PLI-80 permette di usare la tecnica di overlay (sovrapposizione) dei files. E' possibile creare files di tipo overlay usando la procedura "LINK-80".
- PLI - Il file contiene il source di un programma in linguaggio PL/I-80.
- PRL - File rilocabile per pagina. E' un file che non richiede un particolare segmento. Esso puo' essere rilocato in qualsiasi pagina (256 bytes).
- PRN - File di tipo stampabile. Puo' essere displayato a console oppure sulla stampante di sistema.
- REL - File rilocabile prodotto dalla RMAC e dal PL/I-80 che puo' essere linkato con la procedura LINK-80.
- SUB - File di tipo SUBMIT che puo' contenere uno o piu' comandi CP/M. Il programma "SUBMIT" esegue comandi contenuti nei files fornendo un modo di operare batch per il CP/M.
- SYM - File di tipo "Symbol table". Le procedure MAC, RMAC e LINK-80 creano files di tipo "SYM.SIO" e la procedura "ZSIO" legge files di tipo "SYM".
- TEX - File sorgente per il TEX-80 (e' un text-formatter).
- XREF - File "cross-reference" prodotto dalla procedura "XREF".
- \$\$\$ - File temporaneo.

## CP/M - Messages

? (DDT Error Message) - Questo messaggio appare quando la procedura DDT non riconosce l'istruzione assemble, il file non può essere aperto.

ABORTED (PIP) - E' stata interrotta la procedura PIP premendo un tasto.

### ASM Error-Messages

D - Data error: Lo statement di dati non può essere messo nell'area di memoria specificata.

E - Expression error: L'espressione non può essere calcolata durante l'assemblaggio.

L - Label error: La label non può apparire in questo contesto (potrebbe essere una label doppia).

N - Not implemented: Viene richiamata una feature che non è stata implementata.

O - Overflow: L'espressione è troppo complicata per essere calcolata.

P - Phase error: Il valore della label viene modificato durante l'assemblaggio.

R - Register error: Il valore specificato come registro non è compatibile con il codice dell'istruzione.

S - Syntax error: L'espressione formata non è corretta.

U - Undefined label: Viene usata una label che non esiste.

V - Value error: E' stato incontrato in una espressione un operando formato non correttamente.

BAD DELIMITER (STAT) - Controllate la linea comando per gli errori di scrittura.

Bad load - Messaggio di errore del CCP o del SAVE.

Bdos Err On d - E' stato rivelato un errore del BDOS (Basic Disc Operating System) nel drive designato. Questo messaggio è seguito da una delle 4 frasi descritte nel seguito.

Bdos Err On d: Bad Sector - Questo messaggio viene lanciato quando il CP/M non trova nessun disco nel drive, il disco è formattato in modo errato, lo sportello del drive è aperto, oppure il drive è spento. Questo messaggio può indicare anche un hardware-error. Si possono premere i tasti "CTRL-C" per tornare al CP/M oppure il tasto RETURN per ignorare l'errore.

Bdos Err On d: File R/O - E' stato fatto un tentativo di cancellazione, rename o modifica degli attributi di un file che è stato dichiarato R/O. E' necessario impostare il file in modo R/W tramite il comando STAT.

Bdos Err On d: R/O - Al drive "d:" è stato assegnato, tramite la procedura STAT, lo stato R/O oppure il disco è stato cambiato senza poi avere premuto "CTRL-C". Il CP/M termina il programma premendo un qualsiasi tasto.

Bdos Err On d: Select - Il CP/M ha ricevuto un comando che specifica un drive che non esiste. Il programma corrente viene terminato. Premere i tasti "CTRL-C" o return per recuperare l'errore.

Break 'x' at c (ED.) - 'X' è uno dei simboli descritti nel seguito e "c" è il comando che veniva eseguito quando è stato trovato l'errore.

# Search failure - La procedura ED non puo' trovare la stringa specificata nel comando F,S oppure N.

? Unrecognized command lettere C - La procedura "ED" non riconosce la lettera comando indicata, oppure un comando E, H, Q od O non e' presente nella sua linea comando.

O - Il file specificato in un comando "R" non viene trovato.

> Buffer full - La procedura "ED" non puo' mettere altri caratteri in memoria oppure la stringa specificata in un comando "F, N oppure S" e' troppo lunga.

E Command aborted - L' esecuzione del comando e' stata abortita perche' e' stato premuto un tasto sulla console.

F Disc or directory full - Questo messaggio e' seguito dall' indicazione "Disc full" oppure "Directory full". Si faccia riferimento alle procedure di recovery specificate per tali situazioni di errore.

CANNOT CLOSE DESTINATION FILE (filespec) - Tale messaggio viene lanciato dalla procedura "PIP". Il file in output non puo' essere chiuso. E' necessario agire nel modo appropriato dopo aver controllato se nel drive c' e' il disco corretto e se il disco non sia protetto in scrittura.

Cannot close, R/O

CANNOT CLOSE FILES - Il CP/M non puo' scrivere sul file. Cio' accade generalmente quando il disco e' protetto in scrittura.

ASM - Un file in output non puo' essere chiuso. Questo errore termina l' esecuzione del programma ASM. Si deve controllare che il disco sia nel drive e che non sia protetto in scrittura.

DDT - Il file scritto con un comando "W" non puo' essere chiuso. Questo errore pone termine all' esecuzione della procedura "DDT". Controllate che il disco sia inserito nel drive e che non sia protetto in scrittura.

SUBMIT - Questo errore si puo' avere durante il processo dei files tramite la procedura SUBMIT. Controllate che il disco sistema sia nel drive A e che il disco non sia protetto in scrittura. La procedura SUBMIT puo' essere restartata dopo aver caricato nuovamente il CP/M.

CANNOT READ (PIP) - La procedura PIP non puo' leggere il file specificato perche' non e' implementato il lettore.

CANNOT WRITE (PIP) - La destinazione specificata nel comando "PIP" non e' permessa. E' stato specificato probabilmente come output un device di input.

Checksum error:

PIP - E' stato incontrato un errore di parita' sul record esadecimale. Il record che produce tale errore deve essere corretto, per fare cio' e' probabilmente necessario ricreare il file.

```
CHECKSUM ERROR
LOAD ADDRESS hhh
ERROR ADDRESS hhh
BYTES READ:
hhhh:
```

LOAD - Il file contiene dati non corretti. E' necessario generare nuovamente il file "HEX" partendo dal source.

Command Buffer Overflow (SUBMIT) - Il buffer della SUBMIT accetta fino a 2048 caratteri nel file di input.

Command too long (SUBMIT) - La riga comando all'interno del file SUBMIT non può superare i 125 caratteri.

CORRECT ERROR, TYPE RETURN OR "CTRL-Z" (PIP) - E' stato incontrato un errore di parità durante il trasferimento di un file. Il file con tale errore deve essere corretto probabilmente ricreando il file.

DESTINATION IS R/O, DELETE (Y/N): (PIP) - Il file di destinazione specificato in un comando PIP esiste già ed è stato dichiarato R/O. Se a tale domanda si risponde "Y" il file di destinazione viene cancellato prima che venga eseguita la copia.

Directory full -

ED - Non c'è sufficiente spazio di directory perché il file venga scritto sul disco destinatario. Si può usare il comando "OXfilespec" per cancellare dal disco i files non necessari senza lasciare l'ambiente editor.

SUBMIT - Non c'è sufficiente spazio di directory per scrivere il file "\$\$\$\$.SUB" sul drive "A". Cancellare alcuni files oppure cambiare disco e riprovare.

Disc full (ED.) - Non c'è sufficiente spazio disco per il file di output. Questo errore si può avere con i comandi "W, E, H oppure X". Si può ripetere il comando specificando un altro indirizzo di drive.

DISC READ ERROR (filespec) (PIP.) - Il file di input specificato in un comando PIP non può essere letto correttamente. Ciò accade generalmente quando viene incontrato un EOF inatteso. E' necessario correggere il file di output.

