

JAERI-M

6980

データ・プール制御用プログラム“QOPEN”と
“DOYC” コンピュータコードシステムの構造

1977年3月

小山 謹二・中野 鴻*・黒井 英雄

この報告書は、日本原子力研究所が JAERI-M レポートとして、不定期に刊行している研究報告書です。入手、複製などのお問い合わせは、日本原子力研究所技術情報部（茨城県那珂郡東海村）あて、お申しこしてください。

JAERI-M reports, issued irregularly, describe the results of research works carried out in JAERI. Inquiries about the availability of reports and their reproduction should be addressed to Division of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken, Japan.

データ・プール制御用プログラム"QOPEN"と
"DOYC"コンピュータコードシステムの構造.

日本原子力研究所東海研究所原子炉工学部

小山 謙二・中野 鴻^{*}・黒井 英雄

(1977年1月31日受理)

高速炉の炉物理解析用総合コードシステム"DOYC"を組み上げるために作成したデータプール制御用プログラム"QOPEN"およびマルチファイル制御用プログラム"ONO"の機能と使用方法を示す。

さらに"DOYC"の概要とその構造を示し、データプールのしめる役割を明らかにする。なお"QOPEN"、および"ONO"は総べてFORTRANによって記述されており、データプールは直接アクセス可能な大容量記憶装置上に作られている。

*) センチュリー・リサーチ・センタ株式会社

Data Pool Handling Program "QOPEN"
and
Structure of A Computer code system "DOYC"

Kinji KOYAMA, Koo NAKANO* and Hideo KUROI

Division of Reactor Engineering, Tokai, JAERI

(Received January 31, 1977)

A data pool handling code "QOPEN" and a multi file variable unit processing code "ONO" have been developed to link various data dynamically from a computing module to others on a computing system "DOYC" for analyzing fast reactor physics problems.

The basic structure of "DOYC" is shown to indicate a role of the data pool in the code system.

QOPEN and ONO are both described in FORTRAN; the data pool is in a direct access mass storage device.

*) Century Research Center Corp.

目 次

1. ま え が き	1
2. データプール制御用プログラム“QOPEN”	2
2.1 “QOPEN”の概要とデータプールの構造	2
2.2 “QOPEN”の機能	3
2.3 “QOPEN”の使用法	6
2.4 “QOPEN”からのメッセージ	10
3. マルチファイル制御用プログラム“ONO”	11
3.1 “ONO”の概要	11
3.2 マルチファイルの構造	12
3.3 “ONO”の使用法	13
4. DOYCシステムの構造とデータプールの役割	15
4.1 DOYCシステムの概要	15
4.2 DOYCシステムの構造とデータプール	15
5. ま と め	22
付録1 “QOPEN”のプログラムリスト	24
付録2 “ONO”のプログラムリスト	28

1. ま え が き

炉物理の解析用コードの総合化の要請が強まっており、Savannah River の JOSHUA^{1,2)}、ANL の ARC³⁾ 等に代表されるコードシステムが発表されている。我々も高速炉の炉物理解析用コードシステム DOYC を作成し、すでに多くの成果を得ている^{4,5)}。

コードのシステム化は、システムを構成する各計算モジュール間の結合と、データの受授を有機的、動的に行なう事によってなされる。従っていずれのコードシステムもデータプールと言われている機能をシステム内に持たせ、計算モジュール間のデータの受授を行なっている。然しこの機能は、対象としている計算機の持つ機能に直接依存する所が多く、データプールの構造とか制御方式も使用する計算機の機種によって異なる。

この報告は、FACOM 230-60/75を対象機種として開発した高速炉の炉物理解析用のコードシステム DOYC で用いているデータプール制御用コード QOPEN の設計の考え方と機能、その使用法と制限を述べる事を目的とし、データプールを採用する事によって得られる利点を DOYC システムの基本構造を用いて明らかにする。

データプール制御用コード QOPEN は総べて FORTRAN C/D⁶⁾ によって記述されており、使用している主なシステムルーチンは

DFILE, DREAD, . . . そして DWRITE

の3種類であり、その一般性は広く、簡単に他の計算機システムでも用いる事が出来る。

ここに示すデータプール制御用プログラムは、高速炉設計研究室およびCRC (センチュリ・リサーチ・センタ) などでも利用されている。

本報告で扱う事からは、原研で使用している大型計算機 FACOM 230-75 の機能を拡張する部分もあり、用語の選択がむつかしく、あいまいさが残る。したがって本報告をまとめるために使用した計算機用語の中、その定義を明らかにしておく必要があると思われるものを以下に示し定義しておく。

a) File (Logical Unit)

File とは File Name と EOF (End of File) により識別される一連の情報の名称であり、実行時には、JOB 制御文で指定する Logical Unit を介し入出力される。

b) Sub-file (Sub-logical Unit)

一つの File 中に我々が定義した複数の File (Sub-file) であり、一つの Sub-file は一つの Sub-logical Unit に対応する。Sub-file と Sub-logical Unit の関係は、JOB 制御文による指定機能を除いて File と Logical Unit の関係と等価である。

2. データプール制御用プログラム "QOPEN"

2.1 QOPEN の概要とデータプールの構造

データプール制御用プログラムQOPENは、1~50までの Logical Units (50の Files) を制御し、ダイレクト・アクセス (direct access) 機能をもつランダム・アクセス (random access) 装置上に、一種のデータプールを形成し、データの入出力制御を行なうものであり、すべてFORTRANで記述されている。

QOPENにより制御されているデータプールは、前述の File 群の下に、各々 200 までの Sub-file を持つことのできる 2 階の階層構造に組上げられており、Logical Unit 番号で指定する File 名と、Data 番号で指定する Sub-file 名によりデータの入出力管理を行なっている。データプールの構造は Fig. 1 に示されている。

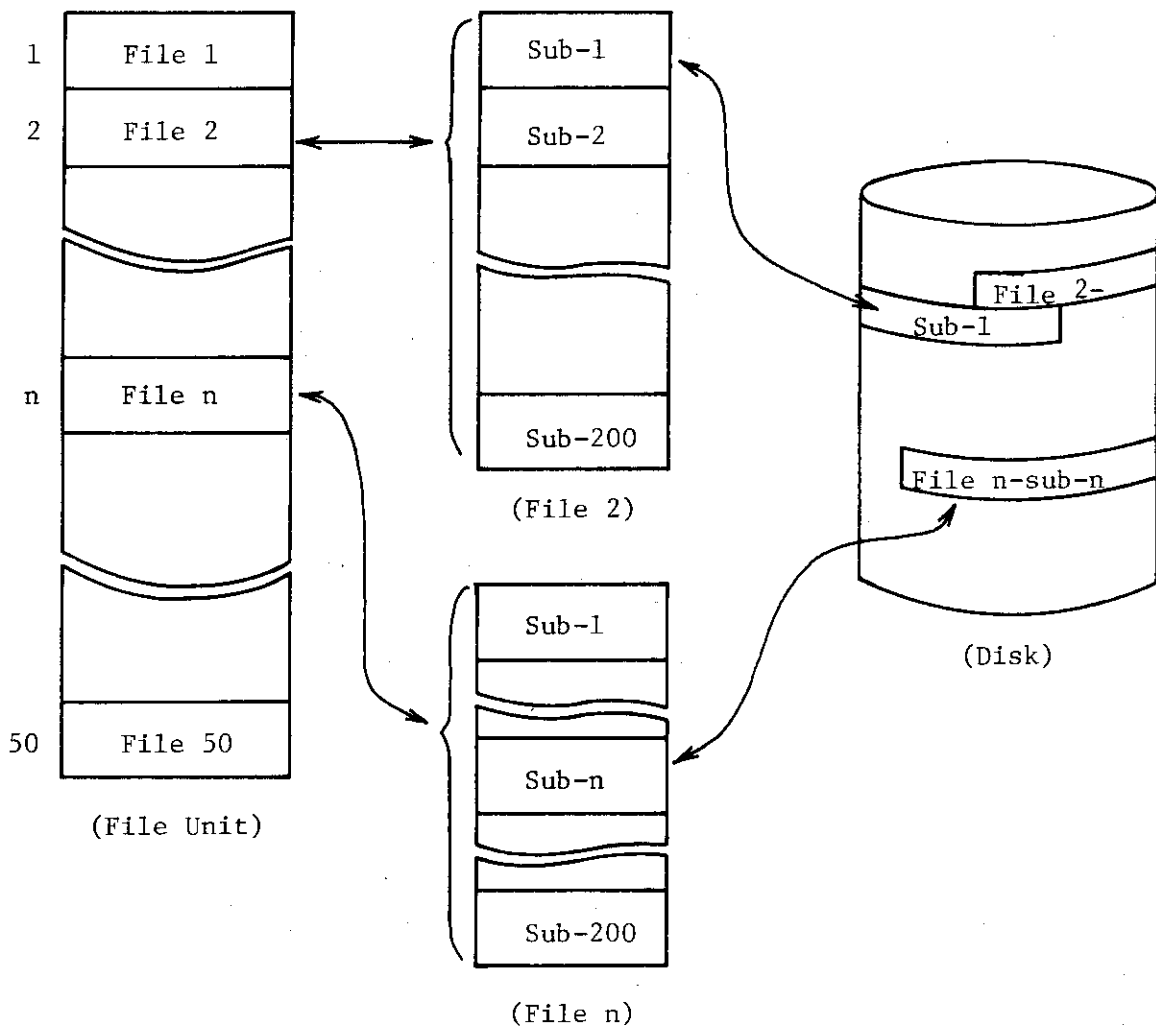


Fig. 1 Structure of Data pool controlled by QOPEN

このデータプールに登録するデータには、総べて File 名と Sub-file 名とによって定義されるデータ名を付ける。そしてこのデータ名を実行時に指定することにより、あらかじめ JOB 制御文で定義した Logical Units 群中の指定した機番に Sub-file 名を登録し、データを書き込む。データの読み出しを行なうためには、そのデータに付けられているデータ名を実行時に指定すればよい。

