

Consiglio Nazionale delle Ricerche

Sottoprogrammi di Libreria OS/VS2

E. Bracci - R. Medves

75

CNUCE

Divisione Servizio Elaborazione Dati

A cura di: E. Bracci
R. Medves

Copyright Maggio 1975

by CNUCE - Pisa

Istituto del Consiglio Nazionale delle Ricerche

Elenco dei sottoprogrammi di libreria
presenti sul sistema operativo OS/VS2

Il presente manuale contiene un elenco delle librerie utilizzabili sotto il sistema operativo OS/VS2.

Si tratta sia delle librerie standard di sistema (SYS1.FORTLIB, SYS1.PL1LIB...) sia di librerie aggiuntive create presso il Centro (SYS1.ASMLIB, SYS1.FORT2).

Nel primo caso ci siamo limitati a fornire il nome delle librerie e, dove possibile, la relativa documentazione specifica (per quelle librerie che non hanno manuali specifici di documentazione, la descrizione e modalità di uso delle subroutines in esse contenute, si trovano nei rispettivi manuali di descrizione del linguaggio al quale si riferiscono), nel secondo caso, invece, abbiamo fatto seguire alla libreria una descrizione più dettagliata dei programmi in essa contenuti e delle loro modalità di uso. Specifiche più dettagliate sull'uso di tali programmi possono essere richieste alla Sezione Consulenza del Servizio Elaborazione Dati del Centro.

Il manuale è intenzionalmente non rilegato per permettere un facile aggiornamento, in modo da seguire costantemente le variazioni che via via potranno essere effettuate sul sistema (inserimenti di nuovi programmi, variazioni su programmi esistenti ecc.). A tal fine non è stata adottata nessuna numerazione delle pagine, limitandoci a mettere i nomi delle librerie, e all'interno di esse i nomi dei programmi contenuti, in ordine alfabetico, in modo da permettere con facilità l'inserimento di nuove pagine, senza dover procedere ad una renumerazione totale.

Libreria

SYS1. ALGLIB

Caratteristiche

E' la libreria standard del compilatore ALGOL-F e contiene sottoprogrammi direttamente richiamabili da un programma Algol.

Manuali

Libreria

SYS1.ASMLIB

Caratteristiche:

E' una libreria di sottoprogrammi scritti in linguaggio assembler e richiamabili direttamente da un programma assembler.

Ciascuno dei programmi in essa contenuti è descritto in dettaglio nel seguito.

Manuali

SYS1.ASMLIB (RMDUMP)

Sottoprogramma per ottenere durante l'esecuzione di un programma assembler un dump in forma esadecimale di una zona di memoria indicata o dei 16 registri generali.

La chiamata della subroutine è facilitata con l'uso di una macro istruzione che porta lo stesso nome del sottoprogramma: RMDUMP.

Tale macro istruzione può essere codificata in due maniere distinte:

a) con passaggio di parametri: RMDUMP (AREA 1, AREA2).

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

b) Senza passaggio di parametri: RMDUMP

In tal caso la macro provvede a memorizzare in una zona di memoria temporanea i 16 registri generali e successivamente a richiamare la routine di dump per l'analisi di tale area.

La stringa alfanumerica di intestazione sarà in tal caso '(REGS 0-16)'.

In entrambi i casi, la macro provvede preventivamente al salvataggio dei 16 registri generali in un'area di comodo e al caricamento nel registro 13 di un'area di salvataggio temporanea: ciò permette la massima affidabilità sull'uso della routine e l'inserimento della macro in un punto qualsiasi di un programma, a scopo di debug, senza il pericolo di danneggiare i registri generali o il contenuto della eventuale "save area" del programma chiamante.

Unica avvertenza è quella di non usare, nel programma chiamante, i registri 1,13,14 e 15 quali registri base, in quanto essi vengono utilizzati nella espansione della chiamata alla subroutine.

Per quanto riguarda l'uscita la subroutine è stata predisposta a saltare a pagina nuova solo la prima volta che essa viene chiamata, mentre le chiamate successive saltano solo due righe: ciò è ottenuto rendendo il programma non rientrante e non serialmente riu

sabile a livello del singolo task; solo una nuova compilazione riporta il programma alle condizioni iniziali.

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

Nella parte destra del tabulato, dopo ciascuna riga di cifre esadecimali, comparirà inoltre l'interpretazione in formato EBCDIC (20 caratteri) in modo da facilitare l'interpretazione del dump (i caratteri non stampabili sono sostituiti, nell'uscita, dal carattere punto (.)).