DISC WRITE ERROR (filespec)

DDT - Una operazione di scrittura su disco non può essere portata a termine con pieno successo durante un comando "W". Ciò accade probabilmente a causa di una situazione di disco pieno. Si può agire in due diversi modi:

- 1- Cancellare alcuni files non necessari per liberare spazio disco
- 2- Usare un altro disco con più spazio libero.

PIP - Una operazione di scrittura su disco non può essere eseguita con successo durante un comando PIP, ciò accade probabilmente per una situazione di disco pieno. Si può liberare spazio disco cancellando i files non necessari oppure prendere un altro disco con maggiore spazio libero.

SUBMIT - La procedura SUBMIT non può scrivere il file "\$\$\$\$.SUB" sul disco. Create spazio disco cancellando alcuni files oppure prendete un altro disco.

ERROR: BAD PARAMETER (PIP) - In un comando "PIP" è stato introdotto un parametro non corretto. Riscrivere il comando in forma corretta.

ERROR: CANNOT OPEN SOURCE, LOAD ADDRESS hhhh (LOAD) - Tale messaggio viene lanciato se la procedura "LOAD" non trova lo specificato file oppure se non è specificato nessun filename.

ERROR: CANNOT CLOSE FILE, LOAD ADDRESS hhhh (LOAD) E' provocato da un codice di ritorno che segnala un errore dalla chiamata alla funzione BDOS. Il disco potrebbe essere protetto in scrittura.

ERROR: CANNOT OPEN SOURCE, LOAD ADDRESS hhhh (LOAD) Il file non viene trovato.  
Controllate il directory del disco.

ERROR: DISC READ, LOAD ADDRESS hhhh (LOAD) E' provocato da un codice di ritorno che segnala un errore restituito da una chiamata al BDOS.

ERROR: DISC WRITE, LOAD ADDRESS hhhh (LOAD) Il disco di destinazione del file e' pieno.

ERROR: INVERTED LOAD ADDRESS, LOAD ADDRESS hhh (LOAD) L' indirizzo di un record era troppo distante dal record processato in precedenza. E' questo un limite interno della procedura "LOAD", esso pero' puo' essere facilmente ovviato. Si usi la procedura DDT per leggere il file all' interno della memoria, quindi si usi il comando SAVE per trasferirne l' immagine su disco.

ERROR: NO MORE DIRECTORY SPACE, LOAD ADDRESS hhhh (LOAD) Il directory del disco e' pieno.

Error on line nnn message (SUBMIT) - Il programma SUBMIT displaya i suoi messaggi nel formato visto in precedenza, dove nnn rappresenta il numero della linea all' interno del file SUBMIT.

FILE ERROR (ED) - Si ha una situazione di disco o di directory pieno e la procedura "ED" non puo' scrivere nient' altro sul disco. In seguito a tale errore si ha la perdita del controllo dell' editor.

FILE EXISTS - E' stato chiesto al CP/M di creare o di fare il rename usando come identificazione del file una identificazione che era gia' associata ad un altro file. E' possibile cancellare il file esistente oppure usare un' altra identificazione per il file da creare.

REN - L' identificazione del file specificata nel comando "RENAME" e' la stessa di quella di un file gia' esistente. Non e' possibile assegnare come nuovo nome di un file il nome di un file che esiste gia'. Se si intende sostituire un file esistente con una nuova versione dello stesso file e' necessario fare il rename o la cancellazione del file esistente, oppure usare la procedura PIP.

File exists, erase it (ED) - Il filename di destinazione esiste gia' quando si va' a mettere il file di output su un disco diverso da quello da cui era stato preso. Esso puo' essere cancellato oppure e' possibile selezionare un altro drive per ricevere il file di output.

\*\* FILE IS READ ONLY \*\* (ED) - Il file richiamato dalla procedura "ED" e' stato dichiarato R/O. La procedura puo' percio' leggere il file ma non puo' modificarne il contenuto.

File Not Found -

CP/M - Il CP/M non trova lo specificato file. Controllate che sia stato fornito il corretto indirizzo di drive e che il drive contenga il disco voluto.

ED - L' editore non trova il file specificato nel comando. Controllate che sia stato introdotto il corretto indirizzo di drive e che il drive contenga il disco voluto.

STAT - La procedura STAT non trova il file specificato nel comando. Il messaggio puo' apparire se viene omissso l' indirizzo del drive, si controlli anche che il disco voluto sia all' interno del drive specificato.

FILE NOT FOUND (filespec) - (PIP) Il file di input che e' stato specificato nel comando non esiste.



- Filename Required - (ED) E' stato introdotto il comando "ED" senza specificare nessuna identificazione di file. Si reintroduca il comando seguito dall' identificazione del file che si intende editare.
- hhh??=dd - (DDT) I caratteri "???" indicano che la procedura "DDT" non sa come rappresentare in linguaggio Assembler 8080 la configurazione esadecimale "dd" incontrata all' indirizzo "hhhh". Cio' accade quando la configurazione "dd" non e' un codice macchina dell' 8080.
- Insufficient memory - (DDT) Non c'e' sufficiente memoria per caricare il file specificato in un comando "R" od "E".
- Invalid Assignment - (STAT) E' stato specificato un drive od una identificazione di file non valido. Questo messaggio puo' essere seguito da una lista delle identificazioni valide che possono seguire un filename. Se e' stato usato un assegnamento non valido viene displayato il messaggio "Use: d:=RO" che mostra la sintassi corretta per l' assegnamento dei drives.
- Invalid Control Character - (SUBMIT) I soli caratteri di controllo validi nei files SUBMIT sono quelli compresi fra "A" e "Z". Si ricordi che nei files SUBMIT il tasto "CTRL" viene rappresentato specificando l' accento circonflesso.
- INVALID DIGIT (filespec) - (PIP) Nella lettura di un file esadecimale e' stata incontrata una codifica esadecimale non valida. Il file sorgente dovra' essere corretto creandolo di nuovo.
- Invalid Disc Assignment - (STAT) Tale messaggio puo' apparire se la specificazione del drive viene fatta seguire da una qualsiasi cosa diversa da "= R/O".
- INVALID DISC SELECT - Il CP/M ha ricevuto una linea comando che specifica un drive non esistente oppure il disco presente nel drive e' stato formattato in maniera errata. Il programma in corso viene abortito.
- INVALID DRIVE NAME (Use A,B,C,D,E or F,G,H,I see bootstrap rom) - (SYSCOPY) La procedura SYSCOPY riconosce come destinazione valida per la generazione del sistema, solamente i drives supportati dalla PROM di bootstrap.
- Invalid File Indicator - (STAT) Questo messaggio viene lanciato dalla procedura STAT se non viene specificato RO, RW, DIR o SYS.
- INVALID FORMAT -
- PIP - Il formato del comando lanciato e' non valido.  
 INVALID HEX DIGIT  
 LOAD ADDRESS hhhh  
 ERROR ADDRESS hhhh  
 BYTES READ:  
 hhhh
- LOAD - Il file contiene una configurazione esadecimale errata.
- INVALID MEMORY SIZE - (MOVCPM) Si deve specificare un valore minore di 64K o minore della dimensione attuale della memoria del computer.
- INVALID SEPARATOR - (PIP) E' stato usato un carattere non valido come separatore fra due identificazioni di files di input.

INVALID USER NUMBER - (PIP) E' stato specificato uno "user-number" maggiore di 15. Gli "user-numbers" sono nel range 0-15).

n? - (USER) E' stato specificato un numero maggiore di 15 per una "user-area". Per esempio se viene inviato il comando "USER 18", sul video verra' scritto: "18?".

NO DIRECTORY SPACE - (ASM) Il directory del disco e' pieno. E' necessario cancellare alcuni files per fare spazio per i files "PRN" ed "HEX".  
Il directory ha spazio per soli 64 files.

NO DIRECTORY SPACE (filespec) - (PIP) Non c'e' piu' spazio nel directory per il file di output. Le soluzioni possibili sono 2, la prima e' quella di cancellare alcuni files per creare lo spazio libero necessario, la seconda e' quella di cambiare dischetto.

NO FILE (filespec) -

DIR,ERA,REN,PIP - Il CP/M non trova il file specificato oppure sul disco non c'e' nessun file.

