



Comando COMPILE

Funzione

Converte un file testo presente in memoria principale in un programma BASIC.

Formato

COM |PILE|

Azione

Il comando ordina al sistema di tradurre nel formato oggetto ogni linea del file testo, presente in memoria principale, che sia conforme alla sintassi del linguaggio BASIC. Al termine in memoria principale si ha un programma BASIC non ancora preseguito.

Note

1. Il file testo deve essere costituito di istruzioni BASIC, altrimenti il sistema stamperà un messaggio d'errore che identificherà ciascuna linea errata. Le linee errate restano in memoria e conservano il precedente numero di linea e sono così richiamabili nel buffer di tastiera per le necessarie operazioni di editing (comando FETCH).
2. Per effettuare la preesecuzione del programma in memoria principale, è necessario inviare un comando PREPARE o RUN.



Comando CONFIGURE

Funzione

Definisce una dimensione logica della memoria utente inferiore a quella fisica effettivamente presente e permette l'impiego di periferiche esterne con funzione ausiliaria a quelle integrate.

Formato

CON[FIGURE] [EXD] [EVD] [EP= n_1] [MS= n_2] [PRO]

dove:

EXD

specifica che viene utilizzato un video display Olivetti DSM 6680

EVD

specifica che viene utilizzato un display video esterno (collegato mediante interfaccia RS 232-C)

n_1

è un numero intero compreso tra 8 e 15 o 24 e 31 che identifica una stampante esterna (collegata mediante interfaccia IPSO)

n_2

è un valore compreso fra 16 e 48. Il valore indica la capacità logica, in K byte, da assegnare alla memoria utente.

PRO

specifica che deve essere eseguita la procedura catalogata *SETUP ogni volta che il sistema è inizializzato.

Azione

Il comando completo di tutti gli operandi ordina al sistema di inicializzarsi configurando la memoria utente con la capacità specificata in byte con l'operando MS = n_2 . Ordina inoltre che le operazioni di stampa riferite alla stampante integrata vengano eseguite sulla stampante il cui nome logico è specificato con l'operando EP = n_1 , e che sullo schermo del display video o del video Olivetti DSM 6680 siano visualizzati tutti i testi che appaiono sul display integrato e sulla stampante integrata. Infine ordina che la procedura catalogata *SETUP sia eseguita ogni volta che

il sistema è inizializzato.

Se non è specificato l'operando MS, il comando CONFIGURE ordina al sistema di inicializzarsi configurando la memoria utente con la capacità fisicamente presente nella unità centrale.

Se l'operando EP noon è specificato, il comando CONFIGURE ordina al sistema che le operazioni di stampa comandate da sistema o da programma utente (istruzioni PRINT e PRINT USING) non siano più eseguibili su di una stampante IPSO ma bensì sulla stampante integrata.

Se l'operando EXD o EVD non è specificato, i testi inviati alla stampante integrata e al display integrato non vengono trasmessi al video display.

Note

1. L'esecuzione del comando CONFIGURE comporta una re-inizializzazione del sistema con le modalità specificate dal comando che vengono registrate sul disco sistema o sul floppy disk sistema e rimangono valide finchè non è eseguito un nuovo comando CONFIGURE. Per stampare le informazioni che definiscono la configurazione di sistema attualmente operante, si esegua il comando ENVIRONMENT.
2. Per informazioni sull'uso del video Olivetti DSM 6680 e sulle sue prestazioni, vedere il manuale "P6066 Prestazioni grafiche - Manuale del programmatore".
3. La possibilità di configurare il sistema assegnando capacità di memoria utente minore di quella reale permette di verificare la corretta esecuzione di programmi destinati ad essere eseguiti su sistemi con capacità di memoria utente pari a quella simulata.
4. Le stampanti IPSO, PR 1350 e PR 1370 possono essere utilizzate in alternativa alla stampante integrata.
5. I caratteri che non rientrano nel set di quelli stampabili dalla stampante IPSO specificata con il comando CONFIGURE provocano la stampa del carattere `|||`. Alcuni dei caratteri compresi fra i primi 32 caratteri del set ISO sono, per le stampanti IPSO, dei comandi; per cui, se uno di questi caratteri è compreso nella linea da stampare, sarà interpretato dalla stampante come comando e la stampante eseguirà le azioni ad esso relative. Gli errori relativi

alla stampante IPSO specificata nel comando CONFIGURE sono segnalati visualizzando ABN PRT, come per la stampante integrata. Perché non ci siano errori usando una stampante IPSO tutte le operazioni di stampa devono essere completate prima che finisca il programma. Può quindi essere necessario inserire un ritardo dopo l'ultima istruzione PRINT di un programma.

6. Se viene eseguito CONFIGURE con EXD o EVD, sul video display esterno sono visualizzate le informazioni che in condizioni normali vanno solo alla stampante integrata e al display integrato.

Più precisamente, nella prima riga dal basso sono visualizzati i messaggi emessi dal sistema e le informazioni relative alle istruzioni DISP e DISP USING, nella seconda riga è visualizzata la linea introdotta da tastiera, nella terza riga non sono visualizzati caratteri e le righe superiori del quadro presentano le informazioni normalmente destinate alla stampante integrata, rispettando la logica di funzionamento del tasto PRINT ALL.

Se il tasto NO PRINT non è attivo, le informazioni sono presentate contemporaneamente su stampante integrata e su video display esterno. Attivando il tasto NO PRINT si blocca il funzionamento della stampante.

Le righe più in alto della terza compongono la "pagina" del display. La pagina viene riempita dal basso verso l'alto e quando è "piena" il tasto di console CONTINUE lampeggia per avvertire l'utente che vi sono altri caratteri che compongono l'informazione da visualizzare. Premendo il tasto **CONTINUE** la pagina viene riempita, dal basso verso l'alto, con i rimanenti caratteri dell'informazione e i caratteri precedentemente visualizzati sono spostati verso l'alto mentre quelli che erano nella linea più in alto sono man mano sostituiti.

7. L'uso degli operandi EXD, EVD e EP fa sì che le corrispondenti routine del sistema operativo vengano caricate in memoria principale. Queste routine richiedono la seguente occupazione in Kbyte:

EXD 1 Kbyte
EVD 1 Kbyte
EP 0,5 Kbyte

Quando EXD è presente con la routine GDI (vedi comando OPTIONS) l'occupazione di memoria totale sarà di 3 Kbyte invece di 3,5 Kbyte.

8. Quando è specificato l'operando EVD, il video display che può essere utilizzato è il TCV 415V oppure qualunque video display i cui caratteri di controllo siano compatibili con esso.
9. Per informazioni sulle procedure catalogate si veda più avanti il paragrafo "PROCEDURE CATALOGATE" ed il comando PROCEDURE.

Esempio

Il seguente esempio mostra l'impiego del comando CONFIGURE per ridurre la capacità della memoria utente a 16 Kbyte.

Si preme **C O N M S = 1 6**

Comando CREATE

Funzione

Alloca, in una libreria, lo spazio necessario a contenere un file dati.

Formato

CRE[ATE] filename, [lib-ref], $\left[\begin{array}{c} S \\ R \\ Z \end{array} \right] [n]$

dove:

filename

indica il nome di un file

lib-ref

può essere uno dei seguenti operandi:

(lib-name, unit-name)

lib-name

(,unit-name)

con lib-name che specifica il nome di una libreria e unit-name che specifica il nome di una unità a disco od a floppy disk

S

indica file dati di tipo sequenziale

R

indica file dati ad accesso diretto inizializzato con dati numerici in singola precisione che non hanno un valore assegnato

Z

indica file dati ad accesso diretto inizializzato con dati numerici in singola precisione il cui valore è zero

n

indica il numero di byte assegnati al file dati.

Azione

Il comando completo con tutti gli operandi ordina al sistema di allocare, per il file dati di nome filename,

n byte nella libreria specificata e di registrarne il nome nel catalogo della libreria.

Il comando privo del quarto operando ordina al sistema di allocare, per il file dati di nome filename, 4096 byte nella libreria specificata.

Note

1. In una libreria non vi possono essere due file con lo stesso nome.
2. Non si possono "creare" nuovi file dati in una sottolibreria package protetta.
3. Non si possono creare in una sottolibreria più file di quanti dichiarati per essa durante l'esecuzione del programma di utilità LBCREATE (vedi appendice A).
4. Il numero di byte richiesto viene arrotondato al successivo multiplo di 128 (per file su floppy disk) o di 256 (per file su disco).
5. La libreria specificata con lib-name deve essere aperta.
6. Se come operando lib-ref è specificato lib-name, lo spazio richiesto per il file è allocato nella prima libreria aperta di nome lib-name.
7. Se lib-name è omesso lo spazio richiesto per il file è allocato nella prima libreria aperta sul disco o floppy disk identificato dal nome simbolico unit-name.
8. Se l'operando lib-ref è omesso lo spazio richiesto per il file è allocato sulla prima libreria aperta.

Esempi

1. Si crei un file dati ad accesso diretto di 1024 byte nella sottolibreria comune della libreria MAT residente nell'unità UD.

Premere

{ C R E + C A L , I M A T , U D I ,
R T D 2 4 END OF LINE }

2. Si crei un file dati sequenziale di 4096 byte nella sottolibreria utente della prima libreria aperta.

Premere

Comando DATE

Funzione

Permette di datare le operazioni sui file registrati nelle librerie.

Formato

DAT [E] date

dove:

date

è una sequenza di 6 caratteri che indica una data.

Azione

Il comando comunica al sistema di registrare sul disco sistema o sul floppy disk sistema la data introdotta da tastiera come operando del comando stesso.

Si devono sempre introdurre 6 caratteri e nessuno di essi può essere lo spazio.

Note

1. Durante l'inizializzazione del sistema, (dopo l'accensione o durante l'esecuzione dei comandi OPTIONS e CONFIGURE) la sequenza di caratteri registrata per ultima sul disco sistema o sul floppy disk sistema è caricata in memoria principale e quindi utilizzata per datare le operazioni sui file dati esterni.
2. La sequenza introdotta sarà, in generale: ggmmaa (giorno, mese ed anno).

Esempi

1. Si registri la data: 1 gennaio 1980.

Premere **D** **A** **T** **0** **1** **1** **0** **8** **0** **END OF LINE**



Comando DCHANGE

Funzione

Permette di sostituire un disco, od un floppy disk, mentre il sistema è in funzione, senza cancellare il contenuto della memoria principale.

Formato

DCH[ANGE] unit-name

dove:

unit- name

indica il nome simbolico di una unità a disco od a floppy disk.

Azione

Il comando comunica al sistema che si vuol sostituire il disco, o floppy disk, presente nella unità con nome simbolico unit-name.

Quando il comando è eseguito vengono chiuse tutte le librerie che erano aperte sul disco o floppy disk che deve essere sostituito ed è visualizzato il messaggio:

LOAD DISK ON unit-name

per avvertire l'utente di sostituire il disco, o floppy disk, presente nell'unità con nome simbolico unit-name. Dopo aver inserito il nuovo disco, o floppy disk, si deve premere il tasto di console **CONTINUE**.

Se la prima libreria presente nel disco inserito, o la libreria presente nel floppy disk inserito, non è stata creata associando ad essa una parola d'ordine e se vi sono meno di 6 librerie già aperte attualmente è visualizzato il messaggio:

LIBR. lib-name OPEN ON UNIT unit-name

per avvertire l'utente che la libreria con nome simbolico lib-name è stata aperta dal sistema.

Note

1. Se si sostituisce un disco sistema ad un floppy disk sistema, il nuovo disco o floppy disk deve contenere la stessa release di sistema operativo e le stesse caratteristiche e funzioni (selezionate durante la generazione di sistema).
2. Il comando deve essere usato ogni qual volta si vuole sostituire un disco od un floppy disk con un altro oppure inserire un secondo disco, mentre la macchina è accesa.
3. Se si apre lo sportello di un trascinatore di una unità a disco, senza prima aver introdotto il comando DCHANGE, il sistema visualizza il messaggio: ABN unit-name - DCH OMITTED, quando esegue una operazione di accesso al disco presente nel relativo trascinatore. L'esecuzione dell'eventuale programma sarà terminata.

Esempi

1. Si introduca un floppy disk in sostituzione di un altro già presente nell'unità con nome simbolico FDU1, il cui spazio libero sia insufficiente a contenere un programma presente in memoria principale.

Premere

Introdurre il disco e quindi premere .

2. Si sostituisca il disco presente nell'unità con nome simbolico UD, senza spegnere il P6066.

Premere

Dopo aver sostituito il disco, si preme .

Comando DECOMPILE

Funzione

Converte un programma BASIC presente in memoria principale in un file testo.

Formato

DEC [OMPILE]

Azione

Il comando comunica al sistema di convertire il programma presente in memoria principale in un file testo.

Note

1. Non si possono convertire in file testo i programmi che sono stati protetti mediante il comando SECURE.
2. Le linee del file testo hanno un formato diverso dalle linee introdotte durante la generazione del programma, poichè il sistema esegue operazioni di editing sulle linee durante l'esecuzione del comando DECOMPILE.
3. Il comando DECOMPILE è utile nel caso in cui in un programma sono state cancellate alcune variabili o richiami di funzione, o riferimenti a linee di programma (es. istruzioni GOTO), durante la fase di editing, perchè tali riferimenti permangono allo interno del sistema ma sono definitivamente cancellati se il programma è decompilato mediante il comando DECOMPILE e poi compilato con il comando COMPILE.





Comando DELETE LINE

Funzione

Cancella una o più linee di programma o di testo presenti in memoria principale.

Formato

DEL [DELETE LINE][line-num₁[, line-num₂]]

dove:

line-num₁

indica il numero della linea da cancellare o la prima linea di un insieme di linee da cancellare

line-num₂

indica l'ultima linea di un insieme di linee da cancellare.

