

Consiglio Nazionale delle Ricerche

Descrizione del

Sistema di Misura

Hardware Monito MS32

R. Baraglia

193

**CNICE**

A cura di : Ranieri Baraglia

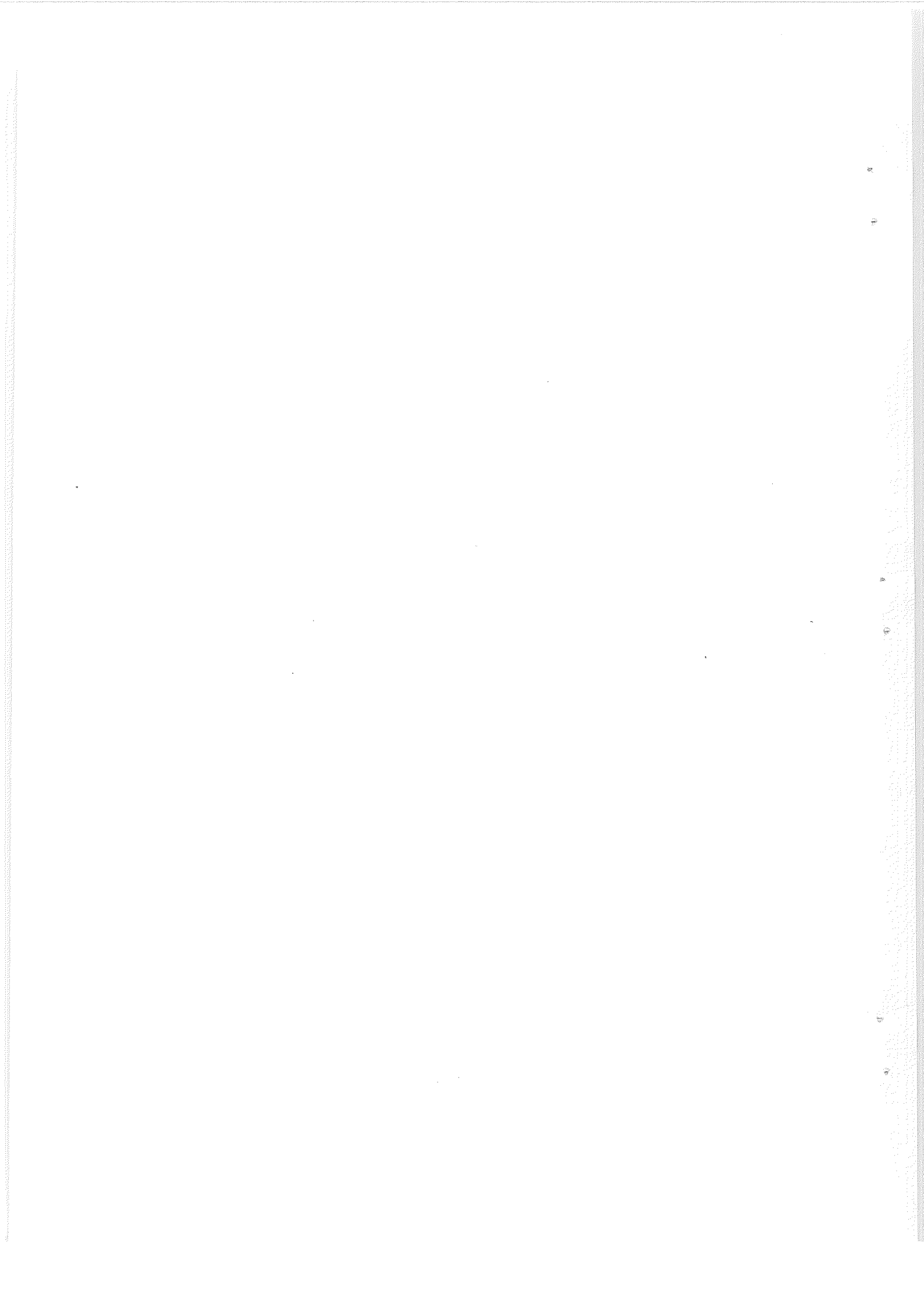
Copyright - Gennaio 1983

by - CNUCE - Pisa

Istituto del Consiglio Nazionale delle Ricerche

# INDICE

INTRODUZIONE	2
HARDWARE MONITOR CONFIGURAZIONE	3
SYSTEMA OPERATIVO TDOS	9
INIZIALIZZAZIONE DEL SYSTEMA TDOS	13
DESCRIZIONE DEI COMANDI TDOS	16
UTIL - GESTIONE FLOPPY DISK	36
COPY - COPIA FILE	41
COPY FD - COPIA FLOPPY DISK	43
FORMFD - FORMATTAMENTO FLOPPY DISK	45



## INTRODUZIONE

La presente nota da una breve descrizione del sistema di misura hardware monitor MS32, installato al CNUCE. La sua realizzazione e' dovuta, in particolare, alla necessita' di mettere a disposizione degli operatori uno strumento di rapida consultazione per l'utilizzo del sistema stesso.

8

9

10

11

12

13

## HARDWARE MONITOR CONFIGURAZIONE

Il sistema Hardware Monitor MS32 installato al C.N.U.C.E. prevede la seguente configurazione:

- 72 sonde
- 2 distributori
- 4 concentratori
- 2 patch panel
- 2 unita' floppy disk
- 1 mini-computer
- 1 terminale telescrivente

### Sonde

Le sonde o sensori permettono di rilevare segnali significativi dal sistema di elaborazione da misurare.

Le sonde sono raggruppate in coppie. Ogni sonda, tramite la sua estremita', viene collegata ad un punto preciso del computer da misurare.

L'insieme dei punti a cui le sonde possono essere collegate, in base naturalmente alla misura da fare, e' riportato in appositi manuali forniti dalla Tsdara casa costruttrice dello strumento di misura.

Ogni punto e' indicato con una sigla del tipo:

B-C2C6-B02      dove:

- B    indica il piano su cui cercare i punti da collegare. Normalmente con A viene indicato il primo piano che si incontra quando si apre l'armadio, contenente i circuiti a cui si vuole accedere.
- C2   indica la tavola o board su cui si trova il piedino a cui deve essere collegata la sonda. Di solito il board si cerca partendo dalla cerniera alla quale e' fissato il piano e contando A,B,C, da destra a sinistra, ved. fig.1.
- C6   indica la scheda all'interno del board. Ogni board e' costituito da piu' schede.

B02 indica la posizione del piedino nella scheda. Su ogni scheda ci sono cinque file di piedini di cui solo la seconda e la quarta sono usate e sono indicate con le lettere B e C rispettivamente. I piedini si contano partendo da due, in quanto il primo non e' usato, andando dall'alto in basso.

L' utilizzo previsto per le 72 sonde e' il seguente:

- 15 sonde per il controllo globale della CPU 3033
- 15 sonde per il controllo globale della CPU 3168
- 11 sonde per il controllo globale di tutti i canali
- 8 sonde per la rilevazione dei rapporti di utilizzo
- 3 sonde per il controllo del lavoro dei tamburi
- 31 sonde per il controllo delle control unit e dei device piu' usati

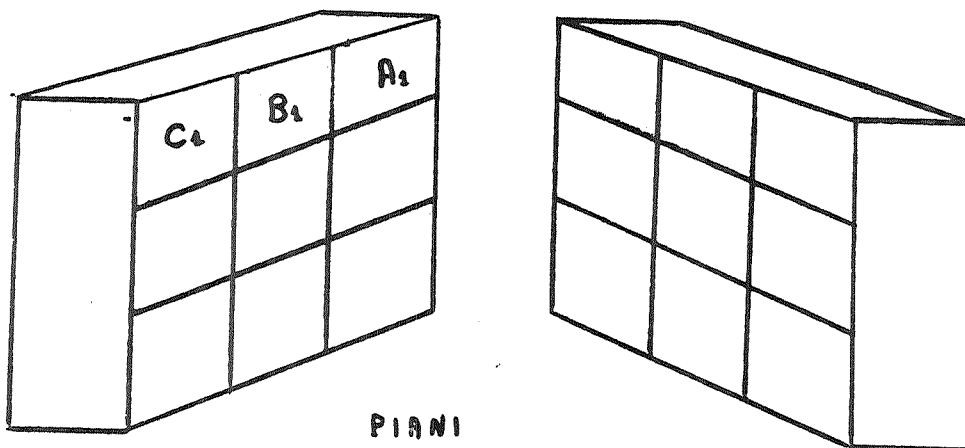


fig.1 - piani contenenti i punti di collegamento del sonde.

