

(1977. 9. 22)

SPOT-6:高信頼性ソフトウェア開発のための言語システム

岩元莞二 紫合治 藤林信也
(日本電気(株) 中央研究所)

1. はじめに

SPOT-6は、プログラミング方法論の言語化、表現の高水準化、及びプログラミング標準の言語化を通して、中大形ソフトウェアの信頼性向上と開発保守の効率化を狙って開発されたプログラミング言語システムである。SPOT-6は次の様な特長をもつ。

・抽象化、情報の隠蔽、局所化のサポート

一つの対象的オブジェクトに対する複数の抽象的オペレーション、あるいは互に関連する幾つかのオペレーションを一つの GROUP モジュールとして定義する機能をもつ。

・モジュール仕様の記述

モジュール構造、パラメータと外部変数の使い方、モジュール内データの宣言等が明白に記述でき、文書性を高めるとともに広域なインタフェースの正当性チェックを可能にしている。

・文書性の高いプログラム記述

簡単なユーザ定義のデータタイプ、式マクロの導入による高水準化と、*goto less programming*のための制御構造により理解性の高い表現ができる。

・論理構造と物理構造の分離

抽象化の原則を現実のソフトウェア開発に適用するときの大きな問題であるオペレーションコールのオーバヘッドを解決するため、論理的意味を保ったままインラインコードに展開するサポート機能をもつ。

・自動インデンテーション機能

論理構造を視覚化したリストを提供するために、字下りを自動的に制御するリフオーマをもつ。

SPOT-6はインプリメンテーションの容易さのため PL/I コンパイラーのプリプロセッサとして、NEAC ACOS-6上で実現した。そのため PL/I の記法との融和性を考慮した記法となっている。以下、SPOT-6の言語上の特長を中心に、SPOT-6の概要を述べる。

2. モジュール構造

SPOT-6プログラムは、従来の外部手続きの機能をもつ procedure と、データやリソースの抽象化のための機能をもつ group から構成される。これらはコンパイル単位となる外部モジュールであり、それを仕様節(spec-section) を含む。group は複数の operation をまとめたものであり、各 operation はその group によって表され、一つの抽象データに対するアクセス手段を提供する。内部手続きに対応する

モジュールとして routine がある。 operation & routine は、内部データの宣言をまとめたデータ節 (data-section) を含むことができる。実行文列はプログラム節 (prog-section) にまとめられ、データ等の宣言 (仕様節がデータ節に置かれると分離してある)。これらのモジュールの構文の概要は次の通りである。

```

procedure ::= p-head spec-section [routine...] prog-section p-end
group ::= g-head spec-section [routine...] operation... g-end
operation ::= o-head [data-section] [routine...] prog-section o-end
routine ::= r-head [data-section] [routine...] prog-section r-end

```

ここで p-head, p-end 等はそれぞれモジュールのヘッダ文、エンド文を示す。

3. 抽象化や情報の隠蔽のためのモジュール化構造

複雑な問題が与えられた場合、まずその問題領域の概念に最も適した抽象的な機能やデータを選び出し、それらを使って問題を解き、次にそれらの抽象的オブジェクトを具体化するといふ stepwise refinement [1] 方式が有効である。この様な抽象を導入する場合、抽象的な機械やデータを使う側と、それを具体化する側を明確に分離し、使用者には使うために必要な情報（抽象オブジェクトの意味、アクセス方法）のみを与える、実現に関する情報は不要な詳細として隠す（情報隠蔽の原則 [2]）ことが重要となる。

従来のプログラミング言語に用意されている手続きやサブルーチンは機能の抽象化には有効であるが、データやリソースの抽象化には不充分である。データやリソースの抽象化では、実現のためのデータ構造とそれに付する複数の抽象的オペレーション（例えば、file の open, get, put, close ; table の clear, add, search など）の定義が必要となるが、オペレーションを従来の外部手続き等で実現すると、オペレーション間で共通に使われるデータが手続き間インターフェース データとしてシステムに広く分散し、隠蔽の原則や局所化の原則に反した理解しにくく複雑な構造となる。

```

G1 : GROUP ;
SPEC ;
_____
PARM P1--, P2-- ; _____ ①
OPER
  OP1( P1 IN, P2 OUT ), } ②
  OP2(-- RETURN (--), }
  ____ ;
GROUPVAR ____ ; _____ ③
_____
ENDSPEC ;
OP1 : OPER( P1, P2 ) ; } ④
ENDOPER OP1 ;
OP2 : OPER(-- RETURN (--)) ;
ENDOPER OP2 ;
_____
ENDGROUP G1 ;

```

Fig.1 Group definition

```

P : PROC ;
SPEC ;
_____
SUBGROUP
  G1(
    OP1(____)
    OP2(-- RETURN (--)) ; } ①
_____
ENDSPEC ;
PROG ;
_____
  G1.OP1(X, Y) ; _____ ②
_____
ENDPROG ;
ENDPROC P ;

```

Fig.2 Usage of group

Fig. 1 に group 記述例の構組を示す。仕様節 (SPEC; と ENDSPEC; で囲まれた部分) には、この group G1 が表す抽象データの使用法の記述 (パラメータ (①) やオペレーション (②) の宣言) やオペレーション間共通な内部データの宣言 (group 変数宣言 (③)) が含まれる。オペレーションの実現は④に書かれ。group を使う側にとっては、このうち使用法の記述のみが必要な情報である。

group を使ったプログラムの記述例を Fig. 2 に示す。group を参照する場合、そのモジュールの仕様節の subgroup 宣言文 (①) で、group名とそこでの参照するオペレーション名を宣言し、実行文中では group 名を修飾した形のオペレーションの呼び出しを行う (②)。但し、オペレーション名がそこでは一意的な名前なら group 名による修飾を省略でき。