Azione

Il comando, completo di tutti gli operandi, comunica al sistema di cancellare le linee del programma o del file testo presente in memoria principale comprese tra i numeri di linea indicati con il primo e secondo operando (estremi inclusi).

Il comando, privo del secondo operando, comunica al sistema di cancellare la linea specificata con il primo operando.

Il comando, privo di operandi, comunica al sistema di cancellare una delle seguenti linee:

- l'ultima linea corretta introdotta da tastiera
- l'ultima linea visualizzata con il comando FETCH od i tasti e
- l'ultima linea stampata mediante un comando LIST
- l'ultima linea di un programma eseguito mediante un comando RUN

Note

1. Non si possono cancellare le linee di un programma protette con il comando SECURE.
2. La parola chiave DELETE LINE può essere introdotta premendo il tasto **DELETE LINE** insieme con **SHIFT**.
3. Se durante la fase di editing di un programma BASIC si cancellano delle variabili, dei richiami di funzioni definite dall'utente o dei richiami ad altre istruzioni di programma, tali riferimenti permangono all'interno del sistema; per cancellarli definitivamente si decompili il programma con il comando DECOMPILE e poi lo si compili con il comando COMPILE.

Esempi

1. Si cancellino le istruzioni di un programma, presente in memoria principale, dal numero di linea 50 al numero di linea 150.

Premere **D** **E** **L** **5** **0** **.** **1** **5** **0** **END OF LINE**

2. Si cancelli la linea 100 del file testo presente in memoria principale.

Premere **SHIFT** **DELETE LINE** **1** **0** **0** **END OF LINE**

Comando ENVIRONMENT

Funzione

Stampa le informazioni che definiscono la configurazione di sistema attualmente operante.

Formato

ENV [IRONMENT]

Azione

Il comando comunica al sistema di stampare le seguenti informazioni:

- codice della release di sistema operativo residente su disco sistema o floppy disk sistema
- data
- capacità della memoria principale
- governi di I/O presenti nel P6066
- situazione dichiarata dall'ultimo comando CONFIGURE eseguito
- opzioni richieste dall'ultimo comando OPTIONS eseguito
- nome simbolico delle unità a disco e floppy disk, nell'ordine NAME (nome), PERIPH CODE (codice interno dell'unità) e PERIPH TYPE (tipo di unità)
- su quale disco risiede il sistema operativo
- elenco delle librerie aperte quando è eseguito il comando
- elenco delle librerie che sono aperte ogni qual volta si inizializza il sistema

Esempio

Si richieda la stampa delle informazioni che definiscono la configurazione di sistema presente.

Premere **E** **N** **V** **END OF LINE**

```
K06066-R 1.0 *      SYSTEM ENVIRONMENT      * DATE:10-01-80

MEMORY PHYSICAL SIZE : 48 K
I/O INTERFACES : IPS0-1
                  DCU

MEMORY LOGICAL SIZE: 24 K
EXTERNAL PRINTER : NO
EXTERNAL VIDEO DISPLAY : NO
SOFTWARE OPTIONS : STRING, MATR, EXT INTPT

DISK UNITS:

NAME      PERIPH CODE      PERIPH TYPE
LD        A0              DCU
UD        A1              DCU
FDU1      C0              FDU
FDU2      C1              FDU

OPERATING SYSTEM ON DISK UNIT LD

OPEN LIBRARIES :
NAME      DISK UNIT
TLX       LD
PROU      UD
P&FSYS    FDU2
INSTAL    FDU1

STORED LIBRARIES :
NAME      DISK UNIT
TLX       LD
PROU      UD
```

Si noti che la capacità di memoria utente è stata ridotta da 48K a 24K eseguendo un comando CONFIGURE con MS = 24.

Comando EXEC

Funzione

Carica in memoria ed esegue un programma di utilità.

Formato

EXE [C] utility [, parameter [, parameter]...]

dove:

utility

è il nome di un programma di utilità

parameter

specifica un operando associato con il programma di utilità specificato.

Azione

Il comando comunica al sistema di caricare in memoria principale ed eseguire il programma di utilità di nome utility.

Nota

1. Nella libreria di software di base sono disponibili i seguenti programmi utilità:

DCOPY
DINIT
FLCOPY
FLPRINT
LBCREATE
LBEMPTY
LBPROTECT
LBRENAME
LBSCRATCH
LIBCOPY
RESTRUCT
VOLLABEL

Per ulteriori informazioni si veda l'appendice A.

Comando FETCH

Funzione

Trasferisce nel buffer di tastiera una linea di programma o di file testo presente in memoria principale.

Formato

FET [CH] [line-num]

dove:

line-num

è il numero di linea della linea da trasferire nel buffer di tastiera.

Azione

Il comando comunica al sistema di trasferire nel buffer di tastiera e visualizzare sul display la linea del programma o file testo presente in memoria principale con numero line-num.

Il comando senza operandi comunica al sistema di trasferire nel buffer di tastiera e di visualizzare sul display una delle seguenti linee:

- l'ultima linea corretta introdotta da tastiera
- l'ultima linea di un file caricato in memoria principale mediante il comando OLD
- l'ultima linea stampata con il comando LIST
- l'ultima istruzione di un programma appena eseguito ed ancora presente in memoria principale
- l'ultima linea richiamata nel buffer di tastiera con il comando FETCH o con i tasti e .

Note

1. Se si introduce il comando FETCH con un operando line-num il cui valore non è presente in memoria principale, la linea con il numero di linea im-

diatamente inferiore a quella specificata viene trasferita nel buffer di tastiera. Se line-num è inferiore al più piccolo numero di linea presente in memoria principale, nel buffer di tastiera viene trasferita la linea con il più grande numero di linea.

2. La linea visualizzata sul display non ha lo stesso formato di quella introdotta da tastiera, perchè il sistema la modifica prima di trasferirla nel buffer di tastiera; così, ad esempio, quando si introduce la istruzione: 10A=B il comando FETCH la visualizzerà sul display come: 0010 LET A=B. Se l'editing effettuato dal sistema produce una linea con più di 80 caratteri, la linea viene compattata eliminando tutti gli spazi.
3. Se l'eliminazione degli spazi non significativi non riporta la linea ad 80 caratteri viene visualizzato un messaggio di errore e la linea è stampata sulla stampante integrata. Premendo il tasto di console RECALL la linea appare sul display (nel buffer di tastiera è troncata ai primi 80 caratteri) e si può editare; quando è introdotta nuovamente, premendo il tasto END OF LINE, sostituisce la linea precedente che era presente, integralmente, nel working-file.
4. Non si può usare il comando FETCH riferito a linee di un programma protette mediante il comando SECURE.
5. La parola chiave può essere introdotta premendo il tasto  con .

Esempi

1. Trasferire nel buffer di tastiera la linea 50 di un programma presente in memoria principale.

Premere      

2. Trasferire nel buffer di tastiera l'ultima linea di un file testo caricato in memoria principale con il comando OLD.

Premere   

Comando LBCLOSE

Funzione

Chiude l'accesso ad una o più librerie.

Formato

LBC [LOSE] { lib-ref }

dove:

lib-ref

può essere uno dei seguenti operandi:

(lib-name, unit-name)

lib-name

(,unit-name)

con lib-name che specifica il nome di una libreria e unit-name che specifica il nome di una unità a disco od a floppy disk

*

indica che devono essere chiuse tutte le librerie attualmente aperte.

Azione

Il comando comunica al sistema di chiudere la libreria specificata con lib-name, residente nel disco o floppy disk presente nell'unità con nome simbolico unit-name.

Il comando con lib-ref = (,unit-name), comunica al sistema di chiudere tutte le librerie aperte residenti nell'unità con nome simbolico unit-name.

Il comando con lib-ref = lib-name, comunica al sistema di chiudere la prima libreria aperta di nome lib-name.

Il comando con * come operando comunica al sistema di chiudere tutte le librerie attualmente aperte.

Nota

Eseguendo il comando LVTOC con l'operando * si possono stampare i nomi delle librerie attualmente aperte.

Esempio

Si chiuda la libreria MAT residente nella unità con nome simbolico LD.

Premere (L) (B) (C) ((M) (A) (T) (,) (L) (D)) (END OF LINE)

Comando LBOPEN

Funzione

Aprire l'accesso ad una libreria specificata.

Formato

LBO [PEN] lib-ref [, password]

dove:

lib-ref

può essere uno dei seguenti operandi:

(lib-name, unit-name)

lib-name

(,unit-name)

con lib-name che specifica il nome di una libreria e unit-name che specifica il nome di una unità a disco od a floppy disk

password

è una stringa di al massimo 6 caratteri uguale alla stringa password specificata quando è stata creata la libreria con il programma di utilità LBCREATE o uguale alla parola d'ordine di sistema

Azione

Il comando comunica al sistema di aprire la libreria specificata con lib-name, residente nell'unità con nome simbolico unit-name.

Se come lib-ref è specificato lib-name, viene aperta la libreria di nome lib-name residente su disco sistema o su floppy disk sistema.

Se come lib-ref è specificato (unit-name), viene aperta la prima libreria residente nell'unità con nome simbolico unit-name.

Note

1. Se la libreria è stata creata associando ad essa una parola d'ordine, questa deve essere specificata quando si introduce il comando.
2. Eseguendo il comando VTOC con l'operando * si possono stampare i nomi delle librerie attualmente aperte.

Esempio

Si apra la libreria MAT residente nell'unità UD avente &PASS come parola d'ordine.

Premere

L B O (M A T , U D) . & P A S S END
OF
LINE

Comando LBRESTORE

Funzione

Riapre le librerie specificate nell'ultimo comando LBSTORE eseguito e chiude quelle attualmente aperte.

Formato

LB[ESTORE]

Azione

Il comando ordina al sistema di chiudere le librerie attualmente aperte e di aprire, successivamente, le librerie specificate nell'ultimo comando LBSTORE eseguito.

Note

1. Se il comando LBSTORE non è mai stato eseguito, l'esecuzione del comando LBRESTORE apre la prima libreria residente sul disco sistema o la libreria residente su floppy disk sistema, se questa è stata creata senza assegnarle una parola d'ordine.
2. Per sapere quali librerie sono aperte dopo che è stato eseguito il comando LBRESTORE, si esegua il comando LVTOC con l'operando *.

Comando LBSTORE

Funzione

Memorizza le librerie da aprire ogni volta che il sistema P6066 sarà inizializzato.

Formato

LBS [TORE]

Azione

Il comando LBSTORE richiede al sistema di riaprire automaticamente tutte le librerie attualmente aperte ogni qual volta il sistema P6066 è inizializzato (accensione, esecuzione del comando OPTIONS o del comando CONFIGURE). Dopo che il comando è eseguito, il sistema stampa il nome delle librerie che devono essere aperte, nell'ordine con cui saranno aperte.

Note

1. Se si esegue LBSTORE senza che vi siano librerie aperte, o se non si esegue mai LBSTORE, ad una successiva inizializzazione del sistema P6066 non viene aperta alcuna libreria e si ha una segnalazione di errore.
2. Il comando LVTOC permette di conoscere quali sono le librerie che vengono aperte ogni qual volta il sistema è inizializzato e con quale ordine sono aperte.
3. Se una libreria protetta da una parola d'ordine è aperta quando si esegue un comando LBSTORE, quella libreria sarà poi aperta automaticamente senza bisogno della parola d'ordine. Per evitare che questo succeda, si deve per prima cosa chiudere la libreria per mezzo del comando LBCLOSE e quindi inviare il comando LBSTORE.

Esempio

Si esegua il comando LBSTORE.

Premere **L** **B** **S** **END OF LINE**

LBS NAME	PERIPHERAL
LIBR3	LD
INSTAL	LD
LIB1	UD
MAT	LF

Come si vede viene stampato il nome delle librerie da aprire ogni qual volta il sistema è inizializzato. L'ordine con cui sono stampati i nomi coincide con l'ordine con cui le relative librerie saranno aperte.

Comando LDKEYS

Funzione

Riassegna ai tasti funzione il contenuto assegnato precedentemente ed attualmente memorizzato sul disco sistema o su floppy disk sistema.

Formato

LDK [EYS]

Azione

Il comando comunica al sistema di riassegnare ad ogni tasto funzione la stringa di caratteri che ad esso era già stata assegnata in precedenza ed era stata successivamente registrata sul disco sistema o sul floppy disk sistema da un comando STKEYS.

Note

1. Il comando è usato perchè il contenuto dei tasti funzione può essere modificato da programma, mediante l'istruzione FKEY #, oppure durante gli stati esecuzione calcoli immediati e debugging mediante FKEY # (vedi capitoli 6 e 7).
2. Il contenuto presente su disco sistema o su floppy disk sistema è riassegnato ai tasti funzione ogni volta che il sistema è inizializzato, quindi: (1) quando il sistema è acceso, (2) quando si esegue il comando OPTIONS e (3) quando si esegue il comando CONFIGURE.

Esempio

Si registri sul disco sistema o sul floppy disk sistema un contenuto per i tasti funzione F1, F2, F3. Dopo aver assegnato ai medesimi tasti funzione un contenuto diverso, si ripristini il precedente.

Premere 

Il sistema è nello stato esecuzione di calcoli immediati.

Premere **SHIFT** **FKEY#** **1** **R** **E** **P** **L** **A** **C** **E** **END OF LINE**

SHIFT **FKEY#** **2** **P** **U** **R** **G** **E** **END OF LINE**

SHIFT **FKEY#** **3** **O** **L** **I** **V** **E** **T** **T** **I** **P** **6** **0** **6** **6** **END OF LINE**

ai tasti funzione F1, F2, F3 sono assegnati i caratteri a destra della virgola.

Premere **CALC MODE**

Il sistema è nello stato comandi.