## Concentratori

Le sonde collegate al sistema di elaborazione da misurare sono a loro volta connesse ad unita' dette concentratori. Ad ogni concentratore possono essere collegate fino a 9 coppie di sonde. Per ognuna, il concentratore prevede un potenziometro il cui scopo e' quello di calibrare il segnale rilevato.

L'operazione di calibrazione del segnale e' necessaria in quanto i differenti tipi di computer che uno stesso hardware monitor puo' misurare lavorano con logiche diverse e quindi da ciascuno di essi, per le stesse operazioni, vengono rilevati segnali differenti. Quindi affinche' tali differenti segnali siano interpretati dallo strumento di misura, questi debbono essere convertiti in un insieme comune di valori riconosciuti dallo strumento stesso.

La calibrazione di un segnale avviene usando un suo valore di aggiustamento, che e' detto BIAS.

Il BIAS viene impostato tramite un voltmetro digitale situato nella parte posteriore dello strumento.

Da ogni concentratore parte un cavo che permette di portare, tramite i connettori, i segnali rilevati e opportunamente corretti, al patch panel.

I connettori disponibili sono 4 e si trovano nella parte posteriore dello strumento come mostrato nella fig.2.

Si possono impostare tre diversi valori bias: Va, Vb e Vc.

Gli interruttori Ib permettono di associare una o piu' coppie di sensori ad un bias, se tutti i sensori annessi ad un concentratore prevedono lo stesso bias, allora basta associare tutte le coppie di sensori usate ad uno solo dei valori Va, Vb, Vc.

Il valore Bias e' calcolato usando la seguente relazione:

$$B = |T| \times 3,3 \quad \text{dove} \quad T = R - (VL+VH)/2$$

R, VL, VH sono valori gia' precalcolati e riportati sulle tabelle descrittive i punti di collegamento di ciascun computer.

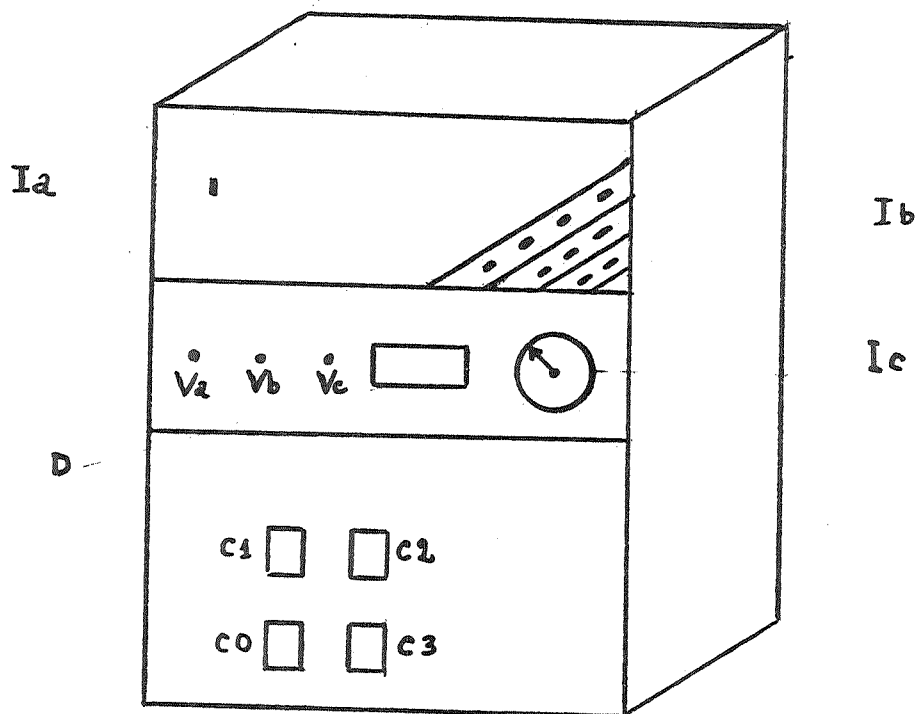


fig.2 - Ia = interruttore per attivare sensori,  
 Ib = interruttore per assegnare il bias ai sensori,  
 Ic = interruttore scelta bias e controllo tensione,  
 $V_a, V_b, V_c$  = regolatori di tensione,  $C_0, C_1, C_2, C_3$  =  
 connettori, D = display.

### Distributor

I distributor sono costituiti da 32 registri in cui vengono memorizzate le informazioni rilevate dalle sonde sotto forma di segnali elettrici.

L'input ai distributor e' controllato dal patch panel. Su quest'ultimo vi sono due distributor ognuno costituito da 16 fori che costituiscono i punti di input per i distributor stessi. I segnali inviati dai sensori al patch panel e disponibili ai fori "SENSOR", eventualmente elaborati secondo qualche funzione logica prevista dallo stesso patch panel, vengono inviati ai distributor passando per i fori omonimi.

Da questo punto in poi l'elaborazione dei dati misurati avviene sotto il controllo del minicomputer.

La frequenza con cui i dati vengono registrati nei distributor dipende dalla modalita' di rilevamento segnali scelta nella sessione di misura.

#### Patch Panel

Il patch panel e' l'unita' hardware alla quale arrivano, tramite i connettori, i segnali rilevati dai sensori. Esso e' costituito da una tavola sulla quale sono riportati 440 fori, raggruppati secondo il tipo di funzione logica che realizzano.

Il patch panel permette di combinare e/o confrontare con valori predefiniti i segnali rilevati, tramite funzioni quali ad esempio AND o OR; inoltre permette di impostare l'intervallo di tempo di campionamento. Ai due gruppi di fori denominati "SENSOR" e "DISTRIBUTOR", e' disponibile input e output, rispettivamente, del patch panel.

Con l'uso di uno o piu' ponticelli e' possibile contare i segnali rilevati nei vari circuiti logici del patch panel; l'output di quest'ultimo costituisce l'input per il distributor.

I patch panel sono situati in basso, nella parte anteriore dello strumento.

#### Unita' floppy disk

I floppy disk costituiscono l'unico supporto di memoria esterna usata dall'hardware monitor MS32 installato al CNUCE. Le due unita' floppy disk hanno, indirizzo 90 quella di sinistra (drive 0) e indirizzo 91 quella di destra (drive 1). I floppy disk che vi possono essere montati sono del tipo a singola faccia, doppia densita' e settorizzati via hardware.

#### Minicomputer

L'attivita' di misurazione e' controllata da un minicomputer a 16 bit del tipo Interdata mod. 7/16, con dimensione di memoria di 32KB.

Il codice di volta in volta in esso caricato permette l'esecuzione di operazioni quali, inizializzazione dei buffer, caricamento moduli (che sono preliminari ad una sessione di misura), esecuzione di programmi per la stampa dei risultati ecc.

Il software di base preposto alla gestione del minicomputer e' costituito dal sistema operativo TDOS.

#### Terminale

Il terminale telescrivente viene usato sia come console di operatore sia come device di stampa. Nel primo caso, dopo l'inizializzazione del sistema di misura (fase di IPL) la console e' pronta per l'immissione

dei comandi quando su di essa compare il carattere "." a colonna uno. I comandi sono sottomessi al sistema premendo il tasto RETURN. I tasti DELETE, BACKSPACE e BREAK permettono rispettivamente di cancellare una riga, di cancellare un carattere e di ritornare a capo riga. Nel secondo caso e' usato come device di output dai programmi applicativi previsti per la produzione dei report, descriventi le misure fatte.

## SYSTEMA OPERATIVO TDOS

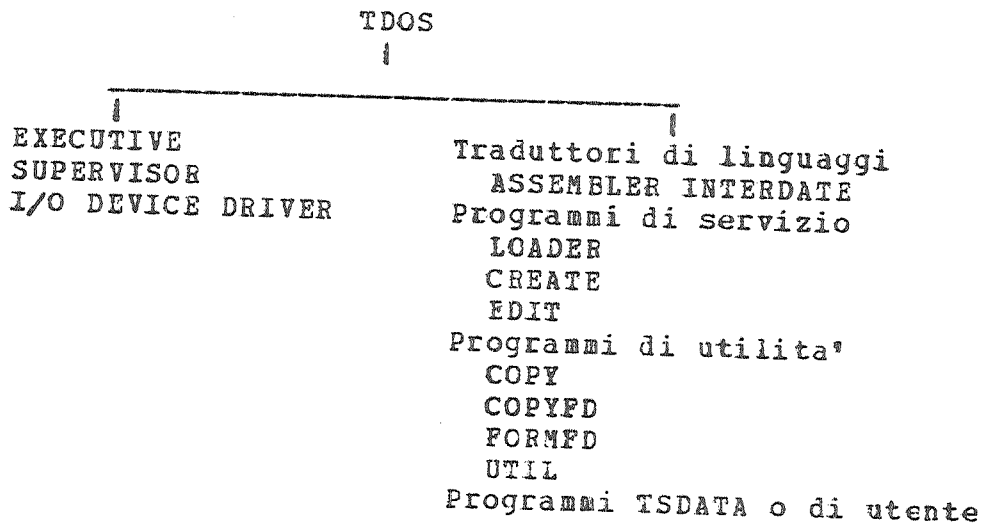
L'Hardware Monitor MS32 utilizza come software di base il sistema operativo TDOS (Tesdata Disk Operating System) versione II.