4. モジュール仕様の記述

外部モジュールの仕様は内部データの宣言とともに仕様節に書かれ。文書性向上のため、同じ性質をもつ名前を類別して宣言し、さらにモジュール仕様書としてビのプロジェクトでも記述すべき内容を鍵語付き注釈文で記述できようとした。宣言文の見出しになら鍵語として下記のものがある。

FUNCTION, AUTHOR, DATE, REMOTEPROCEDURE, TYPE, EMACRO, SUBGROUPL,
SUBPROCEDURE, EXTERNALVAR, INTERNALVAR, PARAMETER, GROUPVAR, CONSTANT,
BUILTIN, CONDITION, EXTERNALFILE, INTERNALFILE, DEFAULT

これらには省略形が許されています。これらの鍵語は SPOT-6 プロセッサに対する指示を与えたりますが、良いプログラミング標準を提供しています。以下、特徴的な機能を説明する。

・ 鍵語付き注釈文

モジュールの機能概要、作成者、作成・修正日は、それぞれ、FUNCTION, AUTHOR, DATE で始まりセミコロンで終る自由形式で表現できる。これによりソース プログラムからこの様な項目だけを自動的に抽出することが可能である。

・ インタフェースの記述

外部モジュールとの間でビのようにデータが受け渡されるかを明記できる。これらのインターフェースは、下記の対応モジュールとの間でパラメータと外部変数の受け渡し方で記述する。

- 当該モジュールを呼び出すモジュール
- 当該モジュールが呼び出す下位モジュール
- 呼び出しによる上下位の関連はないが外部変数を通して関連する横のモジュール

データの受け渡し方を示すために属性の一つとしてアクセス属性を導入した。上位モジュールとの関係は parameter 宣言文、external 宣言文の中でアクセス属性として宣言する。下位モジュールとの関係は subprocedure 宣言文や subgroup 宣言文の中で宣言する。パラメータの使われ方にについてはパラメータ属性の一部として

アクセス属性を宣言する。外部変数の使われ方に関しては、パラメータ属性宣言に続く下記の形式の extref オプションを宣言する。

extref-option ::= EXTERNALREF (variable access-attribute [, variable access-attribute]...)

横のモジュールとの関係を記述するためには *remote-procedure* 宣言文がある。この宣言文の本体の形式はパラメータ属性宣言がない處を除いて *subprocedure* 宣言文と同じである。但し、不必要的複雑さを避けるために手続き名の代りに任意の手続きという意味の ANY を記述できる。

アクセス属性は呼び出される側の立場から定義され、次の意味を持つ。

IN : 値を参照するだけで変更しない

OUT : 入口では値が未定義で内部で値をセットする

INOUT : 入口で値が定義され内部で値を変更することがある

INX : 入口では値が定義されないが出口では未定義

PASS : 参照、変更をせず下位モジュールへの受け渡しだけを行う

PASS はその他と組合せて使用できる。上位モジュールとの関係では一般にこの様な使い方をするという宣言があり、その他との関係では、このモジュールの文脈ではその様な働きをしてくることを宣言する。Fig. 4 は Fig. 3 におけるモジュール構造での手続き P におけるインターフェースの記述である。

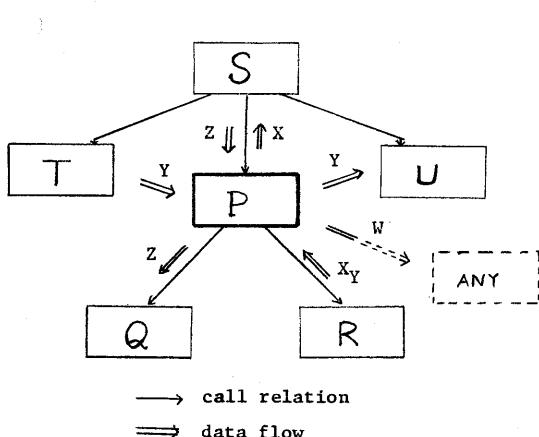


Fig. 3 Illustration of intermodule interface

```

P : PROC ;
SPEC ;
---
EXTVAR
  W BIN,
  X BIN OUT,
  Y CHAR(10),
  Z BIT(6) IN PASS ;
SUBPROC
  Q(--)
    EXTREF( Z IN ),
  R(--)
    EXTREF( X OUT, Y OUT ) ;
REMPROC
  T EXTREF( Y OUT ),
  U EXTREF( Y IN ),
  ANY EXTREF( W IN ) ;
---
ENDSPEC ;
---
ENDPROC P ;

```

Fig.4 Interface description
for the module P

5. モジュール内記述

プログラム節には PL/I の実行文が全て許されると、SPOT-6では group のオペレーションが使用できることはなく、ユーザ定義のデータタイプと式マクロによる表現の高水準化、制御構造の標準化のための goto-less 制御文が用意されており。

5.1 データタイプと式マクロ

マクロ機能はある記号列を他の記号列に置き換える便宜として多くのプログラミングシステムでサポートされてくる。しかし、構造が簡便で強力なことから、規則性を利用するとプログラムを理解する際に結局はマクロ展開した文書に頼ることになる。このため SPOT-6では、マクロ機能を言語要素として理解できるよう次の構造を導入した。

(1) ユーザ定義のデータタイプ

問題領域の概念に近い言葉をデータ扱う道具として、ユーザ定義のデータ属性を許す。機能は Pascal [3] のスカラタイプと同程度である。タイプ定義は仕様節で例えば

```
TYPE COLOR = ( WHITE, BLUE, YELLOW, RED );
INTVAR CAR-BODY-C COLOR;
```

