NUMERO 4d / 1984

DATA BASE PER LA GESTIONE DI CATALOGHI
ASTRONOMICI

----- PRIMA PARTE -----

M.NANNI

RAPPORTO VAX 11/780

CONSIGLIO NAZIONALE DELLE RICERCHE

ISTITUTO DI RADIOASTRONOMIA

c/o ISTITUTO DI FISICA « A. RIGHI »
Via Irnerio, 46 - 40126 BOLOGNA (Italy)

INDICE

INTRODUZIONE.....	1
CONCETTI GENERALI DEL DATA BASE.....	1
DATA BASE DI SISTEMA DEI CATALOGHI ASTRONOMICI.....	4
PERSONAL DATA BASE.....	7
NOTE CONCLUSIVE.....	9
APPENDICE A DESCRIZIONE DELLE ROUTINES.....	10
APPENDICE B ESEMPI DI PROGRAMMI.....	30

INTRODUZIONE

=====

Il lavoro che viene presentato in questa sede fa riferimento al progetto per la costruzione di un Data Base per la gestione di cataloghi astronomici attorno al quale sta operando il gruppo di lavoro Astronet 'Data Base e Documentazione'. In particolare vengono presentate una struttura di documentazione dei cataloghi astronomici disponibili presso il centro, routines per facilitare l'accesso ai dati ed una struttura di utente, compatibile con quella di sistema, che faciliti l'uso di cataloghi, tabelle, liste in genere e che renda possibile la generalizzazione dei programmi applicativi che su queste operano.

CONCETTI GENERALI DEL DATA BASE

=====

I cataloghi di dati astronomici sono normalmente costituiti da tabelle bidimensionali di dati dove i records che si riferiscono ai singoli oggetti del catalogo sono suddivisi in campi contenenti le grandezze fisiche. Data questa semplice struttura le informazioni necessarie e sufficienti, da un punto di vista logico, per poter gestire queste tabelle di dati si riducono alla conoscenza dei soggetti del catalogo e delle grandezze fisiche utilizzate nel lavoro astronomico.

Quindi, e' estremamente semplice procurarsi tutti gli strumenti per poter utilizzare un catalogo o una lista di dati, le difficoltà nascono nel gestire files di dati spesso di grandi dimensioni (centinaia di migliaia di records) tra loro disomogenei in quanto a formati di scrittura, e per l'uso di linguaggi, quali il FORTRAN, che mal si prestano ad un lavoro di tipo gestionale.

Lo scopo a cui abbiamo teso con questo lavoro e' stato quello di creare una struttura software, chiamata DATA BASE, che riconducesse il problema dell'uso delle tabelle e dei cataloghi alla conoscenza dei soli elementi logicamente indispensabili; così facendo si e' ottenuta la possibilità di generalizzare programmi applicativi di utente all'insieme di tutte le liste e tabelle indipendentemente dalle specifiche grandezze fisiche in esame e dai formati con cui queste sono scritte.

Le Grandezze Fisiche

Per utilizzare i cataloghi facendo riferimento unicamente alle grandezze fisiche e' stato necessario costruire un vocabolario delle variabili maggiormente utilizzate in astronomia ed associare

ad esse un codice mnemonico di riferimento. Il vocabolario, e' quindi l'elemento base dell'intera struttura in quanto le letture dai cataloghi avverranno utilizzando i codici mnemonici in esso definiti.

Tra i dati che compaiono nei cataloghi, fin qui indicati genericamente come grandezze fisiche, spesso troviamo anche descrittori, note particolari, codici etc. Queste grandezze 'anomale' di tipo non numerico hanno impedito la costruzione di una semplice struttura basata sulla gestione di insiemi di numeri decimali. Il vocabolario definisce in modo univoco quali sono le grandezze rappresentate come numeri decimali (le grandezze fisiche propriamente dette), quali con numeri interi (i numeri d'ordine i codici, ed alcune grandezze particolari), e quali con stringhe (descrittori referenze e note)

Un discorso a parte riguarda il caso di grandezze fisiche espresse da piu' campi numerici quali le coordinate. In questo caso esistono alcune convenzioni diverse utilizzate da vari autori di cataloghi. Nel nostro caso e' stato utile ricondursi ad un unico tipo di rappresentazione; in particolare e' stata adottata la convenzione dell'uso di Ore Minuti Secondi, Segno Gradi Primi Secondi, dove le prime 2 grandezze numeriche di ogni terna sono considerate intere e la terza decimale.

I Files dei Dati

I files di dati utilizzati dal Data Base possono risiedere su nastro in formato 'VAX/VMS ANSI-labelled magnetic tape' o su disco, possono essere scritti con records di lunghezza fissa o variabile, e non hanno limitazioni per quel che riguarda le dimensioni.

Per quanto riguarda i dati contenuti nei cataloghi le uniche condizioni che abbiamo posto sono:

Tutte le informazioni relative allo stesso oggetto devono risiedere nello stesso record fisico; cioe' non sono ammesse organizzazioni di dati che prevedono di utilizzare piu' records per uno stesso oggetto.

Non sono ammessi records disomogenei con la tabella dei valori (es. intestazioni)

Se si vogliono utilizzare tutte le routines della struttura le coordinate spaziali devono essere esplicitamente espresse tramite la terna (ore minuti secondi etc.)

Si e' deciso di fare uso di una rappresentazione dei numeri decimali con il punto esplicitamente presente.

Il data base potra' processare tabelle contenenti fino a 80 grandezze fisiche indipendenti ordinate in colonne potendone estrarre un massimo di 40 ad ogni lettura operando su un massimo di 30 cataloghi simultaneamente aperti.

Archivi di Servizio del Data Base

La struttura che permette di gestire il Data Base e' costituita da alcuni files su disco che contengono le informazioni relative ai singoli cataloghi, ai formati dei dati in essi contenuti, ed ai vocabolari. Questi files fungono da archivi di servizio alla struttura del Data Base, vengono letti e scritti da routines trasparenti all'utente essendo files ad accesso a chiave, quindi non gestibili in modo immediato.