ASM - Non viene trovato nel drive indicato nel comando il file sorgente oppure e' stata specificata l'opzione "include" nel comando.

DDT - Il file specificato nel comando R od E non viene trovato nel disco.

NO INPUT FILE PRESENT ON DISC - (DUMP) Il file che e' stato specificato nel comando non esiste sul disco.

No memory - Non c'e' sufficiente memoria disponibile per il caricamento del programma specificato.

NO SOURCE FILE ON DISC - (SYSCOPY) La procedura SYSCOPY non trova il CP/M ne' nella forma CPMxx.COM ne' sulle tracce sistema del disco sorgente.

NO SOURCE FILE PRESENT - (ASM) L' assemblatore non trova il file che e' stato specificato nel comando. Si possono avere due possibilita':  
1- E' stata sbagliata la scrittura del filename sulla riga comando  
2- Il filetype del "filename" specificato non e' "ASM".

NO SPACE - (SAVE) Sul disco ci sono gia' troppi files oppure sul disco non e' rimasto sufficiente spazio per salvare le informazioni.

No SUB File Present - (SUBMIT) La procedura SUBMIT, per operare in modo corretto, necessita di un file avente filetype "SUB". Tale file deve contenere comandi CP/M. E' inoltre necessario che tali comandi risiedano uno per linea.

NOT A CHARACTER SOURCE - (PIP) La sorgente specificata per il proprio comando PIP non e' permessa. Probabilmente e' stato specificato come file sorgente un device di output.

\*\* NOT DELETED \*\* - (PIP) La procedura PIP non puo' cancellare il file che e' stato dichiarato R/O.

NOT FOUND - (PIP) La procedura PIP non trova il file specificato.

OUTPUT FILE WRITE ERROR - (ASM) E' stato specificato un dischetto protetto in scrittura come destinazione per i files "PRN" ed "HEX", oppure il dischetto specificato come output non ha spazio sufficiente. E' necessario risolvere il problema prima di assemblare il programma.

Parameter Error - (SUBMIT) All' interno dei files SUBMIT i parametri validi sono quelli che vanno da \$0 a \$9.

QUIT NOT FOUND - (PIP) Non e' stato trovato nel file di input il parametro Q nella stringa degli argomenti.

Read Error - (TYPE) E' stato incontrato un errore mentre veniva letto il file specificato nel comando TYPE. Si controlli il disco e si provi nuovamente. Il comando "STAT filespec" puo' diagnosticare il problema.

READER STOPPING - (PIP) L' operazione del lettore e' stata interrotta.

Record too long - (PIP) Il PIP non e' in grado di processare un record piu' lungo di 128 bytes.

START NOT FOUND - (PIP) Nel file sorgente non viene trovato l' argomento S nella stringa degli argomenti.

SOURCE FILE NAME ERROR - (ASM) Nella definizione del filename da assemblare non sono permessi i caratteri "\*" e "?".

SOURCE FILE READ ERROR - (ASM) L' assemblatore non riconosce l' istruzione che si trova all' interno del file da assemblare. Si usi il comando TYPE per analizzare il file.

SYNCHRONIZATION ERROR - (MOVCPM) L' utility MOVCPM e' stata usata con la versione errata del CP/M.

"SYSTEM" FILE NOT ACCESSIBLE - E' stato fatto un tentativo di accedere un file impostato come "SYS" con il comando "SET".

\*\* TOO MANY FILES \*\* - (STAT) Non c' e' sufficiente memoria per la procedura STAT per fare il sort dei files specificati oppure sono stati specificati piu' di 512 files.

UNEXPECTED END OF FILE (filespec) - (PIP) E' stata incontrata la fine del file prima del record di terminazione file. Il file senza il record di terminazione deve essere corretto ricreandolo nuovamente.

Unrecognized Destination - (PIP) Si controlli la linea comando per cio' che concerne una destinazione valida.

USE: STAT d:=RO - (STAT) E' stato fornito un drive non valido per la procedura STAT. Il solo assegnamento valido per la procedura STAT e': STAT d:=RO.

VERIFY ERROR:-(filespec) - (PIP) Copiando con l' opzione "V" la procedura PIP rivela una differenza fra i dati riletti e quelli presenti in memoria (che erano stati appena scritti). Cio' indica un errore del disco o del drive.

XSUB ACTIVE - (SUBMIT) E' stata richiamata la procedura XSUB.

XSUB ALREADY PRESENT (SUBMIT) La procedura "XSUB" e' stata richiamata quando essa era gia' presente in memoria.

Your Input? Se il CP/M non riconosce il comando che e' stato specificato esso ci restituisce la linea comando ricevuta seguita dal carattere "?". Si controlli che il comando sia stato scritto correttamente o che esista sul disco attivo un file avente il filename del comando e filetype "COM"

## CP/M BDOS Functions

NO	HEX	FUNCTION NAME	INPUT	OUTPUT
0	00H	SYS RESET		NONE
1	01H	CON INPUT		NONE
2	02H	CON OUTPUT		E = CHAR
3	03H	READER INPUT		NONE
4	04H	PUNCH OUTPUT		E = CHAR
5	05H	LIST OUTPUT		E = CHAR
6	06H	DIRECT CON I/O (1)		E = OFFH INPT OFEH STAT CHAR OUTPT
7	07H	GET I/O BYTE		NONE
8	08H	SET I/O BYTE		E = IOBYTE
9	09H	PRINT STRING		DE = BUFFER
10	0AH	READ CON BUF		DE = BUFFER
11	0BH	GET CON STATUS		NONE
12	0CH	RTN VERSION #		NONE
13	0DH	RESET DISC SYS		NONE
14	0EH	SELECT DISC		E = DISC NUM
15	0FH	OPEN FILE		DE = FCB
16	10H	CLOSE FILE		DE = FCB
17	11H	SEARCH FOR FIRST		DE = FCB
18	12H	SEARCH FOR NEXT		NONE
19	13H	DELETE FILE		DE = FCB
20	14H	READ SEQ		DE = FCB
21	15H	WRITE SEQ		DE = FCB
22	16H	MAKE FILE		DE = FCB
23	17H	RENAME FILE		DE = FCB
24	18H	RTN LOGIN VECT		NONE
25	19H	RTN CUR DISC		NONE
26	1AH	SET DMA ADDR		DE = DMA
27	1BH	GET ADDR (ALLOC)		NONE
28	1CH	WRITE PROT DISC		NONE
29	1DH	GET R/O VECT		NONE
30	1EH	SET FILE ATTRIB		DE = FCB
31	1FH	GET ADDR (DISC PARM)		NONE
32	20H	SET/GET USR CODE		E = OFFH GET E = O-OFFH SET
33	21H	READ RAN		DE = FCB
34	22H	WRITE RAN		DE = FCB
35	23H	COMPUTE FILE SIZE		DE = FCB
36	24H	SET RAN REC		DE = FCB
37	25H	RESET DRIVE (3)		DE = DRV VECT

A = FF IF CODE NOT FOUND

40 28H WRITE RAN W/FILL

DE = FCB

A = ERR CODE

- 1) Questa questa funzione deve essere usata con le funzioni 1, 2, 9 o 11.
- 2) Si noti che A = 11, e B = H al momento del RETURN.
- 3) Il drive di sistema non puo' essere resettato.

