FONTAC 8040 MONITOR

辻ヶ堂　信*　伊沢喜三男*　竜田和彦*

1. 概　　論

1.1. 設計の基本原理

FONTAC 8040 (CENTRAL) は Core 記憶装置 64 K 語 (1語 42 bits), 同時動作する Data channel 7 台 (各 Data channel には8台までの入出力機械, 大容量記憶装置または衛星計算機が付着可能) を最大構成とする計算機である. FONTAC 8040 MONITOR はこの計算機を動作させるすべての Program, 入出力機械および Console typewriter を制御する管理 Program であり, 現在主記憶装置 12 K 語を占有している.

この MONITOR を設計する際の基本方針は次のようなものである.

1) FONTAC CENTRAL COMPUTER の構成より大容量記憶装置としては Sequential 型を基本とし, Random Access Device が付着した場合も, これを Sequential なデータ編成で扱うこと.

2) 最大 28 個までの多重 Program を制御すること.

3) すべての入出力機械および大容量記憶装置の動作を制御すること.

4) Console typewriter, Console reader および IO device の制御に関して, 直接には Monitor が起動するが完了を待つため Monitor 中で停止することなく, 直ちに制御を Program にもどし, 完了割込によって再び Monitor に link する.

5) Programmer は自己の Program 中の File の媒体を識別するために, Logical unit name を使用し, 自己の Program と同時動作する他の Program の存在を全く考えないでその Coding を行なえること. このため Logical unit に Physical unit (具体的には Card reader, Line printer, Magtape などで, 特に Random access device の場合にはその一部) を対応させることを Monitor が行なうこと.

Makoto Tsujigado, Kimio Izawa and Kazuhiko Tastuta (Program Section Computer Technology Department Fujitsu Kimited)
* 富士通信機製造株式会社電算機技術部プログラム課

6) 計算機の動作効率を上げることを最大の目標とし, このため Operator に多少の労力をかけるのもやむをえない.

1.2. MONITOR の制御下にある PROGRAM について

すべての Program は FONTAC 8040 MONITOR の制御下で動作する. これを図示すると次のようになる.

```
MONITOR
  ├—COBOL        : COBOL Compiler
  ├—ALGOL        : ALGOL Compiler
  ├—FORTRAN IV   : FORTRAN IV Compiler
  ├—FOCUS        : Utility Program Generator
  ├—FONAS        : FONTAC Assembler
  ├—LIBE         : Library Editor
  ├—EXACT        : Relocatable Binary Loader
  ├—SORT         : Sort/Merge Program
  ├—Utility      : Peripheral to/from Large Capacity Storage
  └—Object Program
```

第1図

Compiler language, Assembler language および Generator language は, それぞれ対応する Compiler, Assembler または Generator により翻訳されて Relocatable binary language となる. 一度に Compile または Assemble される Program の単位は Element と呼ばれ, Relocatable binary となった複数個の Element を集めまた必要な Library subroutines および IOCS を加えて一つの完全に実行できる Program を作り出し, 磁気 tape 上に書き出し, さらに複数個の Program よりなる Program tape を作り出すものを EXACT (Executable coded tape producer) と呼ぶ. Program tape 上にある Program は Monitor 中にある System loader により Core 中に Load され, 実行可能の状態となる.

FONTAC 8040 MONITOR により管理される場合, 各 Program は Monitor の次の位置にある

Core の Lower location より Higher location に向って逐次 Load されていく．またコア中で動作していた Program の一つが終了すると直ちに Program の In-execution Relocation が行なわれる．そして次の Program が必要なら Load される．

1.3. Program Tape の形

EXACT run により任意の Program の順序をもつ Program Tape が作成され得るが，最も普通なものは

1) BOOT (Bootstrap loader)
2) MONITOR
3) 以下必要な任意個の System Program または User's Program

のようなものである．そして BOOT および MONITOR 以外の Program はすべて次のような型式をもつ[1]．

1) Program identification block
   (1) この Program の名前
   (2) この Program の所要語数
   (3) 次に実行すべき Program の名前
   (4) Common segment の最初の Program block の Record definition word (word count および相対番地)
2) Unit assignment block