A) - MASTERDB.BAK contiene le informazioni generali sui files dei cataloghi

- 1) Numero progressivo del catalogo
- 2) Nome del catalogo
- 3) Localizzazione : se il file risiede su disco solo il nome fisico dell'unita' disco.
Altrimenti anche la label nella forma: MT:LABEL
- 4) Nome del file che contiene il catalogo
- 5) Eventuale nome del file di documentazione
- 6) Soggetto del catalogo
- 7) Eventuale specificatore del soggetto
- 8) Tipo di catalogo (riferimento CDS)
- 9) Banda elettromagnetica osservata
- 10) Numero di grandezze fisiche contenute nel catalogo
- 11) Numero di records del catalogo
- 12) Tipo di accesso consentito :
 - S: sequenziale
 - SALF: sequenziale ordinato in R.A.
 - SDEC: sequenziale ordinato in Dec
 - I: diretto
 - IALF: diretto ordinato in R.A.
 - IDEC: diretto ordinato in Dec
- 13) Equinozio

B) - FORMATDB.BAK contiene le informazione sui formati

- 1) Codice mnemonico delle grandezze fisiche del catalogo
- 2) Formati con cui le grandezze sono fisicamente scritte nel file

C) - VOCABOL.BAK contiene il vocabolario delle grandezze fisiche

- 1) Codice mnemonico
- 2) Tipo generale di formato della grandezza
- 3) Descrizione della grandezza fisica

D) - SPECKEY.BAK contiene l'elenco delle parole chiave ammesse nella ricerca per argomenti dei cataloghi

In riferimento agli ultimi due files esiste una versione .VOC dei files stessi che possono essere aggiornati con una normale operazione di edit. Questi files vengono trasformati in files ad accesso a chiave tramite comandi nelle procedure di gestione.

DATA BASE DI SISTEMA DEI CATALOGHI ASTRONOMICI
 =====

I cataloghi astronomici di uso generale sono a disposizione degli utenti in formati compatibili con la struttura descritta, il servizio del centro si occuperà di acquisirne il maggior numero e di renderli fruibili mantenendo il loro formato il più coerente possibile alla pubblicazione originale. L'utente potrà verificare in ogni momento quanti e quali cataloghi di dati sono a sua disposizione utilizzando le routines qui descritte.

Il comando DATABASE abilita le routines e deve essere fornito come prima istruzione anche in processi Batch che volessero far uso questa struttura.

Il comando DB_INFO permette di ottenere informazioni attraverso un menu' di chiavi di ricerca sui cataloghi. E' quindi possibile farsi mostrare l'elenco e le caratteristiche di tutti i cataloghi del centro, oppure selezionare tra questi quelli che contengono solo stelle o galassie o dati X. Il comando permette interrogazioni al vocabolario circa le grandezze fisiche, i simboli mnemonici ad esse associate ed i formati generali con cui le grandezze vengono lette.

Routines per l'utilizzo dei Cataloghi

Sono state messe a punto una serie di routines, richiamabili in FORTRAN, che permettono di accedere ai dati contenuti negli archivi di servizio ed ai records dei cataloghi dei dati; tali routines svincolano l'utente dalla necessita' di conoscere dove si trova il file che si vuole leggere e come questo e' internamente organizzato.

A fianco di routines che funzionano 'alla cieca' indipendentemente dal fatto che i cataloghi si trovino su nastro o disco, sono state scritte routines che ottimizzano la lettura dai files in cui e' permesso l'accesso diretto (files su disco con records a lunghezza fissa). In particolare, per ricerche su grandezze fisiche ordinate nel catalogo, si hanno a disposizione routines basate su tecniche dicotomiche che diventano particolarmente vantaggiose nella ricerca su grosse quantita' di dati.

A) Gestione dei files dei dati

DB_OPEN	Individua la posizione fisica del catalogo su nastro o disco, provvede ad assegnare un numero di unita' logica ed opera una 'open' del file
DB_CLOSE	Chiude il file
DB_INDEX	Chiude un file gia' aperto in modo

sequenziale, lo riapre per un accesso diretto (quando e' consentito dalla struttura del file)

DB_SEQUEN Chiude un file gia' aperto per un accesso diretto e lo riapre per un accesso sequenziale

B) Routines di servizio

DB_MSG Abilita o disabilita la scrittura di messaggi che descrivono lo stato di esecuzione di alcune routines.

DB_NAME Permette di assegnare i nomi ai files di output che si ottengono dalle routines.

C) Routines reperimento del formato

DB_FORM Specificando in input i codici delle grandezze fisiche che si vogliono leggere dai cataloghi si ottiene il formato di lettura sul catalogo.

DB_FORM_ALL Ottiene il formato di tutte le grandezze del catalogo.

D) Routines di posizionamento (per files ad accesso diretto ordinati secondo una particolare grandezza fisica)

DB_PUNT_CRD Ottiene il numero di records in cui compare la coordinata cercata

DB_PUNT_CAR Ottiene il numero di records in cui compare la grandezza fisica (tipo stringa) cercata

DB_PUNT_VAL Ottiene il numero di records in cui compare la grandezza fisica (tipo numerico) cercata

E) Routines di estrazione

DB_WIND_CO Seleziona da un catalogo le grandezze fisiche desiderate all'interno di una finestra definita in R.A. e DEC. e crea un file di output

DB_WIND_REC Seleziona da un catalogo le grandezze fisiche desiderate all'interno di una finestra definita in R.A. e DEC. e le passa al programma chiamante.

Le routines sono meglio definite nella appendice A mentre nella appendice B vengono listati alcuni programmi dimostrativi.

Ci preme invece sottolineare il modo in cui queste routines possono essere praticamente utilizzate all'interno di programmi di utente. In particolare la lettura dei cataloghi puo' avvenire,

ad accesso sequenziale :

```
CALL DB_OPEN(1,NOMECAT,UNIT,IER)
- - - -
CALL DB_FORM (1,FHISVAR,FORMATI,IER)
READ(UNIT,FORMATI) A,B,C,I1,I2,I3
- - - -
```

ad accesso diretto :

```
CALL DB_OPEN(1,NOMECAT,UNIT,IER)
- - - -
CALL DB_FORM (1,FHISVAR,FORMATI,II)
CALL DB_INDEX(1,NL,IER)
- - - -
READ (UNIT'IPX)BUFF(1:NL)
DECODE(NL,FORMATI,BUFF(1:NL))A,B,C,I1,I2,I3
- - - -
```

Dove IPX, il numero del record che deve essere letto puo' essere conosciuto dall'utente o ricavato attraverso routines del tipo: DB_PUNT_VAL

I dati sono quindi accessibili all'utilizzatore attraverso normali chiamate di READ, data la generalita' delle routines utilizzate e' possibile scrivere programmi applicativi completamente indipendenti dal catalogo in esame.