In tale tabella sono state usate le seguenti abbreviazioni:

ADDR = ADDRESS  
ALLO = ALLOCATION  
ATTRIB = ATTRIBUTE  
BUF = BUFFER  
CHAR = ASCII CHARACTER  
CON = CONSOLE  
CUR = CURRENT  
DIR = DIRECTORY  
DSK = DISC  
ERR = ERROR  
PARM = PARAMETER  
PROT = PROTECT  
RAN = RANDOM  
REC = RECORD  
RTN = RETURN  
SEQ = SEQUENTIAL  
STAT = STATUS  
SYS = SYSTEM  
USR = USER  
VECT = VECTOR  
VERS = VERSION

## Un' occhiata ai comandi

### Keyboard Control Keys:

Control + C	Warm boot CP/M
Control + E	Continue on next line
Control + H	Delete character
Control + P	Printer ON/OFF
Control + R	Retype line
Control + U	Delete line
Control + X	Delete line and backup
BACKSPACE	Delete character
DELETE or RUBOUT	Delete and echo character

### Built-in Commands:

DIR name.typ	Directory of disc
ERA name.typ	Erase file
Type name.typ	Type file on console
REN new.type = old.typ	Rename a file
User n	User number set
SAVE b name.typ	Save a file

### MTX Special CP/M Commands

Startup	Initialises a startup command string
Coldboot	Generates a cold start
Baud	Sets RS232 baud rate
Rcheck	Read checks drive
Config	Configures CP/M for disc drive variations
Enter	Used inside submit or batch files.
Eraq	Erase with query

### Standard Utility Programs:

FORMAT	Prepare a new disc
MOVCPM	Adapt CP/M to memory size
PIP	Peripheral Interchange Program
STAT	Display and set system status
SYSCOPY	System generator to copy CP/M

### Programmer utilities:

ASM	Assembler
DUMP	Dump HEX code
DDT	Debugger
ED	Editor
LOAD	Load HEX file
SUBMIT	Submit work
XSUB	Extend SUBMIT
SUB	Modified SUBMIT

## Copying a program with PIP:

PIP d: destination = d: source1,...,d: sourcen(options)

### PIP Options:

B	Block mode
Dn	Delete characters
E	Echo in terminal
F	Form-feed remove
Gn	Get from user
H	Hex transfer
I	Ignore end record
L	Lowercase only
N	Number lines
O	Object file
Pn	Page eject add
Qx Z	Quit copying
R	Read system file
Sx Z	Start copyiing
Tn	Tab set
U	Uppercase only
V	Verify data
W	Write over R/O
Z	Zero parity bit

### PIP Standard Devices:

CON:	Console
LST:	List
RDR:	Reader
PUN:	Punch
INP:	Input
OUT:	Output

### PIP Special Devices:

EOF:	End of file
NUL:	Null
PRN:	Print

### Displaying Status:

STAT d:	Remaining disc space on drive d:
STAT DEV:	Present device assignments
STAT d.DSK:	Display disc parameters for drive d:
STAT NAME.TYP	Display file space
STAT VAL:	Available commands
STAT USR:	Active users

### Setting Status:

STAT NAME.TYP \$R/O	Set file to Read Only
\$R/W	Set file to Read or Write
\$SYS	Set file for system use
\$DIR	Set file for directory use
STAT d:R/O	Set disc to Read Only until warm boot

## Disco BASIC e programmi di utilita'

### FDXB.COM

I comandi che vedremo nel seguito devono essere intesi come un' aggiunta ai comandi MTX-BASIC standard.

Tali comandi si riferiscono alla manipolazione dei files.

Si ricorda che i files contenenti programmi BASIC devono avere filetype "BAS".

Tutti i comandi che maneggiano files sono prefissati con la scritta "DISC"  
(Es. 10 DISC SAVE "PROG.DAT").

In questo disco vi sono due versioni di BASIC e cioe':

- 1- MTX.COM
- 2- FDXB.COM

L' MTX converte il computer all' indietro in un MTX512. La versione allineata con il computer e' la FDXB.

Per far girare il BASIC su di un monitor ad 80 colonne il comando da lanciare e' il seguente.

FDXB <RET>

Per far girare il basic usando come monitor un televisore oppure un monitor a 40 colonne e' necessario usare invece il comando:

FDXB 40 <RET>

Si noti che la procedura "FDXB" supporta solamente dischetti da 5".

## PROGRAM FILE COMMANDS

### Program file manipulation

#### SAVE

Formato: SAVE <filename >

Serve per scrivere su disco il programma che e' correntemente in memoria. Al file cosi' creato verra' associata la identificazione specificata nel comando stesso. Se esiste gia' un file avente tale identificazione esso verra' ricoperto dal nuovo.



Es. 10 DISC SAVE "B: PROG.BAS"  
10 DISC SAVE "C: STAT.BAS"

## LOAD

Formato: LOAD <filename>

Serve per caricare un file (quello avente identificazione "filename") dal disco in memoria. Le istruzioni che erano presenti in memoria prima del comando vengono cancellate.

E' possibile passare informazioni ai programmi attraverso files su disco.

Es. 10 DISC LOAD "B: PROG.BAS"  
10 DISC LOAD "C: STAT.BAS"

## OPEN

Formato: OPEN #<channel no>, <filename>, <type>, (<reclen>)

Questo comando serve per permettere le operazioni di I/O su di un file su disco. Perche' sia possibile eseguire una qualsiasi operazione di I/O su un file e' necessario averne fatto la OPEN.

Questa istruzione determina il modo di accesso usato per il file.

Il parametro "<type>" e' una espressione il cui primo carattere e' uno dei seguenti:

O = Output sequenziale  
I = Input sequenziale  
R = Input/Output ad accesso random

Il parametro "<channel no>" e' una espressione intera il cui valore e' compreso fra 1 e 4. Il numero viene associato con il file per una lunghezza pari a quella definita nella OPEN, e viene usato come riferimento per altre istruzioni di I/O.

Il parametro <filename> e' una espressione che contiene un filename CP/M.

Il parametro "<reclen>" e' una espressione intera che, se usata, specifica la lunghezza record per files ad accesso random.

Es.: 10 DISC OPEN #2,"PROG.BAS","O"

## KILL

Formato: KILL# <channel no>

Serve per chiudere o cancellare un file correntemente aperto. Il parametro "<channel no>" e' una espressione intera il cui valore e' compreso fra 1 e 4. Il numero viene associato con il file per una lunghezza pari a quella specificata nella "OPEN" e viene usato come riferimento per altre istruzioni di I/O.

Es. 10 DISC OPEN #2,"TEMP.DAT","O"  
20 DISC KILL #2

## CLOSE

Formato: CLOSE # <channel no >

Questo comando serve per chiudere le operazioni di I/O su di un file su disco.

Il parametro "<channel no >" e' una espressione intera il cui valore e' compreso fra 1 e 4. Il numero e' associato con il file per una lunghezza pari a quella specificata nella OPEN e viene usato per fare riferimento al file con altre istruzioni di I/O. Tale parametro e' lo stesso con cui il file era stato aperto.

L' associazione fra il particolare file ed il numero di canale termina dopo l' esecuzione della CLOSE. A questo punto il file puo' essere riaperto usando un numero di canale diverso oppure con lo stesso numero di canale.

Una CLOSE specificata per un file di output sequenziale aggiunge un EOF al file che viene chiuso.

Es. 10 DISC CLOSE #2

## INPUT #

Formato: INPUT # <channel no > , arguments

Questo comando serve per leggere dati da un file su disco ed associare i dati letti a variabili del programma.

Il parametro <channel no > e' una espressione intera il cui valore e' compreso fra 1 e 4. Tale numero e' lo stesso con cui il file era stato aperto.

Gli "arguments" sono i nomi di variabili a cui si intende assegnare i dati letti dal file.

Il tipo di variabile deve essere concorde con il tipo specificato nel nome della variabile.

Non viene stampato nessun carattere "?" (richiesta di dati in input).

I dati presenti nel file debbono essere registrati nello stesso modo di come essi verrebbero forniti in risposta all' istruzione INPUT.

Se viene raggiunto un EOF quando veniva letta una stringa od un dato numerico, l' operazione viene terminata.

Con dati di tipo numerico o di tipo stringa il dato termina con un carriage-return ed un line-feed oppure con una virgola.

Il carattere line-feed seguito dal carriage-return non viene riconosciuto come delimitatore di stringa.

Es. 10 DISC INPUT #2,A\$,B\$

## LINE INPUT #

Formato: LINE INPUT # <channel no > ,arguments

Questo comando serve per leggere una intera linea da un file su disco dentro una variabile di tipo stringa.

Il parametro <channel no> e' una espressione intera il cui valore e' compreso fra 1 e 4. Tale parametro e' lo stesso con cui il file era stato aperto.