このような機能を持たせるために、QOPENの制御下にある各 File には、File 当り200迄の Sub-File の登録を許すカタログ領域をもうけ、データの入出力管理情報を記録している。この File の構造を Fig. 2 に示す。各 Sub-file の持つカタログ内には、データの入出力を直接制御するための書き込み開始番地と終了番地、データの参照回数、完了情報、データの優先順位、そしてデータの登録年月日等が記入されている (Fig. 3 参照)

さらにQDPENには、上記のようなデータの入出力の直接的な機能ばかりでなく、すでに登録されているデータ群 (Sub-file 群) のアップデート機能を初めとし、エディット、カタログリスト等の機能をも備えている。

2.2 QOPEN の機能

データプール制御プログラムQOPENはFACOM 230/60, 75用FORTRAN C/Dにより書かれており、使用しているシステム・ルーチンはDFILE, DREADおよびDWRITEであり、制御パラメータの受授にはLAELED COMMON/DTPL/ が用いられている。

QOPENとQOPENに含まれているエンタリー名は、QOPEN, QCHAN, QBIGR, QBIGW, QEND, QINITおよびQDATEである。次に各エンタリーの持つ機能を示す。

a) QOPEN (IIOO) $1 \leq IIOO \leq 50$

QOPENで用いるダイレクト・アクセス・ファイルの Logical Unit のオープンであり IIOOはオープンする Logical Unit (File 名に対応) を示す。すでにオープンされている場合はこの命令は無視される。

現状のQOPENで使用可能な IIOOは1~50までであり、QOPENによりオープンされた File の1レコードの物理語長は50語である。

b) QCHAN (IIOO)

現在指定されている Logical Unit IO 即ち File (IO) を File (IIOO) に変更する。書き込み又は読み込みのモードと旧機番 IO のインデックス (ダイレクト・アクセス・ファイルの READ/WRITE 開始番地) の値 IX は保存され、新機番 IIOO のモードと IX の値がセットされる。

c) QBIGR (NCALL) $1 \leq NCALL \leq 200$

NCALLが Sub-file に相当するものであり、カタログテーブルを参照し、NCALL によって定義される Sub-file のデータの読み込みの準備を行なう。インデックス IX に読み始めるレコードの番地がセットされる。

d) QBIGW (NCALL)

カタログテーブルを参照し、NCALL によって定義される Sub-file のデータの書き込み準備を始める。インデックス IX に書き始めるレコードの番地がセットされる。

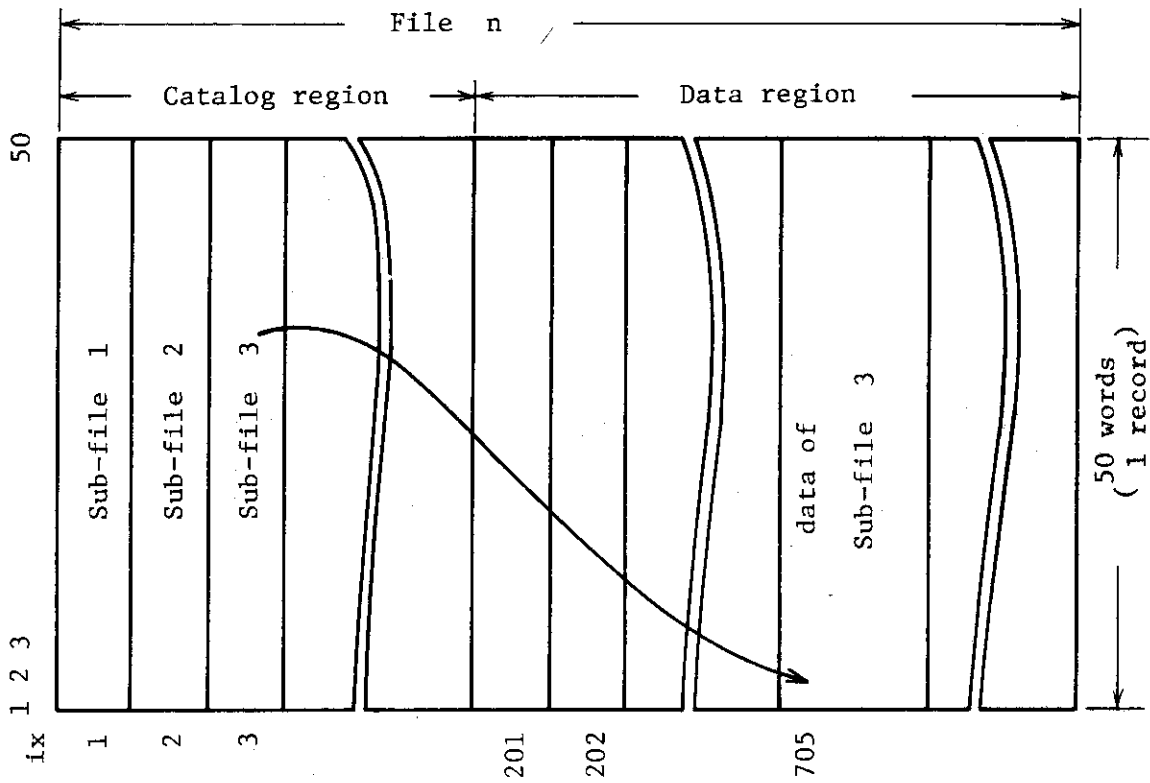


Fig. 2 Data structure of File n on random access device controlled by QOPEN

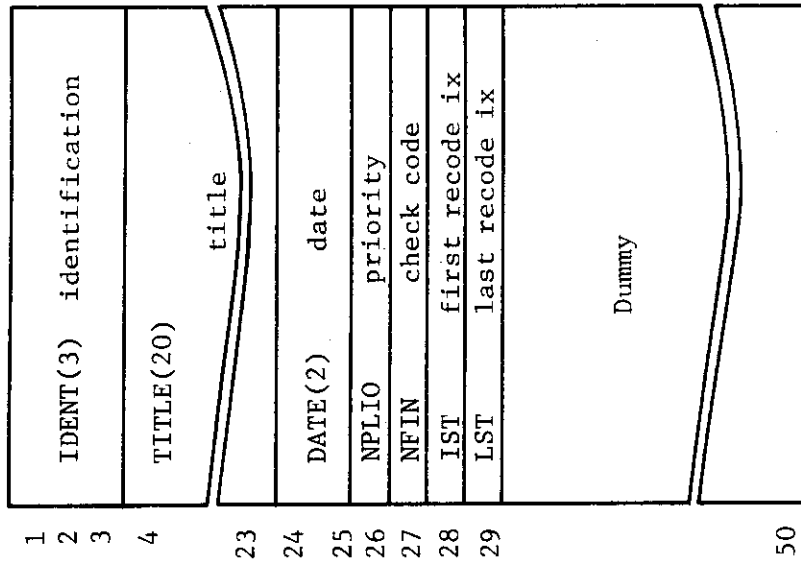


Fig. 3 Structure of catalog table

e) QEND

現在指定されている File (IIOO) のモード (READ/WRITE) に従ってカタログテーブルの書き換えを行なう (Sub-file のクローズ)。

- WRITE モードの場合, LABELED COMMON/DTPL/ の最初からの50語をカタログテーブルに書き込む。書き込み終了時には必ず呼びださなければならない。

f) QINIT

現在指定されている File (IIOO) のカタログテーブルの初期設定を行なう。カタログテーブル内に書かれていた Sub-file の情報は総べて消去される。

g) QDATE (NCALL)

現在指定されている File (IIOO) 内の Sub-file のアップデートを行なう。

- NCALL < 0

NCALL の絶対値に等しい Sub-file (NCALL) のデータを消去し, カatalogテーブルの内容をプリントする。

- NCALL = 0

カタログテーブルの内容をプリントする。

- NCALL > 0

NCALL に等しいデータの優先順位を持つ Sub-file のデータを総べて消去し, カatalogテーブルの内容をプリントする。

- NCALL \approx 0

削除されたデータ部分には, 後に続く他のデータが詰められる。

h) QDENT (IDENT, TITLE, NPL)

QDENT は, QBIGW により指定した Sub-file によりコメントおよびアップデートに用いるデータの優先順位を指定する NPL を設定し, 制御パラメータに記録するためのものであり, QEND の直前に呼びださなければならない。しかし単純な WRITE モードのみによるデータの書き込みを行なう場合は, QEND を呼びだす以前に, 後に示す LABELED COMMON/DTPL/ の中の COMMON 変数 IDENT, TITLE および NPLIO を定義しておけばよい。NPL < 0 として QDENT を呼びだすと, 現在実行中の Sub-file への書き込みの直前で, データの読み出しを行なっていた Sub-file の IDENT, TITLE および NPLIO がそのまま出力中の Sub-file の制御表に書き込まれる。

以上の様な機能を持つデータプール制御プログラムにより制御されているデータプール上の File の記録形式を示すと Fig. 2 のようになる。

1 レコード50語で構成されている File の最初の 200 レコードは 200 個の Sub-file 用のカタログテーブルとして専有されている。そして 201 レコード以後のレコードが実際のデータを記録する部分であり, カatalogテーブル内の情報によって制御されている。