Premere **S** **T** **K** **END OF LINE**

La suddetta associazione fra tasti funzione e relativo contenuto è registrata sul disco sistema o sul floppy disk sistema

Premere **CALC MODE**

Premere **SHIFT** **FKEY#** **1** **R** **U** **N** **D** **E** **M** **O** **END OF LINE**

SHIFT **FKEY#** **2** **R** **E** **S** **E** **Q** **U** **E** **N** **C** **E** **END OF LINE**

SHIFT **FKEY#** **3** **L** **I** **N** **K** **END OF LINE**

Si modifica il contenuto dei tasti funzione F1, F2, F3.

Premere **CALC MODE**

Il sistema è di nuovo nello stato comandi.

Premendo **F1** sul display si vede: RUN DEMO

Premendo **F2** sul display si vede: RESEQUENCE

Premendo **F3** sul display si vede: LINK

Premere **L** **D** **K** **END OF LINE**

Premendo **F1** sul display si vede: REPLACE

Premendo **F2** sul display si vede: PURGE

Premendo **F3** sul display si vede: OLIVETTI P6066

Come si vede è stato ripristinato il contenuto dei tasti funzione che era stato registrato sul disco sistema o sul floppy disk sistema.



Comando LINK

Funzione

Inserisce un insieme di istruzioni BASIC od una definizione di funzione, registrati su una libreria come file testo, in un programma presente in memoria principale.

Formato

LIN[K] filename, [lib-ref], line-num [,a]

dove:

filename

è il nome di un file testo contenente linee di programma BASIC o una definizione di funzione

lib-ref

può essere uno dei seguenti operandi:

(lib-name, unit-name)

lib-name

(,unit-name)

con lib-name che specifica il nome di una libreria e unit-name che specifica il nome di una unità a disco od a floppy disk

line-num

indica il numero di linea che sarà assegnato alla prima linea del testo all'atto dell'inserimento nel programma presente in memoria principale

a

è una lettera maiuscola che completa il nome con cui sarà chiamata la definizione di funzione inserita nel programma

Azione

Il comando, con l'operando *a*, ordina al sistema di aggiungere al programma presente in memoria principale la definizione di funzione registrata nella libreria lib-name, residente nell'unità con nome simbolico unit-name, come file testo col nome filename. Il primo numero di linea della definizione di funzione inserita nel programma dal comando LINK è uguale al

valore specificato con line-num; i numeri di linea successivi mantengono l'incremento che essi avevano nel file testo presente nella libreria suddetta. Il sistema assegna alla definizione di funzione il nome FNa o, nel caso di una definizione di funzione di tipo stringa, $FNa\$$. Al termine della esecuzione del comando LINK, in memoria principale si ha un nuovo programma.

Il comando, senza l'operando α , ordina al sistema di inserire nel programma presente in memoria principale l'insieme di istruzioni BASIC registrato come file testo di nome filename nella libreria lib-name, residente nell'unità con nome simbolico unit-name. Le istruzioni inserite sono rinumerate a partire dal numero di linea line-num, mantenendo fra un numero di linea ed il successivo lo stesso incremento già presente nel file file-name. Al termine della esecuzione del comando, in memoria principale si ha un nuovo programma.

Note

1. Il numero di linea specificato con line-num non deve essere già presente in memoria principale.
2. Se il numero di linea più alto del file da inserire in memoria principale è n e l'operando line-num ha valore m , allora in memoria principale non devono già essere presenti numeri di linea compresi tra m ed $m+n$.
3. Il comando LINK non può essere eseguito se il programma presente in memoria principale è stato protetto mediante il comando SECURE.
4. E' bene verificare che dopo l'esecuzione del comando LINK il programma presente in memoria principale abbia una ed una sola istruzione END con il più alto numero di linea.
5. Se come operando lib-ref è specificato lib-name, il sottoprogramma o la definizione di funzione, di nome filename, è prelevato dalla prima libreria aperta di nome lib-name.
6. Se come operando lib-ref è specificato $(,unit-name)$, il sottoprogramma o la definizione di funzione, di nome filename, è prelevato dalla prima libreria aperta residente nell'unità con nome simbolico

unit-name.

7. Se l'operando lib-ref non è specificato, il sottoprogramma o la definizione di funzione, di nome filename, viene ricercato nelle librerie attualmente aperte secondo l'ordine di apertura.

Esempi

1. Si aggiunga ad un programma presente in memoria principale (iniziando da line-num = 100) una definizione di funzione numerica MAT1 registrata, come file testo, nella libreria MAT residente nell'unità con nome simbolico UD. Alla definizione di funzione si assegni il nome FNR.

Premere

```
{ LIN MAT1.(MAT.UD).  
  100.FNR END  
  OF  
  LINE
```

2. Inserire un sottoprogramma, registrato come file testo di nome FISICA nella prima libreria aperta di nome SCUOLA, in un programma presente in memoria principale iniziando da line-num uguale a 600.

Premere

```
{ LIN FISICA.SCUOLA.  
  600 END  
  OF  
  LINE
```


Con l'operando X specificato, il comando ordina al sistema di stampare le linee specificate del file testo presente in memoria principale, senza il numero di linea.

Note

1. Poichè le istruzioni di un programma sono editate durante la stampa eseguita con il comando LIST, le linee specificate nel comando possono essere stampate con un formato diverso da quello con cui sono state introdotte. Per esempio, se si introduce 40A=B, il sistema stampa ØØ40 LET A=B. Se tale editing produce una linea con più di 80 caratteri, allora la linea viene compattata eliminando gli spazi.
2. Lo stesso editing è attuato sulle linee di testo di un file testo prodotto mediante un comando DECOMPILE partendo da un programma.
3. Il comando LIST non stampa le linee di un programma protette mediante il comando SECURE.
4. La parola chiave del comando LIST può essere introdotta premendo il tasto **LIST**.
5. Il comando LIST può visualizzare le linee di un programma o di un file testo su un video Olivetti DSM 6680 o su un altro video se si è eseguito un comando CONFIGURE specificando l'operando EXD o EVD.
6. Se la configurazione di sistema installata è priva di stampante e di unità video display, le linee di programma o di file testo sono visualizzate sul display. In questo caso, per leggere una linea completa, si devono utilizzare i tasti **→**, **REPEAT** e **SHIFT** come spiegato nel cap. 1, § "Tastiera". Per leggere ogni linea si deve premere il tasto **CONTINUE**: ad ogni pressione appare sul display una diversa linea e quando, alla pressione di **CONTINUE**, il sistema produce un segnale acustico, significa che non vi sono più linee da visualizzare.

Esempi

1. Si richieda la stampa delle linee di un file testo presente in memoria principale. Non si stampino i numeri di linea.

Premere

2. Si richieda la stampa di una istruzione di programma presente in memoria principale con il numero di linea 45.

Premere

Comando LVTOC

Funzione

Stampa alcune informazioni relative alle librerie presenti su una unità a disco (od a floppy disk) o l'elenco delle librerie aperte attualmente e quello delle librerie aperte ogni qual volta il sistema è inizializzato.

Formato

LVTOC [**unit-name** *]

dove:

unit-name

è il nome simbolico associato ad una unità a disco od a floppy disk

*

indica di stampare l'elenco delle librerie attualmente aperte e di quelle memorizzate come librerie da aprire a ogni inizializzazione del sistema default

vengono stampate alcune informazioni relative alle librerie residenti su disco sistema o su floppy disk sistema.

Azione

Il comando ordina al sistema di stampare le seguenti informazioni, relative ad ogni libreria residente sul disco o floppy disk nell'unità con nome simbolico unit-name:

- nome della Release del sistema operativo, con l'identificatore di numero di Release e di livello, ad esempio: KØ 6066-R 1.0 dove 1.Ø identifica la prima distribuzione di livello Ø
- etichetta di identificazione del supporto fisico (volume), stampata come VOLLABEL = XXXXXX dove XXXXXX è l'etichetta (vedi in appendice A il programma di utilità VOLLABEL)

- il formato della traccia, ossia il numero di byte che compongono ogni settore segnalato come: TRACK FORMAT = nnn BYTE, dove nnn è 256 nel caso di disco e 128 nel caso di floppy disk
- la data registrata con il comando di sistema DATE, stampata accanto alla stringa DATE:
- per ogni libreria è stampato nell'ordine: il nome della libreria, la data di creazione, la posizione di inizio della libreria sul disco (nel formato TTT SSS, con TTT numero di traccia ed SSS numero di settore), la posizione di fine della libreria sul disco nel formato TTT SSS), il numero di settori che sono allocati per contenere i file della libreria, il numero di settori che sono ancora liberi per la registrazione di altri file nella libreria
- il numero di settori che sono liberi per la creazione di nuove librerie, utilizzabili dopo aver eseguito il programma di utilità RESTRUCT che riordina lo spazio su disco dedicato alle librerie, nel formato FREE SPACE (TOTAL) = nnnnnn SECTORS
- il numero di settori che sono liberi per la creazione di nuove librerie senza riorganizzare la struttura delle informazioni sul disco, nel formato MAXIMUM BLOCK OF FREE SPACE = nnnnnn SECTORS.

Il comando LVTOC senza operandi ordina al sistema di stampare le suddette informazioni per le librerie residenti sul disco sistema o sul floppy disk sistema.

Se il comando è specificato con * come operando, è stampato l'elenco delle librerie attualmente aperte, nell'ordine di apertura, e l'elenco delle librerie memorizzate come librerie da aprire a ogni inizializzazione del sistema, nell'ordine di apertura. Accanto al nome delle suddette librerie è stampato il nome simbolico dell'unità in cui risiede la libreria.

Esempi

1. Si stampino le informazioni relative alle librerie residenti sul disco nell'unità con nome simbolico UD.

Premere L V T U D END
OF
LINE

Il sistema stampa:

```
K05055-R 1.0 * VOLLABEL=UPPER * TRACK FORMAT=256 BYTE * DATE:10-01-80

LIBRARY CREAT  BEG OF EXT  END OF EXT  SECTORS  EMPTY SECTORS
LIB3    02-11-77  001000     001039     40        32
LIB1    02-11-77  001040     002031     40        27
FREE SPACE (TOTAL) = 19072 SECTORS
MAXIMUM BLOCK OF FREE SPACE = 19072 SECTORS
```

2. Si stampi l'elenco delle librerie attualmente aperte e di quelle aperte quando il sistema è inizializzato.

Premere

Il sistema stampa:

```
OPEN LIBRARIES

NAME          DISK UNIT
LIBR3         LD
INSTAL        LD
LIB1          UD
MAT           LF

STORED LIBRARIES

NAME          DISK UNIT
LIBR3         LD
INSTAL        LD
LIB1          UD
MAT           LF
```

(

(

(

(

(

(

(

(

Comando MERGE

Funzione

Effettua la fusione di un file testo in memoria principale con uno precedentemente registrato in una libreria.

Formato

MER[GE] filename, [lib-ref], [line-num], [/]

dove:

filename

specifica il nome di un file testo presente in una libreria

lib-ref

può essere uno dei seguenti 3 operandi:

(lib-name, unit-name)

lib-name

(,unit name)

dove lib-name specifica il nome di una libreria aperta e unit-name specifica il nome simbolico di un disco o di una unità floppy disk

line-num

è il numero di linea da assegnare alla prima linea del filename

/

richiede al sistema di non fornire una stampa di tutte le linee sostituite.

Azione

Il comando MERGE, se specificato con tutti gli operandi, effettua la rinumerazione del file filename presente nella libreria specificata iniziando dal numero di linea line-num e mantenendo l'incremento originale. Successivamente il file testo filename viene aggiunto al testo che è in memoria principale. Se una linea del file filename e una linea del testo in memoria hanno lo stesso numero, la linea nuova sostituirà quella vecchia. Il risultato sarà un file testo aggiornato e modificato in memoria principale.

Se MERGE è specificato senza il quarto operando, /, il sistema fornirà la stampa di ogni linea sostituita insieme alla linea che la sostituisce. Il listing avrà il seguente formato:

```
#### PREVIOUS LINE (linea precedente)
#### PRESENT LINE (linea attuale)
```

dove #### è un numero di linea.

Se MERGE non specifica l'operando line-num, il file filename non viene rinumerato.

Note

1. Se viene specificato (lib-name, unit-name) come operando di lib-ref, il file testo richiesto sarà quello nella libreria specificata sull'unità identificata da unit-name.
2. Se viene specificato soltanto il nome di una libreria come operando di lib-ref, la ricerca del file testo richiesto incomincia dalla prima libreria aperta che ha il nome specificato.
3. Se viene specificato (,unit-name) come operando di lib-ref, la ricerca del file testo richiesto inizia dalla prima libreria aperta sull'unità identificata da unit-name.
4. Se viene omissso lib-ref, la ricerca del file testo richiesto inizia dalla prima libreria aperta e continua nell'ordine in cui sono state aperte le librerie.
5. Il file testo filename è rinumerato solamente in memoria principale mentre rimane invariato nella libreria.

Esempi

1. Un file testo TEX1, registrato sull'unità FDU1 nella libreria P6FSYS, viene fuso ad un file testo che è in memoria principale con il nome TEX2. Per chiarire meglio l'operazione di fusione, i due file testo sono stati listati per prima cosa. Il comando MERGE richiede al sistema di rinumerare il file TEX1 prima di fare la fusione (iniziando dal numero di linea 85). Si vede il risultato della fusione nel listing che segue il comando inviato.