Gli argomenti sono nomi di variabile di tipo stringa cui vengono assegnati i valori letti nel file.

Questo comando legge tutti i caratteri presenti nel file fino alla sequenza carriage-return/line-feed. Il successivo comando leggerà tutti i caratteri compresi fra il punto in cui si era rimasti con la lettura ed il successivo carriage-return/line-feed. La sequenza carriage return / line feed di fine record non entra a far parte dei dati letti. Nel caso che vengano incontrate due sequenze di tal tipo consecutive (senza nessun dato fra di esse) la seconda viene conservata e considerata come valore da assegnare alla variabile.

## **PRINT #**

Formato: PRINT # <channel no> , <list of expressions>

Questo comando serve per scrivere dati su di un file ad accesso sequenziale oppure ad accesso random.

Il parametro <channel no> e' una espressione intera il cui valore e' compreso fra 1 e 4 ed e' lo stesso numero con il quale era stato aperto il file. Le espressioni nella "list of expressions" espressioni numeriche o di tipo stringa che vengono scritte sul file.

## **REC #**

Formato: REC # <channel no> , <logical record no>

Questo comando posiziona il puntatore del record all'interno del file nella posizione specificata nel comando stesso.

Il parametro "channel no" e' un numero compreso fra 1 e 4 ed e' lo stesso con cui il file era stato aperto. Il successivo record letto con una READ sarà quello specificato nel parametro "logical record no".

Es. 10 DISC REC #1,7

## **EOF**

Formato: EOF ( <channel no> )

Questo comando pone un EOF dopo l'ultimo record del file.

Il parametro "channel no" e' una espressione intera il cui valore e' compreso fra 1 e 4, tale numero e' lo stesso con cui il file era stato aperto.

Es. 10 DISC EOF #1,200

## WRITE

Formato: WRITE "<filename>", <start address>, <no of bytes>

Questo comando serve per scrivere un blocco di memoria sul disco.

Il parametro "filename" specifica l' identificazione del file CP/M che verra' scritto. I parametri "start address" e "no of bytes" sono valori espressi in codifica decimale e possono essere espressioni numeriche.

Es. 10 DISC WRITE "TEST.DAT",17000,100

## READ

Formato: READ "<filename>", <start address>

Questo comando serve per leggere un blocco di memoria da disco. Il parametro "filename" e' l' identificazione di un file CP/M. Il parametro "start address" e' espresso in codifica decimale.

Es. 10 DISC READ "TEST.DAT",17000,100

## DIR

Formato: DIR (string expression)

Questo comando serve per avere la lista dei files presenti sul disco. E' questo un comando BASIC che svolge le stesse funzioni del comando di CP/M DIR e viene usato nello stesso modo.

Es. 10 DISC DIR "\*.BAS"

## ERA

Formato: ERA (string expression)

Questo comando serve per cancellare files dal disco. Esso viene usato nello stesso modo in cui veniva usato il comando di CP/M "ERA".

Es. 10 DISC ERA "\*.DAT"

## Formati di display del "Disc Basic"

La risoluzione del display dell' FDXB e' di 80 caratteri per 24 righe, se opera in modo testo, e di 90x160 in modo grafico. Si hanno a disposizione 8 colori e 4 gruppi di caratteri.

Usando l' output RGB i colori diponibili sono:

0	Nero
1	Rosso
2	Verde
3	Giallo
4	Blue
5	Cremisi
6	Cyan
7	Bianco

Se viene selezionato l' output Nero e Bianco i comandi INK e PAPER hanno i seguenti risultati.

### INK

0	Normale
1	Sottolineato
2	Normale
3	Sottolineato
4	Luminoso
5	Luminoso e sottolineato
7	Luminoso
8	Luminoso e sottolineato

### PAPER

0	Normale
1	Normale
2	Sfondo 1
3	Sfondo 1
4	Sfondo 2
5	Sfondo 2
6	Reverse video
7	Reverse video

Usando il comando CRVS viene creato uno schermo grafico con 80 colonne.

Es. CRVS 6,3,0,0,80,24,80

Uno schermo di tipo 3 non ha caratteri definiti dall' utente SPRITES. Anche se caratteri utente e SPRITES possono essere definiti, e gli SPRITES controllati dall' interno di uno schermo di tipo 3. Gli SPRITES non appariranno su uno schermo di tipo 3, ma appariranno su uno schermo di tipo 1 (lo schermo grafico dell' MTX).

In uno schermo di tipo 3 i comandi "COLOUR" e "ATTR" sono leggermente modificati.

COLOUR n,m

n = 0	Print ink
1	Print paper
2	Non print ink
3	Non print paper
4	No effect

m = 0-7 secondo quanto descritto nel seguito.

ATTR n.1

Attributi di stampa

n = 0	Inverse print
1	Colour inverse
2	Alternative character set
3	Flash
4	Graphics set

Attributi che non sono di stampa

n = 5	Unplot
6	Exclusive or plot (overplot)
7	Flash

Tutti i caratteri "inverse" sono stampati normalmente nel "logical BQW inverse" (incluso il cursore). L'effetto del comando "ATTR 1.1" e' che tutti i caratteri "inverse" vengono stampati nel loro vero colore "inverse".

## CHARACTER SETS

In ambiente "FDXB.COM" sono disponibili 4 character sets:

- 1-
 

ASCII standard-	Codici da 32 a 127.
Grafici speciali-	Codici da 128 a 159.
ASCII alternato-	Codici da 160 a 255.
- 2- ASCII alternato piu' caratteri grafici speciali.
- 3- Standard grafics.
- 4- Alternate graphics

I primi tre gruppi possono essere selezionati usando una combinazione degli attributi di stampa 2 e 4.

	ATT 4,0	ATT 4,1
ATT 2,0	Chr set 1	Chr set 3
ATT 2,1	Chr set 2	Chr set 3

Il character set 4 (alternate graphics) puo' essere acceduto direttamente dal BASIC, dato che i codici che esso usa (da 0 a 31) vengono catturati dall' interpreter del BASIC. Questo perche', in ambiente BASIC, i codici da 0 a 31 costituiscono i caratteri di controllo per operazioni sullo schermo.

Per accedere questi caratteri la RAM del video deve essere scritta direttamente. Nel seguito viene riportato un esempio di programma che svolge tale funzione.

Si veda esempio a pagina 15.7 del manuale originale.

Per ulteriori informazioni riguardo a cio' si faccia riferimento alla parte "80 Column Board Section" del manuale tecnico.

In una schermata BASIC e' possibile mischiare i caratteri di tutti i quattro character sets. La procedura NODDY permette di displayare solamente i caratteri relativi ad un character set.

In tutte le versioni del BASIC sono disponibili tutti i tipi di schermo (tipi 0, 1 e 3).

Tipo 0: Testo su 40 colonne su un monitor a 40 colonne oppure su apparecchio televisivo.

Tipo 1: Testo su 32 colonne e grafica 256x192 su monitor a 40 colonne od apparecchio televisivo.

Tipo 3: Monitor su 80 colonne.

Da cio' si vede che e' possibile usare contemporaneamente un apparecchio televisivo ed un monitor su 80 colonne.

Si vedano gli esempi di programmi BASIC sul manuale originale.

### **TVI.COM - Televideo Terminal Emulator**

TVI.COM e' un emulatore di terminale TVI 912/920. Esso puo' essere usato con qualsiasi programma CP/M che sia stato installato per un terminale Televideo 912/920. Questo programma migliora le possibilita' del sistema SDX in quanto permette di far girare una notevole quantita' di software Televideo in formato CP/M senza la necessita' di personalizzare i programmi per adattarli alla serie MTX.

Per usare l' emulatore e' sufficiente farlo girare una volta e quindi far girare i propri programmi.

Es. A > TVI  
A > XXXTVI

dove XXXTVI e' un qualsiasi programma installato per un terminale Televideo 912/920.

L' emulatore TVI rimarra' in macchina fino al prossimo reset o partenza a freddo. Nelle seguenti tabelle sono riportati i caratteri di controllo e le sequenze di escape supportate dall' emulatore.