PERSONAL DATA BASE
 =====

A fianco della struttura per la gestione dei cataloghi di uso generale e' stata pensata una identica struttura per trattare cataloghi e tabelle personali di utente .

I programmi per la gestione di questo Personal Data Base sono gli stessi che vengono utilizzati per la gestione del Data Base di sistema. In questo caso l'utente dovra' provvedere all'aggiornamento degli archivi di servizio descrivendo le sue liste e tabelle; le chiamate PDB_OPEN e PDB_CLOSE sostituiranno le DB_OPEN e DB_CLOSE quando si vorra' operare sul Personal, ma chiamate a cataloghi di sistema e a tabelle personali potranno coesistere nello stesso programma, con l'unica avvertenza di considerare che il numero massimo di cataloghi e tabelle simultaneamente accessibili sono 30 in totale. Va ricordato inoltre che cataloghi residenti su nastro non possono rimanere simultaneamente aperti per l'uso concorrente dell'unita' fisica, e l'apertura di un catalogo causa l'immediata chiusura del precedente.

La chiamata DB_STORE inoltre permette di includere nel Personal Data Base la descrizione di cataloghi ottenuti facendo uso della DB_FORM. La stessa opzione esiste nella DB_WIND_CO dove un parametro abilita la scrittura della descrizione del file selezionato dalla routine nel Personal Data Base, quindi i dati diventano immediatamente utilizzabili tramite successive chiamate al PDB.

Gestione del Personal Data Base

La gestione degli archivi di supporto del personal data base e' lasciata agli utenti attraverso una serie di routines che permettono l'accesso ai files di servizio descritti precedentemente, che devono risiedere nella stesso directory da cui si fara' girare il programma. I comandi abilitati sono:

PDB_GENERA Copia nel directory dell'utente il vocabolario delle grandezze fisiche ed il vocabolario delle parole chiave di ricerca dal Data Base di sistema, affinche' l'utente stesso possa modificarli a seconda delle specifiche esigenze.

Crea il file MASTERDB.BAK e FORMATDB.BAK con tanti records quanti sono necessari per contenere i cataloghi e tabelle dell'utente (un record per catalogo)

PDB_AMPLIA Aggiunge altri records ai files di servizio.

PDB_GEST Permette di aggiungere cancellare, correggere i records che descrivono i cataloghi.

Nella compilazione di questi descrittori e' utile ricordare che dati obbligatori sono: nome del catalogo, localizzazione, nome del file, numero di grandezze

fisiche contenute nel catalogo, numero di records (obbligatorio solo per i files ad accesso diretto), tipo di accesso, grandezze fisiche e relativi formati. I codici mnemonici delle grandezze fisiche sono composti da un massimo di 5 caratteri alfanumerici e vanno separati da uno spazio o da una virgola.

Sono disponibili routines di correzione a tutta pagina che permettono di posizionarsi sulle grandezze e sui formati da modificare.

Quando un nuovo record viene accettato si opera un controllo sulle grandezze fisiche e vengono segnalate quelle che non compaiono nel vocabolario.

Una opzione di questa procedura trasforma il file del vocabolario in un file ad accesso diretto.

PDB_INFO E' praticamente identico a **DB_INFO** con la sola differenza che opera sul personal data base.

PDB_DELEX Cancella nel PDB tutti i records che si riferiscono ad un certo catalogo.

Le procedure di gestione del Data Base fanno largo uso della libreria VTLIB e richiedono di essere lanciate da un terminale della classe VT100; le procedure **PDB_INFO** e **DB_INFO** invece possono essere utilizzate anche da altri terminali in quanto prevedono diversi tipi di output a seconda della periferica in uso da parte dell'utilizzatore.

Allo stato attuale l'unica procedura di ricerca sui dati dei cataloghi disponibile e' il comando :

DB_WINDCO Estrae un sottoinsieme di un catalogo limitato in R.A., in Delta ed in numero di variabili e crea un file a disposizione dell'utente.

NOTE CONCLUSIVE

=====

Con questo primo lavoro si e' voluto formulare una indicazione in merito alla possibilita' di trattare cataloghi e tabelle svincolandosi dalla loro organizzazione fisica. Le routines che vengono messe a disposizione dell'utente sono ancora di basso livello, ma permettono di iniziare a pensare alla costruzione di una serie di programmi ad alto livello svincolati dalle applicazioni specifiche; anche a questo sta lavorando il Gruppo Astronet Data Base e Documentazione.

Per quanto concerne lo sviluppo immediato, in sede Bolognese, di questo lavoro, si prevede di poter mettere a disposizione in tempi brevi:

1) Routines per le conversioni di grandezze fisiche frequentemente utilizzate, in particolare trasformazioni di coordinate.

2) Routines per operare sort dal fortran riferendosi alle grandezze fisiche

3) Programma per il reperimento di dati extragalattici facendo riferimento alla 'Master List of non stellar optical objects'.

REFERENZE

=====

- BENNACCHIO L. :1982, MEM.SOC.ASTR., II. 53, 155
"DATA BASE CREATION FROM ASTRONOMICAL CATALOGUES
CURRENT PROBLEMS AND SUGGESTIONS FOR SOLUTION"
- R.H.HARTEN,P.GROSBOL,K.P.TRITTON :1983, PREPRINT
"AN EXTENSION OF FITS FOR TABLES AND CATALOGUES"
- NANNI M. : 1983 , MEM.SOC.ASTR.
"SOFTWARE INTERFACE TO A DATA BASE ON TAPE"
- FICARRA A. : 1983, MEM.SOC.ASTR.
"SUBROUTINES FOR THE 'FULL SCREEN' MODE OF THE
DEC_VT100 VIDEO TERMINAL"

APPENDICE A
DESCRIZIONE DELLE ROUTINES FORTRAN

CONTENTS

DB_CLOSE	1
DB_FORM	2
DB_FORM_ALL	3
DB_INDEX	4
DB_MSG	5
DB_NAME	6
DB_OPEN	7
DB_PUNNT_CAR	8
DB_PUNNT_CRD	9
DB_PUNNT_VAL	10
DB_SEQUEN	11
DB_STORE	12
DB_WIND_CO	13
DB_WIND_REC	14
PDB_CLOSE	15
PDB_OPEN	16

MODULE	DB_CLOSE	DB_CL
TYPE	SUBROUTINE	
VERSION	1.1.1	
AUTHOR(S)	M.NANNI	

DESCRIPTION

CHIUDE L'UNITA' ASSOCIATA AD UN SINGOLO CATALOGO , SE E' L'ULTIMO CATALOGO TRA QUELLI APERTI CHIUDE ANCHE GLI ARCHIVI DEL DATABASE.