Per poter ottenere l'uscita del dump su carta, occorre inserire tra le schede DD del passo di esecuzione, le schede:

```
//GO.ZZZZZZOO DD SYSOUT=A,DCB=(RECFM=FA,BLKSIZE=133)
```

(sei caratteri zeta e due caratteri zero)

Nota: il programma può essere richiamato direttamente con l'istruzione CALL. In tal caso è compito del programmatore provvedere preventivamente al salvataggio dei registri in un area di comodo e a fornire una "save-area" temporanea puntata dal registro 13. La chiamata diretta può avere una delle seguenti due forme:

a) CALL RMDUMP (AREA1,AREA2), VL

se non si ha interesse ad avere, in uscita, la stringa alfanumerica di intestazione

b) CALL RMDUMP, (AREA1,AREA2,TITOLO,LUNG), VL

TITOLO DC 'DUMP DI AREA'

LUNG DC H'12'

se si desidera avere in uscita la stringa alfanumerica 'DUMP DI AREA' quale intestazione al dump tra gli indirizzi AREA1 e AREA2. In tal caso il terzo argomento (TITOLO) contiene la stringa alfanumerica di caratteri che vogliamo far comparire come intestazione sull'uni

tà e il quarto argomento (LUNG) sarà una mezza voce che conterrà la lunghezza, in binario, della stampa precedente.

Libreria

SYS1 . CBLIB

Caratteristiche:

E' la libreria standard del compilatore
COBOL-ANS e contiene sottoprogrammi diret
tamente richiamabili da un programma Cobol

Manuali

Libreria: SYS1, FORT2

Caratteristiche: E' una libreria di sottoprogrammi scritti in lin
guaggio Fortran e richiamabili direttamente da
un programma Fortran. Ciascuno dei programmi in
essa contenuto è descritto in dettaglio nel se-
guito.

Manuali:

SYS1.FORT2 (DBGEN-CNVRT)

L'uso dei due programmi DBGEN e CNVRT, consente la conversione di nastri binari (cioè scritti senza FORMAT) a 7 piste scritti da calcolatori con parole di 36 bits (tipo 7090) in nastri binari a 9 piste utilizzabili sui calcolatori della serie 360, 370 sotto i sistemi operativi OS e CMS.

SUBROUTINE DBGEN (P1,P2)

Legge un record fisico (blocco) del nastro binario 7 piste: in genere la prima parola di tale record è una parola di controllo del Fortran 7090 e non fa parte dei dati (a volte le parole di controllo sono 2).

P1: variabile intera che conterrà l'indirizzo di memoria a partire dal quale viene posto il blocco letto

P2: codice di errore di ritorno, dalla subroutine secondo i seguenti valori:

0 : lettura corretta

1 : EOF

4 : errore in lettura

16: blocco di lunghezza maggiore di BUFL

20: combinazione degli errori 4 e 16

SUBROUTINE CNVRT (P1,P2,P3,P4,P5,P6)

Converte il blocco letto secondo le specifiche indicate dai seguenti parametri:

- P1 : la stessa variabile intera fornita alla DBGEN
- P2 : variabile (o insieme) che conterrà i dati dopo la conversione
- P3 : numero di voci da convertire
- P4 : tipo di conversione secondo i seguenti valori

	7090	360
1	reale 36 bits	reale 32 bits
2	reale 36 bits	reale 64 bits
3	reale 72 bits	reale 32 bits
4	reale 72 bits	reale 64 bits
5	intero 36 bits	intero 32 bits
6	intero 36 bits	reale 32 bits
7	intero 36 bits	reale 64 bits
8	BCD	EBCDIC
9	decrement	fixed point
10	muove 36 bits senza conversione nei primi 36 bits di una doppia voce 360	

P5 : codice di errore di ritorno dalla subroutine in caso di con versione di tipo 5, secondo i seguenti valori:

- 0 : nessun errore
- 1 : risultato maggiore di $2 * 2^{31} - 1$
- P6 : numero della voce nell'ambito del blocco letto a partire da cui deve avere inizio la conversione. Si ricordi che un numero in doppia precisione occupa 2 voci e inoltre che la prima parola del blocco è una parola di controllo e non fa parte dei dati.

Note ulteriori:

- il nastro di input deve essere identificato da una scheda DD del tipo

```
// GO.GENTAP DD DSN=...,DISP=(OLD),KEEP)
//          UNIT=TAPE7,VOL=SER=...,LABEL=(,NL),
//          DCB=(BUFL=n,BLKSIZE=n,DEN=...)
```

dove il valore n di BUFL e BLKSIZE deve essere maggiore o uguale al numero massimo di voci 7090 di cui è composto il blocco (comprese le parole di controllo) moltiplicato per 6.