Tramite il TDOS l'utente puo' ampliare o modificare il software applicativo previsto dalla casa fornitrice del sistema per eseguire varie operazioni di misura.

Alcune sue caratteristiche principali sono:

- indipendenza dai device di input/output
- gestione di speciali interruzioni
- possibilita' di allocare file su floppy disk
- possibilita' di memorizzare piu' versioni di un file
- metodi di accesso ai file random e sequenziale
- possibilita' di creare moduli eseguibili anche con struttura overlay
- uso di procedure catalogate
- vasto insieme di comandi di operatore e di utility

I componenti del sistema TDOS possono essere suddivisi in due gruppi: programmi di controllo e programmi elaborativi che a loro volta sono suddivisi secondo lo schema seguente



### PROGRAMMI DI CONTROLLO

#### EXECUTIVE

Questo componente del sistema risiede permanentemente in memoria, le sue funzioni sono: gestione delle interruzioni,

fare da interfaccia tra i programmi di utente ed i programmi del sistema previsti per la gestione dei device di I/O e provvedere alla esecuzione delle procedure catalogate. Le interruzioni previste sono:

- supervisor call
- arithmetic fault
- illegal instruction
- machine check
- power-up

Al verificarsi di una interruzione il modulo EXECUTIVE passa il controllo alle routine dei metodi di accesso ai file nel caso di interruzione di SVC, ad un programma utente, nel caso di interruzione di arithmetic fault, oppure, negli altri casi, viene scritto un messaggio sulla console di sistema ed il controllo passa al modulo SUPERVISOR.

Il sistema TDOS prevede l'uso di uno o più file catalogo in cui vi possono essere memorizzate più procedure catalogate. Una procedura è costituita da un insieme di comandi che informano il sistema sul tipo di elaborazione da eseguire. In esse è possibile usare variabili parametriche, il cui valore viene richiesto a console ogni qualvolta viene eseguita la procedura stessa.

Le procedure vengono richiamate per l'esecuzione con un comando di operatore.

## SUPERVISOR

Il programma SUPERVISOR non risiede permanentemente in memoria, una sua copia vi viene caricata ogni qualvolta ne viene richiesta l'esecuzione. Esso mette a disposizione dell'utente un vasto insieme di funzioni, richiamabili con comandi di operatore, utili a controllare l'attività del sistema, a connettere logicamente programmi di sistema e di utente con i device di I/O e ad eseguire procedure catalogate.

Il sistema TDOS prevede un insieme di indirizzi logici, detti unità logiche, a cui di volta in volta in base alle elaborazioni da fare, debbono essere associati, tramite il loro indirizzo i device di I/O.

Le unità logiche utilizzabili con la nostra installazione sono:

UNITA' LOGICA	UTILIZZO
0	TDOS messaggi di errore
1	Input file
2	Output file
3	Print output
4	File catalogo
5	Comunicazioni all'operatore
6	Comunicazioni all'operatore

## I/O DEVICE DRIVER

Queste routine sono previste dal sistema per eseguire operazioni con i device di input/output. Ogni device di input/output e' controllato da un modulo software detto DRIVER.

I driver relativi alla console ed a floppy disk risiedono permanentemente in memoria, gli altri vengono caricati in memoria al momento del loro utilizzo.

I file memorizzati su floppy disk possono avere due formati: bloccato e non bloccato.

Per ognuno di essi e' possibile avere su di un volume piu' versioni con lo stesso nome.

La scelta del formato e' fatta dal sistema operativo e dipende dalla dimensione dei record da registrare.

### Formato bloccato

Il sistema registra con questo formato i file i cui record hanno dimensione minore o uguale a 255 byte. In questo caso la dimensione di un record fisico e' uguale alla dimensione di un settore del floppy disk; un numero variabile di record logici viene memorizzato in un record fisico. In ogni record logico e' previsto un byte in cui e' memorizzata la lunghezza del record stesso. Ai file registrati secondo il formato bloccato vi si puo' accedere solo in modo sequenziale.

### Formato non bloccato

Questo formato e' usato per tutti i file i cui record sono piu' lunghi di 255 byte.

Se la lunghezza dei record e' minore di un multiplo della dimensione di un settore di floppy disk lo spazio e' sprecato alla fine del settore stesso. Tutti i record iniziano dal byte 0 di un settore, e l'accesso ad essi puo' essere sia sequenziale che random come definito tramite un comando /AL o /AT descritti nel seguito.

Con il metodo di accesso random si deve specificare a programma il numero e la dimensione del record che deve essere letto o scritto. I record sono numerati a partire da zero. Mentre per i file sequenziali la fine del file corrisponde all'ultimo record registrato nel file stesso, per i file random la dimensione di un file e' definita con il parametro SIZE previsto nei comandi /AL o /AT.

Se un floppy disk e' nuovo prima di potervi allocare file esso deve essere formattato (utility FORMFD) e successivamente inizializzato (utility UTIL). Con l'operazione di inizializzazione, viene assegnato un nome al volume floppy disk e vi viene allocata la VIOC o directory di volume. Quest'ultima e una sua copia di backup

sono registrate nella traccia zero. La VTOC puo' richiedere piu' di una traccia, cio' dipende dal tipo di supporto usato e dal numero di entrate di file previste nella VTOC stessa. Quest'ultimo valore puo' essere impostato dall'utente al momento della inizializzazione di un volume. In questa fase vengono anche rilevate le tracce rotte, cioe' quelle tracce che non e' piu' possibile leggere dopo una operazione di scrittura, ed al loro posto vengono usate tracce alternate previste in un data set di nome \$\$\$BAD.

#### PROGRAMMI ELABORATIVI

A differenza dei programmi di controllo che sono interamente gestiti dal sistema operativo in modo quasi automatico, i programmi elaborativi sono quelli la cui elaborazione e' richiesta all'utente con l'uso di alcuni comandi previsti dal modulo SUPERVISOR.

Riportiamo di seguito una breve descrizione dei programmi elaborativi che non sono trattati in questa nota.

**ASSEMBLER** permette di convertire il modulo sorgente, memorizzato secondo la codifica ASCII, in un modulo oggetto rilocabile con o senza riferimenti esterni o moduli assoluti di tipo Interdata(1).

**LOADER** permette di convertire un modulo oggetto rilocabile, in un modulo eseguibile oppure in un modulo oggetto del tipo Interdata.

**CREATE** permette di costruire un modulo eseguibile, da un modulo oggetto del tipo Interdata.

**EDIT** e' un editor interattivo che permette di creare o modificare file registrati secondo la codifica ASCII e costituiti da record con lunghezza minore od uguale a 131 byte. Tali file possono essere procedure catalogate, programmi sorgenti o di altro tipo.

---

NOTA 1) Un modulo di tipo Interdata e' un modulo assoluto non rilocabile.

## INIZIALIZZAZIONE DEL SYSTEMA TDOS

La TSDATA fornisce il sistema operativo memorizzato su due floppy disk labellati SYSRES e SYSMES. Una copia di alcuni programmi si trova memorizzata su ambedue i dischetti. L'inizializzazione del sistema di misura, fase di IPL, puo' essere fatta utilizzando uno qualsiasi dei due floppy disk prima detti.