USE:

CALL DB_CLOSE (N)

ARGUMENTS:

>>> N INTEGER
NUMERO DEL CATALOGO DA CHIUDERE

COMMON:

NONE

REMARKS:

IL NUMERO DI CATALOGO DA CHIUDERE DEVE ESSERE COMPRESO TRA
1 E 30 .

MODULE	DB_FORM
TYPE	SUBROUTINE
VERSION	3.1
AUTHOR(S)	M.NANNI

DESCRIPTION

COMPONE IL FORMAT RELATIVO ALLE VARIABILI CHE SI VOGLIONO
LEGGERE

USE:

CALL DB_FORM (N, VAR_FIS, FORMA, IERR)

ARGUMENTS:

>>> N REQUIRED INPUT INTEGER
NUMERO DEL CATALOGO SU CUI SI OPERA

>>> VAR_FIS REQUIRED INPUT CHARACTER
STRINGA CONTENENTE IL NOME IN CODICE DELLE
GRANDEZZE FISICHE DI CUI SI VOGLIA OTTENERE
IL FORMAT

>>> FORMA REQUIRED OUTPUT CHARACTER*600
FORMAT PER LA LETTURA DELLE GRANDEZZE IN ESAME

>>> IERR REQUIRED OUTPUT INTEGER
CODICE DI ERRORE

IERR =0 O.K.
IERR >0 NON TROVATA IN QUEL CATALOGO
GRANDEZZA FISICA DI POSIZIONE
IERR IN VAR_FIS

COMMON:

/DB_UNIT/ PARAMETRI DEI CATALOGHI APERTI
/DB_OUT / PARAMETRI DA PASSARE AL PERSONAL DB

REMARKS:

LE GRANDEZZE FISICHE CHE SI VOGLIONO OTTENERE DAL CATALOGO
DEBONO ESSERE SPECIFICATE NELLA STRINGA VAR_FIS UTILIZZANDO
I CODICI DEL VOCABOLARIO.
DUE CODICI POSSONO ESSERE SEPARATI DA UNA VIRGOLA, DA UNO
SPAZIO OPPURE ESSERE CONTIGUI SE IL CODICE CHE PRECEDE E'
LUNGO 5 CARATTERI. SE NELLA STRINGA FIS_VAR VI SONO PIU'
DI 4 SPAZI CONTIGUI I CODICI CHE SEGUONO VENGONO IGNORATI.
CON UNA OPERAZIONE DI LETTURA DEI FORMATI POSSONO ESSERE
ESTRATTI FINO A 40 GRANDEZZE INDIPENDENTI, SU UN
MASSIMO DI 80 GRANDEZZE FISICHE.

MODULE	DB_FORM_ALL
TYPE	SUBROUTINE
VERSION	1.1
AUTHOR(S)	M.NANNI

DESCRIPTION

METTE A DISPOSIZIONE L'INTERO FORMAT PER LA LETTURA DI UN CATALOGO.

USE:

CALL DB_FORM_ALL (N, FORMA)

ARGUMENTS:

>>> N REQUIRED INPUT INTEGER
NUMERO DEL CATALOGO IN ESAME

>>> FORMA REQUIRED OUTPUT CHARACTER *600
STRINGA CONTENENTE L'INTERO FORMAT PER LA
LETTURA DEI DATI

COMMON:

/DB_UNIT/ PARAMETRI DEI CATALOGHI APERTI

MODULE DB_INDEX
 TYPE SUBROUTINE
 VERSION 1.1
 AUTHOR(S) M.NANNI

DB_INDI

DESCRIPTION

CHIUDE UN FILE APERTO AD ACCESSO SEQUENZIALE E LO RIAPRE
 AD ACCESSO DIRETTO: DOPO QUESTA CHIAMATA E' POSSIBILE
 EFFETTUARE UNA LETTURA DEL TIPO:

```
READ (NFOR'IPX)BUFF(1:NL)
DECODE (NL,FORM,BUFF(1:NL)) V1,V2,V2,...,Vn
```

DOVE NFOR : NUMERO DI UNITA' FISICA RICAVATO DA DB_OPEN
 BUFF : BUFFER AUSILIARIO CHARACTER*600
 NL : LUNGHEZZA TOTALE DEL RECORD
 FORM : FORMATO RICAVATO DALLA DB_FORM

UN FILE COSI' APERTO NON PUO' ESSERE LETTO IN MODO SEQUENZIALE
 SE NON RICHIAMANDO PRIMA LA DB_SEQUEN

USE:

```
CALL DB_INDEX ( N ,NL, IERR )
```

ARGUMENTS:

```
>>> N ..... REQUIRED INPUT INTEGER
      NUMERO DEL CATALOGO

>>> NL ..... REQUIRED OUTPUT INTEGER
      LUNGHEZZA IN BYTE DEL FILE DEL CATALOGO

>>> IERR ..... REQUIRED OUTPUT INTEGER
      CODICE DI ERRORE 0 =OK
                      10 =IL FILE NON AMMETTE ACCESSO DIRETTO
```

COMMON:

DB_UNIT PARAMETRI RELATIVI AL CATALOGO APERTO
 DB_MESS FLAG DEI MESSAGGI

REMARKS:

I FILES SCRITTI IN MODO SEQUENZIALE POSSONO ESSERE APERTI AD ACCESSO
 DIRETTO SOLO SE SONO DEFINITI COME FILE A LUNGHEZZA FISSA.
 NEI FILES DI GESTIONE DEL DATABASE IL PARAMETRO ACCESS MODE
 DEVE ESSERE: I OPPURE IALF O IDEC .
 SE NON E' DEFINITO QUESTO TIPO DI ACCESSO LA ROUTINE USCIRA' CON
 CODICE DI ERRORE =10