1 ~ 200 レコードまでのカタログテーブル内の 1 つのレコードは, 1 つの Sub-file の制御情報を持っており, 1 レコード50語の制御情報は, 前に示した LABELED COMMON/DTPL/ の持つ最初の50語の COMMON 変数と一致している。次に /DTPL/ の COMMON 変数とその役割を示しておく。

NDENT (3) ; データの識別記号
 TITLE (20) ; データの表題
 NDATE (12) ; データの作成年月日
 NPLIO ; データの優先順位
 NFIN ; 完了コード
 IST ; データの最初のレコード番号
 LST ; データの最後のレコード番号
 NBUF (21) ; ダミー変数
 IO ; 入出力の指定をされている論理機番 (File (IO))
 IX ; 次の指令で書き込み又は読み込みを行なうためのダイレクト・アクセス・
 ファイル上のレコード番号

以上52語の変数の内最初の50語、NBUF (21) までの各Sub-file ごとにカタログテーブル中に記録されている。

以上に示した各エントリと制御情報により制御されているデータテーブルの使用法は次節に示すが、その機能にゆらりする制限は、同一File 内の異なるSub-file の内容をロジカルレコード単位で交互に入力、出力又は入出力する事は出来ない。しかし、1つのSub-file の入力または出力が完了した時点で他のSub-file の入出力を行なう事は出来る。一方、File の異なるSub-file の入出力には、何ら上記のような制限はない。

2.3 “QOPEN” の使用法

前節で示した機能を持つデータテーブル制御プログラム“QOPEN”の基本的な使用法を以下に示す。

a) File のオープンと初期設定

```

IIOO = 1
CALL QOPEN (IIOO) ; File (1) のオープン
CALL QINIT
  
```

Logical Unit IIOO = 1 のFile をオープンし、カタログテーブルの初期設定を行なう。このFile (1)には200迄のSub-file の登録が可能であり、Sub-file 群の使用するデータ領域はFD文で指定したトラック数の許す全レコード数から200レコード(カタログ領域)を差引いたレコード数である。

b) データの書き込み

```

IIOO = 2
CALL QOPEN (IIOO) ; File (2) のオープン
NCALL = 76
CALL QBIGW (NCALL) ; Sub-file = 75
DWRITE (IO, IX) LIST ; File (2), Sub-file (75) にLISTを書き込む。
CALL QEND ; カatalogテーブルの設定とクローズ
  
```

この例の場合、File (2) のカタログテーブルはすでに初期設定を完了しているものとする。

c) データの読み出し

```

IIOO = 3
CALL QCHAN (IIOO) ; File (3) に機番変更
NCALL = 51
CALL QBIGR (NCALL) ; Sub-file (51) からの読み込み準備。
DREAD (IO, IX) LIST1 ; 3-51 から LIST1 の読み込み
DREAD (IO, IX) LIST2 ; 3-51 から LIST2 の読み込み

```

この例では File (3) のオープンがすでになされており、Sub-file (51) に LIST1, LIST2 のデータが書き込まれている場合である。

d) "DO" ループによる連続書き込みと読み出し

通常の FORTRAN の READ/WRITE 文による LIST (J) の書き込みが続く読み出し方と、QOPEN による記述法を対比させると次のようになる。

通常の READ/WRITE	QOPEN による READ/WRITE
IIOO = 8	IIOO = 8
REWIND IIOO	CALL QCHAN (IIOO)
	CALL QBIGW (NCALL)
DO 1 J=1, 10	DO 1 J=1, 10
WRITE (IIOO) LIST (J)	DWRITE (IO, IX) LIST (J)
1 CONTINUE	1 CONTINUE
END FILE IIOO	CALL QEND
REWIND IIOO	⋮
⋮	CALL QBIGR (NCALL)
DO 2 J=1, 10	DO 2 J=1, 10
READ (IIOO) LIST (J)	DREAD (IO, IX) LIST (J)
2 CONTINUE	2 CONTINUE

この例の右側の列に示されている QOPEN によるデータプールへの READ/WRITE では Sub-file 番号 NCALL はすでに定義されているものとする。当然のことながら通常の FORTRAN による READ/WRITE では、この Sub-file 番号の指定はできない。

e) Sub-file の管理

① カタログテーブルのプリント

```

NCALL = 0
CALL QOPEN (IIOO) ; IIOO はデータプールの作られている Logical Unit 番号
CALL QDATE (NCALL)

```

この QDATE (NCALL = 0) によりプリントされたカタログテーブルの 1 例を Table 1. に示しておく。

② Sub-file のアップデート

Table 1 Sample list of catalog table in the data pool

NO.	IDENT	DATE	NPT	NFI	IST	LST	IRE	TITLE	
1	V12T HOMO N	76-12-22	2	1	201	276	0	FCA VI-2 TEST HOMO , NORMAL,TEST REGION	*** 001 ***
2	V12T HOMO N	76-12-24	2	1	318	393	0	FCA VI-2 TEST HOMO , NORMAL,DRIVER REGION	*** 002 ***
3	V12T HOMO N	76-12-24	2	1	468	543	0	FCA VI-2 TEST HOMO , NORMAL,RADIAL BLANKET	*** 003 ***
4	V12T HOMO N	76-12-24	2	1	393	468	0	FCA VI-2 TEST HOMO , NORMAL,AXIAL BLANKET	*** 004 ***
5	FP-LIBRARY	76-12-24	1	1	276	318	0	FP-LIBRARY OF DUVC-SYSTEM BY WOPEN 51-12-24	
11	VII-2 HOMO	77-01-05	2	1	621	699	0	FCA VII-2 TEST HOMO , NORMAL , INNER CORE (T1)	*** 011 ***
12	VII-2 HOMO	77-01-05	2	1	543	621	0	FCA VII-2 TEST HOMO , NORMAL , INNER CORE (D1)	*** 012 ***
13	VII-2 HOMO	77-01-05	2	1	699	777	0	FCA VII-2 TEST HOMO , NORMAL , INNER CORE (C/R,S/R)	*** 013 ***
14	VII-2 HOMO	77-01-06	2	1	933	1011	0	FCA VII-2 TEST HOMO , NORMAL , OUTER CORE (T2)	*** 014 ***
15	VII-2 HOMO	77-01-06	2	1	855	933	0	FCA VII-2 TEST HOMO , NORMAL , OUTER CORE (D2)	*** 015 ***
16	VII-2 HOMO	77-01-06	2	1	777	855	0	FCA VII-2 TEST HOMO , NORMAL , OUTER CORE (D21')	*** 016 ***
17	VII-2 HOMO	77-01-06	2	1	1167	1245	0	FCA VII-2 TEST HOMO , NORMAL , OUTER CORE (D21(61))	*** 017 ***
18	VII-2 HOMO	77-01-06	2	1	1089	1167	0	FCA VII-2 TEST HOMO , NORMAL , BLANKET (DEP,U**)	*** 018 ***
19	VII-2 HOMO	77-01-06	2	1	1011	1089	0	FCA VII-2 TEST HOMO , NORMAL , BLANKET (NAT,U***)	*** 019 ***
20	VII-2 HOMO	77-01-07	2	1	1245	1323	0	FCA VII-2 TEST HOMO , NORMAL , INNER CORE (T1')	*** 020 ***
21	VII-2 HOMO	77-01-11	2	1	1365	1443	0	FCA VII-2 TEST HOMO , NORMAL , INNER CORE (TIN)	*** 021 ***
30		77-01-08	1	1	1323	1365	0	FCA VII-2 TEST HOMO , (FP + T1')	*** 030 ***

```

NCALL = -20
CALL QOPEN (IIOO)
CALL QDATE (NCALL)
NCALL = 20
CALL QBIGW (NCALL)
DWRITE (IO, IX) LIST
CALL QEND

```

この例においては Sub-file 名, すなわち NCALL = 20 で指定した Data No. 20 の内容を消却し, 新たに LIST を NCALL = 20 として登録している。

③ Sub-file の編集

複数個の File に分散している Sub-file を新たな 1 つの File に編集するためには, Sub-file 単位のコピーが可能であればよい。以下にその方法を示す。

```

1  QCHAN (F 1)
2  QBIGR (SF1)
3  II = NIOT ,   IL = NLST
4  QCHAN (F 2)
5  QBIGW (SF2)
6  DO 11 I = II, IL
7  QCHAN (F 1)
8  DREAD (IO, IX) (A(J), J = 1, 50)
9  QCHAN (F 2)
10 DWRITE (IO, IX) (A(J), J = 1, 50)
11 CONTINUE
12 QDENT (IDENT, TITLE, NPLIO)
13 QEND

```

ステートメント番号 1, 2 では File (F 1), Sub-file (SF1) を READ モードでオープンする。ステートメント番号 3 の II 及び IL は, データ名 F 1-SF 1 のデータが記録されている最初と最後のレコード番号を保存するために必要である。

ステートメント番号 4, 5 はデータ名 F 2-SF 2 を WRITE モードでオープンする。

ステートメント番号 6 ~ 11 が実際のデータのコピーを行なう部分であり, コピーするデータのレコード数は, $IL - II + 1$ レコードである。ステートメント番号 12 で, $NPLIO < 0$ とするとあらたに作られた Sub-file SF 2 の IDENT, TITLE および NPLIO は Sub-file SF 1 のそれと同じとなる。