で行き、プログラム節の中では、CAR-BODY-Cに値 WHITE, BLUE, YELLOW, RED を代入したり、値をそれらと比較することができる。

(2) 式マクロ

パラメータ付加可能なマクロで、文書性の低い式や参照を意味のある名標で表現するためのものである。構文上は式の許される位置で使用できる。宣言は仕様節 / データ節で例えれば下記の様に行う。

```
EMACRO NEPHEW( PERSON, I ) =
  PERSON.SON(I) → PERSON.
  NAME;
```

5.2 goto-less 制御構造

goto-less programming を支援するため、PL/I の制御文に加えて下記の文を導入した。

(1) DO FOREVER文とUNTIL文

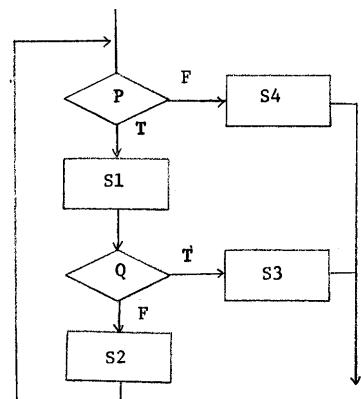
これにより無限ループとループの出口における脱出判定ができる。

(2) EXIT文とEPILOGUE オプション

ループの途中からの脱出を指示する EXIT 文と、脱出後の後処理を指示する EPILOGUE オプションが使える。Fig. 5(a) は Fig. 5(b) の構造を表現している。

```
DO WHILE(P);
  S1;
  EXIT [ABT] WHEN(Q);
  S2;
END EPILOGUE(
  [ABT]: S3;
  ENDED: S4; );
```

(a) Example program



(b) Flowchart for (a)

Fig. 5 Illustration of EXIT statement and EPILOGUE option

(3) CASE文

多岐選択構造として CASE構造文が使われる。選択条件の種類により 3つの変形がある(Fig. 6)。RANGE指定は効率の良いオブジェクト(FORTRANの計算型GOTO文的な実現)を指示するために使われる。

```
CASE e    RANGE[ lo : up ]
  [ C1, C2 ] : --- ;
  [ C3 ] :   --- ;
  ---
ELSECASE --- ;
ENDCASE ;
```

e はスカラ式、 C_i と e は比較可能な定数。 $RANGE$ 指示はオプションで、 lo 、 up 、 C_i が 10進整定数かスカラタイプ値の時に指定できる。

```
CASE ;
  [e1, e2] : --- ;
  [e3] : --- ;
  ---
ELSECASE --- ;
ENDCASE ;
```

e_i は論理値。

Fig. 6 Case structure

6. モジュール展開機能

SPOT-6 プログラムでは、従来外部変数を直接参照していた部分が group のオペレーション呼び出しとなることが多い、パラメータの受け渡し等の呼び出しオペヘッドを無視できない場合が生じる。プログラムの作成のし易さや理解性の高さから、プログラムの作成時には group や procedure で論理を表現し、プログラムの正しさの確認後必要に応じてモジュールを展開する方式をとる。この展開は論理的意味は同じた procedure や group の operation と同じ意味を保存するようを行う。展開処理はソースプログラムのレベルで行うことで展開後さらに言語プロセッサの最適化処理により効率の改善が行える。また、この展開機能は下記の点で通常のマクロ展開機能と異なる。

- operation の展開では group 変数をシステム全体で一意的な名前の外部変数に書き換え、呼び出しへりごとに宣言も書き換え。
- 埋め込まれた本体の RETURN 文を GOTO 文に修正する。これに伴う一意的なラベル名の生成も必要である。
- 式を返す RETURN 文はその呼び出しの位置によって(例えは、代入文の右辺か DO WHILE 文の条件式か)処理方法を変える。
- IF 文の THEN 節や ELSE 節なら DO ; と END ; で本体を囲む等。

モジュール展開機能はこれらの処理を行なうもので、展開処理がなされたプログラムは、一般的のプログラムが直接見ることの出来ない一時ファイルに入り、言語プロセッサによってオブジェクトに変換される。ユーザはひとつのモジュールの記述をもとにプログラムの保守管理をしてくれく。

7. 自動インデンテーション機能

プログラムの構造に従って字下りや改行を行なリフォーマと呼ばれる自動インデンテーション機能をもつ。このインデンテーションは仕様節とプログラム節の両方にわたって行い、詳細フローチャートなしにプログラムの構造が解きやすくなっている。リフォーマには次の2種類がある。

- S-リフォーマ ソースファイルそのものを INCLUDEマクロの展開なしに整形する。結果は新しいソースマスターファイルになる。
- I-リフォーマ ファイルはそのままでINCLUDEマクロ展開後のソースプログラムイメージを整形し印字出力する。この場合はプログラムのネスト構造の対応を分り易くするために、対応するレベルの始めと終りを縦線で結ぶ。

付録のプログラムリストはI-リフォーマの出力である。出力媒体に合せて出力イメージの行の最大長と字下りの単位長をリフォーマに対するパラメータとして指定できる。改行、改頁の指示やINCLUDEマクロをリスト上で展開させないなどの指示がソースプログラム中に書け、体裁の良いプログラムドキュメントを得ることができる。