この Program で使用されている Logical unit すべてについての Physical unit との対応に関する情報が収められている．

3) Common segment

一つの Program の実行中，主記憶装置中に常駐し，全 Segment により共通に使用される部分

4) Segment name block とそれに対応する Segment

Segment とは一度に主記憶装置中に Load されて実行される単位のことであり，一つの Program は複数個の Segment よりなる．

1.4. Execution phase における Object program の構造

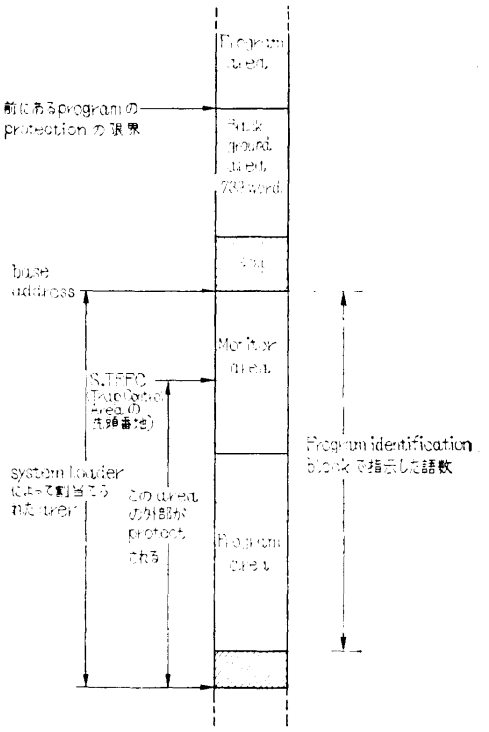
各 Program が Monitor と種々の情報を交換し得るために，Monitor の制御下に動作する Program は一定の構造を有しており，大別して，Background area, Monitor area および Program area よりなる．

Background area は各 Program ごとに確保される Monitor の Working area であって，System input および System output のための Buffer, System unit に対する Label 処理のための領域，入出力の割りあてに関する状態を記憶する表など 739 語よりなり，この領域の設置および管理は Monitor により行なわれる．

Monitor area は前記 Program identification block に対応する情報を含むほか，Monitor からの情報伝達領域，割込時の状態記憶領域，Unit Assignment Table[2]，割込制御領域および Segment 制御領域などからなり，その大きさは使用する入出力の数および Segment の数によって変化する．Monitor area は EXACT run により作成され Program Tape 中では Common segment として存在する．

Program area は Programmer により作成された Source Program を Compile または Assemble し，その結果を Library element と接続し，Executable code 化することによりその内部構造は決定される．

主記憶装置中では番地の若い方から Background area, Monitor area および Program area の順で



第2図

割りあてられる．Program の Load は Monitor 中の Sub-routine の一つである System loader により次のように行なわれる．まずすでに主記憶装置中にある Program の最後尾の位置を求める．System loader はこの位置に接して 739 語をとり Background area とする．次にこれに接して Monitor area を割りつけてみる．このとき，もし割込制御領域（Trap control area）の先頭番地の下位7ビットがゼロでないなら，これがゼロになるように Monitor area だけをずらす．次にこれに接して Program area を割りつける．Program identification block には所要語数が与えられているので[1]，これを含むように記憶保護領域の一方の限界を定める．他方の限界は Trap control area の先頭番地である．Monitor area の先頭番地がこの Program の相対的なゼロ番地とされる（**第2図**参照）．

この Monitor では Over flow, Under flow および Flag による割り出しの結果の処理は Programmer の責任で行なうこととし，Trap control area 中に割り出しの原因を表示すると共に，もしそこに処理 Routine の指示があれば，そこに制御を移す．

2. Multi-Program の制御

2.1. Flow と Program

一般に Program は自己の name と共に次に遂行される Program の name をもてるので，芋蔓式に一連の Program を遂行することができる．この一連の Program を Scheduled process mode における Flow という．

System input 上におかれた Monitor control card によって一連の Program を指定し遂行させることもできる．これを Batch process mode における Flow という．

Console typewriter に Program name を与え，Next program 指定を無視して1個の Program だけを遂行して仕事を終えることもできる．これを Instant process という．

Flow の開始は Console typewriter からの指令による．

Scheduled, Batch, Instant process に対してそれぞれ：

(RQ) CALL program-name physical-unit-number (CR)

(RQ) BATCH physical-unit-number (CR)

(RQ) INSTANT program-name physical-unit-number (CR)

と Type-in される．ただし（RQ）および（CR）はそれぞれリクエストボタンを押すこととキャリジーリターンキーを押すことを意味する．

Flow の Termination はそれぞれ，Next program 指定のなかったこと，¥ FINIS card を検出したこと，および Program が終了したことによって行なわれる．

2.2. Multi-flow と主記憶装置の割りあて

FONTAC 8040 MONITOR システムにおいては，複数個の Program が連って一つの Flow を形成するが，同一の Flow に属する複数個の Program が同時に実行されることはない．そこで Operator が可能な限り多くの Flow を開くことにより，別々の Flow に属する Program の同時動作が行なわれるので，本 Monitor でいう Multi-program とは Multi-flow ということになる．