### Procedura di IPL

La prima operazione da eseguire, dopo aver acceso lo strumento consiste nella inizializzazione del sistema TDOS. Questa fase permette di caricare, nella memoria del minicomputer, una copia del sistema TDOS da floppy disk e di eseguire la procedura di restart. Per eseguire l'IPL, montato il dischetto SYSRES o SYSMES sul driver 0, premere i seguenti tasti sul pannello di controllo(1):

```
DTA
5-0 (prima 5 e poi 0)
ADD
INI
AUTO-LOAD(2)
RUN
```

A questo punto ha inizio la procedura di re-start la quale:

- determina la configurazione della memoria
- cancella eventuali operazioni pendenti
- carica e trasferisce il controllo al modulo supervisor del TDOS
- attiva il file CAT assegnandolo alla unita' logica numero 4
- richiede a console la data e l'ora

Se tali operazioni sono eseguite con successo compaiono a console i seguenti messaggi (3):

---

NOTA 1) Il pannello e' disattivato quando la chiave si trova in posizione LOOK.

NOTA 2) Il bottone si trova dentro lo sportellino dei dischetti sul lato destro.

NOTA 3) Nel seguente esempio le righe battute dall'operatore sono quelle che iniziano con il punto.

\*\*\*INITIAL PROGRAM LOAD\*\*\*

MS32 TDOS=2;01

PLEASE ENTER DATE, TIME AS MM/DD/YY,HHMMSS

.05/05/82,112200 ==> in questa risposta si puo' omettere il numero dei secondi

\*\*\*SYSTEM READY\*\*\*

Quest'ultimo messaggio indica che la procedura di IPL e' terminata ed il sistema e' pronto per l'elaborazione. Se l'IPL del sistema non termina correttamente, allora bisogna rimemorizzare il programma di autoload. Esso e' costituito dal numero minimo di istruzioni richieste per iniziare il caricamento del sistema TDOS da floppy disk. Il programma di autoload e' il seguente:

Locazione	Contenuto esadecimale
22	8000
50	24F0
52	95EF
54	D500
56	0180
58	4300
5a	80
78	9002

Per memorizzare il programma autoload si premono i seguenti tasti del pannello di controllo:

```
5-0          ==> locazione di indirizzo 50
ADD
DTA
24F0 (2-4-F-0) ==> contenuto da memorizzare nella
                    locazione di indirizzo 50
WRT
```

Nello stesso modo si memorizzano le altre situazioni.

Per controllare quanto memorizzato nelle varie locazioni, usare i seguenti tasti:

```
DTA
5-0 (indirizzo di memoria di cui vogliamo vedere il
    contenuto)
ADD
RD
RD
"
```

Nella parte del pannello di controllo indicata con DA (Display Address) viene mostrato l'istruzione counter che punta sempre alla prossima locazione di memoria da leggere.

Nella parte indicata con DD (Display Data) viene mostrato il contenuto della locazione di memoria richiesta.

Dopo avere ricostruito il programma di autoload si puo' rieseguire la procedura di IPL.

## DESCRIZIONE DEI COMANDI TDOS

I comandi che il sistema operativo TDOS mette a disposizione dell'utente possono essere distinti in tre gruppi: comandi per il controllo del sistema, per la gestione dei floppy disk e per il caricamento ed esecuzione dei moduli eseguibili.

I comandi vengono trasmessi al sistema tramite la console oppure da file memorizzati su floppy disk, in tal caso essi fanno parte di una procedura catalogata.

Un comando è costituito da una parola chiave e da uno o più parametri. Come carattere di separazione tra un campo e l'altro del comando è possibile usare la virgola, lo spazio o l'eguale. Ad esempio sono accettati i comandi: ASb290; AS,290; AS=290.

Il modulo di sistema che verifica l'esattezza di un comando esamina solamente il suo codice mnemonico e ignora, accettandolo, qualsiasi carattere che segue tale codice. Esso verifica che il codice mnemonico sia esatto, e che tra ogni campo compaia un carattere di separazione permesso. Ad esempio i comandi: /LI C, /LIST C e /LISTTT C sono accettati dal sistema.

Quando i comandi sono passati al sistema tramite una procedura catalogata, questa viene letta dal file catalogo il quale deve essere sempre stato aperto (comando /AC).

### Comandi di controllo

comando: ASSIGN

Permette la connessione tra una unità logica, con la quale un programma riconosce un device di I/O, e l'indirizzo fisico del device stesso. Quanto detto vale per tutti i device di I/O ad esclusione dei device dischetto magnetico.

formato:

AS uhh

u = unità logica, valori possibili da 1 a 9  
hh = indirizzo fisico del device

osservazioni:

Entrambi i valori dei parametri sono espressi in notazione esadecimale. Gli indirizzi dei device di I/O previsti dalla nostra installazione sono i seguenti:

Indirizzo	Device
00	Null device
12	Console
90	Floppy disk-Drive 0
91	Floppy disk-Drive 1

La console e' di norma associata alla unita' logica numero 3. Essa, nel nostro caso e' usata anche come stampante per la produzione di report.

Null device puo' essere usato per rendere dummy una operazione di I/O.

comando: BIAS

Permette di specificare un indirizzo, in notazione esadecimale che deve essere aggiunto all' indirizzo specificato in un comando OPEN per determinare l'indirizzo che deve essere aperto.

formato:

BI hhhh

hhhh = indirizzo di memoria

osservazioni:

Quando il sistema e' inizializzato l'indirizzo del bias e' impostato a zero. Questo comando viene usato per realizzare i moduli eseguibili.

comando: ECHO

La funzione attivata con l'esecuzione di questo comando fa si che quanto viene scritto a console compaia anche sul device di stampa.

formato:

EC f

f = 1, funzione attiva

f = 0, funzione non attiva

osservazioni:

La nostra installazione prevede come console e come stampante lo stesso device, il cui indirizzo e' uguale a 12 esadecimale.

comando: CONTINUE

Permette di riprendere l'esecuzione di un programma la cui elaborazione e' stata temporaneamente interrotta.

formato:

CO

osservazioni:

Il sistema TDOS prevede la possibilita' di interrompere l'elaborazione di un programma (SVC 2), eseguire dei comandi e riprendere l'esecuzione con il comando CONTINUE.

comando: CLOSE CATALOG

Questo comando permette di chiudere l'unita' logica numero 4. A quest'ultima e' associato il file catalogo del sistema.

formato:

CC

osservazioni:

Il file catalogo del sistema contiene le procedure catalogate per l'esecuzione di programmi di utente o di sistema. Ogni qualvolta il catalogo viene chiuso, esso dovra' essere riaperto quando si vuole eseguire un comando

" > ", il quale permette di richiamare per l'esecuzione una procedura catalogata.

comando: DATE

Permette di impostare e/o variare la data nel hardware monitor. La data specificata comparirà in ogni report prodotto.

formato:

DA mm/dd/yy mese/giorno/anno

comando: DISPLAY TIME/DATE

Stampa la data e l'ora correnti sulla console.

formato:

DT

comando: EJECT

Permette di avanzare di una pagina.

formato:

EJ

osservazioni:

L'unità logica interessata è la numero 3 a cui è associata l'unità periferica di stampa.

comando: ERROR PROCEDURE NAME

Permette di specificare il nome di una procedura catalogata che deve essere eseguita se si verifica un errore di sistema oppure quando viene interrotta dall'utente l'esecuzione di un programma.

formato:

EP aaaa

aaaa = nome di procedura

osservazioni:

La procedura da eseguire viene cercata nel file catalogo quindi esso deve essere precedentemente attivato (comando /AC). Il catalogo e' associato alla unita' logica numero 4. In questo caso i risultati prodotti vengono stampati sulla unita' logica numero 0.

comando: FREE

Questo comando permette di liberare la memoria occupata da un driver non residente permanentemente in memoria, cancellando il driver dalla memoria stessa.

formato:

FR aa

aa = codice di edintificazione driver

osservazioni:

L'effetto di questo comando e' nullo per il sistema installato al C.M.U.C.E. perche' esso prevede i due soli driver che debbono risiedere permanentemente in memoria.

comando: GO

Esegue la stessa funzione del comando CONTINUE.

formato:

GO

comando: MAP

Questo comando stampa a console la descrizione di ogni buffer di I/O e di ogni driver presente al momento in memoria. Esso permette di valutare in qualsiasi momento l'occupazione della memoria.

formato:

MAP

osservazioni:

Come abbiamo visto ogni device di I/O e' controllato da un modulo software detto driver.

Nella descrizione che si ottiene con questo comando non compaiono i driver relativi alla console e ai floppy disk. Questi sono gli unici driver che risiedono permanentemente in memoria.