8. システム構成

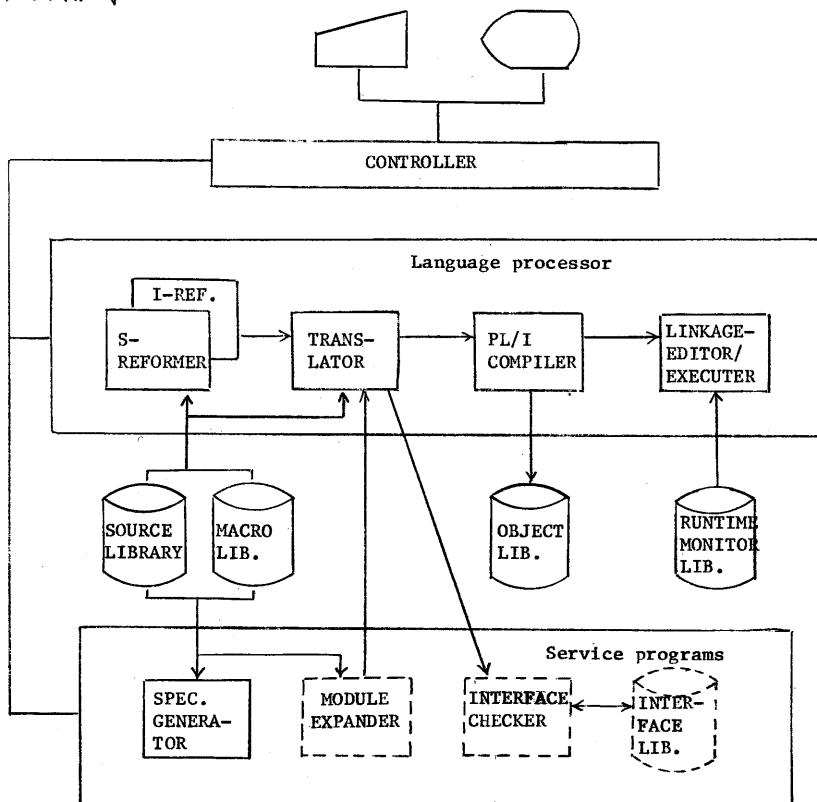


Fig. 7 SPOT-6 processor organization

SPOT-6言語で書かれたソースプログラムは(必要ならリフォーマをへて)トランスレータによってPL/Iプログラムに変換され、連続したアクティビティとしてPL/Iコンパイラを起動することによりオブジェクトプログラムに変換される(Fig. 8)。I-リフォーマとS-リフォーマは一部の出力関係を除いて、同じモジュールで構成されている。プロセッサの使用にあたっては図の機能を選択できるようになつてゐる。プログラムの診断はSPOT-6で導入された構文以外はPL/Iコンパイラにまかせていいので、コンパイラ出力のエラーメッセージとSPOT-6ソースプログラムとの対応はライン番号でつけられ様にして、保守はSPOT-6ソースプログラムで行えるようにしてある。実行モニタライブラリは、外部モジュールの呼び出し回数、CPU時間(絶対値と百分率)を測定するライブラリである。モジュール仕様生成はプログラムから仕様部分を抽出編集して内部仕様書を自動生成する機能である。点線のボックスは現在開発中のものでその他は稼動中である。

SPOT-6プロセッサはSPOT-6で書かれてゐる。トランスレータ、リフォーマ、コントローラの各サブシステム間のインタフェースはパラメータとgroupでとらえられる。Fig. 8 はプロセッサとシステムファイルの関連の一部を示したものである。図でSS, RS, RI はそれぞれ S-リフォーマ、I-リフォーマ、トランスレータの入力となる抽象ファイルであり group で実現されてゐる。その他2文字のファイルコードで示されるファイルはSPOT-6プロセッサで用いられるシステムファイルである。.L はユーザ指定のマクロライブラリである。ソースプログラムの入力ファイルとして通常は*.S ファイルを用いるが、FILEEDIT ジョブ内でプロセッサを起動するときはGK を使用する。プロセッサはNEAC ACOS-6 オペレティングシステムの環境に適合して構成されており、ACOS-6 上の標準ライブラリ管理、テキストエディタ(TSS)、リンクエディタ等がそのまま使える。