Multi-program が行なわれる場合，どの Program も，主記憶装置中で動作する場所を Coding 時に知ることはできないので，Program 中絶対番地に依存する量は扱われ得ない．個々の Program の内部では，番地とはその Program の先頭番地を原点とする相対的番地である．このように書かれた Program は Dynamic に Relocatable な Program と呼ばれる．

FONTAC 8040 の主記憶装置中，0番地に始まる連続する領域には Monitor が駐在し，各 Flow に属する Program はそれよりも番地の大きい部分に Load される．各プログラムは一つの連続する領域を占める．その後に開かれた Flow に属する Program はすでに他の Flow により占有されている領域に接して，さらに大きい番地をもつ領域を占める．

主記憶装置中にある Program 相互間の番地の大小関係は各 Flow の進行中は変化しないが，終了時にはその Flow に属する次の Program が Load されて，この時その Program 領域の大きさが増減する．また Flow の終了時にはその Flow により占められていた領域が不要となるため，この領域よりも大きい番地にある Program の Dynamic relocation が行なわれる．この時，動かされる Program に属する Typewriter や I/O unit の動作の完了を待って Relocation が行なわれるのであるが，この間，他の Program の I/O unit の動作は続行させる．

2.3. Multi-program と Program Switch

Multi-program ということは静的に眺めると，記憶装置および多数の入出力装置を複数個の群に分割して，おのおのに一つづつの Program を割りあてるということであるが，動的に眺めると，主記憶装置中に存在する複数個の Program が中央処理装置を時間的に分割して使用することにより，中央処理装置と入出力装置との間の情報処理の速さについて生じる不均衡をなくすることにある．

本 Monitor の制御下で Multi-program が行なわれる場合，一つの Program より別の Program への制御の遷移は次の条件により行なわれる．

1) CPU-release

CPU を占有している Program が入出力使用状況のため，それ以上進むことができなくなった場合には，Monitor 中の CPU-release Routine に制御を移すことにより，その Program は CPU 使用を放棄し，MONITOR は主記憶装置中にある他の Program に制御を移す．なお IOCS をはじめ富士通の供給する Program は常に進行可能性を検査し，進行不能の時は直ちに CPU-release を行なうという論理となっている．

また一般の User's program が Monitor 中の入出力と関係する Routine を使用するときは，CPU-release を行なったのと同じように Program Switch が行なわれる．

2) IO Operation の完了

Datachannel Completion または Attention Trap により Monitor に制御が移るが，この時 Monitor は制御を Dispatcher（入出力命令処理 Routine）にいったん移し，Error の Recovery または他の入出力を動作させることにより，Datachannel を Busy として後，制御を再び受けとる．この時 Monitor は，もし IO Operation が完了していれば，この入出力機械を使用している Program に制御を移し，またもし未完であれば，この Trap のため中断された Program に制御を帰えす．

3) Datachannel Attention の場合

もし On-line real-time device より，Data 出現を知らせる Attention trap がかかると，その Data を処理する Routine に制御を移す．この時，この Online real-time data を処理する Program が主記憶装置中にあることが必要条件である．Real-time device 以外の Device よりの Attention trap は上記 2) の場合となる．

4) 時計による Program Switch

Program によっては，相当長時間入出力機械を使用しない状態になるものがある．この時，入出力機械の使用効率がさがるのをさけるため，640 m sec たってなお入出力命令が発せられなければ，Program switch が行なわれる．

3. Peripheral Unit の割あて

3.1. Logical Unit について

入出力装置を使用する場合，IOCS を使うにせよ最終的には BLCP (Block Level Control Program, 入出力制御 Routine) を使用して入出力制御を行なう．この場合入出力装置は Logical unit で指示される．

Logical unit name は各 Program ごとに，使われている File の媒体を識別するために使用されるものである．これに実際に存在する入出力装置を対応させるのは，EXACT run において Unit assignment card により行なうか，あるいは Flow を開いたり，Flow 中の Program が交代する際に作用する Monitor control cards の指示または Operator の Console typewriter よりの指示により行なわれる．

なお一般に一つの Physical unit には一つの Logical unit が対応させられるのであるが，Random Access device の場合には複数個の Logical unit が対応させられる．これはたとえば磁気 Drum などの場合，適当に分割して，そのおのおのに対して一つの File を割りあてられ得るようにするためである．