- I dati possono essere convertiti una sola volta
- Con la conversione di tipo 9 si avranno 17 bits significativi
- Ciascuna chiamata alla CNVRT converte solo dati contigui dello stesso tipo (tutti reali o tutti interi). Se per esempio un record è composto di 10 interi, 5 reali e 7 interi, si devono usare le chiamate

```
DIMENSION IVET(10)
DOUBLE PRECISION VET1 (5), VET2 (7)
...
CALL DBGEN (IADDR,IERR)
...
CALL CNVRT (IADDR, LGTH,1,5,IERR1,1)
CALL CNVRT (IADDR,IVET,10,6,IDUM,2)
CALL CNVRT (IADDR,VET1,5,2,IDUM,12)
CALL CNVRT (IADDR,VET2,7,7,IDUM,17)
...
```

Libreria: SYS1.FORT2 (IFLOOR-ICEIL-IEXT-ITRUNC)

Caratteristiche: sono quattro funzioni che permettono di ottenere per un dato numero reale fornito positivo o negativo rispettivamente

IFLOOR: l'intero ,con segno, immediatamente inferiore (es. IFLOOR (3.5)=3; IFLOOR(-3.5)=-4)

ICEIL ; l'intero, con segno, immediatamente superiore (es.ICEIL (3.5)=4; ICEIL (-3.5)=-2

IEXT : l'intero, in valore assoluto, immediatamente inferiore e col segno del numero indicato (es. IEXT (3.5)=4 ; IEXT (-3.5) = -4)

ITRUNC: l'intero,in valore assoluto, immediatamente superiore e col segno del numero indicato (es. ITRUNC (3.5)=3 ; ITRUNC (-3.5)= -3)

```
INTEGER FUNCTION IFLOOR (S)
N=INT(S)
IF (FLOAT(N)-S) 2,2,3
ENTRY ICEIL (S)
N=INT(S)
IF (FLOAT(N) -S) 1,2,2
ENTRY IEXT(S)
N=INT(S)
IF (FLOAT(N) -S) 1,2,3
ENTRY ITRUNC(S)
N=INT(S)
GO TO 2
1 IFLOOR=N+1
RETURN
2 IFLOOR=N
RETURN
3 IFLOOR=N-1
RETURN
END
```

A causa della particolare struttura rozza della funzione, non si pretenda di ottenere risultati sicuri e corretti al 100%. In special modo negli intorni di numeri interi (es. 5.0; 7.0 ecc.) a causa della approssimazione delle cifre decimali (es. 4.99999 ecc.) si possono ottenere risultati che si scostano di una unità dal valore corretto.

Si consiglia pertanto l'uso di tali funzioni solo a livello indicativo, rimandando ad algoritmi di controllo più complessi per necessità di un maggiore rigore nei risultati.

SYS1.FORT2 (PLOTS)

La libreria SYS1 FORT2 contiene tra l'altro anche le sobroulines per l'uso del plotter-Calcomp : PLOTS, PLOT, NUMBER, SCALE, LINE, AXIS

Si ricorda qui che:

- a) dal momento che la scrittura del plotter avviene su un nastro a 9 piste, è necessario codificare, dopo la JOB, la scheda SETUP e la scheda MESSAGE che indichino l'uso di una unità a 9 piste quale unità plotter.
- b) la procedura catalogata che permette di poter eseguire programmi Fortran per il plotter è la procedura PLOTG.
Tale procedura fa uso del compilatore Fortran-G1 per la compilazione del programma Fortran.

```
// EXEC PLOTG
/*SETUP TPE9=1
/*MESSAGE TPE9= PLOT
//COMP.SYSIN DD *
    ...fortran...
/*
//GO.SYSIN DD *
    ...dati...
/*
```

L'uso di tale procedura catalogata indica all'operatore il montaggio di un nastro scratch su una unità a 9 piste, che verrà successivamente plottato e poi distrutto.

Se l'utente vuole eseguire un plot su un nastro personale o riservato, dovrà esplicitamente codificare il parametro "Vn=volume" nella procedura(dove n=1, 2, 3, 4 permette fino a 4 nastri plotter)!

```
// EXEC PLOTG,V1=volume1,V2=volume2,V3=volume3,V4=volume4
```

```
/*SETUP      TPE9=1
/*MESSAGE    TPE9=volume
    ...fortran...
//GO.SYSIN  DD *
    ...dati...
/*
```

dove "volume n" indica i nomi dei nastri (di grandezza media) su cui si vuol ottenere il plot.

In tal caso è cura del programmatore avvisare con un messaggio da programma (PAUSE) l'operatore di plottare i nastri al termine dell'esecuzione.