OLD TEX1

LIST
FILE TEX1

0010 TEXT 1
0020 TEXT 1
0030 TEXT 1
0040 TEXT 1
0050 TEXT 1
0060 TEXT 1
0070 TEXT 1
0080 TEXT 1
0090 TEXT 1
0100 TEXT 1

END OF LISTING

OLD TEX2
LIST
FILE TEX2

0010 TEXT 2
0020 TEXT 2
0030 TEXT 2
0040 TEXT 2
0050 TEXT 2
0060 TEXT 2
0070 TEXT 2
0080 TEXT 2
0090 TEXT 2
0100 TEXT 2
0110 TEXT 2
0120 TEXT 2
0130 TEXT 2

END OF LISTING

REP TEX1. (P9FSYS.FDU10.85

LIST
FILE TEX2

0010 TEXT 2
0020 TEXT 2
0030 TEXT 2
0040 TEXT 2
0050 TEXT 2
0060 TEXT 2
0070 TEXT 2
0080 TEXT 2
0095 TEXT 1
0099 TEXT 2
0095 TEXT 1
0100 TEXT 2
0105 TEXT 1
0110 TEXT 2
0115 TEXT 1
0120 TEXT 2
0125 TEXT 1
0130 TEXT 2
0135 TEXT 1
0145 TEXT 1
0155 TEXT 1
0165 TEXT 1
0175 TEXT 1

END OF LISTING

2. In questo esempio si vede che al file testo TEX1 precedente è assegnato come numero di linea della prima linea 70. Durante la fusione, le linee da 70 a 130 del file TEX2 vengono sostituite dalle linee del file TEX1 con lo stesso numero di linea. Avendo specificato l'operando /, il sistema non ci fornirà un listing che mostra la sostituzione.

```
OLD TEX2

LIST
FILE      TEX2

0010 TEXT 2
0020 TEXT 2
0030 TEXT 2
0040 TEXT 2
0050 TEXT 2
0060 TEXT 2
0070 TEXT 2
0080 TEXT 2
0090 TEXT 2
0100 TEXT 2
0110 TEXT 2
0120 TEXT 2
0130 TEXT 2

END OF LISTING

MER TEX1, (P6FSYS,F0U1),70,/

LIST
FILE      TEX2

0010 TEXT 2
0020 TEXT 2
0030 TEXT 2
0040 TEXT 2
0050 TEXT 2
0060 TEXT 2
0070 TEXT 1
0080 TEXT 1
0090 TEXT 1
0100 TEXT 1
0110 TEXT 1
0120 TEXT 1
0130 TEXT 1
0140 TEXT 1
0150 TEXT 1
0160 TEXT 1

END OF LISTING
```

3. Questo esempio fa vedere cosa succede se viene omissso l'operando /. Ci verrà dato un listing di tutte le linee sostituite.

```
OLD TEX2

LIST
FILE      TEX2

0010 TEXT 2
0020 TEXT 2
0030 TEXT 2
0040 TEXT 2
0050 TEXT 2
0060 TEXT 2
0070 TEXT 2
0080 TEXT 2
0090 TEXT 2
0100 TEXT 2
0110 TEXT 2
0120 TEXT 2
0130 TEXT 2

END OF LISTING

MER TEX1, (P6FSYS.FDU1), 50

REPLACED LINES:
#### PREVIOUS LINE
#### PRESENT LINE

0050 TEXT 2
0050 TEXT 1

0060 TEXT 2
0060 TEXT 1

0070 TEXT 2
0070 TEXT 1

0080 TEXT 2
0080 TEXT 1

0090 TEXT 2
0090 TEXT 1

0100 TEXT 2
0100 TEXT 1

0110 TEXT 2
0110 TEXT 1

0120 TEXT 2
0120 TEXT 1

0130 TEXT 2
0130 TEXT 1

LIST
FILE      TEX2

0010 TEXT 2
0020 TEXT 2
0030 TEXT 2
0040 TEXT 2
0050 TEXT 1
0060 TEXT 1
0070 TEXT 1
0080 TEXT 1
0090 TEXT 1
0100 TEXT 1
0110 TEXT 1
0120 TEXT 1
0130 TEXT 1
0140 TEXT 1

END OF LISTING
```


Comando MODIFY

Funzione

Modifica il nome di un file e/o il numero di byte riservati in una libreria ad un file dati.

Formato

MOD [IFY] old-filename, [lib-ref], {new-filename [, n]}

old-filename

indica il nome di un file programma, testo o dati

lib-ref

può essere uno dei seguenti operandi:

(lib-name, unit-name)

lib-name

(,unit-name)

con lib-name che specifica il nome di una libreria e unit-name che specifica il nome di una unità a disco od a floppy disk

new-filename

indica il nuovo nome da dare al file programma, testo o dati

n

è un numero che indica il numero di byte da riallocare nella libreria per il file specificato

Azione

Il comando, completo di tutti gli operandi, ordina al sistema di assegnare al file dati registrato con il nome old-filename nella libreria lib-name, sul disco contenuto nella unità con nome simbolico unit-name, il nome new-filename e di riservare ad esso n byte nella libreria lib-name.

Il comando, senza l'operando n, ordina al sistema di sostituire al file programma, testo o dati registrato nella libreria lib-name con il nome old-filename il nome new-filename.

Il comando, con n come terzo operando e senza l'operando new-filename, ordina al sistema di riservare n byte al file dati registrato nella libreria lib-name

con il nome old-filename.

Note

1. L'operando new-filename può essere costituito solamente da un massimo di 6 caratteri alfanumerici di cui il primo alfabetico.
2. In una stessa libreria non possono esistere file con lo stesso nome.
3. Non si può modificare il nome di un file della sottolibreria package o della sottolibreria comune dopo che sono state protette con il programma di utilità LBPROTECT.
4. Si possono modificare i nomi dei file di una sottolibreria package o comune prima che siano state protette, specificando come new-filename una stringa di al massimo 6 caratteri alfanumerici di cui il primo alfabetico. Il sistema aggiungerà automaticamente * o +.
5. Se n non è multiplo di 256, nel caso di disco o di 128, nel caso di floppy disk, il sistema rialloca per il file dati specificato un numero di byte pari al successivo multiplo di 256 o di 128.
6. Per un file dati sequenziale n può essere minore della dimensione di allocazione specificata in precedenza, ma non può essere minore della sua dimensione attuale.
7. Per un file ad accesso diretto n non può essere minore della sua dimensione di allocazione.
8. La libreria contenente il file old-filename deve essere aperta.
9. Se come lib-ref è specificato solamente lib-name, il file interessato alla modifica è cercato nella prima libreria aperta, in ordine di apertura, di nome lib-name.
10. Se come lib-ref è specificato (,unit-name), il file interessato alla modifica è cercato nella prima libreria aperta, in ordine di apertura, sull'unità con nome simbolico unit-name.
11. Se l'operando lib-ref non è specificato, il file

interessato alla modifica è cercato nelle librerie aperte secondo l'ordine di apertura.

12. Il comando MODIFY non può essere usato con programmi o file dati protetti mediante il comando SECURE.

Esempi

1. Modificare il nome del file dati NUM1 in NUM2 e riallocare 1735 byte per esso sul floppy disk con nome simbolico FDU1.

Premere

```
{ M O D N U M 1 , ( , F D U 1 ) ,  
  N U M 2 , 1 7 3 5 } END  
                           OF  
                           LINE
```

2. Modificare lo spazio riservato al file dati STAT, nella libreria MATEM residente sul disco nell'unità con nome simbolico UD, da 4096 byte a 2560 byte.

Premere

```
{ M O D S T A T , ( M A T E M ,  
  U D ) , , 2 5 6 0 } END  
                           OF  
                           LINE
```


Comando NEW

Funzione

Predisporre il sistema ad accettare l'introduzione di un programma da tastiera.

Formato

NEW

Azione

Il comando ordina al sistema di allocare spazio in memoria principale per permettere di introdurre un programma, linea per linea, da tastiera.

Note

1. L'esecuzione del comando NEW cancella il precedente contenuto della memoria utente; quindi, prima di introdurre NEW, si deve introdurre SAVE o REPLACE per registrare il programma o file testo, eventualmente presente in memoria principale, in una libreria.
2. Dopo l'inizializzazione o reinizializzazione (comando OPTIONS e CONFIGURE) del sistema, si può introdurre un programma da tastiera senza digitare il comando NEW.
3. Introducendo il comando AUTO# si può avere la numerazione automatica delle linee introdotte successivamente.

Esempio

Generare un nuovo programma da tastiera mentre in memoria principale esiste già un programma.

Premere       

così il programma è registrato col nome +MAT nella prima libreria (è inserito nella sottolibreria comune).

Premere  

Ora si può introdurre il nuovo programma, linea per linea.

Comando OLD

Funzione

Carica in memoria principale un programma o file testo registrato in una libreria.

Formato

OLD filename [,lib-ref]

filename

è il nome del programma o file testo da caricare in memoria principale

lib-ref

può essere uno dei seguenti operandi:

(lib-name, unit-name)

lib-name

(,unit-name)

con lib-name che specifica il nome di una libreria e unit-name che specifica il nome di una unità a disco od a floppy disk

Azione

Il comando ordina al sistema di caricare in memoria principale il programma o file testo filename, registrato nella libreria lib-name, residente sul disco o floppy disk della unità unit-name.

Note

1. La libreria specificata con lib-name deve essere aperta.
2. Se come operando lib-ref è specificato solamente lib-name il file filename è cercato nella prima libreria aperta, in ordine di apertura, di nome lib-name.
2. Se come operando lib-ref è specificato (,unit-name) il file filename è cercato nella prima libreria aperta, in ordine di apertura, residente sul disco nella unità con nome simbolico unit-name.

4. Se l'operando lib-ref non è specificato il file filename è cercato nelle librerie aperte secondo l'ordine con cui sono state aperte.
5. La parola chiave del comando può essere introdotta premendo i tasti **SHIFT** e **OLD** contemporaneamente.

Esempio

Caricare in memoria principale il programma +MAT, presente nella sottolibreria comune, della libreria SCUOLA, residente sul disco dell'unità con nome simbolico UD.

Premere

{ **SHIFT** **OLD** **+** **M** **A** **T** **,**
(**S** **C** **U** **O** **L** **A** **.** **U** **D** **)** **END OF LINE**

Comando OPTIONS

Funzione

Definisce le opzioni del sistema operativo che devono essere caricate in memoria principale ogni volta che il sistema è inizializzato.

Formato

OPT [IONS] [option [,option]...]

dove:

option

Può essere DEB, MAT, STR, RS232, XNT o alternativamente PLO o GDI; nessuna opzione può essere ripetuta e vi possono essere al massimo 6 operandi. Le opzioni possono essere specificate in qualsiasi ordine.

Azione

Il comando ordina al sistema di inicializzarsi caricando in memoria le routine del sistema operativo specificate dagli operandi.

L'operando DEB permette l'esecuzione del debugging di un modulo assembler richiamato da un programma BASIC: in particolare sono caricate in memoria utente le routine di sistema operativo relative che occupano 3,5K byte (K = 1024). Per ulteriori informazioni si veda la pubblicazione: "P6066 Ambiente di programmazione in Assembler," codice 3978380 T).

L'operando MAT permette di eseguire programmi contenenti istruzioni BASIC di calcolo matriciale: in particolare vengono caricate in memoria utente le routine di sistema operativo relative che occupano 1,5K byte.

L'operando STR permette di eseguire programmi contenenti istruzioni BASIC che impiegano le funzioni BLN\$ e REP\$: in particolare vengono caricate in memoria utente le routine del sistema operativo relative che occupano 1K byte.

L'operando RS232 permette di eseguire programmi contenenti istruzioni BASIC di input-output che indirizzano periferiche esterne collegate al sistema mediante interfaccia seriale (EIA RS232-C). In particolare vengono caricate in memoria utente le relative routine del sistema operativo che occupano 1 Kbyte.

L'operando XNT richiede che il sistema sia inizializzato in modo che l'utente possa gestire le interruzioni che arrivano da unità periferiche esterne. Le routine caricate per la funzione occupano 1 Kbyte di memoria utente.

L'operando PLO richiede che si possano eseguire programmi contenenti istruzioni BASIC che permettano di tracciare immagini utilizzando come plotter la stampante integrata. In particolare vengono caricate in memoria utente le routine del sistema operativo relative che occupano 2K byte.

L'operando GDI richiede che il sistema sia inizializzato in modo che si possa plottare su uno schermo video Olivetti DSM 6680 tramite un programma in BASIC. Le routine caricate per la funzione occupano 2,5K byte di memoria utente. Se viene caricata anche la routine per la funzione EXD (vedi CONFIGURE) lo spazio complessivo occupato sarà di 3K byte invece di 3,5K byte.

Il comando, senza operandi, comunica al sistema di cancellare dalla memoria utente le routine del sistema operativo corrispondenti alle opzioni eventualmente presenti.

Quando viene specificato solo con qualche operando, OPTIONS richiede al sistema che cancelli dalla memoria utente le routine precedentemente caricate e che carichi quelle specificate.

Note

1. L'insieme di opzioni richiesto da ogni comando OPTIONS è memorizzato su disco sistema o floppy disk sistema così che ogni volta che il sistema viene acceso ricarica le opzioni richieste dall'ultimo comando OPTIONS eseguito.
2. Quando si esegue il comando OPTIONS, il precedente contenuto nella memoria principale è cancellato. Se si vuole registrare il contenuto della memoria utente, prima di introdurre il comando OPTIONS, si

introduca il comando SAVE od il comando REPLACE.

3. Se è specificato PLO, non si deve specificare GDI e viceversa.
4. E' possibile introdurre un programma che tratta le matrici, che traccia grafici o impiega le periferiche seriali, anche se la configurazione del sistema non ne permette l'esecuzione. Tale programma può essere registrato in una libreria con il comando SAVE ed eseguito dopo aver introdotto il comando OPTIONS con le opzioni necessarie.
5. Per sapere quali opzioni sono attualmente attive si esegua il comando ENVIRONMENT.
6. Per ulteriori informazioni si vedano i manuali: "P6066 I/O con periferiche esterne" e "P6066 Prestazioni grafiche".