Lo strumento installato attualmente al CNUCE ha come unici device di I/O la console e i floppy disk, quindi nel nostro caso con questo comando si ottengono solo informazioni relative ai buffer di memoria che sono:

FMA = indirizzo assoluto di inizio buffer

LMA = indirizzo assoluto di fine buffer

CODE = codice driver o I/O BUFFER

comando: OPEN

Mostra il contenuto di una locazione di memoria.

formato:

OP hhhh

hhhh = indirizzo di memoria

osservazioni:

Ogni locazione di memoria ha dimensione 2 byte, ed il suo contenuto e' stampato in notazione esadecimale. Quando il sistema viene caricato in memoria, fase di IPL, l'indirizzo di open e' impostato a zero. Questo comando prevede i parametri + e -, che sono usati per stampare la locazione di memoria successiva e precedente a quella specificata nel comando OPEN.

comando: POWER FAIL PROCEDURE

Permette di specificare il nome di una procedura catalogata che deve essere eseguita al momento del restart dopo caduta del sistema operativo.

formato:

PF aaaa

aaaa = nome di procedura catalogata

osservazioni:

Valgono le osservazioni viste per il comando ERROR PROCEDURE NAME.

comando: REPLACE

Permette di variare il contenuto di una locazione di memoria precedentemente aperta con il comando OPEN.

formato:

RE hhhh

hhhh = nuovo valore

osservazioni:

L'esecuzione di questo comando di solito e' preceduta dalla esecuzione di un comando OPEN.  
L'uso dei tre comandi OPEN, REPLACE, BIAS permette di esaminare e variare il contenuto della memoria. Cio' rende possibile memorizzare programmi, scritti in codice assoluto, tramite la console.

comando: START

Permette di specificare un indirizzo di memoria a cui il sistema operativo deve trasferire il controllo. Questo indirizzo costituisce l'indirizzo della prima istruzione di un programma precedentemente caricato in memoria.

formato:

ST hhhh

hhhh = indirizzo di memoria

osservazioni:

Il sistema TDOS prevede la costruzione di moduli eseguibili sia in formato rilocabile che non rilocabile. Quest'ultimi una volta caricati in memoria (comando /DL) vengono eseguiti con un comando START.

L'indirizzo di memoria da specificare in questo comando viene determinato al momento della creazione del modulo oggetto (programma ASSEMBLER). Esso compare anche, campo Fwa, nel report che si ottiene con il comando LIST.

comando: TIME

Permette di impostare il clock interno dello strumento di misura.

formato:

TI hhmmss

hhmmss = ore/minuti/secondi

comando: >

Questo comando e' usato per eseguire una procedura catalogata.

formato:

>aaaa

aaaa = nome procedura catalogata.

osservazioni:

La procedura specificata e' cercata nel file catalogo.

Questo comando puo' essere dato da console oppure puo' far parte di una procedura catalogata.

### Comandi gestione floppy disk

Descrizione dei parametri previsti da questo insieme di comandi.

aaaaaa nome del file o label di un volume dischetto magnetico

u unita' logica a cui e' associato il file. Valori possibili da 1 a 9.

Class indica la classe a cui puo' essere associato un file. Valori ammessi 0-9,A-S,U-Y.

Date permette di variare la data

Direct permette di specificare il codice di un modulo eseguibile. Viene usato per il caricamento in memoria di moduli eseguibili.

END permette il posizionamento alla fine di un file. Valore default 0 .

First Byte definisce l'indirizzo del primo byte di memoria per un modulo eseguibile

Last Byte definisce l'indirizzo dell'ultimo byte di memoria per un modulo eseguibile

No Buf permette la allocazione di memoria per i buffer di I/O. Questi sono richiesti solamente per i file registrati secondo il formato bloccato valore default 0.

Protect Code definisce il codice di protezione per una specifica classe di file. Questo parametro e' richiesto se la classe a cui appartiene il file prevede un codice di protezione, ogni qualvolta si vuole accedere al file per apportarne delle modifiche.

Protect New Class permette di assegnare un nuovo codice di protezione ad una classe di file.

Pack definisce l'indirizzo fisico della unita' disco, nella quale il floppy disk e' montato. Se questo parametro non viene specificato il suo

valore default e' 90 (Drive 0). Su tale unita' viene normalmente montato il disco di sistema.

- RElease** indica se lo spazio allocato ad un file e non usato deve essere rilasciato. L'operazione di rilascio dello spazio non usato viene eseguita al momento della esecuzione del comando CLOSE.
- SIze** indica, in KB, lo spazio che deve essere allocato ad un file.  
Lo spazio allocabile e' limitato dalla disponibilita' di spazio contiguo disponibile nel volume disco.  
L'allocazione dello spazio prevede l'arrotondamento per eccesso con allineamento alla traccia.
- TYpe** definisce il tipo di accesso ad un file.  
Valori ammessi: 0 = accesso sequenziale, 1 = accesso random. Valore default 0.
- VERsion** indica la versione di un file. Per ogni file e' possibile averne piu' versioni sullo stesso volume. Valori possibili da 1 a 255.
- VNew** permette di cambiare la versione di un file. Valori possibili da 1 a 255.
- VOlume** indica il nome del volume disco nel nostro caso del floppy su cui il file e' allocato.

I formati ammessi dai parametri previsti dai comandi elencati nel seguito sono:

f = flag (1=si,0=no)  
d = cifra decimale  
a = carattere alfanumerico  
h = cifra esadecimale

Il numero di lettere a, d e h rappresentano il numero di caratteri ammessi nella definizione di ciascun parametro.

#### Descrizione dei comandi

##### comando: ACTIVATE

Permette di attivare, cioe' di associare ad un certa unita' logica un file precedentemente allocato su disco magnetico. In tal modo si rende disponibile un file ai programmi che lo elaborano.

formato:

```
/AC aaaaaa,u,EN = f,  
      NB = f,  
      PC = aa,  
      PK = hh, n  
      RL = f,  
      VE = ddd,  
      VO = aaaaa
```

osservazioni:

I programmi accedono ai file tramite l'indirizzo di una unita' logica.

E' possibile associare ad una unita' logica un solo file per volta, quindi terminata l'elaborazione di un file per utilizzarne un altro sulla stessa unita' logica, il primo deve essere chiuso (comando /CL).

Il file contenente il catalogo del sistema deve sempre essere assegnato all'unita' logica numero 4.

Se si hanno piu' cataloghi, ogni volta che si vuole passare da un catalogo ad un altro, l'assegnazione corrente deve essere chiusa e deve essere creata la nuova assegnazione.

Le unita' logiche ed il loro uso possibile, relativamente ai dispositivi di I/O di tipo floppy disk, previste dalla nostra installazione sono:

unita' logica	uso
1	file in input
2	file in output
4	file catalogo

comando: ALLOCATE

Permette la allocazione di un file su un volume floppy disk ed associa tale file ad una specifica unita' logica.

formato:

```
/AL aaaaaa,u,CL = a,  
NB = f,  
PC = aa,  
PK = hh,  
RE = f,  
SI = f,  
TY = f,  
VE = ddd,  
VO = aaaaaa
```

osservazioni:

Lo spazio richiesto e' allocato in Kb ed e' arrotondando per eccesso ad un multiplo di 8.  
Ad esempio per valori del parametro SI che vanno da 1 a 8 lo spazio riservato e' 8Kb, mentre per valori che vanno da 9 a 16 lo spazio riservato e' di 16 Kb.  
E' consigliato l'uso del parametro RL, il quale permette di rilasciare lo spazio allocato ma non usato.  
IL parametro VE permette l'allocazione di file con lo stesso nome sullo stesso volume, che comunque sono distinti gli uni dagli altri in base alla loro versione.

comando: ATTRIBUTE

Permette di definire o modificare gli attributi di un file precedentemente allocato su dischetto magnetico.

formato:

```
/AT aaaaaa,u,CL = a,  
DA = dddy,  
DI = a,  
FB = hhhh,  
LB = hhhh,  
PC = aa,  
PK = hh,  
TY = f,  
VE = ddd,  
VN = ddd
```

osservazioni:

In ogni momento della vita di un file si possono apportare variazioni ai valori dei suoi attributi.

Puo' essere utile variare la classe del file qualora lo si voglia inserire in una classe di file soggetti ad operazioni comuni.

Quando si voglia variare la versione, debbono essere definiti ambedue i parametri VE e VN. Se invece gli attributi del file non prevedono alcun valore per la versione e' sufficiente specificare il parametro VN. Con il parametro DA e' possibile variare la data di creazione di un file. Essa e' espressa nel formato:

- ddd = numero progressivo del giorno nell'anno
- y = ultima cifra dell'anno corrente

Se la data impostata non corrisponde a nessun giorno dell'anno, viene ignorata, ed essa non comparira' nella lista degli attributi associati al file.

comando: CLOSE

Permette di chiudere tutti i file, precedentemente aperti con i comandi ACTIVATE o ALLOCATE, oppure solo i file associati con una specifica unita' logica.

formato:

```
/CL u,---,u
```

```
o
```

```
/CL All
```

osservazioni:

L'esecuzione del comando secondo il primo formato chiude tutti i file associati alle unita' logiche specificate nel comando stesso.

Il parametro ALL, secondo formato, comporta la disattivazione di tutti i file.

Un file al termine di una sua elaborazione deve essere sempre chiuso, altrimenti, se usato in operazioni di scrittura, non sara' possibile usarlo in elaborazioni successive.

comando: DELETE

Permette di cancellare un file allocato su un volume dischetto magnetico.

formato:

```
/DE aaaaaa,PC = aa,  
    PK = hh,  
    VE = ddd,  
    VN = ddd,  
    VC = aaaaaa
```

osservazioni:

I file possono essere raggruppati in classi e ad ognuna di esse puo' essere assegnato un codice di protezione. In tal caso il codice di protezione associato alla classe deve comparire nel comando DELETE.

comando: LIST

Permette di ottenere i seguenti report:

- lista dei nomi delle procedure catalogate
- lista degli attributi dei file appartenenti ad una classe
- lista degli attributi dei file memorizzati su un volume dischetto magnetico

formato:

```
/LCAT  
  
0  
  
/LI CL = aa,  
    PK = hh,  
    VO = aaaaaa
```

osservazioni:

Se i parametri PK e VC non sono definiti il volume considerato e' quello montato sull'unita' floppy disk di indirizzo 90 (DRIVE 0). L'esecuzione del comando secondo il primo formato produce la lista del directory del file catalogo.

Gli attributi di file che compaiono in una lista sono:

- nome file
- versione
- spazio allocato
- spazio non usato
- classe a cui appartiene il file
- modalita' di accesso al file
- indirizzo primo byte, solo per file contenenti moduli eseguibili
- indirizzo ultimo byte, solo per file contenenti moduli eseguibili
- codice assegnato al modulo eseguibile
- traccia di inizio memorizzazione file su floppy disk
- data di allocazione file

comando: LOGICAL UNIT QUERY

Permette di stampare le associazioni, valide al momento, tra device e unita' logiche.

formato:

/LU

osservazioni:

Un controllo di questo tipo e' particolarmente utile prima di operazioni, quali ad esempio quelle di copia che prevedono associazioni rigide tra unita' logiche e device di I/O.

comando: QUERY

Produce un report riportante lo stato di un floppy disk.

formato:

/QU VO = aaaaaa,  
PK = hh

osservazioni:

Le informazioni che si ottengono con l'esecuzione di questo comando sono:

- numero di entrate di file libere
- numero di entrate di file occupate
- numero di errori di lettura
- numero di errori di posizionamento
- numero di errori di cui non e' stato possibile fare il recovery

Al momento della inizializzazione di un floppy disk viene definito il numero massimo di entrate di file, cioe' il numero massimo di file che possono venire allocati su di un floppy disk.

comando: PARAMETER

Permette di specificare, cancellare, verificare od aggiungere parametri in una procedura catalogata.

formato:

```
/PA  PP=aaaa,pp=aaaa,-----  
o  
/PX  pp=aaaa,pp=aaaa,-----  
o  
/PX  pp  
o  
/PV  pp,x,x,x,x,x,x,xxx,(PROC)
```

dove :

aaaa = numero variabile di byte costituenti il valore del parametro

pp = nome del parametro

x = campo alfanumerico che deve essere confrontato per uguale prima di sostituire il parametro pp.

(PROC) = nome della procedura catalogata che deve essere eseguita se l'esito del

confronto risulta negativo. Parametro  
opzionale.

osservazioni:

Le procedure catalogate prevedono l'uso di valori parametrici. In esse i parametri hanno la forma: &pp, dove pp e' il nome del parametro di lunghezza due byte, oppure @@@ ---, dove i caratteri speciali @ indicano il numero di caratteri di cui e' costituito il parametro.

Quando una procedura catalogata viene eseguiti i parametri del tipo &pp, vengono cercati, tramite il loro nome nella tabella di sistema PARMT. In essa i parametri vi sono stati precedentemente memorizzati con i comandi /PA o /PX.

I parametri del tipo @@@... vengono passati alla procedura tramite la console.

/PA permette di memorizzare parametri e loro valori nella tabella PARMT. Ogni esecuzione di comando comporta prima la cancellazione e poi la ricostruzione della tabella PARMT.

/PX permette di aggiungere un parametro nella tabella, aggiungendo una entrata nella tabella stessa.

/PD permette la cancellazione di un parametro dalla tabella PARMT. Ad ogni cancellazione la tabella viene compattata.

/PV permette di verificare il valore assegnato ad un parametro.

comando: PROTECT

Permette di specificare il codice di protezione da associare ad una classe del file.

formato:

```
/PR CL = a,  
    PC = aa,  
    PK = hh,  
    PN = aa,  
    VO = aaaaaa
```

osservazioni:

Il codice di protezione associato ad una classe di file dovrà essere specificato in ogni comando la cui esecuzione comporta una variazione ad un componente della classe stessa. Il parametro PC e' richiesto in questo comando soltanto quando si vuole variare il codice di protezione di una classe.

## Comandi per la gestione dei moduli eseguibili

comando: //aaaaaa

Permette il caricamento e l'esecuzione di un modulo eseguibile memorizzato su un volume dischetto magnetico oppure nella tabella di sistema dei moduli caricabili.

formato:

```
//aaaaaa
```

aaaaaa = nome del modulo eseguibile

osservazioni:

Un load module viene creato usando nell'ordine i seguenti programmi: Assembler, Loader e Create.

comando: //CREATE

Questo programma permette di convertire un modulo assoluto in formato INTERDATA in un load module.

formato:

```
//CREATE DI = d,  
        FB = hhhh,  
        LB = hhhh,
```

osservazioni:

L'input di questo programma e' costituito dall'output ottenuto con l'esecuzione del programma LOADER.  
Il parametro DI permette di assegnare un codice, detto codice di caricamento o di load, ad un modulo eseguibile. Esso e' usato nel comando DLOAD per caricare in memoria uno o piu' moduli prima che ne venga richiesta l'esecuzione. Le unita' logiche usate dal CREATE sono:

unita' logica	utilizzo
1	input file - modulo INTERDATA
2	output file - load module
5	console

comando: DLOAD

Permette di memorizzare in una tabella di sistema (Direct Load Table) il codice assoluto di programmi o parte di programmi usati in overlay, aventi tutti lo stesso codice di caricamento.

formato:

```
/DLOAD  DI = a  
        PK = hh  
        VO = aaaaaa
```

osservazioni:

Il codice di load viene definito al momento della creazione del modulo eseguibile (programma CREATE) oppure con il comando ATTRIBUTE. Ad ogni esecuzione di questo comando, vengono cancellate tutte le entrate precedentemente memorizzate nella tabella DLT. Questo metodo fa sì che al momento della esecuzione di un programma, il sistema non debba esaminare il directory dei volumi floppy disk per individuarlo.

## UTIL - GESTIONE FLOPPY DISK

Il programma di utilita' UTIL permette la gestione dei volumi disco e dischetto magnetico. Esso prevede diversi comandi per eseguire dump di file da disco a nastro, per cancellare e rinominare file. Di questo insieme di comandi solo quelli utili a cancellare e rinominare file sono disponibili presso la nostra installazione in quanto essa non prevede ne unita' disco ne unita' nastro magnetico.

Nel nostro caso salvare un file significhera' farne una copia sullo stesso o su un altro floppy disk. Sara' cura dell'utilizzatore dello strumento di misura far si che qualora si guasti un volume od un file ne sia sempre disponibile una sua copia funzionante.

Questo programma non prevede nessuna associazione tra unita' logiche ed i file da elaborare. Esso riconosce i file tramite il loro nome.

Una copia del programma UTIL e' memorizzata su entrambi i dischetti di sistema SYSREP e SYSMES. Dopo aver inserito nella unita' floppy di indirizzo 90 (drive 0) uno dei due dischetti di sistema, il programma puo' essere caricato in memoria per l'esecuzione con il comando:

//UTIL

L'esecuzione di ogni comando si basa sul valore dei parametri previsti nel comando stesso.