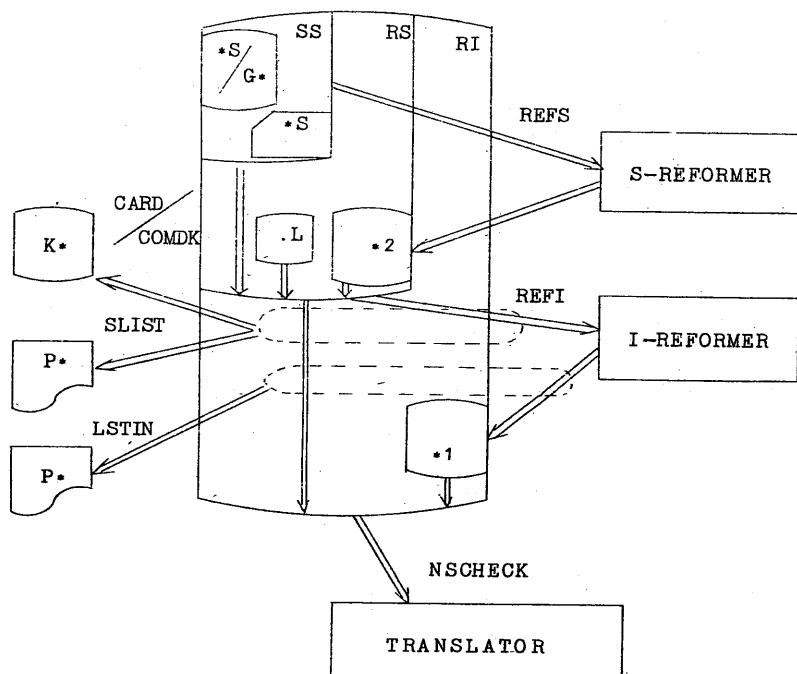


Fig. 8 File structure in SPOT-6 processor

9. まとめ

以上、SPOT-6の主な特徴を述べてきた。現在、SPOT-6はSPOT-6プロセッサ自身の他は2,3のプログラム開発に用いられている。この使用経験を通してSPOT-6言語の有効性が確認された。抽象化のための構造groupは、CLU[4]のclusterやAlphard[5]のformのような抽象データタイプではなく、一つの抽象オブジェクトを表現するものであるが、タイプに比べインプリメンテーションが容易であるという利点をもち、現実のソフトウェア開発では一つの抽象タイプに対して一つの抽象オブジェクトで間に合う場合が多いので有効に利用できる。SPOT-6プロセッサの開発では、トランスレータ、リフォーマ、コントローラ等のサブシステム間のインターフェースはパラメータとgroupによって取扱い、groupの仕様節がインターフェース仕様書の役割を果した。groupの使用によって抽象のレベルが与えられ、いわゆる単体テストの段階では、ダミーのgroupを用いて上位のモジュールをテストでき、groupを使用してリモジュールとgroupの開発を独立して行えた。groupの使用によるユーザ定義のスカラ値のデータタイプの値入により、通常設計言語と称される疑似言語で表現される設計がそのままSPOTプログラムとして書け、記述水準が大きく向上した。記述水準を高めるためににはこのようなデータ記述の高水準化が必須である。groupのもう一つの効果として外部変数の極端な減少があげられる。これによりシステムのインターフェースが单纯化され保守し易いシステムの開発が支援できること。

この論文では主として言語機能を述べたが、SPOT-6は開発を支援するためには、インターフェースチェック・アナライザを始めとしてモジュール仕様生成機能、モジュール展開機能、実行モニタライブラリ・ユーティリティを含んでいる。これらの詳細については別の機会に報告したいが、このようなユーティリティの中にはSPOT-6の言語機能に直る所が大きいつものが多々あることだけを述べておく。

謝辞 SPOT-6プロセッサのインプリメンテーションを手伝っていただきいた当社コンピュータシステム研究部の伊東君、福田さん、長谷川さんに謝意を表す。また、同研究部藤野部長には終始励ましと有益な助言をいただきた。ここに深く感謝する次第である。

参考文献

- [1] N. Wirth: "Program development by stepwise refinement", CACM, Vol.14, No.4(1971).
- [2] D.L. Parnas: "On the criteria to be used in decomposing systems into modules", CACM, Vol.15, No.12 (1972).
- [3] N. Wirth: "The programming language Pascal", Acta Informatica, Vol.1, No.1 (1971).
- [4] B.H. Liskov and S. Zilles: "Programming with abstract data types", SIGPLAN Notices, Vol.9, No.4 (1974).
- [5] W.A. Wulf, R.L. London and M. Shaw: "An introduction to the construction and verification of Alphard programs", IEEE Trans. Software Eng., Vol.SE-2, No.4 (1976).

[付録] SPOT-6 ソースプログラムリストの例(リファーム結果)

```

2703T 01 03-03-52 13.919 *** SPOT-6 PROCESSOR *** 770301 520303HCHC PAGE 2
ALT# LINE# ----- MSCHECK_AND_COPY_SPECIAL_OPTIONS_AND_SETUP -----
 6   1      MSCHECK_AND_COPY_SPECIAL_OPTIONS_AND_SETUP : PROCEDURE ;
 7   2
 8   3
 9   4
10  5      /****** EXTERNAL SPECIFICATION *****/
11  6      FUNCTION
12  7          PRINT STAR CARD IMAGES ON OPTION LISTING FILE.
13  8          SCAN STAR CARDS IN SPOT-SRC-FILE AND CHECK SPECIAL OPTIONS;
14  9          THEN SETUP OPTION TABLE; ALSO COPY STAR CARDS IMAGE INTO REF-SRC-
15 10          FILE IF PLL-SRC-FILE IS CMMB IS SPECIFIED.
16 11          PLL-SRC-FILE IS CMMB IS SPECIFIED.
17 12
18 13      POST CONDITION:
19 14          SPOT-SRC-FILE IS POSITIONED AT THE BEGINNING OF SPOT-6 PROGRAM
20 15          TEXT (NON STAR CARD) IMAGES.
21 16          SPECIFY THE PURPOSE TO COPY STAR CARD IMAGES INTO REF-SRC-FILE IS
22 17          MERGE NEW REFORMED SPOT SOURCE PROGRAM TEXT WITH THEM AND
23 18          PUT THEM INTO K-FILE AT LATER L-REFORMER OR TRANSLATOR READ-
24 19          ING TIME.
25 20
26 21      AUTHOR S. FUJIBAYASHI ;     *** END OF FUNCTION ***
27 22      DATE WRITTEN 76/11/08 ;
28x

```

```

2703T 01 03-03-52 13.919 *** SPOT-6 PROCESSOR *** 770301 520303HCHC PAGE 3
ALT# LINE# ----- MSCHECK_AND_COPY_SPECIAL_OPTIONS_AND_SETUP -----
29 23      /****** INTERNAL SPECIFICATION *****/
30 24
31 25      SUBROUTINE
32 26          USPOT-SRC-FILE-PS
33 27          !READ-CARD (IMAGE CHAR(80) ALIGNED OUT),
34 28          !READ-ALTN0 (ALTN0 FIXED BIN OUT),
35 29          !SEND-BACK-CARD
36 30          !USPOT-SRC-FILE-PS
37 31          !W3OPT-FILE-LIST-IL
38 32          !USPOT-SRC-FILE-PS
39 33          !WRITE-CARD (IMAGE CHAR(80) ALIGNED IN),
40 34          !MSOPT-INPUT-TABLE-OT
41 35          !IS-COMPILER
42 36          !IS-REFS
43 37          !SCM-ZSTAR-CARD (IMAGE CHAR(80) ALIGNED INOUT, ANUM FIXED BIN IN, PLIOPT BIT(1)
44 38          !ALIGNED OUT)
45 39          !CHECK-STAR-CARDS-USAGE (IMAGE CHAR(80) ALIGNED IN, ALTN0 FIXED BIN IN);
46 40
47 41      SURGROUP
48 42          NSINPUT-LINE-IMAGE-LIST-IL
49 43          !PUT-STAR-CARD (IMAGE CHAR(80) ALIGNED IN, ALTN0 FIXED BIN IN);
50 44
51 45      MACRO
52 46          TRUE = '1'N
53 47          FALSE = '0'B
54 48          !STAR-CARD-IS-FOUND = SUBSTR(IMAGE,1,1)==""
55 49          !STAR-CARD-TYPE = SUBSTR(IMAGE,2,4);
56 50
57 51      INVAR
58 52          IMAGE
59 53          ANUM
60 54          !REFORMED-SOURCE-IS-REQUESTED
61 55          !OBJECT-PROGRAM-IS-REQUESTED
62 56          !PLI-OPTION
63 57          !CARD-IS-NOT-NOTE
64 58
65 59          !ENDSPECIFICATION ;
66 60
67 61
68 62
69 63
70 64
71 65
72 66
73 67
74 68
75 69
76 70
77 71
78 72
79 73
80 74
81 75
82 76
83 77
84 78
85 79
86 80
87 81
88 82
89 83
90 84
91 85
92 86
93 87
94 88
95 89
96 90
97 91
98 92
99 93
100 94
101 95
102 96
103 97
104 98
105 99
106 100
107 101
108 102
109 103
110 104
111 105
112 106
113 107
114 108
115 109
116 110
117 111
118 112
119 113
120 114
121 115
122 116
123 117
124 118
125 119
126 120
127 121
128 122
129 123
130 124
131 125
132 126
133 127
134 128
135 129
136 130
137 131
138 132
139 133
140 134
141 135
142 136
143 137
144 138
145 139
146 140
147 141
148 142
149 143
150 144
151 145
152 146
153 147
154 148
155 149
156 150
157 151
158 152
159 153
160 154
161 155
162 156
163 157
164 158
165 159
166 160
167 161
168 162
169 163
170 164
171 165
172 166
173 167
174 168
175 169
176 170
177 171
178 172
179 173
180 174
181 175
182 176
183 177
184 178
185 179
186 180
187 181
188 182
189 183
190 184
191 185
192 186
193 187
194 188
195 189
196 190
197 191
198 192
199 193
200 194
201 195
202 196
203 197
204 198
205 199
206 200
207 201
208 202
209 203
210 204
211 205
212 206
213 207
214 208
215 209
216 210
217 211
218 212
219 213
220 214
221 215
222 216
223 217
224 218
225 219
226 220
227 221
228 222
229 223
230 224
231 225
232 226
233 227
234 228
235 229
236 230
237 231
238 232
239 233
240 234
241 235
242 236
243 237
244 238
245 239
246 240
247 241
248 242
249 243
250 244
251 245
252 246
253 247
254 248
255 249
256 250
257 251
258 252
259 253
260 254
261 255
262 256
263 257
264 258
265 259
266 260
267 261
268 262
269 263
270 264
271 265
272 266
273 267
274 268
275 269
276 270
277 271
278 272
279 273
280 274
281 275
282 276
283 277
284 278
285 279
286 280
287 281
288 282
289 283
290 284
291 285
292 286
293 287
294 288
295 289
296 290
297 291
298 292
299 293
300 294
301 295
302 296
303 297
304 298
305 299
306 300
307 301
308 302
309 303
310 304
311 305
312 306
313 307
314 308
315 309
316 310
317 311
318 312
319 313
320 314
321 315
322 316
323 317
324 318
325 319
326 320
327 321
328 322
329 323
330 324
331 325
332 326
333 327
334 328
335 329
336 330
337 331
338 332
339 333
340 334
341 335
342 336
343 337
344 338
345 339
346 340
347 341
348 342
349 343
350 344
351 345
352 346
353 347
354 348
355 349
356 350
357 351
358 352
359 353
360 354
361 355
362 356
363 357
364 358
365 359
366 360
367 361
368 362
369 363
370 364
371 365
372 366
373 367
374 368
375 369
376 370
377 371
378 372
379 373
380 374
381 375
382 376
383 377
384 378
385 379
386 380
387 381
388 382
389 383
390 384
391 385
392 386
393 387
394 388
395 389
396 390
397 391
398 392
399 393
400 394
401 395
402 396
403 397
404 398
405 399
406 400
407 401
408 402
409 403
410 404
411 405
412 406
413 407
414 408
415 409
416 410
417 411
418 412
419 413
420 414
421 415
422 416
423 417
424 418
425 419
426 420
427 421
428 422
429 423
430 424
431 425
432 426
433 427
434 428
435 429
436 430
437 431
438 432
439 433
440 434
441 435
442 436
443 437
444 438
445 439
446 440
447 441
448 442
449 443
450 444
451 445
452 446
453 447
454 448
455 449
456 450
457 451
458 452
459 453
460 454
461 455
462 456
463 457
464 458
465 459
466 460
467 461
468 462
469 463
470 464
471 465
472 466
473 467
474 468
475 469
476 470
477 471
478 472
479 473
480 474
481 475
482 476
483 477
484 478
485 479
486 480
487 481
488 482
489 483
490 484
491 485
492 486
493 487
494 488
495 489
496 490
497 491
498 492
499 493
500 494
501 495
502 496
503 497
504 498
505 499
506 500
507 501
508 502
509 503
510 504
511 505
512 506
513 507
514 508
515 509
516 510
517 511
518 512
519 513
520 514
521 515
522 516
523 517
524 518
525 519
526 520
527 521
528 522
529 523
530 524
531 525
532 526
533 527
534 528
535 529
536 530
537 531
538 532
539 533
540 534
541 535
542 536
543 537
544 538
545 539
546 540
547 541
548 542
549 543
550 544
551 545
552 546
553 547
554 548
555 549
556 550
557 551
558 552
559 553
560 554
561 555
562 556
563 557
564 558
565 559
566 560
567 561
568 562
569 563
570 564
571 565
572 566
573 567
574 568
575 569
576 570
577 571
578 572
579 573
580 574
581 575
582 576
583 577
584 578
585 579
586 580
587 581
588 582
589 583
590 584
591 585
592 586
593 587
594 588
595 589
596 590
597 591
598 592
599 593
600 594
601 595
602 596
603 597
604 598
605 599
606 600
607 601
608 602
609 603
610 604
611 605
612 606
613 607
614 608
615 609
616 610
617 611
618 612
619 613
620 614
621 615
622 616
623 617
624 618
625 619
626 620
627 621
628 622
629 623
630 624
631 625
632 626
633 627
634 628
635 629
636 630
637 631
638 632
639 633
640 634
641 635
642 636
643 637
644 638
645 639
646 640
647 641
648 642
649 643
650 644
651 645
652 646
653 647
654 648
655 649
656 650
657 651
658 652
659 653
660 654
661 655
662 656
663 657
664 658
665 659
666 660
667 661
668 662
669 663
670 664
671 665
672 666
673 667
674 668
675 669
676 670
677 671
678 672
679 673
680 674
681 675
682 676
683 677
684 678
685 679
686 680
687 681
688 682
689 683
690 684
691 685
692 686
693 687
694 688
695 689
696 690
697 691
698 692
699 693
700 694
701 695
702 696
703 697
704 698
705 699
706 700
707 701
708 702
709 703
710 704
711 705
712 706
713 707
714 708
715 709
716 710
717 711
718 712
719 713
720 714
721 715
722 716
723 717
724 718
725 719
726 720
727 721
728 722
729 723
730 724
731 725
732 726
733 727
734 728
735 729
736 730
737 731
738 732
739 733
740 734
741 735
742 736
743 737
744 738
745 739
746 740
747 741
748 742
749 743
750 744
751 745
752 746
753 747
754 748
755 749
756 750
757 751
758 752
759 753
760 754
761 755
762 756
763 757
764 758
765 759
766 760
767 761
768 762
769 763
770 764
771 765
772 766
773 767
774 768
775 769
776 770
777 771
778 772
779 773
780 774
781 775
782 776
783 777
784 778
785 779
786 780
787 781
788 782
789 783
790 784
791 785
792 786
793 787
794 788
795 789
796 790
797 791
798 792
799 793
800 794
801 795
802 796
803 797
804 798
805 799
806 800
807 801
808 802
809 803
810 804
811 805
812 806
813 807
814 808
815 809
816 810
817 811
818 812
819 813
820 814
821 815
822 816
823 817
824 818
825 819
826 820
827 821
828 822
829 823
830 824
831 825
832 826
833 827
834 828
835 829
836 830
837 831
838 832
839 833
840 834
841 835
842 836
843 837
844 838
845 839
846 840
847 841
848 842
849 843
850 844
851 845
852 846
853 847
854 848
855 849
856 850
857 851
858 852
859 853
860 854
861 855
862 856
863 857
864 858
865 859
866 860
867 861
868 862
869 863
870 864
871 865
872 866
873 867
874 868
875 869
876 870
877 871
878 872
879 873
880 874
881 875
882 876
883 877
884 878
885 879
886 880
887 881
888 882
889 883
890 884
891 885
892 886
893 887
894 888
895 889
896 890
897 891
898 892
899 893
900 894
901 895
902 896
903 897
904 898
905 899
906 900
907 901
908 902
909 903
910 904
911 905
912 906
913 907
914 908
915 909
916 910
917 911
918 912
919 913
920 914
921 915
922 916
923 917
924 918
925 919
926 920
927 921
928 922
929 923
930 924
931 925
932 926
933 927
934 928
935 929
936 930
937 931
938 932
939 933
940 934
941 935
942 936
943 937
944 938
945 939
946 940
947 941
948 942
949 943