C) La prima chiamata di inizializzazione della subroutine PLOTS deve essere codificata come segue:

```
CALL PLOTS (IBUF,DIM,codice)
```

dove IBUF e DIM sono i normali parametri descritti nei manuali e 'codice' è l'identificazione del programmatore racchiuso tra apici, nella forma: 'persona-ente' (es. '0004-P800')

D) in particolare i parametri passati tra gli argomenti devono essere espressi in centimetri e non in pollici come specificato sui manuali.

E) Alle chiamate standard della routine PLOT sono state aggiunte le modifiche:

```
CALL PLOT (X,Y,998)
```

Quando il terzo parametro è uguale a 998 viene generato uno stop fisico sul nastro, senza peraltro chiudere il file.

In fase di plottaggio peraltro il nastro si fermerà nel punto prestabilito permettendo all'utente l'eventuale sostituzione della penna, della carta o un qualsiasi intervento diretto e ripartirà premendo il pulsante MULTIPLE PLOT.

E' in tal caso indispensabile avvertire l'operatore del numero di PLOTS che vi sono su nastro mediante l'uso della scheda /* MESSAGE.

```
CALL PLOT (X,Y,Z,N)
```

Il quarto parametro N (variabile intera) conterrà al ritorno della chiamata, il numero progressivo del blocco registrato su nastro, permettendo così il suo eventuale ritrovamento diretto.

Si noti che il numero di blocco viene incrementato ogni volta di una unità a partire da 0 fino a 997 per poi riprendere dal valore 2 in su.

SYS1.FORT2 (RAND)

Subroutine per l'estrazione di numeri pseudocasuali

```
SUBROUTINE RAND (/NGEN/,/R/)
  NGEN=NGEN*65539
  IF (NGEN) 5,6,6
5  NGEN=NGEN+2147483647 +1
6  R=RGEN
  R=R*.4656613E-9
  RETURN
END
```

dove NGEN = numero generatore, deve essere fornito solo la prima volta che la subroutine viene chiamata e può essere un qualsiasi numero dispari tra 0 e 10×9 .

R è il numero a random tra 0 e 1 ricercato.

Esempio: estrazione di 10000 numeri a random

```
  NGEN=7
  DO 1 I = 1,10000
  CALL RAND (NGEN,R)
  ...
1 CONTINUE
```

Si noti che NGEN ha il valore 7 solo per la prima chiamata: successivamente esso viene modificato e conterrà ogni volta un numero dispari a random tra 0 e 2×31 .

SYS1-FORT2 (RMPLTn)

Sono richiamabili da Fortran sotto OS quattro sottoprogrammi che eseguono grafici e istogrammi di funzioni dell'utente.

SUBROUTINE RMPLT1 (F,NCAN,XMIN,PAS,TF,TX,IPL0T)

serve a stampare grafici o istogrammi di una funzione F(X).

Grafico : viene stampato un asterisco in corrispondenza della coppia di valori X e F(X).

Istogramma : in corrispondenza del valore X in ascisse viene riempita di asterischi tutta una linea da $X = F(X)$.

L'opzione che permette di fare questo e' IPL0T:

IPL0T=1 fa il grafico

IPL0T=2 fa l'istogramma

Modo d'impiego:

NCAN : numero di canali lungo l'asse X per i quali vogliamo sia graficata la funzione.

F(I) : e' il vettore che contiene i valori della funzione da plottare. L'indice I varierà sui valori dell'asse delle ascisse, cioè andrà da 1 a NCAN. Dovremo pertanto dimensionare F(NCAN).

XMIN : e' l'estremo inferiore delle ascisse dal quale si vuol partire per plottare il grafico. I valori di X in cui calcolare e plottare la funzione si ricavano dalla relazione:

$$X(I) = XMIN + PAS * (I - 1)$$

In particolare e' : $XMAX = XMIN + PAS * (NCAN - 1)$

Tutti i valori di X sono cioè ugualmente intervallati e PAS e' la grandezza di questo intervallo: $PAS = X(I+1) - X(I)$

TF, TX : contengono rispettivamente il nome della funzione da plottare e il nome della variabile X: tali nomi possono essere al massimo di 8 caratteri alfanumerici.