MODULE	DB_MSG
TYPE	SUBROUTINE
VERSION	1.1
AUTHOR(S)	M.NANNI

DESCRIPTION

LA ROUTINES ABILITA O DISABILITA LA SCRITTURA DI ALCUNI MESSAGGI DURANTE L'ESECUZIONE DEL PROGRAMMA

USE:

CALL DB_MSG (IFLG)

ARGUMENTS:

>>> IFLG		REQUIRED	INPUT	INTEGER
	FLAG PER LA SCRITTURA		0 =	NESSUN MESSAGGIO
			1 =	SCRIVE MESSAGGI

COMMON:

/DB_MESS/ FLAG DI SCRITTURA DEI MESSAGGI

MODULE	DB_NAME
TYPE	SUBROUTINE
VERSION	1.1
AUTHOR(S)	M.NANNI

DESCRIPTION

PERMETTE DI ASSEGNARE UN NOME AI CATALOGHI CHE VENGONO
A FAR PARTE DEL PERSONAL DATA BASE ED UN NOME AI FILES SCRITTI
DA ROUTINES DI SERVIZIO (ES. DB_WIND_CO)

USE:

CALL DB_NAME (FILE_NAME, CAT_NAME)

ARGUMENTS:

>>> FILE_NAME REQUIRED INPUT CHARACTER
NOME DEL FILE CHE SI INTENDE SCRIVERE SU DISCO

>>> CAT_NAME REQUIRED INPUT CHARACTER
NOME DEL CATALOGO NEL PERSONAL DATA_BASE

COMMON:

/DB_OUT_NAM/ NOME DEI FILE APERTI

MODULE	DB_OPEN
TYPE	SUBROUTINE
VERSION	1.5
AUTHOR(S)	M.NANNI

DB_

DESCRIPTION

LA SUBROUTINE APRE IL FILE CONTENENTE IL CATALOGO CERCATO ED ASSOCIA AD ESSO UN NUMERO DI UNITA' FORTRAN, SE IL CATALOGO SI TROVA SU NASTRO LA ROUTINE RICHIEDE IL MONTAGGIO DEL NASTRO.

USE:

CALL DB_OPEN (N, NAMEIN, NFOR, IERR)

ARGUMENTS:

>>> N REQUIRED INPUT INTEGER
 NUMERO DEL CATALOGO DA APRIRE (DA 1 A 30)

>>> NAMEIN REQUIRED INPUT CHARACTER
 NOME DEL CATALOGO

>>> NFOR REQUIRED OUTPUT INTEGER
 UNITA' FORTRAN ASSOCIATA A QUEL CATALOGO

>>> IERR REQUIRED OUTPUT INTEGER
 CODICE DI ERRORE: 0 = O.K.

COMMON:

/DB_UNIT/	PARAMETRI RELATIVI AL CATALOGO APERTO
/DB_MT_UNIT/	DATI RELATIVI ALLA UNITA' NASTRO
/DB_MESS/	SETTA FLAG DEI MESSAGGI

REMARKS:

I CATALOGHI CHE POSSONO ESSERE SIMULTANEAMENTE ACCESSIBILI DALL'UTENTE SONO 30 E SONO INDICATI DA NUMERI COMPRESI TRA 1 E 30. PER POTER RIASSEGNARE LO STESSO NUMERO AD UN ALTRO CATALOGO E' NECESSARIO PRIMA CHIAMARE LA SUBROUTINE DI DISASSEGNAZIONE DB_CLOSE.
LA ROUTINE PUO' ESSERE UTILIZZATA ASSIEME ALLA PDB_OPEN PER APRIRE CATALOGHI CHE FACCIANO PARTE DEL PERSONAL DATA BASE; IN QUESTO CASO E' IMPORTANTE RICORDARE CHE I 30 NUMERI AMMESSI SONO CONDIVISI DALLE DUE DIVERSE CHIAMATE.
LA VARIABILE NFOR RELATIVA AL CATALOGO APERTO VALE 69+N, QUINDI LE UNITA' LOGICHE DA FOR070 A FOR099 SONO IN USO DAL PROGRAMMA. SONO UTILIZZATE INOLTRE DAL PROGRAMMA LE UNITA' FOR001 E FOR002

MODULE	DB_PUNT_CAR	DB_PUNT_CAR
TYPE	SUBROUTINE	
VERSION	1.1	
AUTHOR(S)	M.NANNI	

DESCRIPTION

LA ROUTINE OPERA UNA RICERCA BINARIA SU UN FILE APERTO AD ACCESSO DIRETTO ED ORDINATO SECONDO LA GRANDEZZA FISICA IN ESAME; RITORNA IL NUMERO DEL RECORD CHE CONTIENE, O SEGUE IMMEDIATAMENTE, LA STRINGA CERCATA.

USE:

CALL DB_PUNT_CAR (N, GRF, STRIN, IPX, IFIND)

ARGUMENTS:

>>> N REQUIRED INPUT INTEGER
 NUMERO DEL CATALOGO

>>> GRF REQUIRED INPUT CHARACTER
 CODICE DELLA GRANDEZZA FISICA DA ESAMINARE

>>> STRIN REQUIRED INPUT CHARACTER
 STRINGA DI RICERCA

>>> IPX REQUIRED OUTPUT INTEGER
 NUMERO DEL RECORD TROVATO

>>> IFIND REQUIRED OUTPUT INTEGER
 CODICE DI RITORNO =0 STRINGA TROVATA
 =1 RECORD IMMEDIATAMENTE
 SUCESSIVO

COMMON:

/DB_UNIT/ PARAMETRI RELATIVI AL CATALOGO APERTO

REMARKS:

NONE

```
MODULE          DB_PUNT_CRD
TYPE            SUBROUTINE
VERSION        1.1
AUTHOR(S)      M.NANNI
```

DESCRIPTION

LA ROUTINE OPERA UNA RICERCA BINARIA SU FILE ORDINATI IN ALFA O DELTA RITORNA IL NUMERO DI RECORD ALLA COORDINATA SPECIFICATA.