一方, $NPLIO \geq 0$ であれば, ステートメント番号 12 以前で IDENT, TITLE および NPLIO を定めておかなければならない。ステートメント番号 13 の QEND は File (F 2), Sub-file (SF 2) のクローズであり, 完了状況をカタログ領域に書き込む。

2.4 "QOPEN" からのメッセージ

QOPENを使用してデータの入出力制御を行なった場合、以下に示す情報がプリントされる。

- a) *** ERROR *** DATA NUMBER GT 200 IX = ---
 QBIGR/QBIGWで200以上のSub-file番号が使用された。
- b) *** ERROR *** DATA WRITTEN N = ---
 QBIGWですでに登録されているSub-file番号と同じSub-file番号にデータをWRITEしようとした。
- c) *** ERROR *** DATA NOT WRITTEN N = ---
 QBIGRでいまだ登録されていないSub-file番号N = ---のデータをREADしようとした。
- d) FOLLOWING DATA WAS REQUIRED UNIT = ---
 QBIGRでこのUNIT = ---から以下のSub-fileがREADされる。
- e) FOLLOWING DATA WAS STORED UNIT = ---
 QBIGWでこのUNIT = ---に以下のSub-fileがWRITEされた。
- f) *** ERROR *** LIST LESS THAN DATA N = ---
 データのREAD終了時にQENDを呼び出した場合、実際にWRITEされていたデータの最後までREADしなかった時にプリントされる。
 (READモードでは、読み込み終了時にQENDをCALLする必要はない)
- g) *** ERROR *** LIST EXCEED DATA N = ---
 データのREAD終了時にQENDを呼び出した場合実際にWRITEされていたデータより多くをREADした場合にプリントされる。
- h) DATA PACK HISTORY UNIT = ---
 QDATEを呼びだしLogical Unit UNIT = ---のSub-file群のUPDATEをした場合、アップデート完了後のSub-fileのカタログリストがリストされる。

3. マルチファイル制御用プログラム "ONO"

原研に導入されている FACOM 230/60/75 で使用されている FORTRAN C/D の READ/WRITE 機能を用いる時、1 度オープンされた File は、READ モードと WRITE モード間のモードの変換が不可能である。

このため CITATON 等の大型原子力コードの FORTRAN C/D への変換は不可能に近かった。そこでダイレクト・アクセス機能を用い、READ/WRITE のモード変換を可能にすると同時に、1 つの Logical Unit に多数の Sub-file を作り、更にこの Sub-file に対応する sublogical Unit はすべて Variable Unit として扱うことが出来る SUBROUTINE "ONO" を作った。この "ONO" により制御される File はすべてスクラッチとしてのみ使用されるものであるため、その使用法とパラメータの定義のみを示しておく。

3.1 "ONO" の概要

スクラッチ専用マルチ・ファイル制御用プログラム "ONO" はすべて FORTRAN-C/D によって書かれており、使用しているシステムルーチンは DFILE, DREAD そして DWRITE であり、制御用パラメータの受授は LABELD COMMON /DIRCT/ および /SEPT/ を介して行なう。

"ONO" に含まれているエントリ名は SETPNT 及び NPNT である。SETPNT は、"ONO" で制御する各 File に対して READ, WRITE, SKIP FILE, REWIND, BACKSPACE, EOF, SKIP 等の処理を行なう。また NPNT は File のオープン、制御用テーブル、そしてダイレクト・アクセス機能を持つ File の入出力を制御するインデックス (READ/WRITE 開始番地を示す) IX の指定を行なう。以下に制御用パラメータの定義を示す。

LABELED COMMON 中の制御パラメータ

COMMON /DIRCT/ IIOO (4000)

COMMON /SEPT/ IBUF (24)

EQUIVALENCE (IIOO(6), IX), (IIOO(7), IEND)

/DIRCT/ 及び /SEPT/ は "ONO" を用いるメインプログラムで宣言しなければならない。
/DIRCT/ はダイレクト・アクセス機能を持っている Logical Unit の READ/WRITE 開始番地を指定するインデックス IX および制御用カタログテーブルに使用し、/SEPT/ は、テーブルを作成するためのオプションとして使用する。

IX ; ダイレクト・アクセス用インデックス

IEND ; EOF を READ すると IEND=999 となる。通常は IEND=0

IBUF (1); ダミー

IBUF (2); Logical Unit の数 (= 5)

実際に使用する Logical Unit の数であり最大 5 Units まで割当てる事が

できる。割当てられる順序は F01, F02, F03, F04 そして F08 となっている。

IBUF(3) : 1つの Logical Unit に対するトラック数 (ディスクの使用量) であり, IBUF(2)で指定した全 Units について共通である。ただし JOB制御文(FD文)では IBUF(3)以上のトラック数を宣言しなければならない。

IBUF(4) : DFILEで宣言する時に指定する1レコード長当りの語数

IBUF(5) ~ IBUF(24)

; Sub-fileの専有するレコード数(トラック数)の修正に用いる。すなわち1つの Logical Unit に割当てられた Sub-logical Unit のレコード数を増減する時に用いる。現状では1~23までの Sub-logical Units (5,6,7を除く20 Units) について独立に指定可能である。

ここで示した Sub-logical Unit は, マルチ・ファイル制御用プログラム "ONO" の制御下では, FORTRAN C/Dで処理する場合の logical Unit と同等の, 独立した Unit として扱われている。

3.2 マルチ・ファイルの構造

Logical Unit 中の Sub-logical Unit の割当ての一例を Fig. 4 に示す。

この図に示した Sub-logical Unit の割当ては, IBUF(2)=4 (F01~F04 までの4つの Logical Units を使用する), IBUF(3)=200 (各 Logical Unit の使用するトラック数は各々200トラック), IBUF(4)=100 (1レコード当りの語数は100語) の場合についてである。すなわち各 Sub-logical Unit は40トラックのデスク容量(1039レコード)を使用すること

F01*	1**	8	12	16	20	
F02	2	9	13	17	21	
F03	3	10	14	18	22	
F04	4	11	15	19	23	
record index	1	1040	2080	3120	4160	
		1039	2079	3119	4159	5199
tracks		40	80	120	160	200

* Logical Unit number

** Sub-logical Unit number

Fig. 4 Assignment of Sub-logical Units on Logical Units controlled by "ONO"

になる (1トラック \approx 2600語)。

上記の例は IBUF(5)~FBUF(23)のすべてが零の場合である。Fig. 4 に示されている Sub-logical Unit の割当について Logical Unit F01 の中に割当てられている Sub logical Units 1, 12, を 500 レコードずつ増し, 16を 300, 20を 100 レコード減らすためには, 以下の指定をすればよい。

```
IBUF(5) = 500 ;      500 レコード追加
IBUF(13) = 500 ;
IBUF(17) = -300 ;   300 レコード削減
IBUF(21) = -100 ;   100 レコード削減
```

この場合, JOB制御文 (FD文) で F01 Unit に対して指定したトラック数が 200 トラックでは 600 レコード分だけ不足して来る。このため, $600/26 + 1 = 24$ トラックだけ F01 Unit のトラック指定を増しておく必要がある。

3.3 “ONO” の使用法

マルチ・ファイル制御用プログラム “ONO” には NPNT と SETPNT のエントリーがあるが, この 2 つのサブルーチンの持つ機能とパラメータの定義をし, 使用法を示す。

a) サブルーチン NPNT (NU) 、

NU で指定した Sub logical Unit のインデックス (ダイレクト・アクセス・ファイルの READ/WRITE 開始番地) を IX に入れる。最初に NPNT (NU) が呼ばれた (CALL された) 時 /DIRCT/ の IIOO にカタログテーブルが作られる。

b) サブルーチン SETPNT (NU, NOPT)

NU ; Sub-logical Unit の指定

NOPT; NU Unit の動作の指示

- = 1 DWRITE によるデータの書き込み終了時に呼ぶ,
- = 2 DREAD によるデータの読み込み終了時に呼ぶ,
- = 3 EOF (End of File) を SKIP するために呼ぶ,
- = 4 REWIND
- = 5 1レコード分だけ BACK SPACE
- = 6 EOF まで SKIP
- = 7 1レコード SKIP

次に簡単な例を上げ, その使用法を示しておく。

1. CALL NPNT(2)
2. CALL SETPNT(2, 4)
3. DWRITE(2, IX)(A(J), J=1, 20)
4. CALL SETPNT(2, 1)

この例では 1 において Sub logical Unit 2 を指定し, 2 で REWIND, 3 で Sub-file 2 の先頭からデータ A(J) を J が 1 から 20 まで書き込み, 4 で EOF を書く。

以上の説明で明らかのように, このマルチ・ファイル制御用プログラム “ONO” を用いて使用

可能となった Sub-file (Sub logical Unit) は、バリアブルユニットとしても使用可能なスクラッチファイルであり、FORTRAN C/Dにより使用出来るファイルの機能を大巾に拡張している。また、ここで示した Sub-file と通状の File 指定の併用も可能である。"ONO" を使用した場合に出力される各 File の使用状況を Table 2 に示す。

Table 2 File assignments on the multi file control program "ONO"

DISK OPERATION ASSIGNMENT

```
WRITE ..... 1
READ ..... 1
SKIP FILE ..... 2
REWIND ..... 2
BACK SPACE ..... 2
END OF FILE ..... 2
SKIP ..... 2
```