Esempio : CALL RMPLT1 (F,4,1.,1., 'VOTI', 'ESAMI', 1)

I	VOTI	ESAMI	
1	0.1800E 02	0.1000E 01	*
2	0.2800E 02	0.2000E 01	*
3	0.2800E 02	0.3000E 01	*
4	0.2500E 02	0.4000E 01	*

Se si vogliono eseguire NVAR plots basta chiamare la subroutine come:

DO 1 J = 1, NVAR

CALL RMPLT1 (F(I,J), NCAN(J), XMIN(J), PAS(J), TF(J), TX(J), IPL0T(J))

SUBROUTINE RMPLT2 (F1,F2,NCAN,XMIN,PAS,T1,T2,TX)

Serve a plottare due funzioni sullo stesso grafico.
 Puo' ad esempio essere utile quando si vogliono plottare i
 dati sperimentali e il fit su tali dati.

F1(I),F2(I) sono i due vettori che contengono i valori
 delle due funzioni da plottare. L'indice I
 variera' sui valori delle ascisse, cioè
 andra' da 1 a NCAN. Dovremo pertanto
 dimensionare F1(NCAN),F2(NCAN).

Per NCAN,XMIN,PAS valgono le stesse considerazioni fatte
 per RMPLT1.

T1,T2,TX contengono rispettivamente i nomi delle due
 funzioni da plottare e il nome della
 variabile X: tali nomi possono essere al
 massimo di 8 caratteri alfanumerici.

Esempio : CALL RMPLT2 (F1,F2,4,1.,1.,'MEDIE','VOTI','ESA'),
 * = MEDIE
 X = VOTI
 \$ = OVERLAP

I	(X-*)	ESAMI							
1	0.0000E 00	0.1000E 01	1	\$					
2	0.5000E 01	0.2000E 01	1		*				X
3	0.3333E 01	0.3000E 01	1			*			Y
4	0.2500E 00	0.4000E 01	1				*X		

Se si vogliono eseguire NVAO plots, basta chiamare la
 subroutine come:

```
DO 1 J = 1,NVAO
1 CALL RMPLT2 (F1(1,J),F2(1,J),NCAN(J),XMIN(J),PAS(J),
  XT1(J),T2(J),TX(J))
```

SUBROUTINE RMPLT3 (X,N,NCAN,XMIN,PAS,TX,IFLUX)

Serve per istogrammare una variabile X generata secondo una distribuzione N(X).

NCAN, XMIN, PAS determinano un certo intervallo di valori che la variabile X può assumere: chiamiamo "canale l-mo" l'intervallo $[X(l), X(l+1))$ chiuso a sinistra e aperto a destra. Per la relazione intercorrente tra tali variabili valgono le considerazioni fatte per RMPLT1.

N(I) e' una variabile fittizia che dara' la popolazione in ciascuno dei canali I. Va dimensionata $N(NCAN+2)$ e azzerata. Ogni volta che un valore di X viene passato alla subroutine, questa provvede a testarlo e ad incrementare il contenuto del canale in cui tale X si trova.

Underflow : e' il numero degli X compresi nell'intervallo $[-\infty, XMIN)$

Overflow : e' il numero degli X compresi nell'intervallo $[XMAX, +\infty)$ dove $XMAX = XMIN + PAS * (NCAN - 1)$

Contenuto del plot : e' il contenuto dell'intervallo $[XMIN, XMAX)$

TX contiene il nome della variabile da istogrammare, di lunghezza massima 8 caratteri alfanumerici.

IFLUX dirige il flusso del programma nella subroutine:

IFLUX # (0,1) : la subroutine si limita ad accumulare i valori di X nei vari canali, ogni volta che viene chiamata.

IFLUX = 0 : stampa l'istogramma con scala lineare.

IFLUX = 1 : stampa l'istogramma con scala logaritmica.

Esempio : CALL RMPLT3 (X,N,4,0.,0.5,'PROVA',IFLUX)

CONTENUTO PLOT....		X-UNDERFLOW....		X-OVERFLOW. ..
N	CONT.	LOW	EDGE	
1	5	0.0000E	00	XXXXXXXXXX
2	10	0.5000E	00	XXXXXXXXXXXXXXXXXXXXXXXXXX
3	10	0.1500E	01	XXXXXXXXXXXXXXXXXXXXXXXXXX
4	3	0.2000E	01	XXXXXX

Se si vogliono eseguire NVAR plots basta chiamare la subroutine come

```
DO 1 J = 1, NVAR
1 CALL RMPLT3 (X(J), N(1, J), NCAN(J), XMIN(J), PAS(J), TX(J), IFLUX(J))
```

SUBROUTINE RMPLT4 (X,Y,N,NCANX,NCANY,XMIN,YMIN,PASX,PASY, TX, TY, IFLUX)

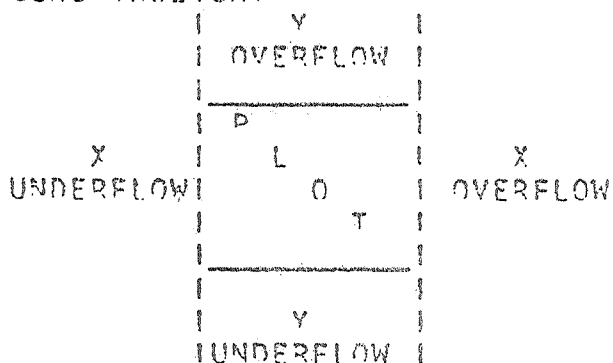
Serve ad eseguire un istogramma bidimensionale nelle variabili X e Y generate secondo una distribuzione N(X,Y).