Esempi

1. Reinizializzare il sistema per poter eseguire programmi che elaborano matrici e che tracciano grafici con la stampante integrata.

Premere **O P T M A T I P L O** **END OF LINE**

2. Cancellare dalla memoria utente le routine opzionali del sistema operativo.

Premere **O P T** **END OF LINE**



Comando PREPARE

Funzione

Esegue la preesecuzione di un programma ed al termine pone il sistema nello stato di DEBUGGING.

Formato

PRE[PARE] [filename [,lib-ref]]

dove:

filename

è il nome di un programma registrato in una libreria

lib-ref

può essere uno dei seguenti operandi:

(lib-name, unit-name)

lib-name

(,unit-name)

con lib-name che specifica il nome di una libreria e unit-name che specifica il nome di una unità a disco od a floppy disk

Azione

Il comando ordina al sistema di caricare in memoria principale il programma con nome filename, registrato nella libreria lib-name residente sul disco o floppy disk nell'unità con nome simbolico unit-name, e di eseguirne la preesecuzione.

Il comando, privo di operando, ordina al sistema di eseguire la preesecuzione del programma presente in memoria principale.

La preesecuzione di un programma BASIC comporta l'esecuzione di diverse operazioni tra le quali:

1. Controllo della corretta compilazione delle funzioni multilinea.
2. Controllo della correttezza dei riferimenti ad al-

tre istruzioni contenuti nelle istruzioni dei cicli FOR/NEXT.

3. Controllo della correttezza dei riferimenti ad altre istruzioni contenuti nelle istruzioni delle funzioni multilinea.
4. Controllo della correttezza delle istruzioni di salto.
5. Controllo della posizione nel programma delle istruzioni FOR rispetto alle istruzioni NEXT e delle istruzioni DEF rispetto alle istruzioni FNEND.
6. Posizionamento del pointer del file dati interno sul primo dato.
7. Caricamento delle routine del sistema operativo necessarie per l'esecuzione del programma.
8. Allocazione di spazio per le variabili.
9. Allocazione di buffer per i file dati esterni riferiti nel programma.
10. Apertura dei file dati riferiti nel programma.
11. Allocazione di buffer per l'impiego di periferiche esterne.

Se dopo l'esecuzione del comando non è segnalato alcun errore, il programma presente in memoria è in formato eseguibile ed il sistema è nello stato di debugging. Viene emesso il messaggio: *** FORMALLY CORRECT PROGRAM *** che è stampato se PRINTALL è attivo. Viene inoltre stampato il messaggio:
:: ROOM = X ::
dove X è il numero di byte ancora disponibile in memoria utente per l'esecuzione del programma. Si osservi che durante l'esecuzione del programma il sistema utilizza questa parte di memoria utente per cui si possono avere segnalazioni di spazio di memoria insufficiente a seconda delle caratteristiche del programma utente stesso. Infine sul display compare il messaggio: PROGRAM nome-programma READY TO RUN. Premendo il tasto **BREAK** il sistema commuta nello stato comandi.

Se dopo l'esecuzione del comando PREPARE non è stato segnalato alcun errore, si può eseguire il programma presente in memoria principale premendo il tasto **CONTINUE** ; oppure premendo il tasto **STEP** si può eseguire una istruzione per volta. Prima di comandare l'esecuzione del programma si possono inizializzare le variabili con valori assegnati da tastiera.

Se durante la preesecuzione si rilevano degli errori, vengono stampati i relativi messaggi di errore. Il sistema è nello stato comandi e si possono effettuare le necessarie correzioni.

Note

1. Se come operando lib-ref è specificato lib-name, il programma di nome filename è cercato nella prima libreria aperta di nome lib-name.
2. Se come operando lib-ref è specificato (, unit-name), il programma di nome filename è cercato nella prima libreria aperta residente sul disco o floppy disk nella unità con nome simbolico unit-name.
3. Se l'operando lib-ref non è specificato il programma di nome filename è cercato nelle librerie aperte secondo l'ordine di apertura.
4. Quando un programma è stato preseguito mediante il comando PREPARE od un comando RUN (vedi RUN) una successiva esecuzione comandata con un comando RUN evita la fase di preesecuzione. Se il programma è registrato su disco o su floppy disk il sistema ricorda che è stato preseguito, per cui ogni successiva esecuzione comandata con il comando RUN evita la fase di preesecuzione. Se in un programma preseguito si eseguono azioni di editing, il programma deve essere nuovamente preseguito.
5. Per poter eseguire un comando PREPARE senza gli operandi si deve avere sulla prima libreria aperta uno spazio libero da registrazioni equivalente a quello occupato dal programma in memoria principale.
6. Se un programma è registrato in una libreria senza che sia prima preseguito sarà preseguito ogni qualvolta se ne comanderà l'esecuzione con il comando RUN. Per evitare la perdita di tempo dovuta

alla preesecuzione si richiami il programma in memoria principale eseguendo il comando OLD, si preesegua il programma eseguendo un RUN o PREPARE e quindi si esegua il comando REPLACE.

Esempio

Completare l'analisi di coerenza tra le istruzioni di un programma presente in memoria principale; si assegnino quindi alle sue variabili dei valori durante lo stato di debugging. Il programma presente in memoria è:

```
10 PRINT A$;B,C$;D
20 A$="AREA"
30 C$="RETTANGOLO"
40 PRINT A$+C$; B*D
50 END
```

Premere

Il sistema è nello stato di debugging. Da tastiera si assegnano i valori:

```
A$="BASE="
C$="ALTEZZA="
B=10
D=100
```

Premere

Il programma è eseguito e stampa:

```
BASE= 10      ALTEZZA= 100
AREA RETTANGOLO 1000
```

Generalità

Una sequenza di comandi di sistema o programmi BASIC, od una combinazione di comandi e programmi, che siano utilizzati di frequente, possono essere registrate in una libreria come procedure catalogate e quindi eseguite come singole unità. L'impiego di una procedura catalogata può abbreviare il tempo totale di esecuzione dei corrispondenti lavori che richiederebbero altrimenti un intervento continuo dell'operatore attraverso la tastiera.

Una procedura catalogata è introdotta da tastiera come un file testo, quindi trascodificata in un file dati sequenziale e registrata in una libreria mediante un comando TRANSCODE. Al termine di queste operazioni, può essere eseguita mediante un comando PROCEDURE. Come primo passo per creare una procedura si introduca il comando TEXT. Poiché ogni linea della procedura inizia con un numero di linea -- ed i numeri devono essere in ordine crescente -- si può successivamente introdurre il comando AUTO# . Il modo con cui si codificano le linee di una procedura dipende dal tipo di procedura che si genera. In generale:

1. Se la procedura si compone di soli comandi, si digiti un comando per volta antepo-
nendo ad esso un numero di linea.
2. Se la procedura è una sequenza di programmi, si introduca una istruzione per linea -- compreso il numero di linea dell'istruzione. Così ogni linea di ogni programma avrà due numeri: il primo corrispondente alla sua posizione nella procedura, il secondo alla sua posizione nel programma. Ogni programma scritto per la procedura deve essere preceduto da un comando NEW e deve terminare con una istruzione END. Vediamo un semplice esempio:

```
0010 NEW
0020 10 PRINT"PROGRAMMA 1"
0030 20 END
0040 RUN
0050 NEW
0060 10 PRINT"PROGRAMMA 2"
0070 20 END
0080 RUN
```

Si noti che quando una linea contiene due numeri essi devono essere separati da almeno uno spazio.

3. Se una procedura è una combinazione di programmi e comandi, valgono le regole espresse al punto 2. Inoltre, si osservi che non è necessario che il comando RUN per l'esecuzione di un programma segua immediatamente l'istruzione END. Esso può essere posto dopo alcuni comandi che seguono l'istruzione END. Tuttavia, tali comandi non possono essere:

```
CCNFIGURE
NEW
OPTIONS
PROCEDURE filename
RUN filename
TEXT
```

Ognuno di questi comandi cancellerebbe il contenuto della memoria principale e quindi renderebbe non valido il comando RUN.

La seguente è un esempio di procedura composta da programmi e comandi:

```
0010 NEW
0020 10 PRINT"PROGRAMMA 1"
0030 20 END
0040 LIST
0050 RUN
0060 RUN NUMBER,LIB2
0070 NEW
0080 10 PRINT"PROGRAMMA 2"
0090 END
0100 LIST
0110 ENVIRONMENT
0120 RUN
```

Si noti che poichè una procedura è registrata come file dati, essa può essere creata mediante un programma BASIC che registri le linee della procedura in forma di stringhe di caratteri in un file dati sequenziale.

Input automatici

I valori richiesti dalle istruzioni INPUT che appaiono sia nei programmi richiamati da una procedura (RUN filename) che in quelli scritti come parte di una procedura, possono essere introdotti automaticamente. Utilizzando questa possibilità, si può evitare l'interruzione dell'esecuzione di un programma e la necessità di introdurre dati da tastiera ogni qualvolta è eseguita un'istruzione INPUT.

Si può predisporre l'input automatico per ogni comando RUN, contenuto in una procedura, che esegue un programma contenente istruzioni INPUT. La prestazione è predisposta facendo seguire immediatamente al comando RUN una o più linee con il seguente formato:

$$IN = \left\{ \begin{array}{l} \text{num-constant} \\ \text{string-constant} \end{array} \right\}, \left[\left\{ \begin{array}{l} \text{num-constant} \\ \text{string-constant} \end{array} \right\} \right] \dots$$

Una linea di tale formato deve essere scritta per ogni istruzione INPUT nel programma corrispondente. I valori specificati a destra del segno uguale sono assegnati automaticamente alle variabili specificate in ogni istruzione INPUT corrispondente, quando è eseguito un comando PROCEDURE che ha come operando A. (Si noti che la stringa IN = non deve contenere uno spazio).

Si osservi che il numero di linee IN = deve essere uguale al numero di istruzioni INPUT del corrispondente programma. Se vi sono più linee IN = di quelle richieste, le linee in eccesso sono ignorate; ma se il numero di linee IN = è inferiore a quello richiesto, viene visualizzato il seguente messaggio d'errore:

ERROR 204 IN LINE line-number

Se, invece, il numero di valori specificati in una istruzione IN = è superiore al numero di variabili della corrispondente istruzione INPUT, i valori in eccesso sono semplicemente ignorati.

Questo messaggio d'errore viene inviato anche se viene data una risposta non valida a una richiesta di input, come ad esempio se si impostano dati stringa invece di dati numerici oppure se si imposta un numero insufficiente di valori per un'istruzione INPUT.

Salti condizionati

Una procedura può contenere linee che ne modificano l'ordine di esecuzione se è soddisfatta una certa condizione. L'esecuzione della procedura riprende da una linea specificata se il programma eseguito in precedenza ha assegnato ad una variabile CC un valore predefinito. La sintassi di una linea di salto condizionato è:

```
IF CC = X, line-number
```

dove X è un valore da 0 a 9 e line-number è il numero di linea di una linea della procedura. Il valore a CC è assegnato, dal programma eseguito in precedenza, per mezzo di una variabile stringa (con lunghezza di allocazione di 1) specificata come primo valore nell'area comune (vedi nel capitolo 5 l'istruzione COMMON).

Procedura avviata alla inizializzazione del sistema

Per creare una procedura catalogata che sarà eseguita automaticamente ogni volta che il sistema è acceso o, comunque, inizializzato (comandi CONFIGURE ed OPTIONS), si eseguano i seguenti passi:

1. Si apra una libreria (se non vi sono librerie aperte) e si registri in essa la procedura presente in memoria principale, eseguendo un comando TRANSCODE con l'operando filename uguale a *SETUP.
2. Si esegua il comando LBSTORE per fare in modo che la libreria contenente *SETUP sia aperta quando il sistema è inizializzato.
3. Si esegua un comando CONFIGURE con l'operando PRO.

Test di una procedura: comando TEST

Il comando TEST permette di controllare l'esecuzione di una procedura per: (1) determinare se vi sono degli errori, (2) procedere nell'esecuzione anche se vi sono degli errori e (3) conoscere il numero totale degli errori. Il comando TEST è descritto qui piuttosto che in una parte ad esso specifica, perchè può essere

usato unicamente in una procedura catalogata.

Il formato del comando è:

$$\text{TEST } \left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \\ \# \end{array} \right\}$$

La prestazione è attivata specificando: TEST ON nel punto della procedura in cui si vuole che inizi il controllo, ad esempio, subito dopo il comando PROCEDURE o subito prima di una serie di comandi di cui si vuol controllare la validità. Subito dopo l'esecuzione di TEST ON, il sistema stampa il codice d'errore di ogni errore che si verifica durante l'esecuzione dei comandi successivi. La procedura sarà eseguita fino alla fine anche se si verifica un errore di programma, ma in questo caso l'esecuzione del programma è interrotta ed è eseguito il comando successivo della procedura.

Per terminare l'effetto prodotto dal comando TEST ON si deve specificare nella procedura un comando TEST OFF o TEST#. Gli errori che si verificano dopo l'esecuzione di TEST OFF o TEST# e prima che sia eseguito un successivo TEST ON sospendono l'esecuzione della procedura catalogata. La differenza tra gli ultimi due comandi sta nel fatto che quando è eseguito TEST# il sistema stampa il numero totale di errori che si è verificato fino a quel momento.

Esempio

Per un esempio che illustri come si crea una procedura catalogata si veda il comando PROCEDURE.

Comando PROCEDURE

Funzione

Inizia l'esecuzione di una procedura catalogata.