USE:

```
CALL DB_PUNT_CRD ( N, V, IPX, IFIND, IERR)
```

ARGUMENTS:

```
>>> IUNT ..... REQUIRED INPUT  INTEGER
      UNITA' FISICA SU CUI DEVE LEGGERE

>>> V ..... REQUIRED INPUT  REAL*8
      VALORE DELLA COORDINATA SU CUI POSIZIONARSI IN SEC.

>>> IPX ..... REQUIRED OUTPUT  INTEGER
      NUMERO DEL RECORD ALLA COORDINATA RICHIESTA

>>> IFIND ..... REQUIRED OUTPUT  INTEGER
      CODICE DI RITORNO      =0 COORDINATA EFFETTIVAMENTE
                              TROVATA
                              =1 COORDINATA MAGGIORE DI QUELLA
                              RICHIESTA

>>> IERR ..... REQUIRED OUTPUT  INTEGER
      CODICE DI ERRORE      =0 O.K.
```

COMMON:

```
/DB_UNIT/          PARAMETRI RELATIVI AL CATALOGO APERTO
FLAG PER MESSAGGI
/DB_MESS/
```

REMARKS:

IL VALORE DELLA COORDINATA SI RIFERIRA' ALLA ASCENSIONE RETTA O ALLA DECLINAZIONE A SECONDA DI COME IL FILE DEI DATI E' ORDINATO E' QUINDI NECESSARIO CONOSCERE IL TIPO DI ORDINAMENTO DEL FILE PRIMA DI APPLICARE QUESTA ROUTINES

MODULE	DB_PUNT_VAL	DB_PUNT_VAL
TYPE	SUBROUTINE	
VERSION	1.1	
AUTHOR(S)	M.NANNI	

DESCRIPTION

LA ROUTINE OPERA UNA RICERCA BINARIA SU UN FILE APERTO AD ACCESSO DIRETTO ED ORDINATO SECONDO LA GRANDEZZA FISICA IN ESAME; RITORNA IL NUMERO DEL RECORD CHE CONTIENE, O SEGUE IMMEDIATAMENTE, IL VALORE CERCATO

USE:

CALL DB_PUNT_VAL (N, GRF, V, IPX, IFIND)

ARGUMENTS:

>>> N REQUIRED INPUT INTEGER
 NUMERO DEL CATALOGO

>>> GRF REQUIRED INPUT CHARACTER
 CODICE DELLA GRANDEZZA FISICA DA ESAMINARE

>>> V REQUIRED INPUT REAL
 VALORE DI RICERCA

>>> IPX REQUIRED OUTPUT INTEGER
 NUMERO DEL RECORD TROVATO

>>> IFIND REQUIRED OUTPUT INTEGER
 CODICE DI RITORNO =0 VALORE TROVATO
 =1 RECORD IMMEDIATAMENTE
 SUCESSIVO

COMMON:

/DB_UNIT/ PARAMETRI RELATIVI AL CATALOGO APERTO

REMARKS:

NONE

MODULE	DB_SEQUEN	DB_SEQUE
TYPE	SUBROUTINE	
VERSION	1.1	
AUTHOR(S)	M.NANNI	

DESCRIPTION

RIAPRE, PER UN ACCESSO SEQUENZIALE UN FILE GIA' ASSEGNATO
E' LA ROUTINES INVERSA DELLA DB_INDEX

USE:

CALL DB_SEQUEN (N , IERR)

ARGUMENTS:

>>> N REQUIRED INPUT INTEGER
 NUMERO DEL CATALOGO

>>> IERR REQUIRED OUTPUT INTEGER
 CODICE DI ERRORE 0 =OK

COMMON:

DB_UNIT PARAMETRI RELATIVI AL CATALOGO APERTO
DB_MESS FLAG PER MESSAGGI

```
MODULE          DB_STORE

TYPE            SUBROUTINE
VERSION        1.1
AUTHOR(S)      M.NANNI
```

DESCRIPTION

LA ROUTINE AGGIORNA IL PERSONAL DATA BASE SCRIVENDO I DATI RELATIVI AL CATALOGO SELEZIONATO ED ALLE VARIABILI E FORMATI DEFINITI NELLA DB_FORM .
QUESTA OPERAZIONE VIENE COMPIUTA DIRETTAMENTE QUANDO SI FA' USO DI ROUTINE DI ESTRAZIONE DI SOTTOCATALOGHI

USE:

```
CALL DB_STORE ( N, IERR )
```

ARGUMENTS:

```
>>> N ..... REQUIRED INPUT    INTEGER
      NUMERO DEL CATALOGO IN ESAME

>>> IERR ..... REQUIRED OUTPUT  INTEGER
      CODICE DI ERRORE:          0 = O.K.
                                  1 = ERRORE
```

COMMON:

```
/DB_UNIT/          PARAMETRI RELATIVI AL CATALOGO APERTO
/DB_OUT/           PARAMETRI RELATIVI AI FILE DA COSTRUIRE
/DB_OUT_NAM/       NOME DEI FILE APERTI
```

MODULE	DB_WIND_CO
TYPE	SUBROUTINE
VERSION	1.1
AUTHOR(S)	M.NANNI

DESCRIPTION

ESTRAE DA UN CATALOGO UN CAMPIONE LIMITATO IN ALFA E DELTA CON UN NUMERO SELEZIONATO DI VARIABILI; CREA UN FILE SEQUENZIALE CON RECORD DI LUNGHEZZA FISSA. ATTRAVERSO IL PARAMETRO IFCAT E' POSSIBILE INSERIRE UN RECORD DI DESCRIZIONE DEL FILE NEL PERSONAL DATA BASE IL NOME DEL CATALOGO NEL PDB ED IL NOME DEL FILE IN SCRITTURA SONO DEFINITI FACENDO USO DELLA ROUTINE 'DB_NAME', CHE VA' UTILIZZATA PRIMA DELLA CHIAMATA DB_WIND_CO, ALTRIMENTI IL CATALOGO AVRA' LO STESSO NOME DEL CATALOGO ORIGINALE, ED IL FILE SI CHIAMERA' PDB00.N ATTAVERSO L'ARGOMENTO IFAPP E' POSSIBILE APRIRE, CON MODALITA' APPEND, UN FILE GIA' ESISTENTE PER AGGIUNGERE NUOVI DATI; IN QUESTO CASO E' OBBLIGATORIO AVER DEFINITO IL NOME DEL FILE ATTRAVERSO LA DB_NAME.