<----- Vedere tabelle sul manuale originale ----->

## Il programma di comunicazione CONTACT

Per l'uso di tale programma nel sistema MDX si faccia riferimento alla fine del paragrafo.

Questo programma fornisce le facilities di comunicazione per l'uso con i sistemi MEMOTECH.

Il programma, anche se e' stato fatto specificamente per tali macchine, puo' girare sotto qualsiasi sistema operativo CP/M. Esso e' stato scritto per facilitare la trasmissione e la ricezione dei files dal memotech all'host.

Quando sta' girando tale programma il computer si trasforma in un terminale che lavora da una delle due porte seriali della macchina. Tutti i caratteri digitati sulla tastiera vengono trasmessi sulla linea e tutti i caratteri in arrivo dalla linea vengono evidenziati sullo schermo. Il sistema e' percio' in grado di svolgere le funzioni di console di un terminale. Una volta che si operi in tale modo i files possono essere trasmessi in linea con uno qualsiasi dei due protocolli disponibili. Inoltre i dati ricevuti dalla linea possono essere depositati all'interno di un qualsiasi file. La macchina puo' operare sia in formato ASCII che in formato esadecimale.

Il programma e' in grado di comunicare con una notevole variet  di sistemi remoti. Il trasferimento di files dalla macchina remota puo' avvenire senza la necessita' di nessun particolare protocollo. I dati possono venire displayati sullo schermo e trasferiti al dischetto alla velocita' con cui vengono ricevuti. Il trasferimento di files dall'MDX all'host deve invece avvenire sotto il controllo di uno dei seguenti protocolli.

- Protocollo XON/XOFF. Tale protocollo e' sempre attivo.
- Protocollo opzionale che implica la trasmissione di una linea per volta in risposta ad un prompt della stazione remota.

Queste opzioni ed altre vengono impostate tramite un "menu". Tale menu viene richiamato digitando i caratteri "CTRL-G". Per tornare all'emulatore e' sufficiente premere RETURN. I dati arrivati dalla linea durante tale operazione non vengono persi ma vengono invece displayati a console quando si ritorna in ambiente emulatore.

Quando viene terminato il programma CONTACT, viene salvato sul disco un file "LAST.LNK" che contiene tutte le informazioni di stato del programma. La successiva volta che verra' richiamato il programma esso andra' a prelevare le informazioni di status da tale file e si trovera' automaticamente impostato come lo era la volta precedente.

E' inoltre possibile specificare diverse configurazioni della macchina facendo il rename del file "LAST.LNK" e richiamando la procedura CONTACT fornendo come parametro il "filename" del file che si e' cosı' creato. Il "filetype" deve essere necessariamente "LNK".



## Uso del CONTACT

Il programma CONTACT puo' essere richiamato da ambiente CP/M in diversi modi. Qualsiasi di questi modi venga scelto verranno salvate le informazioni di "status" nel file "LAST.LNK".

**CONTACT** Questo comando esegue una ricerca sul disco attivo per cio' che concerne il file "LAST.LNK". Se il file viene trovato verra' usato il suo contenuto per impostare il programma. Se il file non viene trovato, per i parametri verranno usati i valori default.

**CONTACT name** Con tale forma del comando il programma viene attivato usando come parametri di inizializzazione quelli contenuti nel file "name.LNK". Se il file non viene trovato viene lanciato un messaggio di avvertimento e verranno attivati i valori default di tali parametri.

**CONTACT d: name** Il file "name.LNK" viene cercato all' interno del disco di indirizzo "d:" ed all' interno di tale disco verra' anche salvato il file "LAST.LNK".

**CONTACT d:** Il file "LAST.LNK" viene cercato all' interno del disco "d:".

Una volta che il programma sta' girando, il computer si comporta come un terminale. Dato che la pressione dei tasti "G" provoca l' apparsa del menu, per trasmettere tale configurazione di caratteri e' necessario premere nuovamente tali tasti. Avremo cosi' la scomparsa del menu e la trasmissione della configurazione relativa a tali tasti.

Quando e' attivo il menu i dati in arrivo dalla linea vengono bufferizzati e vengono displayati non appena ripasseremo al controllo dell' emulatore. Una volta che il buffer e' stato riempito per 3/4, il programma invia un carattere XOFF all' host. Se l' host non riconosce tale carattere di controllo e continua la trasmissione, dopo aver ricevuto alcuni altri caratteri dall' host l' MDX esce dall' ambiente menu ed il contenuto del buffer viene displayato sul video. Se viene riconosciuto l' XOFF il programma, non appena sara' pronto per svuotare il buffer, inviera' il carattere XON per attivare di nuovo la trasmissione dall' host.

## Menu commands and options

- X Per uscire dall' ambiente emulatore e tornare al CP/M. Prima di tornare in ambiente CP/M il comando "X" chiude tutti i files esistenti e salva il file di status "LAST.LNK". Nel caso che si verifichi un errore del BDOS mentre sta' girando l' emulatore, l' utente deve lanciare il comando "CC" per richiedere una chiusura di emergenza della procedura CONTACT. Questa chiusura d' emergenza non provvede alla chiusura dei files ne' al salvataggio del file "LAST.LNK".
- CR Per tornare allo schermo CONTACT. Si digiti un carriage return per tornare allo schermo CONTACT e permettere lo svuotamento del buffer.
- E Per cancellare lo schermo CONTACT. Questo comando cancella lo schermo CONTACT e restora gli attributi default dello schermo, e' implicito un ritorno immediato allo schermo CONTACT.
- G Per trasmettere una codifica "G" alla stazione remota e tornare allo schermo CONTACT.
- DEL Forza un BREAK sulla linea. Dopo che e' stato lanciato il break il controllo torna al CONTACT.
- ESC Serve per passare dalla porta seriale "A" alla "B" e viceversa.
- B Serve per selezionare una nuova velocita' di trasmissione per il canale corrente.

In seguito a tale comando si entra in un menu che permette di selezionare la velocita' di trasmissione fra quelle di 300, 600, 1200, 2400, 4800 e 9600 bps.

- + , - Seleziona la parita' pari o dispari (EVEN or ODD).
- 0, 8 Per selezionare NULL-PARITY o RAW-8-bits.

La parita' EVEN/ODD forza tutti i dati trasmessi ad un codice di 7 bits con la relativa parita', anche se i dati ricevuti vengono convertiti in un codice a 7 bits prima di ogni ulteriore elaborazione. NULL-PARITY (0,8) selezionano rispettivamente una parita' a mark od a space, anche in questo caso i dati ricevuti vengono ridotti ad un codice a 7 bits. Nel modo "RAW-8" il sistema opera con un codice ASCII espanso ad 8 bits, i dati ricevuti vengono passati alla linea senza nessuna modifica, in tale codice e' possibile trasmettere anche la codifica "Z". I dati ricevuti vengono passati al file senza nessuna modifica. In questo modo di operare la codifica "Z" non provoca la chiusura del file. Nel codice a 7 bits la ricezione della codifica "Z" provoca la chiusura del file, in trasmissione tale codice non viene trasmesso.

Per ricevere il codice su 8 bits e' necessario mettere fuori servizio lo "SCREEN FILTER". Non e' raccomandato usare il codice su 8 bits per trasmettere files binari, e' invece preferibile fare uso delle facilities di conversione Hexadecimal.

- P Per introdurre un nuovo carattere di prompt. Digitando tale tasto viene evidenziato un menu che permette la selezione di un nuovo carattere di prompt che viene usato nel flusso controllato della trasmissione di files.

La "Status Display Box" contiene una copia di tale carattere, se tale carattere e' un carattere di controllo esso verra' evidenziato in "reverse video".

- S Per switchare ON/OFF lo "SCREEN FILTER".