NCANX, XMIN, PASX e NCANY, YMIN, PASY determinano le celle (o bins) nelle quali viene suddiviso il grafico bidimensionale. Per ciascuna proiezione lineare sugli assi valgono le stesse considerazioni fatte per la RMPLT3 con la limitazione che NCANX non può superare il valore di 60.

N(I) e' una variabile fittizia che dara' la popolazione in ciascuna delle celle.

Va dimensionata N(NCANX*NCANY+4) e azzerata.

Le zone del plot sono suddivise secondo le denominazioni



TX, TY contengono i nomi delle due variabili X e Y da istogrammare e possono al massimo essere di 8 caratteri alfanumerici.

IFLUX dirige il flusso del programma nella subroutine:
IFLUX ≠ 0 accumula i valori delle coppie X-Y nelle varie celle
IFLUX = 0 stampa il plot richiesto

Se si vogliono eseguire NVAR plots, basta chiamare la subroutine come:

```
DO 1 J = 1, NVAR  
1 CALL RMPLT4 (X(J), Y(J), N(1, J), NCANX(J), NCANY(J), PASY(J), PASX(J),  
  XTX(J), TY(J), IFLUX(J))
```


Note generiche

- : - i plots 1,2 e 3 vengono stampati con l'asse delle ordinate lungo la parte orizzontale del foglio e l'asse delle ascisse in verticale.
- Per facilitare il riferimento nelle ordi-
nate, vengono stampati lungo tale asse de-
gli apostrofi : essi sono messi a partire dal
valore $F=0$ verso i valori negativi e positivi
delle ordinate, ad ogni valore intero di F .
Tali subroutines non contengono all'inizio
l'opzione 1H1 per il salto a pagina nuova.
Questo per lasciare la possibilità di grafi-
care le funzioni di seguito ad eventuali ta-
belle. Tale opzione può facilmente essere
inserita dal programmatore prima della chia-
mata delle subroutines.

Libreria : SYS1:FXLIB (5734-LM3) versione 2.0

● Caratteristiche : E' la libreria standard dei compilatori FORTRAN G1 e X (H esteso) e contiene sottoprogrammi matematici e di servizio richiamabili direttamente da programma Fortran e che permettono di sfruttare appieno le possibilità dei due compilatori quali una più accurata conversione di dati da una forma all'altra, il supporto per l'input/output in forma di lista, le possibilità di uso di variabili in precisione estesa, la convenzione automatica di precisione ecc.

Manuali

C28-C864 Fortran IV (Hextended) Math and Service Subprograms

Libreria : SYS1.MACLIB

Caratteristiche : E' la libreria standard delle macroistruzioni
per l'assembler-F

Manuali

C26-3793

Data Management Macro-Instructions

Libreria : SYS1.MACRO

Caratteristiche : E' una libreria di macro concatenata con la
SYS1.MACLIB, e contiene le macro istruzioni
descritte in dettaglio nel seguito.

Manuali

SYS1.MACRO (REGS)

E' una macro mediante la quale si ottiene l'associazione di 16 reg-
gistri generali, di altrettanti nomi simbolici:

	REGS	
+ R0	EQU	0
+ R1	EU	1
	.	
	.	
	.	
	.	
	.	
	.	
	.	
	.	
+ R15	EQU	15

SYS1.MACRO (RMDUMP)

E' la macro che permette la chiamata dell'omonima subroutine.
Per le spiegazioni dettagliate si faccia riferimento a quella,
nella libreria SYS1.ASMLIB.

SYS1.MACRO (RMSTART-RMRETURN)

Sono due macro istruzioni che facilitano la minuzazione di semplici programmi in linguaggio Assembler, riducendo il numero di schede da perforare.

Le due macro istruzioni hanno la forma:

nome1 RMSTART (base,savearea)

e

nome2 RMRETURN

dove:

nome1: è il nome che si vuol dare alla control section.

Nome2: è un nome opzionale che si vuole dare all'istruzione.