Quando viene passato un comando errato, l'elaborazione del programma termina e deve essere ripresa dall'inizio.

### Descrizione dei parametri:

aaaaaa	nome di file o label di volume, lunghezza prevista da 1 a 6 byte.
All	indica che tutte le versioni di un file presenti su un volume disco magnetico sono coinvolte nella operazione richiesta.
Class	indica la classe di appartenenza di un file.
First Track	indirizzo della prima traccia, da cui ha inizio un file
Name	nome del file corrente, lunghezza prevista da 1 a 6 byte.
New Name	nuovo nome da assegnare ad un file, lunghezza prevista da 1 a 6 byte.

**Pack** indirizzo fisico del device su cui e' montato il floppy disk da elaborare, valore default 90 (driver 0).

**Protect Code** codice di protezione associato ad una classe di file. Permette di proteggere il file da operazioni che tendono a modificarlo.

**Version** indica la versione del file, valori previsti da 1 a 255.

**Volume** nome del volume disco su cui e' allocato il file da elaborare.

I formati ammessi dai parametri previsti dai comandi elencati nel seguito sono:

f = flag (1=si,0=no)  
d = cifra decimale  
a = carattere alfanumerico  
h = cifra esadecimale

Il numero di lettere a, d e h rappresentano il numero di caratteri ammessi nella definizione di ciascun parametro.

Descrizione dei comandi

comando: FILE NAME CHANGE

Permette di assegnare un nuovo nome ad un file o a tutti i file di un classe.

formato:

FN newname,  
NA = aaaaaa,  
AL = f,  
VE = dddd,  
VO = aaaaaa,  
PK = hh,  
PC = aa,  
FT = hhhh

osservazioni:

Il parametro VE permette di individuare un file quando di questo ne esistono piu' versioni sullo stesso volume.

Quando il parametro AL e' definito uguale a 1 la variazione interessa tutti i file appartenenti ad un classe. Se quest'ultima prevede un codice di protezione questo deve essere specificato.

Il parametro FT e' sufficiente ad individuare univocamente un file; i parametri NA, AL e VE se usati con esso sono ignorati. Se il volume richiesto e' montato sull'unita' disco di indirizzo 90 (drive 0) i parametri PK e VO non vanno definiti.

comando: INITIALIZE

Permette di inizializzare un volume floppy disk.

formato:

```
IN  aaaaaa
    BY = f
    DS = d
    PC = aa
    PK = hb
    VO = aaaaaa
```

osservazioni:

L'inizializzazione di un floppy disk consiste nella registrazione di un nuovo directory di volume, senza file allocati sul volume stesso.

Se il volume da inizializzare presenta un directory efficiente, allora il nome del volume viene stampato a console e viene richiesta all'operatore la conferma a continuare l'operazione di inizializzazione. Cio' permette di evitare errori nella scelta del volume da inizializzare.

I valori previsti per il parametro DS sono 1, 2 o 3 ai quali corrispondono rispettivamente 230, 486 e 721 entrate di file. Piu' entrate di file sono richieste piu' spazio viene riservato per il directory.

Il parametro BY, se non definito, permette di rilevare eventuali tracce rotte e quindi non utilizzabili. Ogni traccia rotta viene resa non usabile, allocandovi un file dummy. Questo tipo di controllo e' sempre consigliato. Se il parametro BY e' attivo ogni errore rilevato in scrittura per l'intero volume non viene considerato. Quando il volume da inizializzare e' nuovo esso deve essere prima formattato (programma FORMFD). Se il volume da inizializzare e' montato sull'unita' disco di indirizzo 90 (drive 0) i parametri PK e VO non debbono essere specificati.

comando: PURGE

Permette la cancellazione da un floppy disk di tutti i file appartenenti ad una classe

formato:

PU CL = a  
PC = aa  
PK = hh  
VC = aaaaaa

osservazioni:

In questo comando bisogna specificare la classe di file interessata alla cancellazione e l'eventuale codice di protezione ad essa associato.

I parametri PK e VO non debbono essere usati se l'unita' disco interessata e' quella con indirizzo 90 (drive 0).

comando: PACK MAINTENANCE

Permette l'eliminazione di file temporanei e di vecchie versioni di file memorizzati su un volume dischetto.

formato:

PM AL = f  
PC = aa  
PK = hh  
VO = aaaaaa

osservazioni:

Sono coinvolti nella operazione di cancellazione tutti i file appartenenti alle classi T e \*.

comando: EXIT

Termina l'esecuzione del programma UTIL

formato:

```

TITLE 'HASP USER EXIT 5 -- PROLOG (LOCAL MACRO)'
MACRO -- $CFCVB -- CONVERT TO BINARY
&NAME $CFCVB &TYPE=CALL,&POINTER=(R1),&NUM=2,&INFO=NO,&NOK=, C
      &MAX=9999

```

```

      LCLC &R
      LCLA &CNT
&R SETC 'ESYSNDX'
      AIF ('&INFO' EQ '' OR '&INFO' EQ 'NO')-Z
      AIF ('&TYPE' EQ 'RES')-NOJECT
      EJECT
.NOJECT SPACE 2

```

```

*****

```

```

*
*       COFCVB -- CONVERT TO BINARY
*
*       ROUTINE CONVERTS A PAIR OF NUMBERS OF THE FORM
*
*       TEXT1-N2 WHERE
*       TEXT = OPTIONAL TEXT IDENTIFIERS JOB, PRT, RM ETC.
*       N1   = FIRST OF A SERIES OF NUMBERS LT &MAX IN VALUE
*       N2   = OPTIONAL LAST OF A SERIES OF NUMBERS LT &MAX.
*       IN A SERIES OF VALUES, N1-N2-N3-N4, THE LAST TWO VALUES
*       ARE CONSIDERED TO BE START-STOP VALUES.
*       THE MEANING OF THE START-STOP VALUES FOR EXAMPLE 1-5 ARE
*       THE USER DESIRES AN OPERATION PERFORMED ON JOB OR FACILITY
*       TYPE INDICATED, NUMBERS 1, 2, 3, 4, AND 5.
*
*       NO $WAITS ARE ISSUED.

```

```

* REGISTERS USED
*       R0   = ACCUMULATOR - STOP VALUE
*       R1   = ADDRESS OF OPERAND POINTER -- START VALUE
*       R14  = LINK REGISTER
*       R15  = WORK REGISTER, UPON RETURN, CONTAINS THE ADDRESS
*             OF THE CHARACTER THAT STOPPED THE CONVERSION

```

```

* EXITS
*       R14+2 OPERAND DOES NOT CONTAIN NUMERIC OR NUMERIC TOO LARGE
*       R14+6 NORMAL EXIT

```

```

* NOTES
*       R1 VALUE IS EQUAL TO R0 IF USER REQUESTS NUM=2 AND ONLY
*       ONE VALUE IS PRESENT. IF NUM=1 IS USED R1 VALUE IS
*       UNPREDICTABLE. ( NUM=1 MAY BE IGNORED ).
*
*       MAX IS PASSED VIA AN INLINE PARAMETER LIST GENERATED
*       BY THE $CFCVB MACRO.

```

```

*****

```

```

.Z ANOP
&NAME DS OH
      AIF ('&TYPE' NE 'CALL')-INL
      $DECODE &POINTER
.CAL BAL R14,COFCVB&NUM CONVERT NUMBERS TO BINARY
      AIF ('&NOK' NE '')-NOKOK
MNOTE 4,'NOK KEYWORD MUST BE SUPPLIED'
AGO .XIT
.NOKOK ANOP