Digitando tale tasto si ha il passaggio ON/OFF od OFF/ON dello SCREEN-FILTER. Quando tale feature e' attiva i codici ASCII vengono ridotti a 7 bits ed i soli caratteri di controllo che hanno effetto sono: NULL, BS, TAB, LF e CR. Tutti gli altri caratteri di controllo vengono rappresentati in "reverse video" ma non hanno alcun effetto. Inoltre viene fornita una indicazione relativamente all' attivita' di file transfer. Una alta luminosita' del video indica "nessuna attivita' di trasferimento dati", una bassa luminosita' indica invece che e' attivo un trasferimento dati. Quando la feature e' off i dati vengono passati direttamente al gestore del video che interpreta tutti i caratteri di controllo nel modo usuale. L' indicazione di "attivita' di trasferimento files" viene persa.

- D Seleziona il modo di funzionamento HALF/FULL DUPLEX.

La differenza fra i due modi di operare sta' nel fatto che nel modo HALF-DUPLEX viene svolta la funzione di echo locale della tastiera.

- F Per commutare ON/OFF il controllo del flusso.

Il controllo di flusso puo' essere usato per la trasmissione dei files. Il file da trasmettere deve essere in codice ASCII ed i records del file debbono essere terminati con "CR" oppure "CR-LF". Il trasferimento inizia con il primo record, prima di trasferire i successivi records il programma si mette in attesa della ricezione del carattere di "prompt" dall' host. L' eventuale carattere "LF" alla fine del record viene filtrato e non viene inviato in linea. L' host deve essere in grado di riconoscere il carattere "CR" come fine record e di inviare il suo "prompt" in linea. L' ultimo carattere del "prompt" dovra' essere il "CONTACT-PROMPT-CHARACTER" impostato nel menu.

I successivi quattro comandi "R, T, I ed H" operano tutti su files sul disco locale. Prima che tali comandi vengano eseguiti sara' percio' necessario fornire una identificazione di file. Tale identificazione apparira' nella "status box". Per abortire il comando e' necessario premere "C".

- R Per ricevere all' interno di un file.

I dati in arrivo dalla linea vengono salvati nel file selezionato. Il file puo' essere chiuso sia in seguito alla ricezione dei caratteri "Z" (se si opera in codice ASCII a 7 bits) che in seguito ad un intervento operativo.

Prima che venga iniziato il trasferimento dei dati viene testata la identificazione del file. Se il file esiste già verrà richiesta la scelta di una delle seguenti funzioni:

- Abort command
- Over write the file
- Append new data to the file

Esistono due tipi di "Append" uno "ASCII" ed uno "binario". Nell' "ASCII APPEND" viene rivelata la fine del file non appena viene incontrata la stringa "€Z". Nel "binary append" viene rivelata la fine del file quando è terminato l'ultimo settore allocato al file. In questo secondo caso i caratteri "€Z" eventualmente presenti nel file vengono ignorati.

T Per trasmettere da un file.

Viene aperto il file specificato e viene iniziata la trasmissione. Essa continuerà fino alla fine del file (se si opera in modo 8 bits) oppure fino a che non venga riconosciuto il carattere "€Z". Il file verrà chiuso dall'intervento dell'operatore. Il trasferimento può essere sospeso dall'intervento dell'operatore, dalla ricezione di una stringa "€S" oppure dalla trasmissione del carattere "CR" se il "FLOW-CONTROL" è attivo.

Una volta che il trasferimento è stato interrotto in seguito alla ricezione della codifica "€S", esso può essere riattivato in uno dei seguenti modi:

- Viene ricevuta la codifica "€Q"
- L'utente torna al menu principale digitando "€G" e poi a quello secondario digitando "CR".

Se è attivo il "FLOW-CONTROL" ed il trasferimento è stato sospeso in seguito alla trasmissione di un "CR", esso verrà ripreso solo se si verifica una delle seguenti condizioni:

- Viene ricevuto il carattere "prompt" selezionato nel menu.
- Viene ricevuta una codifica "€Q".
- L'operatore torna al menu principale (digitando "€G") e poi a quello secondario (digitando "CR").

È compito dell'operatore inviare una stringa di fine trasmissione all'host (es. "€Z").

I Per ricevere e decodificare l'immagine esadecimale di un file.

Questo comando viene usato per ricevere files binary in codice standard "INTEL-HEX". Una alternativa all'uso di questo comando è quella di ricevere il file come file ASCII (usando il comando R) e poi usare la procedura LOAD per convertire tale file in binario.

Man mano che i dati vengono ricevuti essi vengono controllati per ciò che concerne la correttezza della trasmissione e quindi convertiti in binario e registrati sul disco. Se viene rivelato un errore il file viene chiuso immediatamente. I caratteri che non siano in codifica esadecimale vengono ignorati e non entrano a far parte del file registrato. La ricezione del file continuerà fino a che non viene ricevuto un record di EOF esadecimale (:00XXXXXXXX), una codifica "€Z" oppure fino a che il trasferimento non viene sospeso da un intervento operativo. Il file può essere chiuso dall'intervento dell'operatore oppure dalla rivelazione di un errore. Questo comando ha la stessa funzione del comando "R" con più la funzione di decodifica da codice ASCII in codice binario.

H Per trasmettere l'immagine esadecimale di un file.

Si usi questo comando per trasmettere l'immagine nel formato "INTEL-HEX" di un file in codice binario od in codice ASCII. Il file viene letto come una serie di bytes (stringhe di 8 bits) fino alla sua fine fisica (non viene tenuta in nessun conto la sequenza EOF "€Z") e convertita la stringa ASCII in una immagine esadecimale prima di trasmetterla in linea. Questo comando opera nello stesso modo del comando "T" con la differenza che esso opera una conversione da binario ad ASCII.

Gli ultimi quattro comandi presi in considerazione (R, T, I ed H) producono l'apertura di un file per la lettura o la scrittura. Una volta che il file è stato aperto sarà necessario provvedere anche alla sua chiusura. Ciò può essere fatto con uno dei due seguenti comandi.

C Per chiudere un file.

Questo comando puo' essere usato per svolgere una delle tre seguenti funzioni:

- Per arrestare prematuramente il trasferimento di un file.
- Per chiudere files ricevuti con codice a 7 bits ma che non erano stati chiusi tramite la stringa "Z".
- Per chiudere files che erano stati ricevuti nel modo "8-bits-RAW-mode".

Quando viene chiuso prematuramente un file che era in corso di ricezione i dati presenti nel buffer non vengono inclusi nel file. Essi rimangono pertanto in attesa di essere prelevati. Sono stati salvati tutti i records che erano stati displayati fino al momento in cui e' stato richiamato il menu principale premendo i tasti "G".

Z Per sospendere o riprendere il trasferimento del file.

Questo comando permette di sospendere temporaneamente il trasferimento del file. Dando nuovamente tale comando il trasferimento verra' ripreso dal punto in cui era stato interrotto.

## The CONTACT Status Display Box

Questa tabella, che appare nell'angolo destro dello schermo, fornisce lo stato del CONTACT. Le informazioni che vengono fornite sono le seguenti:

SOFTWARE RELEASE NUMBER	RELEASE: 041282
BUFFER STATUS	0000/A416
CURRENTLY ACTIVE CHANNEL	CHANNEL A
CURRENT BAUD RATE	9600 BAUD
CURRENT FILE NAME	B: UNIX.DAT
CURRENT FILE STATUS	HEX RECEIVE
TRANSMISSION STATUS IF SUSPENDED	(SUSPENDED)
DUPLEX MODE	FULL DUPLEX
SCREEN FILTER STATE	SCRN FILTER ON
FLOW CONTROL STATE	FLOW CNTRL OFF
PROMPT CHARACTER	PROMPT: >
PARITY	RAW 8-BIT

Alcune di queste informazioni, con il loro valore corrente, vengono salvate nel file "LAST.LNK" una volta che sia richiesta la chiusura del CONTACT. Esse, con i loro possibili valori, sono riportate nel seguito.