Base : è un numero che indica il registro che si vuole utilizzare come registro base.

Savearea: è il nome che si vuol dare ad un'area di salvataggio dei registri.

Le due macro istruzioni si espandono come segue:

```

nome1      RMSTART      (base, savearea)
+nome1     CSECT
+          SAVE        (14,12),,*,
+          BALR        base,0
+          USING       *, base
+          B           savearea + 72
+savearea  DS           18F
+          LA          2, savearea
+          ST          2,8(13)
+          ST          13,4(2)
+          LR          13,2

```

```

nome2      RMRETURN
+nome2     L           13,4(13)
+          LM          14,12,12(13)
+          BR          14

```

esempio:

```

MAIN      RMSTART      (12,SAVE)
          ---programma dell'Utente---
          RMRETURN
          ---istruzioni dichiarative---
          END          MAIN

```

equivale a:

```

MAIN      CSECT
          SAVE        (14,12),,*,
          BALR        12,0
          USING       *,12
          B           SAVE+72
SAVE      DS           18F
          LA          2,SAVE
          ST          2,8(13)
          ST          13,4(2)
          LR          13,2
          ---programma dell'utente---
          L           13,4(13)
          LM          14,12,12(13)
          BR          14
          ---istruzioni dichiarative---
          END          MAIN

```


SYS1.MACRO (SECTION)

Analogamente alla REMSTART, consente l'inizializzazione semi-automatica di un programma assembler.

L'istruzione ha la forma:

```
SECTION (csect, base, start, return)
```

dove:

csect: è il nome simbolico che si vuol dare alla control section

base : è il nome simbolico del registro base che si vuole utilizzare

return: è il nome simbolico a cui si deve saltare alla fine del programma

start: è il nome simbolico dell'indirizzo di partenza del programma

l'espansione relativa è la seguente

```
SECTION (csect, base, start, return)
+csect CSECT
+ USING *,15
+ SAVE (14,12),,*
+ LA 14,SAVEnnnn
+ ST 13,4(14)
+ ST 14,8(13)
+ LR 13,14
+ LR base,15
+ DROP 15
+ USING csect,base
+ B start
+return SR 15,15
+ L 13,SAVEnnnn+4
+ L 14,12(13,0)
+ LM 0,12,20(13)
+ BR 14
+ DS OF
+SAVEnnnDC 9C'SAVEAREA'
```

Libreria : SYS1.PL1LIB

Caratteristiche: E' la libreria standard del compilatore PL/1-F
e contiene sottoprogrammi direttamente richiamabili da un programma PL/1-F

Manuali

C28-6590 PL/1 Computational Subroutines

Libreria : SYS7.PL1LIB2

Caratteristica: E' una libreria di sottoprogrammi scritti in linguaggio PL/1. Ciascuno dei programmi in esso contenuto è descritto in dettaglio nel seguito.

Manuali:

Esempio

```

SECTION      (MYPROG, BASE ,INIZIO,FINE)
BASE        EQU      12
INIZIO      EQU      *
            ---
            ---
            ---
            ---
            ---
            B        FINE
    
```

equivale a:

```

MYPROG      CSECT
            USING    *,15
            SAVE     (14,12),*,*
            LA       14,SAVE0001
            ST       13,4(14)
            ST       14,8(13)

            LR       13,14
            LR       BASE,15
            DROP     15
            USING    MYPROG,BASE
            B        INIZIO
FINE        SR       15,15
            L        13,SAVE0001+4
            L        14,12(13,0)
            LM       0,12,20(13)
            BR       14
            DS       OF
SAVE0001    DC       9C'SAVEAREA'
BASE       EQU      12
INIZIO     EQU      *
    
```

SYS1.PL1LIB2 (CNVTORA)

Consente la conversione di un numero che esprima un tempo in centesimi di secondo in un formato esteso comprendente ore, minuti, secondi. Eventuali messaggi di errore vengono stampati in un file di nome SYSPRINT.

```
DCL          VAR      BIN      FIXED (31);
DCL          VAR1     CHAR (11);
DCL          CNVTORA  ENTRY (BIN FIXED (31)) RETURNS (CHAR (11) );
VAR1 = CNVTORA (VAR);
```

Se VAR contiene il tempo in centesimi di secondo (es.738500), VAR1 conterrà l'ora in forma:

02H 03' 05''

sys&.PL1LIB2 (ESADECI)

Consente di ottenere un dump esadecimale di una variabile carattere specificato.