950 944
951 945
952 946
953 947
954 948
955 949
956 950
957 951
958 952
959 953
960 954
961 955
962 956
963 957
964 958
965 959
966 960
967 961
968 962
969 963
970 964
971 965
972 966
973 967
974 968
975 969
976 970
977 971
978 972
979 973
980 974
981 975
982 976
983 977
984 978
985 979
986 980
987 981
988 982
989 983
990 984
991 985
992 986
993 987
994 988
995 989
996 990
997 991
998 992
999 993
1000 994
1001 995
1002 996
1003 997
1004 998
1005 999
1006 1000
1007 1001
1008 1002
1009 1003
1010 1004
1011 1005
1012 1006
1013 1007
1014 1008
1015 1009
1016 1010
1017 1011
1018 1012
1019 1013
1020 1014
1021 1015
1022 1016
1023 1017
1024 1018
1025 1019
1026 1020
1027 1021
1028 1022
1029 1023
1030 1024
1031 1025
1032 1026
1033 1027
1034 1028
1035 1029
1036 1030
1037 1031
1038 1032
1039 1033
1040 1034
1041 1035
1042 1036
1043 1037
1044 1038
1045 1039
1046 1040
1047 1041
1048 1042
1049 1043
1050 1044
1051 1045
1052 1046
1053 1047
1054 1048
1055 1049
1056 1050
1057 1051
1058 1052
1059 1053
1060 1054
1061 1055
1062 1056
1063 1057
1064 1058
1065 1059
1066 1060
1067 1061
1068 1062
1069 1063
1070 1064
1071 1065
1072 1066
1073 1067
1074 1068
1075 1069
1076 1070
1077 1071
1078 1072
1079 1073
1080 1074
1081 1075
1082 1076
1083 1077
1084 1078
1085 1079
1086 1080
1087 1081
1088 1082
1089 1083
1090 1084
1091 1085
1092 1086
1093 1087
1094 1088
1095 1089
1096 1090
1097 1091
1098 1092
1099 1093
1100 1094
1101 1095
1102 1096
1103 1097
1104 1098
1105 1099
1106 1100
1107 1101
1108 1102
1109 1103
1110 1104
1111 1105
1112 1106
1113 1107
1114 1108
1115 1109
1116 1110
1117 1111
1118 1112
1119 1113
1120 1114
1121 1115
1122 1116
1123 1117
1124 1118
1125 1119
1126 1120
1127 1121
1128 1122
1129 1123
1130 1124
1131 1125
1132 1126
1133 1127
1134 1128
1135 1129
1136 1130
1137 1131
1138 1132
1139 1133
1140 1134
1141 1135
1142 1136
1143 1137
1144 1138
1145 1139
1146 1140
1147 1141
1148 1142
1149 1143
1150 1144
1151 1145
1152 1146
1153 1147
1154 1148
1155 1149
1156 1150
1157 1151
1158 1152
1159 1153
1160 1154
1161 1155
1162 1156
1163 1157
1164 1158
1165 1159
1166 1160
1167 1161
1168 1162
1169 1163
1170 1164
1171 1165
1172 1166
1173 1167
1174 1168
1175 1169
1176 1170
1177 1171
1178 1172
1179 1173
1180 1174
1181 1175
1182 1176
1183 1177
1184 1178
1185 1179
1186 1180
1187 1181
1188 1182
1189 1183
1190 1184
1191 1185
1192 1186
1193 1187
1194 1188
1195 1189
1196 1190
1197 1191
1198 1192
1199 1193
1200 1194
1201 1195
1202 1196
1203 1197
1204 1198
1205 1199
1206 1200
1207 1201
1208 1202
1209 1203
1210 1204
1211 1205
1212 1206
1213 1207
1214 1208
1215 1209
1216 1210
1217 1211
1218 1212
1219 1213
1220 1214
1221 1215
1222 1216
1223 1217
1224 1218
1225 1219
1226 1220
1227 1221
1228 1222
1229 1223
1230 1224
1231 1225
1232 1226
1233 1227
1234 1228
1235 1229
1236 1230
1237 1231
1238 1232
1239 1233
1240 1234
1241 1235
1242 1236
1243 1237
1244 1238
1245 1239
1246 1240
1247 1241
1248 1242
1249 1243
1250 1244
1251 1245
1252 1246
1253 1247
1254 1248
1255 1249
1256 1250
1257 1251
1258 1252
1259 1253
1260 1254
1261 1255
1262 1256
1263 1257
1264 1258
1265 1259
1266 1260
1267 1261
1268 1262
1269 1263
1270 1264
1271 1265
1272 1266
1273 1267
1274 1268
1275 1269
1276 1270
1277 1271
1278 1272
1279 1273
1280 1274
1281 1275
1282 1276
1283 1277
1284 1278
1285 1279
1286 1280
1287 1281
1288 1282
1289 1283
1290 1284
1291 1285
1292 1286
1293 1287
1294 1288
1295 1289
1296 1290
1297 1291
1298 1292
1299 1293
1300 1294
1301 1295
1302 1296
1303 1297
1304 1298
1305 1299
1306 1300
1307 1301
1308 1302
1309 1303
1310 1304
1311 1305
1312 1306
1313 1307
1314 1308
1315 1309
1316 1310
1317 1311
1318 1312
1319 1313
1320 1314
1321 1315
1322 1316
1323 1317
1324 1318
1325 1319
1326 1320
1327 1321
1328 1322
1329 1323
1330 1324
1331 1325
1332 1326
1333 1327
1334 1328
1335 1329
1336 1330
1337 1331
1338 1332
1339 1333
1340 1334
1341 1335
1342 1336
1343 1337
1344 1338
1345 1339
1346 1340
1347 1341
1348 1342
1349 1343
1350 1344
1351 1345
1352 1346
1353 1347
1354 1348
1355 1349
1356 1350
1357 1351
1358 1352
1359 1353
1360 1354
1361 1355
1362 1356
1363 1357
1364 1358
1365 1359
1366 1360
1367 1361
1368 1362
1369 1363
1370 1364
1371 1365
1372 1366
1373 1367
1374 1368
1375 1369
1376 1370
1377 1371
1378 1372
1379 1373
1380 1374
1381 1375
1382
```