USE:

```
CALL DB_WIND_CO ( N, VAR, ALF1, ALF2, DEC1, DEC2, IFCAT,  
                 IFAPP )
```

ARGUMENTS:

- >>> N REQUIRED INPUT INTEGER
NUMERO DEL CATALOGO DA CUI SI OPERA
- >>> VAR REQUIRED INPUT CHARACTER
STRINGA CONTENENTI I CODICI DELLE VARIABILI DA
ESTRARRE
- >>> ALF1 REQUIRED INPUT REAL*8
COORDINATA R.A. IN SECONDI DI TEMPO CHE LIMITA INFE-
RIORMENTE IL CAMPO
- >>> ALF2 REQUIRED INPUT REAL*8
COORDINATA R.A. IN SECONDI DI TEMPO CHE LIMITA SUPE-
RIORMENTE IL CAMPO
- >>> DEC1 REQUIRED INPUT REAL*8
COORDINATA DECLINAZIONE IN SECONDI D'ARCO CHE LIMITA
INFERIORMENTE IL CAMPO
- >>> DEC2 REQUIRED INPUT REAL*8
COORDINATA DECLINAZIONE IN SECONDI D'ARCO CHE LIMITA
SUPERIORMENTE IL CAMPO
- >>> IFCAT REQUIRED INPUT INTEGER

OPZIONE DI SCRITTURA NEL PERSONAL D.B. 0 =NO
 1 =SI
>>> IFAPP REQUIRED INPUT INTEGER
 OPZIONE PER APPENDERE I DATI IN UN FILE ESISTENTE
 0 =NO
 1 =SI

COMMON:

/DB_UNIT/	PARAMETRI RELATIVI AL CATALOGO APERTO
/DB_OUT_NAM/	NOME DEI FILE APERTI
/DB_OUT/	PARAMETRI RELATIVI AI FILE DA COSTRUIRE
/DB_MESS/	FLAG PER I MESSAGGI

REMARKS:

IL PROGRAMMA ACCEDE IN MODO SEQUENZIALE O DIRETTO A SECONDA DE
'ACCESS MODE' DEFINITO NELLA DESCRIZIONE DEL CATALOGO DI INPUT
NEL CASO SI VOGLIANO LEGGERE ZONE DI CIELO INTORNO ALLE ORE
24 IL LIMITE SUPERIORE DEL CAMPO DEVE ESSERE POSTO > 24

COMMON:

/DB_UNIT/	PARAMETRI RELATIVI AL CATALOGO APERTO
/DB_MESS/	FLAG PER I MESSAGGI
/DB_BUFF/	BUFFER

REMARKS:

IL PROGRAMMA ACCEDE IN MODO SEQUENZIALE O DIRETTO A SECONDA DE
'ACCESS MODE' DEFINITO NELLA DESCRIZIONE DEL CATALOGO DI INPUT
NEL CASO SI VOGLIANO LEGGERE ZONE DI CIELO PASSANTI PER LE ORE
24 IL LIMITE SUPERIORE DEL CAMPO DEVE ESSERE POSTO > 24

MODULE	PDB_CLOSE	DB_C
TYPE	SUBROUTINE	
VERSION	1.1.1	
AUTHOR(S)	M.NANNI	

DESCRIPTION

CHIUDE L'UNITA' ASSOCIATA AD UN SINGOLO CATALOGO , SE E' L'ULTIMO CATALOGO TRA QUELLI APERTI CHIUDE ANCHE GLI ARCHIVI DEL DATABASE.

USE:

CALL PDB_CLOSE (N)

ARGUMENTS:

>>> N INTEGER
NUMERO DEL CATALOGO DA CHIUDERE

COMMON:

NONE

REMARKS:

IL NUMERO DI CATALOGO DA CHIUDERE DEVE ESSERE COMPRESO TRA
1 E 30 .

MODULE	PDB_OPEN
TYPE	SUBROUTINE
VERSION	1.5
AUTHOR(S)	M.NANNI

DESCRIPTION

OPERA SUL PERSONAL DATA BASE CON MODALITA'
IDENTICHE ALLA DB_OPEN
LA SUBROUTINE APRE IL FILE CONTENENTE IL CATALOGO
CERCATO.

USE:

CALL PDB_OPEN (N, NAMEIN, NFOR, IERR)

ARGUMENTS:

>>> N REQUIRED INPUT INTEGER
 NUMERO DEL CATALOGO DA APRIRE (DA 1 A 30)

>>> NAMEIN REQUIRED INPUT CHARACTER
 NOME DEL CATALOGO

>>> NFOR REQUIRED OUTPUT INTEGER
 UNITA' FORTRAN ASSOCIATA A QUEL CATALOGO

>>> IERR REQUIRED OUTPUT INTEGER
 CODICE DI ERRORE: 0 = O.K.

COMMON:

/DB_UNIT/	PARAMETRI RELATIVI AL CATALOGO APERTO
/DB_MT_UNIT/	DATI RELATIVI ALLA UNITA' NASTRO
/DB_MESS/	SETTA FLAG DEI MESSAGGI

REMARKS:

I CATALOGHI CHE POSSONO ESSERE SIMULTANEAMENTE ACCESSIBILI
DALL'UTENTE SONO 30 E SONO INDICATI DA NUMERI COMPRESI TRA
1 E 30. PER POTER RIASSEGNARE LO STESSO NUMERO AD UN ALTRO
CATALOGO E' NECESSARIO PRIMA CHIAMARE LA SUBROUTINE DI
DISASSEGNAZIONE DB_CLOSE.
LA ROUTINE PUO' ESSERE UTILIZZATA ASSIEME ALLA DB_OPEN,
IN QUESTO CASO E' IMPORTANTE RICORDARE I 30 NUMERI LOGICI
AMMESSI SONO CONDIVISI DALLE DUE DIVERSE CHIAMATE.
LA VARIABILE NFOR RELATIVA AL CATALOGO APERTO VALE 69+N,
QUINDI LE UNITA' LOGICHE DA FOR070 A FOR099 SONO IN USO
DAL PROGRAMMA.
SONO UTILIZZATE INOLTRE DAL PROGRAMMA LE UNITA' FOR007 E
FOR008

APPENDICE B
ESEMPI DI PROGRAMMI

C
C
C
C
C

LETTURA DI 10 RECORD DA UN CATALOGO A SCELTA ESTRAENDO
LE COORDINATE ED UNA GRANDEZZA FISICA (FLOATING)