PARITY	puo' essere: EVEN, ODD, NULL o RAW 8-BIT
SCREEN FILTER	puo' essere: ON od OFF
FLOW CONTROL	puo' essere: ON od OFF
PROMPT CHAR.	puo' essere un qualsiasi codice ASCII
FILE NAME	puo' essere un qualsiasi identificatore
CHANNEL	puo' essere sia A che B
DUPLEX	puo' essere HALF o FULL
BAUD RATE	puo' essere 300,600,1200,2400,4800 o 9600

## Come si riceve un file ASCII

I passi da seguire per ricevere un file da un remoto, o da una qualsiasi sorgente seriale, al quale il CONTACT e' connesso come terminale o come console sono i seguenti:

- 1- Accertarsi che la procedura CONTACT sia impostata correttamente (parita', baud rate etc.). Si usi il menu di controllo per tale operazione. Per tornare al menu principale e' sufficiente premere "CR".
- 2- Si stabilisca la connessione con la stazione remota e si scriva il comando per abilitare la trasmissione del file da parte di tale stazione (in ogni caso non si deve premere RETURN).
- 3- Si digiti "G" per entrare nel menu secondario, si dia il comando "R". Si introduca il filename richiesto e si diano le informazioni richieste nel caso che tale filename esista gia'. Si usi quindi il "CR" per tornare al menu principale.
- 4- Si digiti ora un nuovo "CR" per inviare il comando alla stazione remota. A questo punto i dati verranno displayati sulla console ed anche trasferiti al file specificato. Per indicare questa attivita' verra' inoltre ridotta la luminosita' dello schermo.
- 5- Una volta che sia terminata la ricezione o la si intenda terminare forzatamente, si ritorni al menu principale e si lanci il comando "C" per chiudere il file. Nel caso che la stazione remota abbia inviato la codifica "Z" quest'ultimo passo avviene automaticamente.

## Trasmissione di un file ASCII

I passi da seguire per trasmettere un file ASCII ad una stazione remota, o ad un qualsiasi device seriale a cui sia connesso il CONTACT come terminale o come console, sono i seguenti:

- 1- Si impostino correttamente i parametri del CONTACT ("channel number", "parity", "flow-control" e "prompt character" se richiesto). Si ritorni al menu principale digitando "CR".
- 2- Si lanci il comando che prepara la stazione remota a ricevere dati.
- 3- Si entri nel menu secondario digitando "G" e si lanci il comando "T". Si introduca quindi l' identificazione del file da trasferire e si ritorni al menu principale digitando "CR". A questo punto inizierà il trasferimento del file. Nel caso che la stazione remota non faccia l' echo dei caratteri ricevuti e' necessario impostare il CONTACT nel modo di operare HALF-DUPLEX, in tal caso infatti l' echo viene fatto dal programma stesso.
- 4- Una volta che e' terminata la trasmissione del file la procedura CONTACT lo chiude automaticamente. Puo' essere necessario introdurre ora, semplicemente digitandola sulla tastiera, la sequenza di terminazione per chiudere il file nella stazione remota.

## Trasmissione e ricezione di Files esadecimali

Cio' viene fatto grosso modo come veniva fatto per i files in codice ASCII. Dopotutto una immagine esadecimale di un file e' un tipo speciale di file ASCII. Questo tipo di trasferimento viene fatto quando si rende necessario avere una immagine binaria esatta del file che deve essere mosso da una macchina all' altra (files in codice binario). Non e' necessario che la macchina remota sia in grado di generare o decodificare tali immagini esadecimali, essa puo' cioe' memorizzarli come files ASCII. Nel caso che si operi pero' con macchine aventi come supporto il "CONTACT" se ne raccomanda l' uso.

## Note varie

### Messaggi di errore

I messaggi di errore della procedura CONTACT vengono displayati solo quando siamo nell' ambiente "menu principale".

### Come si cambiano i dischi

I dischetti possono essere cambiati quando lo si desidera se la procedura CONTACT e' nello stato "FILE INACTIVE". Essi non possono invece essere cambiati se la procedura si trova nello stato "FILE ACTIVE". Per entrare nel menu di controllo e controllare lo stato del sistema e' necessario digitare i tasti "G".

### Terminal Emulation

E' possibile riscrivere i drivers dello schermo del CONTACT per emulare altri terminali.

### CONTACT.COM nel sistema a dischi SDX

Questo programma viene fornito come parte del sistema FDX. Diversamente dalla versione FDX, la piastra controller fornita con l' "SDX" non supporta il DMA (Direct Memory Access), e' quindi possibile che vengano persi dati quando il programma va' a scrivere dati su disco e simultaneamente riceve dati all' interno della RAM.

Per evitare cio' deve essere fatto largo uso del buffer di input costituito dalla RAM dell' MTX.

Cio' puo' essere fatto leggendo blocchi di circa 40 Kbytes direttamente dentro la memoria trattenendo il display del menu di schermo del CONTACT quando avviene il trasferimento. Digitando "G" alla fine del trasferimento si avra' lo scaricamento del buffer all' interno del disco.





## Indice

Introduzione	3
Il sistema SDX CP/M con un dischetto	3
L' Hardware	3
SDX CP/M Single Drive Utilities Programs	4
Disco sistema	4
NewWord Working Disc	4
Dischetto SUPERCALC	5
FRMSDX.COM	6
SYS.COM	6
COPY.COM	6
Come si fa' una copia di backup del proprio disco sistema	7
Che cos'e' il CP/M?	7
Procedure di partenza	7
Regole fondamentali sull' uso dei floppy-discs	9
Comandi di sistema	10
DIR	10
ERA	12
TYPE	13
REN	13
USER	13
SAVE	14
Come si richiama l' esecuzione di un programma	15
Save per il salvataggio dei dati	15
Come si cambia dischetto o drive	16
Come si cambia drive	16
MTX special CP/M utilities	17
Come si fa' una copia del disco CP/M	18
Come si copia un nuovo disco programma	18
Come si crea un disco per uso generale	18
FORMAT	18
SYSCOPY	18
PIP - Che cosa e'?	20
Come si usa il PIP per copiare files	20
Come si usa il PIP per copie fra periferiche	22
Opzioni per la modifica dei files	23
Opzioni per la modifica del testo	24
Opzioni per copie generiche	26
Impieghi speciali della procedura PIP	29
Come si concatenano i files	29
Concatenamento dei files che non sono di tipo testo	30
Devices speciali	31
STAT - Controllo dello stato del sistema	33
Controllo dello spazio disco rimasto libero	33
Controllo della dimensione del file	33
Impostazione dello stato dei files o dei dischi	35
Protezione di un file dalla cancellazione	36
Controllo degli assegnamenti delle periferiche	37
Come si modifica l' assegnamento delle periferiche	38
Automazione del sistema con STAT	40
MOVCPM nn *	40
WRTCPM <d:>	40

WRTBIOS < d: >	40
Generazione di una nuova dimensione del sistema CP/M	41
Funzioni speciali del CP/M 2.2 SDX	42
Logical and physical drives	42
CONFIG	43
STARTUP	44
COLDBOOT < d: >	45
BAUD < channel > < rate >	45
RCHECK < d: >	45
BATCH	45
ENTER	45
ERAQ	46
Utilities di programmazione	46
ED	46
ASM	47
LOAD	47
DDT	48
DUMP	48
SUBMIT	49
Impiego delle variabili con SUBMIT	49
XSUB	50
SUB	50
Caratteri di controllo del CP/M	51
Filetypes CP/M	52
CP/M - Messages	53
CP/M BDOS Functions	60
Un'occhiata ai comandi	62
Copying a program with PIP:	63
Disco BASIC e programmi di utilita'	64
FDXB.COM	64
Program file commnads	64
SAVE	64
LOAD	65
OPEN	65
KILL	65
CLOSE	66
INPUT#	66
LINE INPUT#	66
PRINT#	67
REC#	67
EOF	67
WRITE	68
READ	68
DIR	68
ERA	68
Formati di display del DISC BASIC	69
TVI.COM - Televideo Terminal Emulator	71
Il programma di comunicazione CONTACT	72
Uso del Contact	73
Menu commands and options	73
The Contact status display box	77
Come si scrive un file ASCII	77
Trasmissione di un file ASCII	78
Trasmissione di files esadecimali	78
Note varie	79
Messaggi di errore	79

Come si cambiano i dischi  
Terminal emulation  
CONTACT.COM nel sistema a dischi SDX

79  
79  
79