Formato

PRO [CEDURE] filename, [lib-ref], [A], [line-num₁] [line-num₂]

dove:

filename

specifica il nome di una procedura catalogata (registrata in una libreria come file dati sequenziale)

lib-ref

può essere uno dei seguenti operandi:

(lib-name, unit-name)

lib-name

(,unit-name)

dove lib-name specifica il nome di una libreria e unit-name specifica il nome simbolico di una unità a disco o floppy disk

A

specifica l'impiego degli input automatici

line-num₁

specifica la linea della procedura da cui deve iniziare l'esecuzione

line-num₂

specifica l'ultima linea che deve essere eseguita

Azione

Quando è specificato con tutti gli operandi, il comando PROCEDURE ordina al sistema di eseguire la procedura catalogata filename, registrata nella libreria specificata da lib-ref, iniziando dalla linea line-num₁ e finendo con la linea line-num₂. I valori alle variabili delle istruzioni INPUT di tutti i programmi contenuti nella procedura, o richiamati dalla procedura, saranno assegnati automaticamente per mezzo delle linee IN = comprese nella procedura.

Quando line-num₁ non è specificato nel comando PROCEDURE, l'esecuzione inizia dalla prima linea della procedura.

Quando line-num₂ non è specificato nel comando PROCEDURE, l'esecuzione continua finchè non è eseguita l'ultima linea della procedura.

Note

1. Se come operando lib-ref è specificato (lib-name, unit-name), la procedura deve essere nella libreria lib-name sul disco o floppy disk nell'unità il cui nome simbolico è unit-name.
2. Se come operando lib-ref è specificato lib-name, la procedura è cercata nella prima libreria aperta di nome lib-name.
3. Se come operando lib-ref è specificato (,unit-name), la ricerca della procedura specificata inizia dalla prima libreria aperta residente sull'unità con nome simbolico unit-name.
4. Se l'operando lib-ref è omissso, la ricerca della procedura inizia dalla prima libreria aperta e continua nell'ordine con cui le librerie sono aperte.
5. L'operando line-num₁ non può essere il numero di linea di un'istruzione BASIC.
6. Se la funzione PRINT ALL è attiva, ogni linea della procedura eseguita è visualizzata e stampata.
7. Quando l'esecuzione di una procedura è completa, il sistema commuta nello stato comandi.
8. Si può terminare l'esecuzione di una procedura premendo **BREAK**. Il sistema commuta nello stato comandi.
9. Se una linea di una procedura ha un errore di sintassi (ad esempio LITT invece di LIST), è visualizzato un messaggio di errore ed il sistema commuta nello stato comandi. Introducendo il comando:

TRANSCODE T, filename, lib-ref,#

(dove filename è il nome della procedura), la pro-

cedura è caricata in memoria principale come file testo per le correzioni necessarie. Dopo che la correzione è stata effettuata, la versione modificata della procedura può essere registrata di nuovo, ed essere disponibile per successive esecuzioni, introducendo i comandi:

```
PURGE filename, lib-ref  
TRANSCODE D, filename, lib-ref,#
```

10. Se durante l'esecuzione di un programma che fa parte di una procedura è visualizzato un errore recuperabile, l'esecuzione è interrotta ed il sistema commuta nello stato di debugging. Dopo aver premuto **BREAK** si può correggere l'errore eseguendo le operazioni descritte nel punto 9. Oppure si può continuare l'esecuzione del programma, e quindi della procedura, premendo **CONTINUE**.
11. Se durante l'esecuzione di una procedura è visualizzato un errore non recuperabile riferito ad un programma della procedura, l'esecuzione del programma è interrotta. Il tasto di console **STEP** è acceso per informare l'utente che può verificare il contenuto delle variabili del programma, come quando il sistema è nello STATO DI DEBUGGING. Dopo aver eventualmente verificato il contenuto delle variabili di programma, tuttavia, si deve premere **BREAK**. Il sistema commuta nello STATO COMANDI. Per correggere il programma, si eseguono le operazioni descritte nel punto 9.
12. Se si introduce un comando PROCEDURE senza l'operando A ma nella relativa procedura vi sono linee IN =, tali linee sono ignorate.
13. Se una procedura contiene un comando OPTIONS o CONFIGURE, l'esecuzione della procedura termina dopo l'esecuzione del comando.
14. Una procedura può richiedere l'esecuzione di altre procedure mediante comandi PROCEDURE.
15. Se durante l'esecuzione di una procedura è eseguita una linea del tipo IF CC = x, line-number con line-number più grande del numero di linea line-num₂ specificato nel comando PROCEDURE, l'esecuzione della procedura riprende dal numero di linea

line-number. L'esecuzione della procedura terminerà quando sarà eseguita la linea della procedura con il più grande numero di linea o quella con numero di linea line-num₂.

16. Quando il sistema è nello STATO COMANDI a causa di un errore incontrato durante l'esecuzione di una procedura, si può riprendere l'esecuzione della procedura introducendo un comando PROCEDURE senza operandi. La prima linea eseguita sarà quella che segue la linea errata. Questo metodo di lavoro è comunque sconsigliabile perchè può produrre dei risultati imprevedibili.

Esempio

Vediamo un esempio di procedura, PRO1, composta da tre programmi, alcuni comandi e due linee IF.

```
0010 OLD P1
0020 LIST
0030 RUN
0040 IN=3,5,10
0050 IF CC=0.80
0060 RUN P3
0070 IF CC=1.90
0080 RUN P2
0090 PRO STOP
```

Questa procedura permette di calcolare alcuni valori della funzione $\frac{1}{x-A}$ definita in un intervallo $[X(2), X(3)]$ tale che $X(2) < X(3)$ e che in esso vi sia il punto di ascissa A.

Il programma P1 è composto dalle seguenti istruzioni:

```
0010 DCL 1A$
0020 COMMON A$,X(1)
0030 DIM X(3)
0040 INPUT X(1),X(2),X(3)
0050 IF X(2)>X(3) THEN 100
0060 IF (X(2)<X(1)) AND (X(3)>X(1)) THEN 100
0070 IF (X(2)=X(1))OR(X(3)=X(1)) THEN 100
0080 A$="1"
0090 GOTO 110
0100 A$="0"
0110 END
```

ed è stato registrato nella libreria MAT, specificando l'impiego di un'area comune di 28 byte, con il seguente comando:

```
SAV P1, MAT,, COM = 28
```

Mediante l'istruzione COMMON, linea 20, i valori assegnati ad A\$, X(1), X(2) ed X(3) sono registrati in una area comune. Il valore corrispondente al punto A è registrato nel secondo elemento dell'area comune X(1). Il programma quindi verifica che $X(2) < X(3)$ e che A sia compreso nell'intervallo definito. Se tali condizioni sono soddisfatte un 1, nel formato stringa, è registrato nel primo elemento dell'area comune. Se le condizioni non sono soddisfatte, viene registrato uno \emptyset nel formato stringa. Se le condizioni non sono soddisfatte, la linea 50 della procedura legge lo \emptyset e cede il controllo dell'esecuzione alla linea 80, che provoca l'esecuzione del programma P2.

Il programma P2 è composto dalle seguenti due istruzioni:

```
0010 PRINT "VALORI INTRODOTTI NON VALIDI"  
0020 END
```

Se le condizioni testate dal programma P1 sono soddisfatte, la linea 50 della procedura passa il controllo all'esecuzione del programma P3. Il programma P3 è composto dalle seguenti istruzioni:

```
0010 DCL 1A$  
0020 DIM X(3)  
0030 COMMON A$,X()  
0040 DEF FNA(X,A)=1/(X-A)  
0050 FOR I=X(2) TO X(3) STEP 0.1  
0060 PRINT I,FNA(I,X(1))  
0070 NEXT I  
0080 END
```

ed è anch'esso registrato specificando l'impiego di una area comune di 28 byte. Come si vede vengono stampati i valori assunti dalla funzione $\frac{1}{X-A}$ per valori variabili, con incremento di 0,1, nell'intervallo $[X(2),X(3)]$.

Dopo che uno dei programmi P2 o P3 è eseguito, viene eseguita la procedura STOP che è composta dalle seguenti linee:

```
0010 OLD TEXT,MAT
0020 LIST ,,X
```

con il file testo TEXT composto dalle seguenti linee:

```
0010 Olivetti P6066
0020 PROCEDURA - PRO1
```

Per creare la procedura PRO1, avendo in precedenza creato e registrato su disco (o floppy disk) la procedura STOP ed i programmi P1, P2 e P3, essa deve prima essere introdotta in memoria come file testo e quindi trascodificata in un file dati eseguendo il comando:

```
TRA D, PRO1, MAT
```

Se si esegue la procedura con il comando:

```
PRO PRO1,,A
```

che richiede l'esecuzione di input automatici, i valori assegnati alle variabili X(1), X(2) ed X(3) saranno nell'ordine 3, 5 e 10.

Se invece la procedura PRO1 è eseguita con il comando:

```
PRO PRO1
```

i valori alle variabili suddette potranno essere assegnati da tastiera quando l'esecuzione del programma P1 sarà interrotta per effetto dell'esecuzione dell'istruzione INPUT nella linea 40.

Comando PURGE

Funzione Cancellare un file da una libreria.

Formato **PUR[GE] filename [, lib-ref]**

dove:

filename

è il nome di un file da cancellare

lib-ref

può essere uno dei seguenti operandi:

(lib-name, unit-name)

lib-name

(,unit-name)

con lib-name che specifica il nome di una libreria e unit-name che specifica il nome di una unità a disco od a floppy disk

Azione

Il comando ordina al sistema di cancellare il file specificato con l'operando filename dalla libreria lib-name sul disco o floppy disk nell'unità con nome simbolico unit-name. Lo spazio occupato dal file viene reso disponibile per altre operazioni sulla stessa libreria.

Se è specificato il solo operando filename, il file da cancellare è cercato nelle librerie attualmente aperte secondo l'ordine di apertura.

Note

1. Se come operando lib-ref è specificato lib-name, il file da cancellare è cercato nella prima libreria aperta di nome lib-name.
2. Se come operando lib-ref è specificato (unit-name), il file da cancellare è cercato nella prima libreria aperta residente sul disco o floppy disk nella unità con nome simbolico unit-name.

3. Il comando non è accettato con i file che appartengono alla sottolibreria package ed alla sottolibreria comune, se queste sono state protette con il programma di utilità LBPROTECT.
4. Quando si invia il comando PURGE per cancellare un file definito DEAD, il file viene cancellato solo logicamente; lo spazio, infatti, resta occupato. Vedere il comando CATALOG per ulteriori spiegazioni sulla riallocazione dello spazio.

Esempio

Si cancelli il programma MAT1 dalla libreria MAT presente sul disco nell'unità UD.

Premere **PUR MAT1(MAT,UD)** **END OF LINE**

Comando REPLACE

Funzione

Sostituisce, in una libreria, un programma od un file testo con un altro presente in memoria principale.

Formato

REP [LACE] [lib-ref]

dove:

lib-ref

può essere uno dei seguenti operandi:

(lib-name, unit-name)

lib-name

(,unit-name)

con lib-name che specifica il nome di una libreria e unit-name che specifica il nome di una unità a disco od a floppy disk

Azione

Il comando ordina al sistema di registrare nella libreria lib-name, sul disco o floppy disk nell'unità con nome simbolico unit-name, il programma o file testo presente in memoria principale al posto del programma o file testo già registrato nella libreria con lo stesso nome.

Se come operando lib-ref è specificato lib-name, il file da sostituire è cercato nella prima libreria aperta con nome lib-name.

Se come operando lib-ref è specificato (unit-name), il file da sostituire è cercato nella prima libreria aperta residente sul disco o floppy disk nell'unità con nome simbolico unit-name.

Se l'operando lib-ref non è specificato, il file da sostituire è cercato nella prima libreria aperta, secondo l'ordine di apertura.

Note

1. Non si può utilizzare il comando REPLACE per sostituire un programma o file testo con un programma o file testo appena introdotto da tastiera, quindi senza nome. In questo caso si introduca il comando PURGE per cancellare il file che si vuole sostituire e quindi il comando SAVE per registrare il nuovo file.
2. Eseguendo il comando REPLACE si può verificare una condizione anomala che causa la cancellazione del file originale dalla libreria senza sostituirlo; a questo punto il sistema invia il seguente messaggio:

ERROR cod. errore FILE filename PURGED

Il file in memoria principale non viene cancellato e si può procedere all'azione correttiva. Per esempio si può eseguire un comando SAVE riferito ad una diversa libreria.

Esempio

Si aggiorni il programma EQUAZ presente nella libreria MAT sul disco nell'unità UD.

Premere

Il programma è trasferito in memoria principale.

Premere

E' stampato il listing del programma. Si introducano le modifiche desiderate.

Premere

Il nuovo programma EQUAZ è registrato nella libreria sostituendo il vecchio programma con lo stesso nome.



Comando RESEQUENCE

Funzione

Modifica la numerazione delle linee del programma o del file testo presente in memoria principale.

Formato

RES [EQUENCE] [line-num] [, increment]

dove:

line-num

indica il primo numero di linea da assegnare al programma o file testo presente in memoria principale

increment

è un numero intero positivo che indica il valore che è aggiunto ad ogni numero di linea per ottenere il numero di linea successivo

Azione

Il comando ordina al sistema di assegnare alla prima linea del programma o file testo presente in memoria principale il numero specificato con il primo operando e di aggiungere ad ogni numero di linea l'incremento specificato con il secondo operando, per assegnare il successivo numero di linea.

Il comando, con solamente il primo operando, ordina al sistema di assegnare alla prima linea, del programma o file testo, il numero specificato e di aggiungere 10 ad ogni numero di linea per assegnare il numero di linea successivo.

Il comando, con solamente il secondo operando, ordina al sistema di assegnare il numero increment alla prima linea del programma o file testo e di aggiungere ad ogni numero di linea l'incremento specificato con il secondo operando per assegnare il successivo numero di linea.

Il comando, privo di operandi, ordina al sistema di

assegnare il numero 10 alla prima linea del programma o file testo e di aggiungere 10 ad ogni numero di linea per assegnare il successivo numero di linea.