```

	DC	AL2(&MAX)		@133 01475000
	AIF	('&NOK' (1,1) EQ ')-REG		01480000
	B	&NOK	BRANCH IF OPERAND INVALID	01485000
	AGO	.XIT		01490000
.REG	ANOP			
&CNT	SETA	K'&NOK-2		
&R	SETC	'&NOK' (2,&CNT)		
	B	0(,&R)	BRANCH IF INVALID OPERAND	
	MEXIT			
.INL	L	R15,0(0,R1)	PICK UP FIRST PARAMETER	
COF&R.K	CLI	0(R15),C'0'	CHARACTER NUMERIC...	
	BNL	COF&R.L	IF NUMERIC, CONVERT	
	LA	R15,1(0,R15)	NEXT CHARACTER	
	CLM	R15,7,5(R1)	DID WE GO TOO FAR	
	BL	COF&R.K	LOOP IF NO	
	B	2(,R14)	RETURN INVALID OPERAND	
	EJECT			
*****				
*	CONVERT PAIR OF START STOP VALUES			*
*****				
	SPACE 1			
COF&R.L	CLI	0(15),C'9'	CHECK IF REALLY NUMERIC	
	BH	2(,R14)	INVALID OPERAND IF NOT	
	SLR	R0,R0	ZERO ACCUMULATOR, STOP VALUE	
	ST	R0,\$DOUBLE	CLEAR WORK AREA	
	AIF	('&NUM' NE '2').BA		
	LNR	R1,R15	SET NEGATIVE FLAGS	
.BA	ANOP			
COF&R.C	MVN	\$DOUBLE+1(1),0(R15)	MOVE NUMERIC TO WORK AREA	
	MH	R0,=H'10'	MULTIPLY BY TEN FOR TEN'S DIGIT	
	AH	R0,\$DOUBLE	ADD LOW DIGIT	
	CH	R0,0(,R14)	LOOK FOR MAXIMUM IN INLINE LIST	
	BH	2(,R14)	RETURN INVALID OPERAND	
COF&R.N	LA	R15,1(0,R15)	NEXT CHARACTER	
	CLI	0(R15),C'0'	TEST FOR NUMERIC ( FA-FF INCLUDED )	
	BNL	COF&R.C	IF NUMERIC, CONVERT THE CHARACTER	
	AIF	('&NUM' NE '2').G		
	CLI	0(R15),C'-'	LOOK FOR SEPARATOR	
	BE	COF&R.O	BR IF YES TO CONVERT NEXT VALUE	
	CLR	R1,R0	TEST FOR BOTH VALUES	
	BNH	6(,R14)	RETURN WITH CONVERTED VALUES	
	LR	R1,R0	SET TO SAME	
	B	6(,R14)	RETURN WITH CONVERTED VALUES	
COF&R.O	CLI	1(R15),C'0'	NEXT CHARACTER NUMERIC...	
	BL	2(,R14)	RETURN INVALID OPERAND IF NO	
	LR	R1,R0	SET START OF STRING	
	SLR	R0,R0	PREPARE FOR NEXT	
	B	COF&R.N	CONVERT NEXT VALUE OF SET	
	AGO	.I		
.G	ANOP			
	B	6(,R14)	RETURN	
.I	ANOP			
.XIT	MEND			



## COPY FD - COPIA FLOPPY DISK

Il programma COPYFD permette di copiare un intero volume floppy disk su un altro dello stesso tipo, distruggendo il contenuto del dischetto ricevente.

Il programma e' memorizzato su floppy di sistema SYSREP. Con il sistema ready, dopo avere inserito il floppy SYSREP nell'unita' di indirizzo 90 (drive 0) il programma viene richiamato in memoria per l'esecuzione con il comando:

```
//COPYFD
```

Questo programma con opportune domande descritte da messaggi inviati sulla console dell'operatore, guida quest'ultimo nelle azioni da farsi.

La sequenza dei messaggi previsti da COPYFD e' la seguente(1):

- 1) ENTER 0 FOR COPY FUNCTION  
ENTER 1 FOR COMPARE FUNCTION

.0 <== viene eseguita la copia del volume

.1 <== viene eseguita la verifica della copia

- 2) se richiesta la funzione copy:

YOU HAVE REQUEST COPY FUNCTION  
ENTER GO TO CONTINUE

se richiesta la funzione compare:

YOU HAVE REQUEST COMPARE FUNCTION  
ENTER GO TO CONTINUE

.GO <== continua l'esecuzione del programma

.-GO <== termina l'esecuzione del programma

- 3) se richiesta la funzione copia:

PLACE DISKETTE TO BE COPIED IN LEFT DRIVE  
PLACE BLANK DISKETTE IN CENTER DRIVE

---

NOTA 1) Nella seguente descrizione le righe battute dall'operatore sono quelle che iniziano per il punto.

```
TITLE 'HASP USER EXIT 5 -- OPERAND CONVERSION'
*****
*      INIZIO ELABORAZIONE COMANDI X,Y: TEST SUL FORMATO      *
*****
TESTY  CLI  COMOPRND,C'S'      STC?
      BE  CAJLOOP              SI, CONTINUA ELABORAZIONE
      CLI  COMOPRND,C'T'      TSU?
      BE  CAJLOOP              SI, CONTINUA ELABORAZIONE
TESTXY CLI  COMOPRND,C'J'      JOB?
      BNE ERR649              NO, ERRORE DI FORMATO DEL COMANDO
      SPACE 3
*****
*
*      LOOP DI CONVERSIONE DEI PARAMETRI CON TEST SU FORMATO  *
*
*****
CAJLOOP L    R1,0(0,R5)        R1 PUNTA ALL'OPERANDO
      CLI  0(R1),C'0'        INIZIA CON UN NUMERICO?
      BNL  CAJLOOPC          SI, CONTINUA CONVERSIONE
      L    R4,COFJOBS        SETTA FLAG PER JOB BATCH
      CLI  0(R1),C'J'        C'E' UNA J COME PRIMA LETTERA?
      BE  CAJLOOPC          SI, CONVERTI
      L    R4,COFSTCS        SETTA FLAG PER STC
      CLI  0(R1),C'S'        C'E' UNA S COME PRIMA LETTERA?
      BE  CAJLOOPC          SI, CONVERTI
      L    R4,COFSTSUS        SETTA FLAG PER TSU
      CLI  0(R1),C'T'        C'E' UNA T COME PRIMA LETTERA?
      BNE ERR649              NO, ERRORE FORMATO COMANDO
CAJLOOPC CLI  1(R1),C'='      NON PERMETTERE J= O S= O T=
      BE  ERR650              ERRORE FORMATO OPERANDO
      $CFCVB POINTER=(R5),NOK=ERR650  CONVERTI IN BINARIO
      LTR  R1,R1              ZERO?
      BZ  ERR650              SI, ERRORE OPERANDO
      OR  R1,R4              SETTA TIPO
      STH R0,2(0,R5)          MEMORIZZA VALORE FINALE
      STH R1,0(0,R5)          MEMORIZZA VALORE INIZIALE
CAJNXTOP BXLE R5,R6,CAJLOOP  COMPLETA PER TUTTI GLI OPERANDI
```

## FORMFD - FORMATTAMENTO FLOPPY DISK

Il programma FORMFD esegue un controllo hardware sulla efficienza del supporto, prepara il floppy per la allocazione dei file ed inoltre verifica il funzionamento della coppia drive/dischetto.

FORMFD e' memorizzato su dischetto di sistema SYSREP.

Dopo avere montato il floppy SYSREP sull'unita' di indirizzo 90 (drive 0), il programma viene richiamato per l'esecuzione con il comando:

```
//FORMFD.
```

Opportuni messaggi stampati dal programma sulla console dell'operatore guidano quest'ultimo nelle azioni da eseguire per il formattamento di un dischetto.

Il programma basa la sua esecuzione sul valore di alcuni suoi parametri. Questi valori possono, tramite opportuni comandi, essere variati prima di eseguire il formattamento di un dischetto.

### Comandi

- CL    permette di cancellare i valori precedentemente assegnati ai parametri.
- CO    permette di rilevare le tracce rotte e quindi non utilizzabili. Valori possibili: 0 viene eseguito il controllo, 1 un eventuale difetto rilevato non viene considerato. Valore default 0.
- DR    indica l'unita' floppy disk su cui e' montato il dischetto da formattare. Valoari possibili 0 e 1. Valore default 0 (drive 0).
- GO    viene eseguito il formattamento del dischetto.
- IN    usato per il test del supporto.
- PT    usato per il test del supporto.
- RW    indica se il supporto deve essere verificato per sole operazioni di lettura o di scrittura o per entrambe. Default controllo lettura/scrittura.
- SH    permette di stampare i valori assegnati a ciascun parametro.

EX termina l'elaborazione del programma. Il messaggio EOJ indica che il controllo e' ritornato al sistema.

La sequenza dei messaggi e' la seguente(1):

1) FLOPPY DISK FORMATTER

ENTER DO FOR COMMAND DOCUMENTATION

.DO <== viene stampata a console una breve descrizione dei comandi

.altro comando <== viene accettato un qualsiasi altro comando

2) SUCCESSFULLY FORMATTED

segue un comando di GO ed indica che il formattamento del supporto e' stato eseguito correttamente.

3) EOJ

indica che siamo usciti dal programma FORMPD, ed il controllo e' ritornato al sistema.

---

NOTA 1) Nella seguente descrizione le risposte battute dall'operatore sono quelle che iniziano per il punto.