3.2. Unit Assignment Card について

Logical unit に Physical device を対応させるためには，¥ DEVICE card に続いて Unit assignment card をおく．Unit Assignment card には次の 6 種がある．なお以下〔 〕は Option, alias は 6 桁以下の文字または数字の列で Logical unit の別名となるもの，l は Logical unit, p は Physical unit を表わすものとする．

（1）［alas］ASSIGN $l=p$

この制御 Card により Logical unit l が Physical unit p に対応させられる．

（2）［alias］ASSIGN $l_c=l_p$

ここで l_c は Current logical unit, l_p は Previous logical unit の意であり，前の Program における l_p に対応する Physical unit が現在の Program の l_c に割りあてられる．なお l_p と同じ Lo-

gical unit name が現在の Program 中にも現われているなら，この ¥ DEVICE に続く Unit assignment card 群中に，この Logical unit name に Physical unit を対応させる Card が存在すること．またもし前の Program と現在の Program 中に同じ Logical unit name があり，かつこの Logical ASSIGN の右辺にこの名前が現われぬときには，前の Program で対応させられた Physical unit がそのまま割りあてられる．

（3） [alias] ATTACH l=type

この指示により type で指定された機種の中，使われていないものが Monitor により選ばれて l に割りあてられる．ただし type には α および $\alpha(\beta)$ なる書き方があり，α は機種，β は Logical datachannel (0, 1, 2, ⋯, 7) を表わす．α としては次の文字を書くこと．

CR Card Reader
CP Card Punch
LP Line Printer
PTR Paper Tape Reader
PTP Paper Tape Punch
MT Magnetic Tape

（4） DETACH $l_i, l_j, l_k, \cdots$

Logical unit $l_i, l_j, l_k \cdots$ がこの Program の Flow から切り離され，他の Program の Flow で使うことができるようになる．

（5） [alias] UNITE $l_i=l_j$

Logical unit l_i と l_j は同じ Physical unit を使用するということを指示する．たとえば Logical unit Uo から入力情報を読み込んでいる Program が SYIN (Standard input または System input) より情報を読みたい時に使用される．

（6） [alias] SWITCH $l_i=l_j$

Logical unit l_i に対応する Physical unit と l_j に対応する Physical unit を交換し，それぞれ l_j と l_i に割りあてることを指示する．

3.3. Typewriter による割りあてについて

Program の実行直前までに Logical Unit と Physical unit の対応がついていない場合には，Monitor は Typewriter 経由

$*l=*$

のように訊ねる故，Operator は

$l=p$(CR)

のように Type-In により答えること．ただし (CR) は Carriage return bar を押すことを意味する．

4. Flow の制御について

4.1. 概 論

本 MONITOR では Flow を制御する3種の手段がある．第一のものは処理されるデータの間に制御カードをはさみ，実行時に Monitor に指示を与えつつ制御する方法であり，Batch process と呼称される．

第二は Flow を構成する Program の順序を EXACT run またはそれ以前に計画しておき，EXACT run により Program tape 化し，Program の流れを制御する情報を同時に Program tape 中に入れて，Execute phase にもち込むような処理方法である．実行時に Operator は Typewriter より出る指示に従って動作すればよく，この処理方法は事務計算などで，計算順序もしくは Data の到着順序がきまっている場合に使用される．この処理方法は Scheduled process と呼称される．

第三のものは切迫した (Instant) 事態が発生したとき，この目的のために計算機を動作させること，およびある時点における計算機の一部の遊びをなくすため，Operator の判断により Program を読込んで Data 処理を行なう処理方法のことであり，Instant process と呼称される．

以上三つの処理方法において Program tape は，いずれの場合にも EXACT の Output として作り出されたものが使用される．そして Batch process の場合には SYIN より入る制御情報が最優先で制御を行なう．

4.2. Batch Process について

Batch process をはじめるためには，Typewriter より

(RQ) BATCH pu (CR)

と Type-in を行なう．ここで pu は制御情報ののっている Standard input SYIN に対応する Physical unit である．この SYIN 一つに対応して一つづつ Batch process により Flow を制御することができる．

Batch process を制御する Control cards には次のものがある．

（1） ¥ DEVICE

このカード以下次の ¥ (Monitor 制御) Card までは Unit assignment cards であるということを指示する．

(2) ¥ EXECUTE program-name

Program-name をもつ Program を Program tape SSYS より主記憶装置中に読み込み，動作可能とする．ただし Program-name の中，次の Name は Processor 用に保存されている．PL/I, ALGOL, COBOL, FORTRAN, FONAS, SORTG, LIBE, EXACT, FOCUS. なお Program tape 中に対応 Processor がない場合，User が自己の Program に上記の名前を用いることは自由である．