Note

1. Il sistema modifica automaticamente i riferimenti ad altre istruzioni di programma contenuti in alcune istruzioni (es. GOTO etc.) presenti in memoria principale.
2. Se in un programma una istruzione fa riferimento ad altra istruzione che è stata cancellata, il comando RESEQUENCE rinumeri i numeri di linea con l'incremento specificato, ma stampando il listing del programma (comando LIST) si potrà notare un gap nella numerazione delle linee. Per rimuovere tale anomalia si esegua il comando DECOMPILE e quindi il comando COMPILE.
3. Il comando non può essere usato con i programmi protetti mediante il comando SECURE.

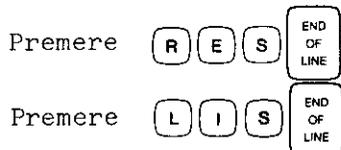
Esempi

1. Si rinumeri il seguente programma presente in memoria principale iniziando da 10 con incremento di 10.

```
FILE

0040 GOTO 55
0045 PRINT "VUOI ESEGUIRE UN'ALTRA MEDIA?"
0048 INPUT A$
0050 IF A$<>"SI" THEN 500
0055 PRINT "DIMMI DI QUANTI NUMERI VUOI FARE LA MEDIA"
0060 INPUT N
0100 LET X=0
0120 FOR I=1 TO N STEP 1
0150 INPUT M
0170 PRINT M
0200 LET X=M+X
0240 NEXT I
0270 PRINT "LA MEDIA E:"
0300 PRINT X/N
0340 GOTO 45
0500 END

END OF LISTING
```



FILE

```
0010 GOTO 50
0020 PRINT "VUOI ESEGUIRE UN'ALTRA MEDIA?"
0030 INPUT A$
0040 IF A$<>"SI" THEN 160
0050 PRINT "DIMMI DI QUANTI NUMERI VUOI FARE LA MEDIA"
0060 INPUT N
0070 LET X=0
0080 FOR I=1 TO N STEP 1
0090 INPUT M
0100 PRINT M
0110 LET X=M+X
0120 NEXT I
0130 PRINT "LA MEDIA E: "
0140 PRINT X/N
0150 GOTO 20
0160 END

END OF LISTING
```

Come si vede i riferimenti ad altre istruzioni contenuti in alcune di esse sono rinumerati automaticamente.

2. Si rinumeri il seguente programma presente in memoria principale come file testo.

FILE

```
0040 GOTO 55
0045 PRINT "VUOI ESEGUIRE UN'ALTRA MEDIA?"
0048 INPUT A$
0050 IF A$<>"SI" THEN 500
0055 PRINT "DIMMI DI QUANTI NUMERI VUOI FARE LA MEDIA"
0090 INPUT N
0100 LET X=0
0120 FOR I=1 TO N STEP 1
0150 INPUT M
0170 PRINT M
0200 LET X=M+X
0240 NEXT I
0270 PRINT "LA MEDIA E: "
0300 PRINT X/N
0340 GOTO 45
0500 END

END OF LISTING
```

Premere

R E S

END
OF
LINE

Premere

L I S

END
OF
LINE

FILE

```
0010 GOTO 55
0020 PRINT "VUOI ESEGUIRE UN'ALTRA MEDIA?"
0030 INPUT A$
0040 IF A$<>"SI" THEN 500
0050 PRINT "DIMMI DI QUANTI NUMERI VUOI FARE LA MEDIA"
0060 INPUT N
0070 LET X=0
0080 FOR I=1 TO N STEP 1
0090 INPUT M
0100 PRINT M
0110 LET X=M+X
0120 NEXT I
0130 PRINT "LA MEDIA E:"
0140 PRINT X/N
0150 GOTO 45
0160 END
```

END OF LISTING

Come si vede i riferimenti ad altre istruzioni non sono modificati.

Comando RUN

Funzione

Comanda l'esecuzione di un programma.

Formato

RUN [filename [, lib-ref]
[line-num]

dove:

filename

è il nome di un programma presente in una libreria

line-num

indica il numero di linea da cui inizia l'esecuzione di un programma presente in memoria principale

lib-ref

può essere uno dei seguenti operandi:

(lib-name, unit-name)

lib-name

(, unit-name)

con lib-name che specifica il nome di una libreria e unit-name che specifica il nome di una unità a disco od a floppy disk

Azione

Il comando ordina al sistema di caricare in memoria principale ed eseguire il programma di nome filename, presente nella libreria lib-name sul disco o floppy disk nell'unità con nome simbolico unit-name.

Il comando, con line-num come operando, comunica al sistema di eseguire il programma presente in memoria principale iniziando l'esecuzione dall'istruzione il cui numero di linea è line-num.

Il comando, senza operandi, ordina al sistema di eseguire il programma che è presente in memoria principale.

1. Se come operando lib-ref è specificato lib-name, il programma da eseguire è cercato nella prima libreria aperta di nome lib-name.
2. Se come operando lib-ref è specificato (unit-name), il programma da eseguire è cercato nella prima libreria aperta residente sul disco o floppy disk nell'unità con nome simbolico unit-name.
3. Se l'operando lib-ref non è specificato, il programma da eseguire è cercato nella prima libreria aperta, secondo l'ordine di apertura.
4. Per poter eseguire un programma devono essere effettuati una serie di controlli di coerenza fra le istruzioni del programma, ad esempio se nel programma vi è una istruzione:

```
55 GOTO 95
```

si deve verificare l'esistenza della istruzione 95. Questa fase (detta preesecuzione) è attuata eseguendo il comando PREPARE (vedi PREPARE) oppure il comando RUN, se il programma non è stato precedentemente registrato in una libreria con l'informazione che è formalmente corretto. Per introdurre tale informazione nel programma si possono seguire due vie:

- interrompere l'esecuzione del programma premendo il tasto di console BREAK e quindi registrarlo in una libreria con il comando SAVE o REPLACE
- preeseguire il programma con il comando PREPARE e quindi registrarlo in una libreria con il comando SAVE o REPLACE

In questo modo i successivi comandi RUN eseguono direttamente il programma evitando la fase di preesecuzione.

5. Se, durante la fase di preesecuzione, il sistema non rileva alcun errore, viene visualizzato il messaggio: ***** FORMALLY CORRECT PROGRAM ****** che è anche stampato se la funzione **PRINT ALL** è abilitata. Viene quindi visualizzato il messaggio: PROGRAM nome-programma RUNNING; il programma è eseguito; al termine della esecuzione viene visualizzato il mes-

saggio: READY.

6. Normalmente, durante l'esecuzione di un programma, appare sul display il seguente messaggio: PROGRAM nome-programma RUNNING ed al termine della esecuzione appare sul display il messaggio: READY. Se si vogliono sopprimere tali messaggi, si deve specificare l'operando MSG nel comando SAVE quando si comanda la registrazione del programma in una libreria (vedi comando SAVE).
7. Per poter eseguire un programma con il comando RUN, nella prima libreria aperta vi deve essere spazio disponibile pari allo spazio occupato in memoria principale dal programma stesso. Se però si introduce il comando RUN filename, il programma è eseguito anche se non è verificata la condizione suddetta. Se il programma è stato appena creato, si modifichi l'ordine con cui le librerie sono state aperte.
8. L'esecuzione di un programma può essere terminata premendo il tasto di console **BREAK** ; il sistema passa nello stato comandi.
9. Si può interrompere l'esecuzione di un programma premendo il tasto di console **STEP** ; il sistema passa nello stato di debugging. Premendo successivamente **STEP** si possono eseguire le istruzioni del programma una per volta oppure, premendo **CONTINUE** si esegue completamente il resto del programma (vedi capitolo 7).
10. Non si può utilizzare come operando line-num un numero di linea di una istruzione interna ad un ciclo FOR/NEXT, ad un sottoprogramma o ad una definizione di funzione multilinea.

Comando SAVE

Funzione

Registra un programma od un file testo, presente in memoria principale, in una libreria.

Formato

SAV[E] filename, [lib-ref] [,MSG=n] [,COM=m]

dove:

filename

indica il nome con cui il programma od il file testo sarà registrato nella libreria

lib-ref

può essere uno dei seguenti operandi:

(lib-name, unit-name)

lib-name

(, unit-name)

con lib-name che specifica il nome di una libreria e unit-name che specifica il nome di una unità a disco od a floppy disk

MSG=n

con n uguale a zero od a uno; inibisce la visualizzazione di uno o tutti i messaggi emessi durante l'esecuzione del comando RUN

COM=m

con m numero intero di al massimo 5 cifre, specifica che all'atto del caricamento del programma in memoria principale m byte della memoria saranno riservati come area comune per lo scambio di dati con altri programmi (vedi istruzione COMMON)

Azione

Il comando ordina al sistema di registrare, con il nome filename, nella libreria lib-name sul disco o floppy disk nell'unità con nome simbolico unit-name il programma o file testo presente in memoria principale.

Se è specificato l'operando MSG=Ø, il comando SAVE inibisce la visualizzazione di tutti i messaggi emessi

durante l'esecuzione del comando RUN.

Se è specificato l'operando MSG=1, il comando SAVE inibisce la visualizzazione del messaggio: PROGRAM nome-programma RUNNING, durante l'esecuzione del comando RUN.

Se il comando SAVE agisce su un programma e se è specificato l'operando COM=m, il programma viene registrato con l'indicazione che all'atto del caricamento in memoria principale (comando OLD, oppure RUN, oppure istruzione CHAIN) dovranno essere riservati m byte all'inizio dell'area utente, subito prima dell'indirizzo da cui parte il caricamento del programma. L'area così riservata potrà essere utilizzata dal programma per depositare o prelevare dati comuni ad altri programmi (vedi istruzione COMMON).

Note

1. Se come operando lib-ref è specificato lib-name, il file è registrato nella prima libreria aperta di nome lib-name.
2. Se come operando lib-ref è specificato (,unit-name), il file è registrato nella prima libreria aperta residente sul disco o floppy disk nell'unità con nome simbolico unit-name.
3. Se l'operando lib-ref non è specificato, il file è registrato nella prima libreria aperta, secondo l'ordine di apertura.
4. Gli operandi MSG=n e COM=m possono essere specificati solamente se in memoria principale è presente un programma.
5. La capacità della zona di memoria specificata in m byte è arrotondata per eccesso al più vicino multiplo di quattro.
6. La dimensione specificata per l'area comune può essere modificata eseguendo la seguente sequenza di comandi:

OLD filename

PURGE filename
SAVE filename, (lib-name, unit-name), COM=m

con m che specifica il nuovo valore.

7. Se il tentativo di caricare in memoria principale un programma registrato con un comando SAVE con l'operando COM=m causa una segnalazione di errore che specifica spazio di memoria utente disponibile insufficiente, si preme il tasto **CLEAR** e si esegua il comando LIST. Se è stampato un listing, il programma è stato caricato in memoria principale ma senza l'area comune richiesta. In questo caso la dimensione dell'area comune può essere ridotta eseguendo un comando PURGE seguito da un comando SAVE che specifica un valore più piccolo di m. Si può quindi ripetere l'operazione che ha provocato il messaggio d'errore.
8. Il comando SAVE non può essere usato per registrare un programma o file testo in una sottolibreria package che sia stata protetta mediante l'esecuzione del programma di utilità LBPROTECT.
9. L'operando filename non può essere uguale al nome di un programma, file testo o file dati presente nella libreria in cui il file testo o programma deve essere registrato.
10. La parola chiave del comando può essere introdotta premendo **SHIFT** con **SAVE**.
11. Per registrare in una libreria dei programmi che sono ottimizzati in occupazione di memoria e velocità di esecuzione si devono eseguire i seguenti comandi nell'ordine con cui sono riportati:

DECOMPILE
COMPILE
PREPARE

Dopo aver eseguito il comando PREPARE si preme il tasto di console **BREAK** e si esegua il comando SAVE.

Esempio

Si registri il programma presente in memoria principale, nella libreria MAT sul disco nell'unità UD, con il nome CAL.

Premere

SHIFT SAVE C A L , (M A T , U D) END OF LINE

Comando SECURE

Funzione

Vieta l'esecuzione di alcuni comandi su parte o tutto un programma o file dati.

Formato

SEC[URE] filename, [lib-ref] [,n]

dove:

filename

indica il nome di un file programma o file dati, presente in una libreria che si vuole proteggere da alcune operazioni

lib-ref

può essere uno dei seguenti operandi:

(lib-name, unit-name)

lib-name

(, unit-name)

con lib-name che specifica il nome di una libreria e unit-name che specifica il nome di una unità a disco od a floppy disk

n

è un numero intero positivo che indica la prima linea di un insieme di linee da proteggere (od il primo dato di un insieme di dati da proteggere)

Azione

Se filename è il nome di un programma presente in una libreria, il comando SECURE ordina al sistema di proteggerlo dall'azione dei comandi:

DECOMPILE

LINK

MODIFY

RESEQUENCE

DELETE LINE

FETCH

LIST

e dall'azione dei tasti \uparrow e \downarrow . Se nel comando è specificato l'operando n, allora l'azione dei comandi:

DELETE LINE
FETCH
LIST

e dei tasti \uparrow e \downarrow , è impedita dall'istruzione con il numero di linea n in poi.

Se filename è il nome di un file dati presente in una libreria, il comando SECURE ordina al sistema di proteggerlo dall'azione dei comandi MODIFY, TRANSCODE e TRUNCATE e di impedire la registrazione di nuovi dati in tutto il file. Il comando SECURE impedisce anche la stampa del contenuto di un file dati effettuata con il programma di utilità FLPRINT; in questo caso, se si specifica l'operando n, tale stampa è impedita dall'n-esimo dato in poi.