1	1	200	1	200
2	2	200	2	200
3	3	200	3	200
4	4	200	4	200
5	5	200	5	200
6	6	200	6	200
7	11	200	7	200
8	12	200	8	200
9	13	200	9	200
10	14	200	10	200
11	15	200	11	200
12	16	200	12	200
13	17	200	13	200
14	18	200	14	200
15	19	200	15	200
16	20	200	16	200
17	21	200	17	200
18	22	200	18	200
19	23	200	19	200
20	24	200	20	200

4. DOYCシステムの構造とQOPENの役割

4.1 DOYCシステムの概要

DOYCシステムは、高速炉の炉物理解析用総合コードシステムであり、微分断面積処理、1950群による中性子スペクトル計算と断面積の縮約、そして2次元、3次元の臨界計算と積分データ処理、さらに実験データ処理と、最小二乗法による1950群断面積セットの修正処理までを含んでいる。この種の総合コードシステムを有機的に構成し、かつその使用法を簡単化するためには、プログラムの構成上からは、いわゆる動的結合 (Dynamic linkage) 機能を活用したモジュラー化が望ましく、データの転送と受授の面からは、データプールの概念を組み入れたシステムである事が望ましい。

しかし、DOYCシステムは原研に導入されている計算機により計算出来る形式で作り上げなければ、コードシステムとしての有用性は半減する。現在原研に導入されている大型計算機^{*}にはプログラムエレメントの動的結合機能がなく、理想的なモジュラーシステムを作ることは出来ない。一方データプールの機能も使用する計算機システムの仕様に大きく依存する。いずれの計算機を用いるにしても、総合コードシステムを組み上げ、有効に利用するためにはデータプールの機能を持たせなければならない。DOYCシステム用に開発したデータプールの構造と機能の詳細はすでに示した。ここでは、DOYCシステムの構造とデータプールのしめる役割を明らかにする。

4.2 DOYCシステムの構造とデータプール

群定数の作成に始まり、実験解析とその評価さらに群定数の修正処理の過程を含む高速炉の炉物理解析用総合コードシステムDOYCは、すべてFORTRANで記述されており、さらに原研に導入されている大型計算機により計算可能なことを条件として作成した。このためDOYCシステムは、4つのセクション (1つのセクションは1つのJOB Stepに相当する) に分割されており、各セクション間のデータの受授はデータプールを介して行なわれる。そして各セクションは互いに関連の強い複数の処理コードにより構成されている。DOYCシステムを構成するこれらのセクションの基本的な構造を図示するとFig. 5の様になる。この図に従って各部分の機能の概要を示す。

a) Editor

セクションに含まれている各処理コードの選択と計算処理の順序を定めるとともに、入力データの補充整備を行なう。

b) Processing codes

セクションに含まれている処理コード群の総称であり、Editorからの指令により実行に

* FACOM 230/75

入る。(Processing Codes 群の概要は Table 3 を参照のこと)

c) Cross section Library

セクションで必要とする基本的な断面積用ライブラリー、すなわち ENDF/B-III, AGLI または JAERI-Fast である。

d) Cross section Data Pool

“QOPEN”で制御されている断面積用データプールであり、すべて DOYC フォーマット (Table 4)により書き込まれ、いずれの処理コードからも READ/WRITE 可能である。

f) Flux Data Pool

“QOPEN”で制御されている中性子束用のデータプールであり、すべて DOYC 内で定められた形式に従って書き込まれている (Table 5)。いずれの処理コードからも READ/WRITE 可能である。

この様に DOYC システムを構成する各処理コードは、データプールを介し基本的なデータの受授を行なっている。マルチファイル制御用プログラム “ONO” は、この基本構造には現われてこないが、各セクションを構成する処理コード群の使用するスクラッチファイルの機番割当 (Logical Units の割当) を統一し、JOB 制御文を簡略化するばかりでなく、処理コード自体が必要とする File の Open 後のモード変換を可能にし、処理速度を速めている。

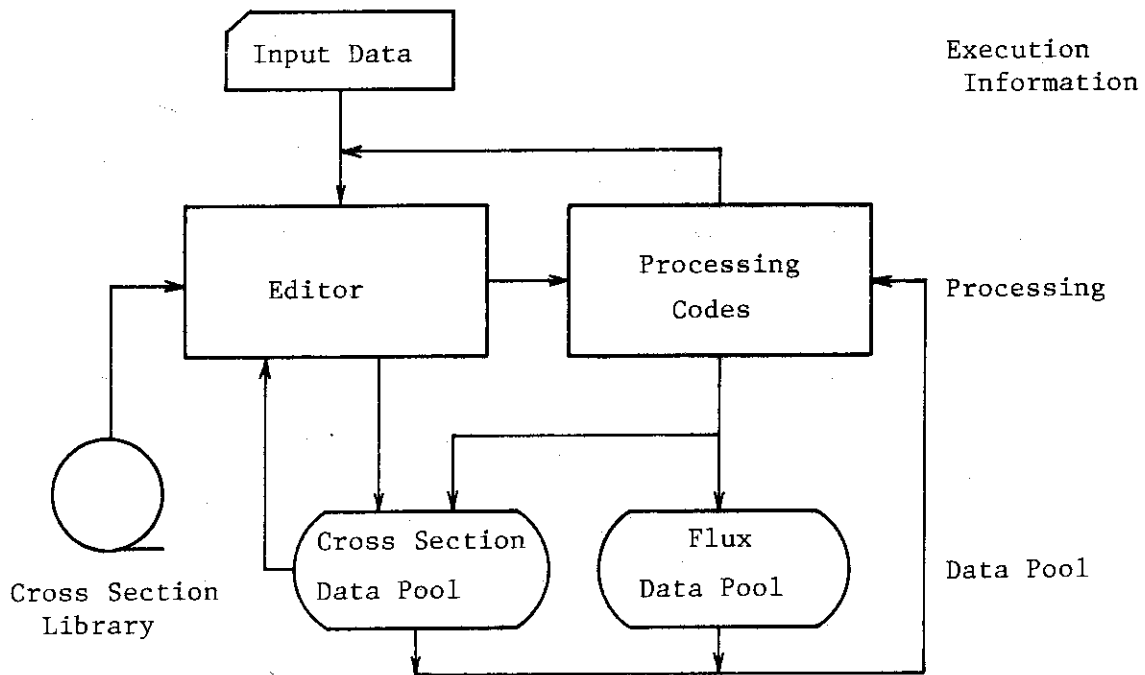


Fig. 5 Basic structure of the DOYC system

Table 3. Processing codes in DOYC computer code system.

Section 1 Generating AGLI-1950 energy group library from ENDF/B-III libraries and the others³⁾.

- POINTXS : Production of point-wise cross section data.
- GXS-2000 : Calculation of histogram parameters in 1950 energy group structure and compilation of the AGLI/A.
- CHEBY : Chebycheff expansions of the Legendre coefficients of scattering angle and the average numbers of neutrons per fission with respect to the incident neutron lethargy.
- FGINXS : Calculation of level-wise inelastic scattering cross section and compilation of the AGLI/B.

Section 2 Calculation of neutron fluxes and collapsing the fine group cross section of AGLI⁹⁾.

- DF-2000 : Calculation of neutron flux and eigen value in one dimensional geometry in about 2000 energy group structure by means of diffusion approximation. And collapsing the fine group cross sections by using the space dependent neutron spectrum as weighting function.
- SP-2000¹⁰⁾ : Calculation of neutron fluxes in a multi-region-cell by means of the collision probability concept, and collapsing the fine group cross sections by using the space dependent neutron spectrum as weighting function.

Section 3 Calculation of neutron fluxes and eigen values in a broad group structure less than 80 energy groups, and calculation of various integral data.

- MGPI : Calculation of neutron flux and eigen value in one dimensional geometry by means of P_1 approximation.
- ATRAS : Calculation of neutron flux and eigen value in one dimensional geometry by means of DSN approximation.

Table 3. continue

- HUMP : Calculation of neutron flux and eigen value in multi-region-cell by means of collision probability concept.
- 2DIF : Calculation of neutron flux and eigen value in two dimensional geometry by means of diffusion approximation.
(modified from CITATION)
- 3DIF : Calculation of neutron flux and eigen value in three dimensional geometry by means of diffusion approximation.
- 2DSN : Calculation of neutron flux and eigen value in two dimensional geometry by mean of DSN approximation.
- THINDA : Calculation of various integral data of sample reactivity worth, reaction rate et al., and collapsing broad group cross sections.

Section 4 Performing sensitivity analysis and adjustment of cross section.

- SALVAGE : Sensitivity analysis of cross section to various integral data.
- LEAST : LSQ adjustment of group cross sections.
- ARCADIA¹¹⁾ : On-line adjustment of the AGLI library.