(3) ¥ FINIS

これにより Program の流れは停止し，この Flow で使用していた Core 領域も，Peripheral unit も Secondary storage もすべて解放される．

(4) ¥ FINIS BATCH pu

これまでの Flow を中止し，次は pu に対応するところを SYIN として Batch process を続けることを Monitor に指示する．

(5) ¥ FINIS program-name pu

これまでの Program の流れを中止し，次は Program tape pu より Program-name をもつ Program を主記憶装置中に読み込み，Scheduled process として処理を続けることを Monitor に指示する．

なお各 Program がその終点に到着すると

*EOP B program-name

が Type-out される．ここで EOP は End Of Program の略であり，B は Batch process の意である．続いて Monitor は SYIN より制御情報を読み，その指示に従って Flow の制御を行なう．

4.3. Scheduled Process について

Scheduled process をはじめるためには，Typewriter より

(RQ) CALL program-name pu (CR)

と Type-in を行なう．ここで pu は Program tape SSYS に対応する Physical unit であり，ここより program-name をもつ Program が主記憶装置中に Load される．もしこの Program において使用されている Logical unit に Physical unit がすでに定められていれば，Console typewriter 上に

*MOUNT [alias] *l* ON *p*

が印刷される．ただし *l* は Logical unit name であり，*p* は Physical unit である．またこの時もし U 00～U 63 について，その Physical unit を他の Program が使用していれば

*[alias] *l* ON *p* CONFLICT

と type-out される．この場合には 3.3 の場合と同じく，Operator が Type-in により対応する Physical Unit を知らせねばならぬ．

もし Logical unit に対して Physical unit が定められていない時には 3.3 と同様のことが行なわれる．

Logical unit と Physical unit の対応がすべてについて定められると

*EXEC OK

が Type-out される故，Operator は Line printer の紙や Card に正しい Deck が積まれていることを確認した上で

EX(CR)

と Type-in する．これにより Program の実行がはじまる．

Program がその終点に到着すると，まず

*EOP S program-name

が Type-out される．ただし S は Scheduled process の意である．また EXACT 制御カードの一つに NEXT program-name がある．もしこの NEXT カードが与えられていないと

*WHAT NEXT

という指示が Type-out されて，Flow は終了する．

また NEXT カードは与えられても，それに program-name が与えられていない時には，Program tape 上その次にある Program が Load される．NEXT program-name が与えられている場合，program-name をもつ Program が Load され，その遂行が行なわれる．

4.4. Instant Process について

Instant process をはじめるためには，Typewriter より

(RQ) INSTANT Program-name pu (CR)

と Type-in を行なう．ここで pu は Program tape SSYS に対応する Physical unit であり，ここより program-name をもつ Program が Load される．以下 Logical unit と Physical unit の対応および Execution の開始については Scheduled Process と全く同じである．そして Program が終ると

*EOP I program-name

が Type-out される．ただし I は Instant process の意である．これに続いて

*WHAT NEXT

が Type-out されて，Flow は終了する．

あとがき

本モニタは 1964 年12月に仕様を決定し，1965 年12 月に設計および製作を完了したものである．本文中にある BLCP と Disptcher および FONTAC 8040 EXACTは富士通久保宏志君の設計である．また本 MONITOR に属する COBOL COMPILER および ALGOL COMPILER は富士通丸山武君が，FONAS ASSEMBLER は久保宏君が製作した．それぞれの立場から貴重な意見を寄せられた上記の両君，FONTAC SOFTWARE ADVISORY GROUP としていろいろ御批判を寄せられた高橋秀俊・森口繁一・米田信夫・和田英一・島内剛一・渋谷政昭・清水留三郎の諸先生および MONITOR の構想作成のため御協力下さった東大物性研井上謙蔵先生に御礼申し上げます．

なおわれわれはこの MONITOR をそのまま FACOM 230-50 用に改造し，FACOM 230-50 MONITOR II とすると共に，この考え方に，大容量 Randum access 形記憶装置中に JOB（本文中での Program flow）の STACKING を行なうことを加えることにより，Operator の介在を最小にしてかつ計算機の使用効率を上げるような MONITOR III を作成中である．

最後に，本 MONITOR 作成の指針となったものは UNIVAC III 用 SUPERVISOR BOSS III であり，Operator message の一部を転用させて頂いたことを明記して，同 Supervisor を作成された方々に謝意を呈します．

参考文献

1) FONTAC 8040 EXACT 仕様書（富士通発行）
2) FONTAC 8040 IOCS 仕様書（富士通発行）

（昭和41年1月4日受付）