Note

1. Se come operando lib-ref è specificato lib-name, il file da proteggere è cercato nella prima libreria aperta di nome lib-name.
2. Se come operando lib-ref è specificato (unit-name), il file da proteggere è cercato nella prima libreria aperta residente sul disco o floppy disk nella unità con nome simbolico unit-name.
3. Se l'operando lib-ref non è specificato, il file da proteggere è cercato nella prima libreria aperta, secondo l'ordine di apertura.
4. Non si può togliere la protezione assegnata ad un programma o file dati mediante il comando SECURE.
5. Si può ampliare ma non si può ridurre l'area di protezione di un programma. Così, se un programma è stato protetto con il comando:

```
SECURE MAT1,,100
```

si può ampliare l'area protetta con il comando:

```
SECURE MAT1,,50
```

ma non si può ridurre l'area protetta; infatti il

comando: SECURE MAT1,,300 non è accettato.

Esempio

Si protegga il programma MAT1, residente nella prima libreria aperta, dall'istruzione 50 in poi.

Premere

S	E	C	M	A	T	.	.	5	0	END OF LINE
---	---	---	---	---	---	---	---	---	---	-------------

Comando SHIFT

Funzione Modifica la numerazione delle linee di un programma o file testo presente in memoria principale, iniziando dalla linea specificata.

Formato **SHI [FT] line-num, increment**

dove:

line-num

è un numero di linea che specifica da quale linea deve essere modificata la numerazione delle linee

increment

è un numero intero positivo che specifica il valore di cui deve essere incrementato ogni numero di linea da modificare

Azione Il comando ordina al sistema di modificare i numeri di linea delle linee del programma, o file testo, presente in memoria principale, iniziando dalla linea con numero di linea line-num. I nuovi numeri di linea sono ottenuti aggiungendo il valore specificato con increment ai precedenti numeri di linea.

Note

1. Se in memoria principale vi è un programma, l'esecuzione del comando SHIFT modifica automaticamente i riferimenti ad altre istruzioni di programma contenuti in alcune istruzioni (es. GOTO). Sono modificati anche i riferimenti ad istruzioni che non sono state ancora introdotte nel programma o che sono state cancellate.
2. Se si cancellano delle linee di programma che contengono un riferimento ad altre istruzioni, il riferimento è ricordato dal sistema e quando si esegue il comando SHIFT, viene incrementato di increment. Per rimuovere tali riferimenti si devono eseguire i comandi DECOMPILE e COMPILE, nell'ordine.

Esempio

Vediamo come i numeri di linea, dal 15 in poi, sono incrementati di 100 nella routine sottostante.

```
LIS  
FILE  
  
0010 INPUT A$  
0015 PRINT A$  
0040 INPUT B$  
0070 PRINT B$  
0080 END  
  
END OF LISTING  
  
SHI 15,100  
LIS  
FILE  
  
0010 INPUT A$  
0115 PRINT A$  
0140 INPUT B$  
0170 PRINT B$  
0180 END  
  
END OF LISTING
```

Comando SPACE

Funzione

Visualizza lo spazio disponibile (in byte) per registrazioni successive in una libreria.

Formato

SPA[CE] [lib-ref]

dove:

lib-ref

può essere uno dei seguenti operandi:

(lib-name, unit-name)

lib-name

(,unit-name)

con lib-name che specifica il nome di una libreria e unit-name che specifica il nome di unità a disco od a floppy disk

Azione

Il comando ordina al sistema di visualizzare lo spazio libero da registrazioni nella libreria di nome lib-name residente sul disco o floppy disk nell'unità con nome simbolico unit-name.

Note

1. Se come operando lib-ref è specificato lib-name, la libreria di cui è visualizzato lo spazio libero disponibile è la prima libreria di nome lib-name aperta.
2. Se come operando lib-ref è specificato (,unit-name), la libreria di cui è visualizzato lo spazio libero disponibile è la prima libreria aperta residente nell'unità con nome simbolico unit-name.
3. Se l'operando lib-ref non è specificato, la libreria di cui è visualizzato lo spazio libero disponibile è la prima libreria aperta.

Esempio

Si richieda la visualizzazione dello spazio libero
nella libreria MAT presente sull'unità floppy disk
FDU1.

Premere

Comando START

Funzione Permette di riprendere l'esecuzione di un programma da una istruzione specificata.

Formato **STA [RT] line-num**

dove:

line-num

è un numero intero positivo che indica l'istruzione da cui riprenderà l'esecuzione del programma.

Azione Il comando ordina al sistema di riprendere l'esecuzione del programma interrotto dalla istruzione con numero di linea line-num.

- Note
1. Il comando può essere usato solamente quando il sistema è nello stato di debugging.
 2. L'esecuzione del programma riprende immediatamente dopo l'esecuzione del comando START.
 3. Non si può avere come operando line-num un numero di linea di una istruzione interna ad un ciclo FOR/NEXT, ad un sottoprogramma o ad una definizione di funzione multilinea.

Esempi Vedi il capitolo 7.

Comando STKEYS

Funzione

Registra su disco sistema o floppy disk sistema il contenuto dei tasti funzione.

Formato

STK [EYS]

Azione

Il comando ordina al sistema di registrare sul disco sistema o floppy disk sistema l'attuale contenuto dei tasti funzione.

Nota

Ogni qual volta il sistema è inizializzato (dopo la accensione, eseguendo il comando CONFIGURE od il comando OPTIONS) od è eseguito il comando LDKEYS, ai tasti funzione è assegnato il contenuto registrato sul disco sistema o floppy disk sistema dall'ultimo comando STKEYS eseguito.



Comando STOP

Funzione Interrompe l'esecuzione di un programma prima della istruzione specificata.

Formato **STO [P] line-num**

dove:

line-num

è un numero intero positivo che indica l'istruzione di programma prima della quale l'esecuzione deve essere interrotta

Azione Il comando ordina al sistema di interrompere l'esecuzione del programma prima che l'istruzione con numero di linea line-num sia eseguita.

- Note
1. Il comando può essere introdotto solamente quando il sistema è nello stato di debugging.
 2. Dopo aver introdotto il comando STOP si può riprendere l'esecuzione del programma premendo il tasto di console **CONTINUE** oppure **STEP**, o introducendo il comando START.
 3. Il comando STOP non deve essere confuso con l'istruzione STOP (vedi capitolo 5).

Esempi Vedi il capitolo 7.



Comando TEXT

Funzione Permette di introdurre un file testo da tastiera.

Formato **TEX [T]**

Azione Il comando comunica al sistema che quanto verrà successivamente introdotto sono linee di un file testo.

- Note
1. Quando il comando è eseguito il precedente contenuto della memoria utente è cancellato.
 2. Utilizzando il comando AUTO# si possono numerare automaticamente le linee del file testo introdotte successivamente.
 3. Per la definizione di file testo si veda il capitolo 2 e per ulteriori informazioni sull'impiego dei file testo si veda la pubblicazione P6066 - Gestione dei file testo - Guida dell'utente.



Comando TRANSCODE

Funzione

Converte un file dati sequenziale in un file testo e viceversa.

Formato

TRA[NSCODE] $\left\{ \begin{matrix} T \\ D \end{matrix} \right\}$, filename, [lib-ref] [,#]

dove:

T

indica che il file dati sequenziale filename presente in una libreria deve essere convertito in un file testo residente in memoria principale

D

indica che il file testo presente in memoria principale deve essere convertito in un file dati residente in una libreria

filename

specifica il nome di un file dati sequenziale

lib-ref

può essere uno dei seguenti operandi:

(lib-name, unit-name)

lib-name

(,unit-name)

con libe-name che specifica il nome di una libreria e unit-name che specifica il nome di una unità a disco od a floppy disk

#

specifica il tipo di numerazione da assegnare al file convertito

Azione

Il comando, con T come primo operando e con l'operando #, ordina al sistema di convertire ogni dato di un file dati, i cui elementi sono stringhe di caratteri, in una linea di file testo assegnando alla linea il numero di linea specificato nel dato stesso. Se nel comando non è specificato l'operando #, i numeri di linea sono assegnati automaticamente con incremento

di 1. Il file testo prodotto è presente in memoria principale.

Il comando, con D come primo operando, ordina al sistema di creare, nella libreria di nome lib-name residente sul disco o floppy disk nell'unità con nome simbolico unit-name, un file dati di tipo sequenziale con il nome filename e di allocare per esso n byte (dove n è il primo multiplo di 256 nel caso di disco, o 128 nel caso di floppy disk, più grande del numero di byte occupati dal file testo presente in memoria principale). Ogni linea del file testo presente in memoria principale viene quindi convertita in una stringa di caratteri che è registrata nel file dati creato. Se nel comando è specificato l'operando #, il numero di linea di ogni linea del testo è compreso nella stringa registrata come elemento del file dati suddetto. Se l'operando # non è specificato, il numero di linea viene omissso nella conversione delle linee del file testo nel corrispondente elemento del file dati suddetto.

Note

1. Il comando TRANSCODE non può convertire in un file testo un file dati che è stato protetto mediante il comando SECURE.
2. Se nel file dati filename non sono presenti i numeri di linea, il comando TRANSCODE con l'operando # non è accettato ed è visualizzato un messaggio d'errore. Mentre, se il comando TRANSCODE è specificato senza l'operando #, il file dati è trascodificato in un file testo numerato automaticamente dal sistema iniziando da 1 con incremento di 1.
3. Non si può trascodificare un file testo in un file dati che deve essere registrato in una sottolibreria package protetta mediante l'esecuzione del programma di utilità LBPROTECT.
4. Se come operando lib-ref è specificato lib-name, la libreria interessata dal comando TRANSCODE è la prima libreria aperta di nome lib-name.
5. Se come operando lib-ref è specificato (,unit-name), la libreria interessata dal comando TRANSCODE è la prima libreria aperta residente sul disco o floppy disk nell'unità con nome simbolico unit-name.

6. Se l'operando lib-ref non è specificato, la libreria interessata dal comando TRANSCODE è la prima libreria aperta, secondo l'ordine di apertura.
7. Se durante la traduzione di un file dati in un file testo viene emesso un messaggio che avverte l'utente che non vi è spazio sufficiente in memoria utente per continuare l'operazione, le stringhe del file dati che sono state tradotte fino a quel momento sono presenti in memoria principale.

Esempi

1. Si converta un file testo presente in memoria principale in un file dati che sia registrato nella prima libreria aperta con il nome MAT2.

Premere

2. Si converta il file dati presente nella prima libreria aperta con il nome MAT2 in un file testo con i numeri di linea contenuti negli elementi del file dati.

Premere

Comando TRUNCATE

Funzione

Modifica la lunghezza di allocazione di un file dati sequenziale eguagliandola alla sua lunghezza attuale.

Formato

TRU[NCATE] filename [,lib-ref]

dove:

filename

indica il nome di un file dati di tipo sequenziale
lib-ref

può essere uno dei seguenti operandi:

(lib-name, unit-name)

lib-name

(,unit-name)

con lib-name che specifica il nome di una libreria
e unit-name che specifica il nome di una unità a
disco od a floppy disk

Azione

Il comando comunica al sistema di eguagliare la lunghezza di allocazione del file dati, specificato con l'operando filename, alla sua lunghezza attuale.

Note

1. Lo spazio non effettivamente occupato dal file dati è reso disponibile per altre registrazioni nella medesima libreria.
2. Se la lunghezza attuale del file dati specificato non è multiplo di 256, nel caso di disco, o 128 nel caso di floppy disk, allora lo spazio allocato dal sistema è uguale al numero di byte pari al multiplo di 256, o di 128, successivo al valore che indica la lunghezza attuale.

3. Se come operando lib-ref è specificato lib-name, il file è cercato nella prima libreria aperta di nome lib-name.
4. Se come operando lib-ref è specificato (unit-name), il file è cercato nella prima libreria aperta residente sul disco nell'unità con nome simbolico unit-name.
5. Se l'operando lib-ref non è specificato, il file è cercato nella prima libreria aperta, secondo l'ordine di apertura.
6. Il comando TRUNCATE non può essere usato con file dati protetti mediante il comando SECURE.

Esempio

Si eguagli la lunghezza di allocazione del file dati sequenziale MAT3, residente nella prima libreria aperta, alla sua lunghezza attuale.

Premere

Comando VALIDATE

Funzione

Rende nuovamente operabile un file dati rimasto aperto a seguito della terminazione anormale di un programma.

Formato

VAL [IDATE] filename [, lib-ref]

dove:

filename

specifica il nome del file da rendere nuovamente operabile

lib-ref

può essere uno dei seguenti operandi:

(lib-name, unit-name)

lib-name

(,unit-name)

con lib-name che specifica il nome di una libreria e unit-name che specifica il nome di una unità a disco od a floppy disk

Azione

Il file dati di nome filename, che è rimasto aperto in seguito alla terminazione in modo anomalo della esecuzione di un programma che predispone il file ad operazione di registrazione od aggiornamento, viene reso nuovamente operabile.

Note

1. Se filename è il nome di un file dati ad accesso sequenziale aperto in registrazione, esso è reso nuovamente disponibile per la registrazione. Se filename è il nome di un file dati ad accesso diretto che era rimasto aperto in aggiornamento, il file è reso nuovamente disponibile in lettura e registrazione. In ogni caso, se per l'effetto della terminazione anormale, alcune informazioni erano state danneggiate, queste non sono recuperabili.

2. Per determinare se un file è aperto si esegua il comando CATALOG.
3. Se come operando lib-ref è specificato lib-name, il file è cercato nella prima libreria aperta di nome lib-name.
4. Se come operando lib-ref è specificato (,unit-name), il file è cercato nella prima libreria aperta residente sul disco o floppy disk nell'unità con nome simbolico unit-name.
5. Se l'operando lib-ref non è specificato, il file è cercato nella prima libreria aperta.