Table 4. Format of cross section on the data pool in the DOYC system
(CROSS-SECTION DATA POOL (F08))

<u>RECORD</u>	<u>VARIABLE</u>	<u>IDENTIFICATION</u>
1.	MDENT(3)	identification of data
	IMAX	number of energy groups
	MESH	number of mesh points
	NMAT	number of materials
	INIG	number of inelastic energy groups
	JFIC	number of inelastic scattering terms
	JFIE	number of elastic scattering terms
	(CHI(i), i=1, IMAX)	fission neutron spectrum
	(FLX(i), i=1, IMAX)	energy spectrum used for collapsing
	(SIGT(i), SIGF(i), SIGI(i), SIGC(i), SIGN(i), D(i), i=1, IMAX)	correspond to Σ_T , Σ_f , Σ_{in} , Σ_c , $v\Sigma_f$ and D respectively
	((TRANI(i, j), j=1, JFIC), i=1, INIG)	$\Sigma_{in}^{i \rightarrow j}$
	((TRANE(i, j), j=1, JFIE), i=1, IMAX)	$\Sigma_e^{i \rightarrow j}$
	((TRMUL(i, j), j=1, JFIE), i=1, IMAX)	$\Sigma_e^{\mu e^{i \rightarrow j}}$
2.	((CIGC(i, j, k), CIGF(i, j, k), CIGN(i, j, k), k=1, NMAT), j=1, MESH), i=1, IMAX)	correspond to σ_c , σ_f and $v\sigma_f$ respectively

Table 5 Format of flux on the data pool in the DOYC system
(FLUX DATA POOL (F40))

<u>RECORD</u>	<u>VARIABLE</u>	<u>IDENTIFICATION</u>
1.	HOL(18)	identification of data
2.	IGE	geometry type
		=0 plane
		=1 cylinder
		=2 sphere
		=10 X-Y
		=11 R-Z
		=12 R- θ
		=13 triangler
		=20 X-Y-Z
		=21
		=22 R- θ -Z
		=23 triangler-Z
	NDA	flux type
		=0 normal flux
		=1 adjoint flux
		=2 normal & adjoint flux
	IGM	number of energy groups
	IM	number of radial intervals
	JM	number of rotational intervals
	KM	number of axial intervals
	NLAY	number of material layers
	IZM	number of material zones
	MT	total number of material
	MO1	number of mixture specification
3.	B01	left boundary condition (i=1)
		=0 vacuum
		=1 reflective
		=2 periodic
	B02	right boundary condition (i=IM)
	B03	bottom boundary condition (j=JM)
	B04	top boundary condition (j=1)
	B05	front boundary condition (k=1)

Table 5. (continue)

<u>RECORD</u>	<u>VARIABLE</u>	<u>IDENTIFICATION</u>
	B06	back boundary condition (k=KM)
4.	ZKEFF	value of K_{eff}
	FLPO	
		<0 total power
		>0 total center point flux
	VF	volume factor
	BUCK	buckling
5.	X1(IM+1)	radial mesh points
	X4(IM)	radial flux points
6.	Y1(JM+1)	rotational mesh points
	Y4(JM)	rotational flux points
	* if IGE > 10	
7.	Z1(KM+1)	axial mesh points
	Z4(KM)	axial flux points
	* if IGE > 20	
8.	LYN(KM)	material layer number
9.	MO(IM*JM)	material gone number
	* DO 1000 N=1,NLAY	
	WRITE(40) (MO(IJ),IJ=1,IM*JM)	
	1000 CONTINUE	
10.	M2(IZM)	material numbers by zone
11.	GAM(IZM)	buckling modifier by zonez
	* if BUCK > 0	
12.	K6(IGM)	fission neutron spectrum
13.	V7(IGM)	neutron velocity (cm/sec)
14.	NO(IM,JM)	normal flux
	* DO 1010 I=1,IGM	
	DO 1010 K=1,KM	
	WRITE(40) (NO(IJ),IJ=1,IM*JM)	
	1010 CONTINUE	
15.	N1(IM,JM)	adjoint flux

5. ま と め

データプールの効用は、計算機が大型化し、共用化が進められている現在、ますます大きくなっている。さらに解析もデータ・アンド・メソッドの確立が重要視されて来た事から、解析コードシステムも大型化し、データの処理と結合が複雑になって来ている。このため、汎用のデータプール処理コードの必要性が高まり、原研内でも原子力コード委員会総合化専門部会において、データプールの詳細設計に入り、試験段階に入ろうとしている。

DOYCシステム用に開発したデータプール処理コードQOPENは、すでに3年間の稼働実績があり、多くの成果を得ているが、残念ながら対象とした計算機FACOM 230/60/75のダイレクト・アクセス・ファイルの機能に不安定な部分があり、一般に公開した場合計算センターの運用に多大の支障をもたらす事が分かったので、今日まで公開する事が出来なかった。然し、この障害も51年8月に修正されたので公開することとした。

このQOPENが処理コードの総合化とシステム化の助けになれば幸いである。

謝 辞

データプールを作製するにあたり、直接アクセス装置の使用法をまとめて下さった計算センター稲見泰生氏（現在茨城大学）に感謝します。又大型計算機の使用の便を計って下さった計算センター平川室長及び斉藤直之氏に感謝致します。

参 考 文 献

- 1) Honeck H.C.; Suich J.E.: "JOSHUA - A Reactor Physics Computational System", Conf-69041.
- 2) Honeck H.C.: "Current Trends in the Development of Computer Systems for Reactor Analysis", Conf-730414-6.
- 3) Just L.C., et al.: "Recent Developments and Capabilities in the ARC System", Conf-69041.
- 4) Kuroi H., et al.: "Adjusted Cross Section Library AGLI and Reliability of Analysis of Integral Data", International Symposium on Physics of Fast Reactor (Tokyo), A-33 (1973).
- 5) Tone T.: "A Numerical Study of Heterogeneity Effects in Fast Reactor Critical Assemblies", J. Nucl. Sci. Technol., 12[8], (1975).
- 6) "FACOM 230 M-V FORTRAN 解説編 (I, II), 富士通
- 7) "FACOM 230-60 MONITOR V システム解説編" (I), 富士通
- 8) Kuroi H., et al.: "Computer Programs to Produce the 1950 Energy Group Cross Section Library AGLI from ENDF/B", EACRP-U-52, P-122, (1973).
- 9) Kuroi H., Tone T.: "Cross Section Data and Specification of AGLI/O for Fast Reactor Analysis (Adjusted Group Library by Integral Data)", JAERI-1230, (1973).
- 10) Kuroi H., Tone T.: "SP-2000 Program for Multi-region Cell and Effective Broad Group Constants", JAERI-1240, (1976).
- 11) Kuroi H., et al.: "ARCADIA a Comprehensive Semi-automated System for Cross Section Evaluation Utilizing Integral Measurements", JAERI-1241, 1976.

Appendix 1 Program list of "QOPEN"

FACOM 230-75 (M7) FORTRAN-C -760820-V06-L08 76.10.12 PAGE 98

* SOURCE STATEMENT *

```

1 C *** SUBROUTINE @OPEN (IIOO )
2 C *** DATA PACK OPERATION ROUTINE 72/06/20 KO,NAKANO @OP00010
3 C *** DATA PACK OPERATION ROUTINE 72/12/20 KO,NAKANO @OP00020
4 C *** DATA PACK OPERATION ROUTINE 73/06/20 KO,NAKANO @OP00030
5 C *** DATA PACK OPERATION ROUTINE 73/12/20 KO,NAKANO @OP00040
6 C *** COMMON/DTPL/IDENT(3),ITL(20),NDATE(2),NPT ,NF,N ,JST @OP00050
7 * COMMON/ETPL/OND (50),JND (50),KND (50),LND (50),MND (50),NND (50) @OP00060
8 * DIMENSION IND (50),NDENT(3),NOL (18) @OP00070
9 * EQUIVALENCE (IDENT ,IND ) @OP00080
10 INTEGER ITL @OP00090
11 DATA LND /50*0 /,MND /50*0 /,NND /50*0 / @OP00100
12 C *** DATA FILE OPEN *** @OP00110
13 IO = IIOO @OP00120
14 IF (LND(I0),GT,0) RETURN @OP00130
15 CALL DFILC(I0, 50,IX) @OP00140
16 LND(I0)= 201 @OP00150
17 RETURN @OP00160
18 C *** FILE INITIALIZE *** @OP00170
19 ENTRY @INIT @OP00180
20 DO 1000 I=1,50 @OP00190
21 JND(I) = 0 @OP00200
22 JND(28)= 201 @OP00210
23 DO 1010 I=1,200 @OP00220
24 IX = I @OP00230
25 DWRITE(I0,IX) JND @OP00240
26 RETURN @OP00250
27 C *** FILE CHANGE *** @OP00260
28 ENTRY @CHAN(IIOO) @OP00270
29 LND(I0)= IX @OP00280
30 MND(I0)= IZ @OP00290
31 NND(I0)= NCALL @OP00300
32 IO = IIOO @OP00310
33 IX = LND(I0) @OP00320
34 IZ = MND(I0) @OP00330
35 NCALL = NND(I0) @OP00340
36 RETURN @OP00350
37 C *** FILE WRITE @BIGW(MCALL) @OP00360
38 ENTRY @BIGW(MCALL) @OP00370
39 NCALL = MCALL @OP00380
40 IX = NCALL @OP00390
41 IF (IX,GT,200) WRITE (6,9006) IX @OP00400
42 IF (IX,GT,200) STOP *** CF @BIGW *** @OP00410
43 FORMAT (//** ERROR ** DATA NUMBER GT 200 IX=,I4/) @OP00420
44 DREAD (I0,IX) JND @OP00430
45 IF (JND(27),EQ,0) GO TO2000 @OP00440
46 WRITE (6,9000) @OP00450
47 FORMAT (//** ERROR ** DATA WRITTEN @OP00460
48 NCALL,(JND(I),I=1,3),(JND(J),J=24,25) @OP00470
49 N=1,I4,4X,3A4,4X,2A4/) @OP00480
50 STOP @OP00490
51 CALL DATE(NDATE) @OP00500
52 IST = JND(28) @OP00510
53 IRE = JND(30) @OP00520
54 IX = IST @OP00530
55 LND(I0)= IX @OP00540
56 IZ = I @OP00550
57 RETURN @OP00560
58 C *** FILE READ @BIGR(MCALL) @OP00570
59 ENTRY @BIGR(MCALL) @OP00580
60 NCALL = MCALL @OP00590

```

* SOURCE STATEMENT (@OPEN)*

```

50 IX = NCALL
51 IF (IX.GT.200) WRITE (6,9006) IX
52 IF (IX.GT.200) STOP
53 JND = JND + 1
54 DREAD (10,IX)
55 IF (JND(27).NE.0) GO TO3000
56 WRITE (6,9001) NCALL
57 FORMAT(' ** ERROR ** DATA NOT WRITTEN N=',I4/)
58 STOP
59 3000 IX = JND(28)
60 IY = JND(29)
61 LND(10) = IX
62 IZ = 2
63 WRITE (6,9008) IO
64 WRITE (6,9005) NCALL,(JND(L),L=1,3),(JND(J),J=24,30),(JND(K),K=
65 * 4,23)
66 * FOLLOWING DATA WAS REQUIRED UNIT=' ,14)
67 RETURN
68 C *** FILE IDENTIFICATION ***
69 DO 2050 I=1,3
70 @DENT (NDENT ,NOL ,NPLIO )
71 IDENT(I)=NDENT(I)
72 CONTINUE
73 DO 2060 I=1,18
74 TITL(I)=NOL(I)
75 CONTINUE
76 TITL(19)=4H
77 TITL(20)=4H
78 NPT = NPLIC
79 RETURN
80 C *** TABLE SET ***
81 ENTRY @END
82 GO TO (4000,5000),IZ
83 4300 NFIN = 1
84 LST = IX
85 IX = NCALL
86 DWRITE(10,IX)
87 DO 4010 I=1,200
88 IX = I
89 DREAD (10,IX)
90 IF (JND(27).NE.0) GO TO4010
91 JND(28) = LST
92 IX = 1
93 DWRITE(10,IX)
94 CONTINUE
95 WRITE (6,9009) IO
96 * NCALL,(IND(L),L=1,3),(IND(J),J=24,30),(IND(K),K=
97 * 4,23)
98 * FOLLOWING DATA WAS STORED UNIT=' ,14)
99 FORMAT('0
100 RETURN
101 IF (IX=1Y)
102 JND(30)= JND(30)+1
103 IX = NCALL
104 DWRITE(10,IX)
105 JND
106 RETURN
107 WRITE (6,9002) NCALL,(JND(I),I=1,3),(JND(J),J=24,25)
108 FORMAT(' ** ERROR ** LIST LESS THAN DATA N=',I4,4X,3A4,4X,2A4/)
109 STOP
110 * NCALL,(JND(I),I=1,3),(JND(J),J=24,25)
111 * FOLLOWING DATA WAS EXCEED DATA N=',I4,4X,3A4,4X,2A4/)
112 9003 FORMAT(' ** ERROR ** LIST EXCEED DATA

```

@OP00600
@OP00610
@OP00620
@OP00630
@OP00640
@OP00650
@OP00660
@OP00670
@OP00680
@OP00690
@OP00700
@OP00710
@OP00720
@OP00730
@OP00740
@OP00750
@OP00760
@OP00770
@OP00780
@OP00790
@OP00800
@OP00810
@OP00820
@OP00830
@OP00840
@OP00850
@OP00860
@OP00870
@OP00880
@OP00890
@OP00900
@OP00910
@OP00920
@OP00930
@OP00940
@OP00950
@OP00960
@OP00970
@OP00980
@OP00990
@OP01000
@OP01010
@OP01020
@OP01030
@OP01040
@OP01050
@OP01060
@OP01070
@OP01080
@OP01090
@OP01100
@OP01110
@OP01120
@OP01130
@OP01140
@OP01150
@OP01160
@OP01170
@OP01180

FACOM 230-75 (MT) FORTRAN-D -760020- V06-L08 76.10.12 PAGE 100

* SOURCE STATEMENT (OPEN)*

```

105 STOP
106 C *** FILE UPDATE ***
107 ENTRY @DATE(MCALL)
108 NCALL = MCALL
109 IX = IABS(NCALL)
110 IF (IX.GT.200) WRITE (6,9006) IX
111 IF (IX.GT.200) STOP ** CF @DATE ***
112 IF (NCALL) 7000,6000,8000
113 6000 WRITE (6,9004) IO
114 9004 FORMAT('11 DATA PACK HISTORY UNIT=',14//', NO, IDENT
115 DO 6010 I=1,200 TITLE',/)
116 IX = I
117 DREAD (10,IX) JND
118 IF (JND(27).EQ.0) GO TO6010
119 WRITE (6,9005) I, (JND(L), L=1,3), (JND(J), J=24,30), (JND(K), K=
120 9005 FORMAT(1H0,13,2X,3A4,2X,2A4,2X,214,216,14,2X,20A4)
121 6010 CONTINUE
122 RETURN
123 7000 IX = IABS(NCALL)
124 DREAD (10,IX) JND
125 JND(26) = NCALL
126 IX = IABS(NCALL)
127 DWRITE (10,IX) JND
128 IST = JND(28)
129 LST = JND(28)
130 GO TO8080
131 8000 IST = 201
132 LST = 201
133 8080 CONTINUE
134 DO 8010 J=1,200
135 LLST = 7168021
136 DO 8020 I=1,200
137 IX = I
138 DREAD (10,IX) JND
139 IF (JND(26).EQ.0) GO TO8020
140 IF (JND(28).EQ.NCALL) GO TO8020
141 IF (JND(29).GE.LLST) GO TO8020
142 IF (JND(29).LE.LLST) GO TO8020
143 LLST = JND(29)
144 LLST = JND(28)
145 IY = I
146 CONTINUE
147 IF (LLST.EQ.7168021) GO TO8040
148 IX = IY
149 DREAD (10,IX) JND
150 JND(28) = IST
151 JND(29) = IST+LLST-111ST
152 IX = IY
153 DWRITE (10,IX) JND
154 DO 8030 I=111ST,LLST
155 IX = I
156 DREAD (10,IX) KND
157 IX = IST
158 DWRITE (10,IX) KND
159 LST = IX
160 8030 CONTINUE
161 IST = JND(29)

```

FACOM 230-75 (M7) FORTRAN-D -760620- V06-L08 76,10,12 PAGE 101

* SOURCE STATEMENT (@OPEN)*

```

161 LST = LLST
162 8010 CONTINUE
163 8040 DO 8050 I=1,50
164 8050 KND(I) = 0
165 KND(28) = 1ST
166 DO 8060 I=1,200
167 IX = I
168 DREAD (10,IX) JND
169 IF(JND(26).EQ.0) GO TO8070
170 IF(JND(26).EQ.NCALL)GO TO8070
171 GO TO8060
172 8070 IX = I
173 0WRITE(10,IX) KND
174 8060 CONTINUE
175 GO TO8000
C *** FILE CLOSE @CLOSE (1100) ***
176 ENTRY
177 RETURN
178 END
@OP01780
@OP01790
@OP01800
@OP01810
@OP01820
@OP01830
@OP01840
@OP01850
@OP01860
@OP01870
@OP01880
@OP01890
@OP01900
@OP01910
@OP01920
@OP01930
@OP01940
@OP01950
@OP01960

```


Appendix 2 Program list of "ONO"

FACOM 230-75 (M7) FORTRAN-D -760820- V00-L08 76.10.12 PAGE 92

* SOURCE STATEMENT *

```

1 SUBROUTINE SETPNI (NU, NLOPT)
2 COMMON/DIRECT/IBUF(4000)
3 DIMENSION I1(4000)
4 EQUIVALENCE (I100,I1)
5 EQUIVALENCE (I1(1), IX1), (I1(2), IX2), (I1(3), IX3), (I1(4), IX4),
6 (I1(5), IX5), (I1(6), IX6), (I1(7), IEND), (I1(8), I1),
7 (I1(9), I2), (I1(10), I3), (I1(11), I4), (I1(12), I5),
8 (I1(13), I6), (I1(14), I7), (I1(15), I8), (I1(16), I9),
9 (I1(17), J1), (I1(18), J2), (I1(19), J3), (I1(20), J4),
10 (I1(21), J5), (I1(22), J6), (I1(23), J7), (I1(24), J8)
11 DATA NV/0/
12
13 8000 FORMAT(1H0,1 LAST =,19/)
14 8001 FORMAT(1H0,1 VARIABLE EXCEED COMMON AREA, 1/)
15 8002 FORMAT(1H0,1 AN UNDEFINED UNIT NUMBER WAS FOUNDED, 1,19/)
16 8003 FORMAT(1H0,1 OUT OF PHYSICAL RECORD, UNIT 1,19,4X, PT 1,19/)
17 8004 FORMAT(1H0,1 OUT OF LOGICAL RECORD, UNIT 1,19,4X, PT 1,19/)
18 8005 FORMAT(1H0,1 EOF READ, UNIT 1,19,4X, PT 1,19/)
19 8006 FORMAT(1H0,1 LIST GREATER THAN DATA, UNIT 1,19,4X, PT 1,19/)
20 8007 FORMAT(1H0,1 LIST LESS THAN DATA, UNIT 1,19,4X, PT 1,19/)
21 8008 FORMAT(1H0,1 DISK OPERATION ASSIGNMENT, 1//
22 WRITE (.....) 1,1/
23 READ (.....) 2,1/
24 SKIP FILE (.....) 3,1/
25 REWIND (.....) 4,1/
26 BACK SPACE (.....) 5,1/
27 END OF FILE (.....) 6,1/
28 SKIP (.....) 7,1//)
29
30 8009 FORMAT(1H0,1 NOT SKIPPED RECORD, UNIT 1,19,4X, PT 1,19/)
31 8010 FORMAT(1H,1 R15,2(5X,815))
32 IJK(I,J) = (J-1)*IFOX+I
33 JKK(I,J) = (J-1)*IUNUN+I+15-1
34 NU = NV
35 IX = I(ENL)
36 IEND = 0
37 GO TO (1000,2000,3000,4000,5000,6000,7000),NOPT
38
39 C WRITE
40 C
41 1000 IF(IX.GE.MINI.AND.IX.LT.MAXI) GO TO1010
42 WRITE (6,8003)
43 STOP
44 1010 IF(I1(J3).LT.ILUN)
45 WRITE (6,8004)
46 STOP
47 1020 I1(J2) = 1
48 I1(J3) = I1(J3)+1
49 I1(J5) = I1(J3)
50 J5 = JKK(I,J,1)
51 I1(J5) = IX
52 RETURN
53
54 C READ
55 C
56 2000 IF(I1(J3).EQ.I1(J4))
57 I1(J5) = I1(J3)+1
58 J5 = JKK(I,J,1)
59 IF(IEND.EQ.999)
60 GO TO2030
61 IF(IX-I1(J5))
62 GO TO2030,2010
63
64 ONO00010
65 ONO00020
66 ONO00030
67 ONO00040
68 ONO00050
69 ONO00060
70 ONO00070
71 ONO00080
72 ONO00090
73 ONO00100
74 ONO00110
75 ONO00120
76 ONO00130
77 ONO00140
78 ONO00150
79 ONO00160
80 ONO00170
81 ONO00180
82 ONO00190
83 ONO00200
84 ONO00210
85 ONO00220
86 ONO00230
87 ONO00240
88 ONO00250
89 ONO00260
90 ONO00270
91 ONO00280
92 ONO00290
93 ONO00300
94 ONO00310
95 ONO00320
96 ONO00330
97 ONO00340
98 ONO00350
99 ONO00360
100 ONO00370
101 ONO00380
102 ONO00390
103 ONO00400
104 ONO00410
105 ONO00420
106 ONO00430
107 ONO00440
108 ONO00450
109 ONO00460
110 ONO00470
111 ONO00480
112 ONO00490
113 ONO00500
114 ONO00510
115 ONO00520
116 ONO00530
117 ONO00540
118 ONO00550
119 ONO00560
120 ONO00570
121 ONO00580
122 ONO00590

```

FACOM 230-75 (MT) FORTHAN-D -760820- V06=L08 76.10.12 PAGE 93

* SOURCE STATEMENT (SETPT)*

42	2010	WRITE (6,8006)	NU,11(J3)	ON000600
43	STOP			ON000610
44	2020	WRITE (6,8005)	NU,11(J3)	ON000620
45	STOP			ON000630
46	2030	11(J2) = 2		ON000640
47	11(J3) = 1			ON000650
48	IX	= 11(J5)		ON000660
49	RETURN			ON000670
	C	SKIP FILE		ON000680
	C			ON000690
50	3000	11(J3) = 11(J4)+1		ON000700
51	11(J2) = 2			ON000710
52	I	= 11(J3)		ON000720
53	J5	= JKK(IJ,1)		ON000730
54	IX	= 11(J5)		ON000740
55	RETURN			ON000750
	C	REWIND		ON000760
	C			ON000770
56	4000	J6 = 11(J2)+1		ON000780
57	GO TO	(4020,4010,4020),J6		ON000790
58	4010	11(J4) = 11(J3)		ON000800
59	11(J3) = 11(J3)+1			ON000810
60	I	= 11(J5)		ON000820
61	J5	= JKK(IJ,1)		ON000830
62	4020	11(J2) = 0		ON000840
63	11(J3) = 1			ON000850
64	J5	= JKK(IJ,1)		ON000860
65	IX	= 11(J5)		ON000870
66	RETURN			ON000880
	C	BACK SPACE		ON000890
	C			ON000900
67	5000	11(J5) = 11(J3)-1		ON000910
68	I	= 11(J3)		ON000920
69	J5	= JKK(IJ,1)		ON000930
70	IX	= 11(J5)		ON000940
71	11(J2) = 2			ON000950
72	RETURN			ON000960
	C	END OF FILE		ON000970
	C			ON000980
73	6000	11(J4) = 11(J3)		ON000990
74	11(J2) = 2			ON001000
75	11(J3) = 11(J3)+1			ON001010
76	I	= 11(J5)		ON001020
77	J5	= JKK(IJ,1)		ON001030
78	11(J5) = IX+1			ON001040
79	IX	= 11(J5)		ON001050
80	RETURN			ON001060
	C	SKIP		ON001070
	C			ON001080
81	7000	11(J3) = 11(J3)+1		ON001090
82	I	= 11(J5)		ON001100
83	J5	= JKK(IJ,1)		ON001110
84	IF (11(J5),GT,1)			ON001120
85	WRITE (6,8009)			ON001130
86	STOP			ON001140

60 T07010
NU,11(J3)

* SOURCE STATEMENT (SETPNT) *

```

87 7010 IX = 11(J5)
88 11(J2) = 2
89 IF(11(J3).GT.11(J4)) IEND=999
90 RETURN
91 ENTRY NPNT (NU ) GO T09020
92 IF(CNV.AE.0) DO 9000 I=1,2000
93 9000 1100(I) = 0
94 IFOX = IBUF(2)
95 JPYS = IBUF(3)
96 JPYS = IBUF(4)
97 IF(11(J5).LE.0) IFOX=5
98 IF(11(J5).LE.0) JPYS=200
99 IF(JPYS.LE.0) JPYS=100
100 INUN = 20
101 JCOL = (INUN+IFOX-1)/IFOX
102 11 = 24
103 11 = 11+JCOL*IFOX
104 12 = 12+INUN
105 13 = 13+INUN
106 14 = 14+INUN
107 15 = 4000-15
108 16 = 16/INUN
109 16 = 15+INUN*ILUN
110 16 = IPYS/JCOL
111 16 = IPYS*1024/JPYS
112 WRITE (6,8000) 16
113 GO TO (9001,9002,9003,9004,9008),IFOX
114 9008 CALL DFIL(8,JPYS,148)
115 9004 CALL DFIL(4,JPYS,144)
116 9003 CALL DFIL(3,JPYS,143)
117 9002 CALL DFIL(2,JPYS,142)
118 9001 CALL DFIL(1,JPYS,141)
119 1X1 = 0
120 1X2 = 0
121 1X3 = 0
122 1X4 = 0
123 1X5 = 0
124 1X6 = 0
125 KEY = -2
126 IF(1100(1).GT.0) KEY=-1
127 KEY = KEY+1
128 IF(KEY.NE.0) WRITE (6,R008)
129 16 = -JPYS+1
130 DO 9010 I=1,INUN
131 IPYST = 0
132 J = I
133 IF(J.LE.IFOX) GO T09011
134 DO 9012 K=1,JCOL
135 IF(J.GT.K+IFCX) IPYST=IPYST+IBUF(1-K*IFOX+4)
136 9012 CONTINUE
137 9011 CONTINUE
138 17 = I-1
139 18 = JKK(1,1)
140 19 = JKK(1,INUN)
141 IF(J.AE.5) J=J+3
142 16 = ((1-1)/IFOX)*IPYS+1
143 IF(KEY.GT.0.AND.MOD(1,5).EQ.0)
144 11(11+17)=J
145 11(12+17)=0

```

* SOURCE STATEMENT (SETPNT)*

```

146      I1(I13+I7)=1
147      I1(I14+I7)=0
148      I1(I18) = I6+IPYSI
149      I1(I19) = I1(I18)+IPYS-1+IRUF(I+4)
150      9009 IF(KEY.EQ.0) GO TO9010
151      WRITE (6,8010) I,J,IPYS,IPYST,I1(I18),I1(I19)
152      9010 CONTINUE
153      IF(KEY.EQ.0) GO TO9020
154      CALL MOJI
155      9020 IF(NV.EQ.NU) GO TO9050
156      IF(NV.EQ.23.AND.KEY.EQ.0) GO TO9013
157      NV = NU
158      DO 9030 I=1,IFOX
159      DO 9030 J=1,JCOL
160      IJ = IKK(I,J)
161      IF(I1(I1+IJ-1).EQ.NU) GO TO9040
162      9030 CONTINUE
163      WRITE (6,8002) NU
164      STOP
165      9040 NL = I
166      9050 NU = NL
167      IF(NU.EG.5)
168      DO 9060 I=1,5
169      DO 9060 J=1,5
170      I1(I+16)= I1(I+7)+IJ-1
171      I1(IJ) = I1(IJ)
172      I1(IJ) = JKK(IJ,I)
173      I1(IJ) = I1(IJ)
174      I1(IJ) = JKK(IJ,I)
175      I1(IJ) = I1(IJ)
176      I1(IJ) = JKK(IJ,I)
177      I1(IJ) = I1(IJ)
178      RETURN
      END

```

```

ONC01780
ONC01790
ONC01800
ONC01810
ONC01820
ONC01830
ONC01840
ONC01850
ONC01860
ONC01870
ONC01880
ONC01890
ONC01900
ONC01910
ONC01920
ONC01930
ONC01940
ONC01950
ONC01960
ONC01970
ONC01980
ONC01990
ONC02000
ONC02010
ONC02020
ONC02030
ONC02040
ONC02050
ONC02060
ONC02070
ONC02080
ONC02090
ONC02100

```