CHARACTER*80 RECORD
CHARACTER*120 PH_VAR,GRG
CHARACTER*600 FORMATI
CHARACTER*30 NAME,NAM
INTEGER ARO,ARM,DEG,DET

C--

TYPE *, 'NOME CATALOGO >>'
ACCEPT 2,NAME
TYPE *, 'GRANDEZZA SELEZIONATA (OLTRE LE COORDINATE) >>'
ACCEPT 2,GRG
PH_VAR='RAO, RAM, RAS, DEG, DEP, DES, '//GRG

C--

CALL DB_OPEN(1,NAME,N,IERR)
CALL DB_FORM(1,PH_VAR,FORMATI,IERR)

C--

DO 100 I=1,10
READ(N,FORMATI)ARO,ARM,ARS,DEG,DET,DES,TTT
WRITE (6,4)ARO,ARM,ARS,DEG,DET,DES,TTT
4
100
FORMAT (2(2I5,F9.2),F9.2)
CONTINUE

C--

CALL DB_CLOSE(1)

C--

END

```

C
C      ESTRAZIONI DI SOTTOCAMPI DAL CATALOGO ZWIGAL
C      CON COSTRUZIONE DI 3 FILE SEPARATI ED AGGIORNAMENTO
C      DEL PERSONAL DATA BASE
C
C      CHARACTER*600 FORM
C      CHARACTER*50 NAME
C      CHARACTER*100 VAL
C      REAL*8 AI,AF,DI,DF
C--
C      VAL='MAG'
C      NAME='ZWIGAL'
C      II=1
C      DI=0.
C      DF=170110.
C      CALL DB_MSG(II)
C      CALL DB_OPEN (1,NAME,NF,IRR)
C--
C-----PRIMA ZONA
C--
C      AI=100.
C      AF=1000.
C      CALL DB_NAME ('PART1.ZW','ZWCAT1')
C      CALL DB_WIND_CO(1,VAL,AI,AF,DI,DF,1,0)
C--
C-----SECONDA ZONA
C--
C      AI=10000.
C      AF=11000.
C      CALL DB_NAME ('PART2.ZW','ZWCAT2')
C      CALL DB_WIND_CO(1,VAL,AI,AF,DI,DF,1,0)
C--
C-----TERZA ZONA CENTRATA SULLE ORE 24
C--
C      AI=82000.
C      AF=82500.
C      CALL DB_NAME ('PART3.ZW','ZWCAT3')
C      CALL DB_WIND_CO(1,VAL,AI,AF,DI,DF,1,0)
C--
C      CALL DB_CLOSE (1)
C      END

```

```

C      PROGRAMMA PER LA LETTURA E PRESENTAZIONE SUL TERMINALE
C      DEI DATI DEL CATALOGO UGC COMPRESI IN UN INTERVALLO
C      LIMITATO DI ASCENSCIONE RETTA E DECLINAZIONE
C
      CHARACTER*600 FORMAT
      CHARACTER*300 BUFFER
      CHARACTER*20 NAME
      REAL*8 AL1,AL2,DC1,DC2
C--
C-----DEFINIZIONE INTERVALLO
C--
      AL1=3600.
      AL2=7200.
      DC1=0.
      DC2=60*3600
C--
C-----DEFINIZIONE ED APERTURA DEL CATALOGO
C--
      NAME='UGC'
      CALL DB_MSG(1)
      CALL DB_OPEN(1,NAME,IUN,IER)
      CALL DB_INDEX(1,IRL,IRR)
C--
C-----LETTURA E SCRITTURA SU TERMINALE
C--
10     CALL DB_WIND_REC(1,' ',AL1,AL2,DC1,DC2,BUFFER,FORMAT,
      1           IRL,IEND)
      WRITE(5,5)BUFFER(1:IRL)
      IF(IEND.NE.1)GOTO 10
C--
5      FORMAT (A80)
      END

```

```

C
C   PROGRAMMA PER LA LETTURA DI 10 RECORDS DAL CATALOGO
C   UGC A PARTIRE DA UN NUMERO DI CATALOGO DATO IN INPUT
C   IN QUESTO ESEMPIO IL CODICE DI RICERCA ED IL NUMERO
C   DI CATALOGO SONO CONSIDERATI STRINGHE ALFANUMERICHE
C   E IL CATALOGO IN ESAME SI SUPPONE ORDINATO SECONDO
C   IL NUMERO DI CATALOGO
C
CHARACTER*600 FORM
CHARACTER*50 NAME
CHARACTER VAL*30,GRF*19,BUFF*500
CHARACTER NN*10,NO*5
C--
TYPE *,'VALORE DI RICERCA >>'
ACCEPT 1,NN
1
FORMAT(A10)
C--
II=1
VAL='MAG,NO,RAO,RAM,RAS'
GRF='NO'
NAME='UGC'
C--
CALL DB_MSG(II)
CALL DB_OPEN (29,NAME,NF,IER)
CALL DB_FORM(29,VAL,FORM,II)
CALL DB_INDEX(29,NL,IERR)
CALL DB_PUNT_CAR(29,GRF,NN,IRC,IPC)
C--
DO 10 I=1,10
    READ(NF,IRC)BUFF(1:NL)
    DECODE(NF,FORM,BUFF(1:NL))FAG,NO,IO,IM,S
    TYPE 2,IRC,NO,FAG,IO,IM,S
2
    FORMAT(1X,I5,A7,F5.2,2I4,F6.3)
    IRC=IRC+1
10
CONTINUE
C--
CALL DB_CLOSE (29)
END

```

NOTE DELL'ULTIMA ORA

E' stata apportata una modifica alla struttura precedentemente descritta per quello che riguarda l'apertura dei cataloghi residenti su nastro: in particolare si e' resa disponibile l'opzione di copiare su disco in una area scratch i cataloghi al fine di poter utilizzare le routines che sfruttano l'accesso diretto.

La subroutine DB_OPEN provvede anche a questa operazione di trasferimento da nastro a disco e la chiamata alla subroutine e' ora :

```
CALL DB_OPEN( N, NAMEIN, IVERS, NFOR, IERR)
```

Dove IVERS = 1 permette di copiare su disco il catalogo.

Nel file MASTERDB.BAK il parametro 13 (Equinozio) e' divenuto un parametro composto dove gli ultime 3 caratteri rappresentano la data a cui le coordinate sono riferite mentre il primo carattere indica (con una V) che il file su nastro puo' essere copiato su disco.

Per mantenere pulito l'area scratch e' stato definito il comando DB_DIXAN che cancella tutti i file creati da un utente.