Il risultato viene memorizzato in una variabile di lunghezza doppia

```
DCL      VAR      CHAR (50) VARYNG;
DCL      VAR1     CHAR (100) VARYNG;
DCL      ESADECI  ENTRY (CHAR (50) VARYNG) RETURNS (CHAR (100) VARYNG);
VAR1=ESADECI (VAR);
```

Se ad esempio la variabile VAR contiene la stringa 'PROVA', VAR1 conterrebbe la stringa:

D7D9D6E5C1

SYS1.PL1LIB2 (OSDATA-DATAOS-F7DATA)

Consentono di ottenere la data del giorno nella forma YYDDD
(dove YY=anno, DDD=giorno dell'anno) e il passaggio tra tale data
e la data in forma DD/MMM/YY, dove:

DD=giorno del mese, MMM=mese in caratteri, YY=anno

Eventuali messaggi di errore vengono stampati in un file di nome
SYSPRINT.

```
DCL      VAR      DEC      FIXED(7)
DC1      VAR1     CHAR (9)
DCL      VAR2     DEC      FIXED(7)
DCL      F7DATA   ENTRY   (CHAR(9)) RETURNS(FIXED(7))
DCL      OSDATA   ENTRY   RETURNS(FIXED(7))
DCL      DATAOS  ENTRY   (FIXED(7)) RETURNS(CHAR(9))
```

```
-----
VAR=OSDATA
VAR1=DATAOS (VAR)
VAR2=F7DATA (VAR1)
```

Se VAR è ad esempio il 2 febbraio 1975, VAR conterrà il valore
75043, VAR1 conterrà la data nella forma:
12/FEB/75 mentre VAR2 avrà lo stesso valore di VAR.

SYS1.PL1LIB2 (PUTDUMP)

Consente di ottenere un dump esadecimale di una variabile formato carattere o binaria specificato. Oltre al dump esadecimale, sulla destra di ciascuna riga, stampa anche l'interpretazione del dump in formato carattere.

DCL STRINGA CHAR (nn); (nn \leq 500)

DCL NUMERO BIN FIXED (m); (m \leq 31)

CALL PUTDUMP (STRINGA);

CALL PUTDUMP (NUMERO);

Libreria: SYS1.PLRLIB

(5734.LM4) versione 1.2

Caratteristiche; costituisce la libreria delle subroutines residenti del PL/1 Optimizing Compiler. I sottoprogrammi in essa contenuti vengono agganciati in fase di Linkage Editor e vanno a costituire insieme al programma sorgente, il modulo eseguibile.

Tra l'altro la libreria contiene le funzioni matematiche (es SIN,COS), quelle per la conversione dei dati (es picture) e quello per l'ingresso/uscita in forma STREAM (es GET,PUT)

Manuali

Libreria:

SYS1,PLTLIB:

(5734-LM5) versione

Caratteristiche:

costituisce la libreria delle subroutines
del PL/1 Optimizing Compiler
e Checkout Compiler. I sottoprogrammi in es-
sa contenuti vengono agganciati dinamicamente
in fase di esecuzione.

Tra l'altro, la libreria contiene i moduli per
la stampa di messaggi di errore, i moduli per
la gestione delle interruzioni, per la aper-
tura e chiusura dei files, per la gestione del
la memoria ecc.

Manuali

Libreria : SYS1.PL1SSP (Scientific Subroutine Package)
Versione 1.1.

Caratteristiche: direttamente richiamabili da PL/1
E' una raccolta di sottoprogrammi PL/1 statistici e matematici atti alla soluzione di numerosi problemi scientifici e di ingegneria. Numerosi sottoprogrammi sono in forma di singola e doppia precisione, per ottenere l'accuratezza voluta.

Manuali: H20-0586 Scientific Subroutine Package (PL/1)

Sottoprogrammi: SYS1.SMLIB (5734-SM1) versione 1.3

Caratteristiche: è la libreria standard del Sort/Merge e con-
tiene tutte le funzioni necessarie al program-
ma durante la sua esecuzione.

Manuali:

Libreria: SYS1. SSPLIB (Scientific Subroutines Package)
Versione 3.1.

Caratteristiche: direttamente richiamabili da Fortran.
E' una raccolta di sottoprogrammi Fortran sta
tistici e matematici, atti alla soluzione di nu
merosi problemi scientifici e di ingegneria. Nu
merosi sottoprogrammi sono in forma di singola
e doppia precisione, per ottenere l'accuratezza
voluta.

Manuali: H20-0205 Scientific subroutines Package (Fortran)

Libreria: SYS1.WATLIB versione 1.4

Caratteristiche: è la libreria standards del compilatore WATFIV
e contiene sottoprogrammi direttamente richia-
mabili da un programma watfiv.

Manuali:

