

この資料は日本エム・テクノロジー学会員専用です。
この資料を学会員以外がコピーしたり、学会員以外に配布することを禁じます。

Copy right : M Technology Association - Japan

日本エム・テクノロジー学会事務局
〒259-1193 神奈川県伊勢原市望星台
東海大学医学部・基礎医学系
大櫛陽一

Tel: 0463-93-1121 ext. 2140

Fax: 0463-96-4301

Email: youichi@keyaki.cc.u-tokai.ac.jp

```

S A=-1,B=-1,C=-1,U=...
120 S A=$N(^MS3(A)) I A<0 D ^MS0(0,1,3)=3~
130 S B=$N(^MS3(A,B)) I B<0 ^MS0(0,1,4)=11000010111
140 S C=$N(^MS3(A,B,C)) G ^MS0(0,2,1)=0
I C#1=0 G 140 ^MS0(0,2,2)=0
D 200 G 140 ^MS0(0,2,3)=512~
Q ^MS0(0,2,4)=10000000000
200 ^PICK UP DATUM AND S ^MS0(0,3,1)=0
S D=^MS3(A,B,C),INF ^MS0(0,3,2)=0
S INF(4)=$P(D,$E(U), ^MS0(0,3,3)=0
I INF(6) L "*" S PNU ^MS0(0,3,4)=2.10~
S INS=^MS1(INU\10, ^MS0(0,4,1)=0
235 S INF(1)=$P(INS,$ ^MS0(0,4,2)=0
D 300 S J=J+1 I 1 ^MS0(0,4,3)=0
Q ^MS0(0,4,4)=0
Q ^MS0(1,1,5)=1
300 ^WRITE EACH DAT ^MS0(1,1,5)=1
F K=1:1:10 S ^MS0(1,1,5)=1
F K=1:1:10 S L ^MS0(1,1,5)=1
S KK=K-1 ^LINE ^MS0(1,1,5)=1
I INF(3)=" S ^MS0(1,1,5)=1
F K=1:1:10 S ^MS0(1,1,5)=1
S KKK=K-1 ^L ^MS0(1,1,5)=1
335 I INF(4)=" ^MS0(1,1,5)=1
F K=1:1:10 ^MS0(1,1,5)=1
S KI=K-1 ^I ^MS0(1,1,5)=1
S ^MS2(1,J) ^MS0(1,1,5)=1
350 F K=1:1:K ^MS0(1,1,5)=1
F K=1:1:K ^MS0(1,1,5)=1
F K=1:1:K ^MS0(1,1,5)=1
S E=INF( ^MS0(1,1,5)=1

```

```

;AH 6-4-78
;CALLED BY I18
S CC=CC*10
I $D(^MS3(AA,BB,CC))=0 S ^MS3(AA,BB,CC)=1,^(CC+0.01)=N
S CO=^(CC) G 200
Q
F I=0.01:0.01:CO/100*100\1/100 D 300 I +FIN=1 G 400
S ^((CO+1/100*100\1/100+CC)=NAZ,CA=CO+1,^(CC)=CA Q
Q
S NAX=^(CC+I),NAY=$P(NAX,"^",1) I NAYJNAM S FIN=1 Q
I NAY=NAM S FIN=1 Q
S FIN=0 Q
Q
F J=CO/100*100\1/100:-0.01:I S ^((J+0.01+CC)=^(J+CC)
      NAZ,CA=CO+1,^(CC)=CA Q

```


Digital Standard MUMPS-11

DSM(Digital Standard MUMPS)-11はANSI標準仕様(X11.1-1979)を拡張したMUMPS言語を利用するマルチユーザデータベースマネジメントシステムでPDP-1134A, PDP11/70のCPUを中心に、最大1,400MBのデータベースをサポートし、63台のユーザーミナルを同時使用できるタイムシェアリングシステムです。

DSM-11は、モジュラー方式のシステム構成が出来るので予算に応じて段階的にシステムを拡張し、各システム間をコミュニケーションネットワークで結合することにより、総合医療情報システムを構成することができます。

京大病院情報処理システム(MUMPS-11)の実例から



システム導入前保険窓口における月はじめの保険チェックの長い行列は10年間以上改善不能であった。病院当局の頭痛の種であった。新患・再来とも行列が強制されていた。



システム導入後オンライン・リアルタイムに他の掛の窓口でも保険情報がチェックできるようになった。行列は新患だけとなり、激減した。システム導入の効果は明白になった。



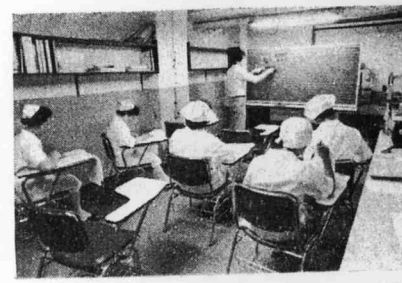
各部門を支えるものは各部門を担当する専門家達であり、特にめざめた若い活動的な事務官の参加はトータルシステムの中心的役割りを果たす。



開発はメーカ、ベンダー任せでは駄目で、アプリケーションの殆んどは夫々の領域の専門家の方がずっと上手である。絶えずミーティングをやり、意見を交換し、開発にぶつかってゆく医事電算機掛職員とDECマン。



一患者一番号を用いた病歴の中央管理により迅速な検索と管理の徹底が可能となり、missing chartは激減し、検索は早くなり窓口での応接も良くなった。



看護部幹部RNに対する300時間教育(DSM-11)での勉強ぶりで、ユーザオリエンテッドなMUMPSは各仕事場での『生き甲斐システム』になりうることを理解が教育のひとつの目標である。

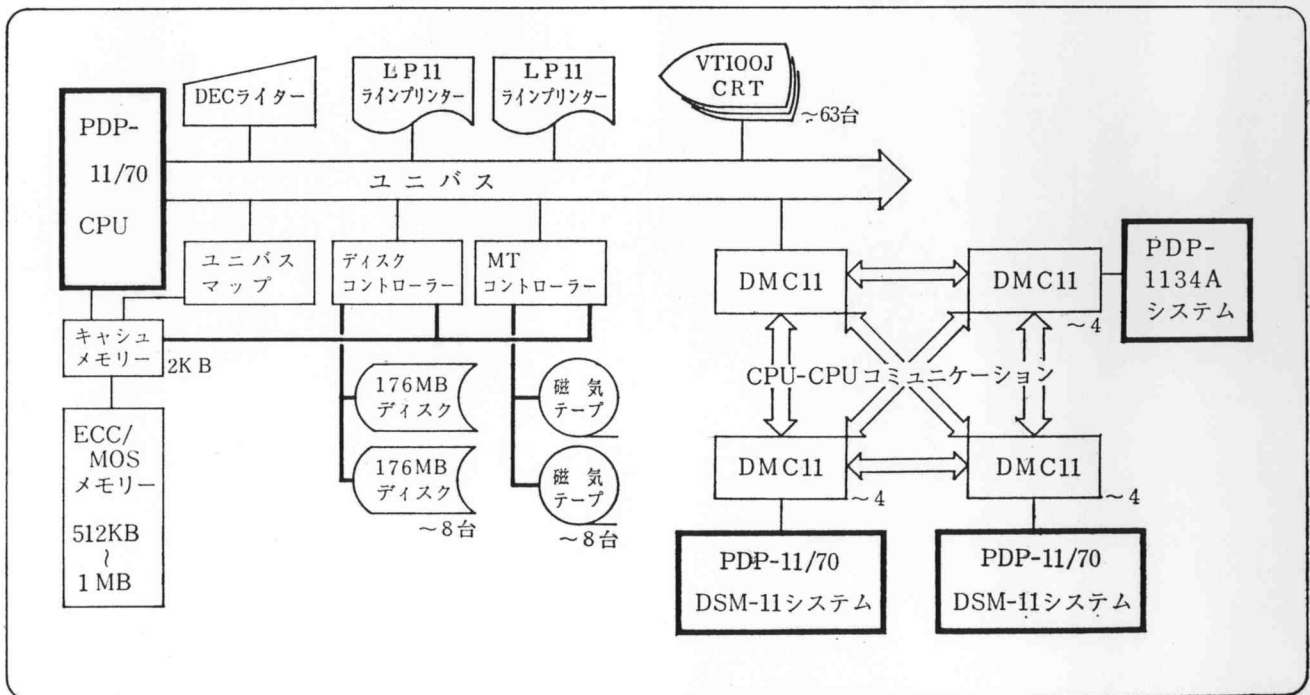
DSM-11の特長と機能

- Mトリーデータファイル構造
- インプットジャーナリング
- アウトプットスプーリング
- インコアジョブコミュニケーション
- CPU-CPUコミュニケーション
- シークエンシャルディスクプロセッサ
- 容易なユーザデバイスとのインターフェイス
- プライバシー保護の為のパスワード
- 255文字の英、数、カナ文字処理
- 豊富なシステムユーティリティプログラム
とライブラリーユーティリティプログラム
- 強力なバックアップユーティリティプログラム
- 標準マンプス言語を拡張した強力なZコマンド

DSM-11アプリケーション

- 外来システム
- 入院システム
- 薬剤システム
- 検査システム
- 病歴システム
- 医学教育システム
- 地域医療システム
- 図書館システム

DEC MUMPS ネットワークシステム



digital DEC日本支社

メディカルシステムズグループ

東京都渋谷区道玄坂2-10-7(新大宗ビル5号館6階)

TEL (03)476-3181 (代) 〒150

大阪市北区西天満 6-7-2 (梅新東ビル4階)

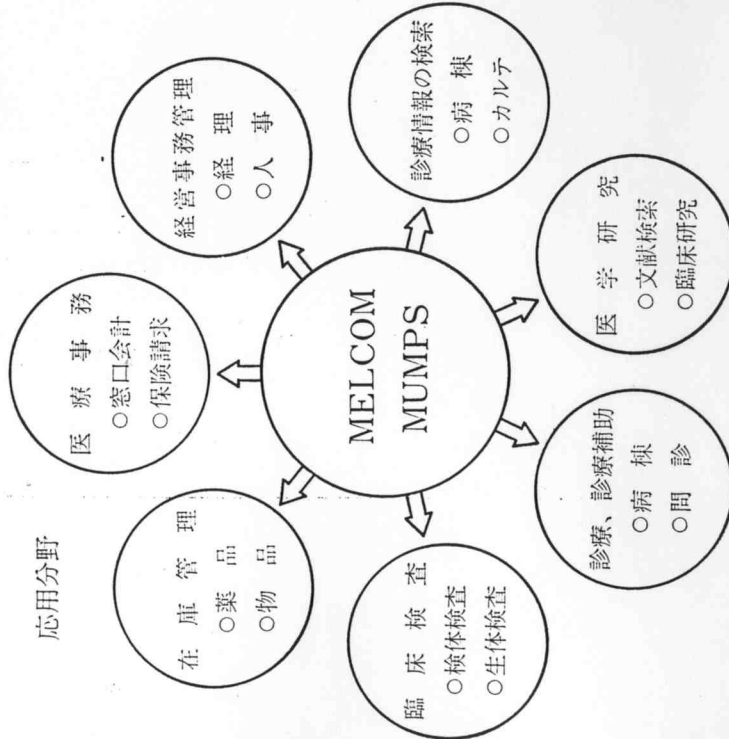
TEL (06)364-2364 (代) 〒530

MELCOM-MUMPS

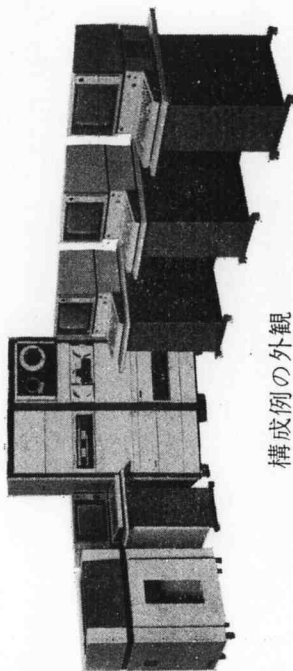
MELCOM MUMPSは、高性能ミニコンMELCOM-70のもとで、すぐれた処理能力、

データベース管理機能、豊富な入出力装置、充実したソフトウェアをもち、医療情

報分野をはじめとし、工業分野、分析研究分野にも広く適用できます。



応用分野



構成例の外観

本社——東京都千代田区丸の内2丁目2番3号〈三菱電機ビル〉——☎100 東京 03〈218〉3216

お問合せ先

大阪営業所	——	大阪市北区梅田町8番地〈西阪神ビル〉	——	☎530	大 阪
名古屋営業所	——	名古屋市中村区広井町3丁目88番地〈大名古屋ビル〉	——	☎450	名 古 屋
九州営業所	——	福岡市中央区天神2丁目12番1号〈天神ビル〉	——	☎810	福 岡
札幌営業所	——	札幌市中央区北二条西4丁目1番地〈北海道ビル〉	——	☎060	札 幌
東北営業所	——	仙台市大町1丁目1番30号〈新仙台ビル〉	——	☎980	仙 台
北陸営業所	——	富山市桜木町1番29号〈明治生命館〉	——	☎930	富 山
広島営業所	——	広島市中町7番32号〈日本生命ビル〉	——	☎730	広 島
四国営業所	——	高松市丸の内2番5号〈ヨシヅビル別館〉	——	☎760	高 松
新潟営業所	——	新潟市東大通2丁目4番10号〈日本生命ビル〉	——	☎950	新 潟
静岡営業所	——	静岡市伝馬町16の3番地〈明治生命静岡支社〉	——	☎420	静 岡
長崎営業所	——	長崎市丸尾町7番8号〈長崎底曳会館〉	——	☎852	長 崎
岡山営業所	——	岡山市駅前1丁目9番地〈明治生命ビル〉	——	☎700	岡 山

第6回 日本MUG学術大会 プログラム および抄録集

◇第2日

----- 昭和54年9月14日(金)

会場: 京大会館 2F 210号室

—— 9:00 開会 ——

A. 一般演題 (9:10~9:40) (座長) 小森 優

A-1 LOG-ON 管理システム ----- 1

..... 大阪府立成人病センター 情報企画室

服部 敏夫・藤江 昭

A-2 マイコンコンピュータ用MUMPSの開発 ----- 5

..... 日立製作所 システム開発研究所

渡辺 坦・大沢 恒春・隈 久雄

..... 日立メテコ

大家 康秀・塚田 涌長

A-3 Computer Assisted Talkie Slide (CATS) ----- 9

..... 獨協医科大学 総合研究施設

林 徹・木村一元

..... 獨協医科大学 第一病理学教室

馬場 謙介・木村 園恵

..... 三洋電機株式会社 開発研究所

武田 和忠

B. パル(1) MUMPSの速さ (9:50~10:50) (座長) 今井 敏雄

B-1 MUMPSの速さを上げるための技術 ----- 11

..... 日本DEC

笹子 能

B-2 MUMPS ベンチマークテスト ----- 19

..... 大阪府立羽曳野病院

今井 敏雄

B-3 MUMPSと他言語との実行速度比較 ----- 39
 大阪府立成人病センター 情報企画室
 藤江 昭・永井 利廣・服部敏夫

B-4 MUMPS-FORTRAN トランスレータ ----- 59
 獨協医科大学 総合研究施設
 獨協医科大学 第一病理学教室
 林 徹・木村一元
 馬場謙介・木村園恵

※ 指定発言者

小森 優・塚本 耕平

C. パネル(2) MUMPSの互換性 (11:00~12:30)
 (座長) 嶋 芳成

C-1 総 論 ----- 69
 マンプスシステム研究所
 嶋 芳成

C-2 標準MUMPSの言語仕様について ----- 77
 日立製作所 システム開発研究所
 渡辺 坦・大沢恒春

C-3 実際の移送の経験から(ユーザの立場から) ----- 85
 大阪府立羽曳野病院
 今井 敏雄

C-4 実際の移送の経験から(メーカの立場から) ----- 101
 日本ミニコンピュータ
 木下博義

C-5 MUMPSにおけるカナ文字の現状と展望 ----- 107
 大阪府立羽曳野病院
 野口 弘・大槲陽一

※ 指定発言者

菅沢 登・錦見 尚道

D. MUMPSの医療への応用について (14:00~16:00)
(座長) 山本和子

D-1 MUMPSによる健康管理システム ----- 117

..... 防衛医科大学

西田 正之・渡辺 隆郁・風間 富栄・砂川 裕之
高谷 治 他 ME同好会員

D-2 全国剖検症例の検索 ----- 121

..... 獨協医科大学 総合研究施設

木村 一元

..... 獨協医科大学 第一病理学教室

木村 園恵・馬場 謙介

..... 東京大学 医学部 病理学教室

浦野 順文

..... 慈恵会医科大学 病理学教室

藍沢 茂雄

D-3 血液透析用病態解析システムについて ----- 123

..... (財)鷹揚郷腎研究所

平山 順朗・山内 裕子・磯野 正子・八木 槁栄子
内山 公一・森 喜代志・石戸谷 豊・舟生 富寿

D-4 産科研究データと産科臨床データの統合データベースの作成 ----- 129

..... 京大産婦人科・国立京都病院

吉村 誠之

..... 京大病院中央情報処理部

平川 顕名

D-5 標準MUMPSによる退院病歴の管理 ----- 135

..... 大阪医科大学 衛生・公衆衛生

山本 和子

D-6 病院管理システム ----- 147

..... 三鷹市・長谷川病院 医事課

佐藤 弘美

..... 三鷹市・長谷川病院 管理室

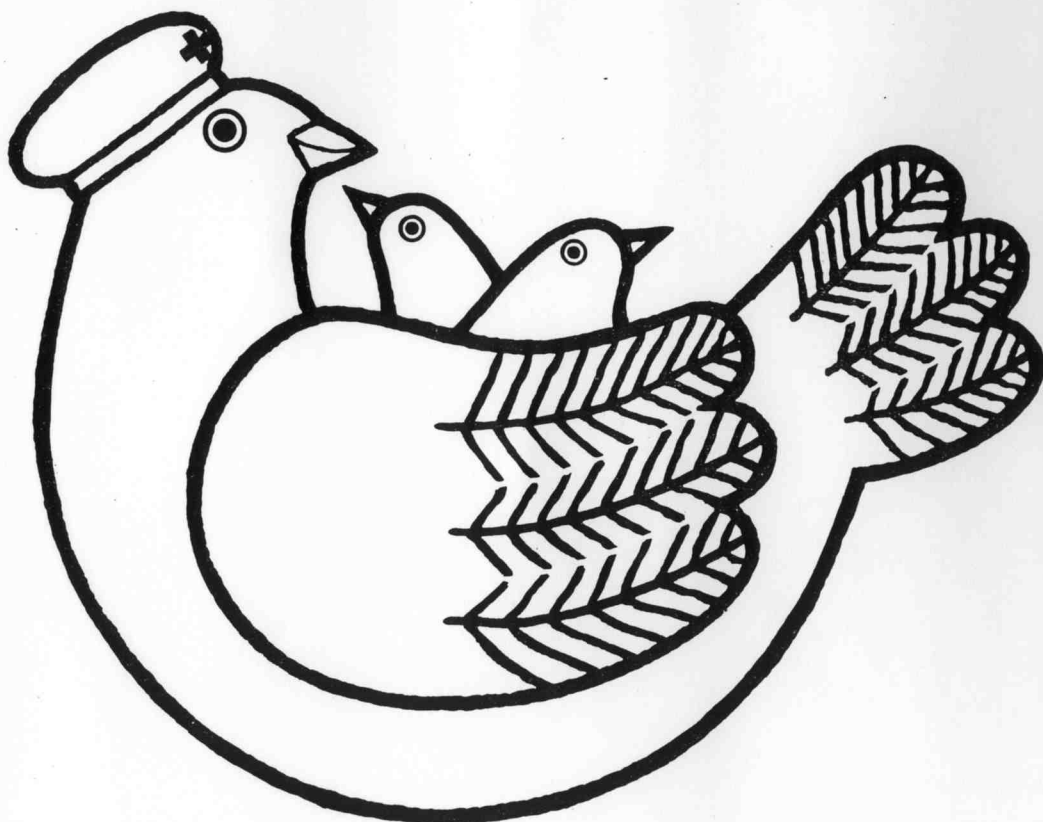
川島 英樹・山浦 寿子

D-7 検査結果問合わせシステムの試み----- 151
..... 大阪府立成人病センター 情報企画室 北川 護・上田 清治
..... 大阪府立成人病センター 臨床検査部 森井卓郎

D-8 MUMPS-Systemを利用した臨床診断のためのLaboratory aid-- 167
..... 和歌山県立医科大学 中央検査部 前田 次郎・宮本 一雄・西村 雅隆
..... 和歌山県立医科大学 消化器内科 山本 實彦
..... 大阪府立羽曳野病院 ME研究室 大槲 陽一・今井 敏雄・野口 弘

E. 日本のMUGと世界のMUG (16:10~17:00)
(座長) 若井 一朗

いま、 医療経営に最新医療機器の 導入が求められています。

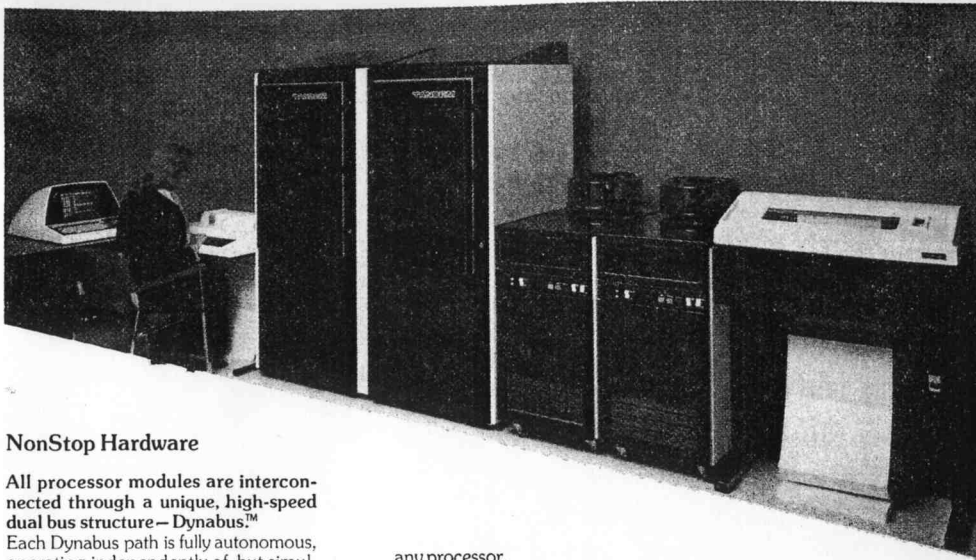


- 取扱品目**
- ① 診断機器
 - ×線テレビジョン装置
 - 超音波診断装置
 - 心電心音計
 - 脳波計
 - ガストロファイバー
 - ② 治療機器
 - 人工腎臓装置
 - 人工心肺装置
 - ③ 能力回復機器
 - 電動ベット
 - リハビリ機器
 - ④ 健康管理
 - 全自動血液分析装置
 - 総合検診システム
 - 検診車
- ★この他、いろいろな医療機器をリースいたします。

必要な時、購入資金を確保せずに医療機器が導入できるリース利用は、患者サービスの向上、病院経営改善に大きな利益をもたらします。

発展へのブレイク
昭和リース

本社	東京都千代田区大手町1-7-2	☎(03) 242-6621
大阪支店	大阪市東区京橋2-15-1	☎(06) 941-7171
名古屋支店	名古屋市中区錦2-15-22	☎(052)211-5271
札幌営業所	札幌市中央区北三条西4-1	☎(011)231-1567
仙台営業所	仙台市中央2-2-10	☎(0222)64-0742
広島営業所	広島市基町11-10	☎(0822)22-0827
福岡営業所	福岡市博多区綱場町2-2	☎(092)271-2187



NonStop Hardware

All processor modules are interconnected through a unique, high-speed dual bus structure—Dynabus.SM

Each Dynabus path is fully autonomous, operating independently of, but simultaneously with, the other bus to insure that two communications paths exist between all processor modules in the Tandem 16 NonStop System. All Dynabus interprocessor transfers are hardware controlled, independent of, but concurrent with, all normal I/O transfers and other CPU activities. Since Dynabus handles interprocessor transfers at 26 megabytes/second, this produces extremely low overhead interprocessor communications at speeds fast enough to simultaneously handle peak transaction and message loads even when the Tandem system is fully expanded to sixteen processors.

Designed to increase transaction throughput and minimize system overhead, the Tandem processors separate I/O processing, from interprocessor communications and CPU workloads. Each processor can handle up to 128 19.2K baud communications lines, a message handling capability miles ahead of other processors in their price category. The CPU segment of each processor is a pipelined microprogrammed unit designed specifically to handle business transaction data applications. With a cycle time of 100 nanoseconds, each CPU can, for example, add two 18-digit numbers in under 2.4 microseconds.

NonStop Software

Guardian controls the traffic and watches over the system.

Guardian is Tandem's complete, transaction-oriented operating system. It resides in each processor, and has the capability to respond positively to a failure anywhere in the system. With Guardian, there is no need for customized or special operating system development by the user. Since programs are geographically independent, the task of applications programming is vastly simplified for the user. Further, Guardian carries the user-assigned priorities for applications programs, handles all communications among programs and between programs and the outside world, and extends the capability to start program execution in any available processor module from

any processor.

Geographical independence of programs and data is a noteworthy feature of Guardian. Programs are not only unaware of which processor, or even which node in a network, is running them; they may well be running simultaneously on all processors or in multiple nodes. And programs can access any device in the system, even those not physically connected to the processor(s) running the program. Because of this geographic independence, the Tandem NonStop system can be expanded all the way to sixteen processors and beyond via the network without any reprogramming.

Multiprocessor message system.

Guardian automatically handles all communications between Tandem processor modules, system processes and applications programs—routing messages to the correct processor, verifying correct receipt and deciding which program receives the message in the destination processor.

Guardian provides protection, performing comprehensive data validation on all transfers. It was designed to detect and isolate any faulty module, preventing corruption of any other module by a "mad" processor. Guardian has the inherent ability to detect an error or point of non-response anywhere in the system, log the failure and disallow access to the faulty module without disturbing system operation. The Guardian operating system, along with the system hardware architecture, is the basis of Tandem's NonStop operation.

Expand: global networks made easy.

Expand extends the basic, single node capabilities of Guardian to a whole network. Under Expand, the network can grow as large as 4,080 processors, each one capable of accessing a geographically distributed data base exactly as if it were located in its entirety in the local system. Each node of the network can be minimal or a fully expanded Tandem NonStop System; and there can be as many as 255 nodes in the network. Since Guardian treats all resources within the system, both hardware and software, as

files, expansion and perhaps reconfiguration of the system is accomplished without reprogramming, even without re-compile. Expand also extends the NonStop capabilities of the single node to the network. Packets of data are passed from node to node automatically and are rerouted automatically if a failure occurs in a communication line.

Tandem's Transaction Application Language.

T/TAL. A powerful block structured language designed for fast, flexible programming. T/TAL is self-documenting and is easy to read, modify and maintain. Developed by Tandem, T/TAL gives the programmer every vehicle to optimize the hardware potentials of its interactive, multi-processor environment. T/TAL promotes the use of structured programming principles.

Tandem/COBOL.

The only COBOL available for multiple processor systems, Tandem/COBOL (ANSI 1974) utilizes all the capabilities of Guardian and Enscribe. Under Guardian and Guardian/Expand; Tandem/COBOL features NonStop operation: shared, re-entrant code; virtual memory; geographic independence of I/O devices; and checkpoint/checkmonitor facilities. Under Enscribe, Tandem/COBOL provides key-sequenced, entry-sequenced and relative file structures; logical file size up to four billion bytes; up to 255 alternate keys per file; and optional mirror data base recording.

Tandem/FORTRAN.

This is the only FORTRAN available and optimized for multiple processor systems. Tandem/FORTRAN (ANSI 77). It utilizes all of our Guardian and Guardian/Expand Operating System features including NonStop operation, re-entrant code, interprocessor communications, virtual memory, and Enscribe data base record management facilities for keyed, relative and sequential access, multi-

keyed data paths, and concurrent record access. Benchmarks consistently show Tandem/FORTRAN to be exceptionally fast, and fully capable of running efficiently in a multi-language environment. This is comprehensive FORTRAN with a host of extensions utilizing and capitalizing on the special features of the Tandem NonStop System.

Tandem/MUMPS.

Like our COBOL and FORTRAN, this is the only MUMPS available for a multiple processor system. Tandem/MUMPS both complies with and vastly exceeds MUMPS ANSI Standard 1977. This is the only MUMPS which allows concurrent execution of COBOL, FORTRAN and T/TAL. It can access files created by other programs, and it can share global variables with other language systems. And since it operates under Guardian, Guardian/Expand and Enscribe, it offers all of Tandem's NonStop System features and support. Program development is made remarkably easy through the use of the Tandem Editor incorporated into the T/MUMPS capability. And no one else offers a MUMPS package with near the Tandem available memory.

Enscribe: versatile and efficient data base management system, providing high level access to and manipulation of records in data bases. Operating in distributed fashion across multiple processors as a part of Guardian or Guardian/Expand, Enscribe ensures the integrity of the data base in case a processor, an I/O channel, disc drive or communication line fails during transaction processing. During operations, Enscribe performs internal checkpointing to ensure integrity of the file structure and ensure that no user data is lost. Control information and data are maintained in two processors controlling a disc volume; if a failure occurs in one processor, Enscribe completes the operation using the alternate processor. And Enscribe maintains all indices; when a new record is added to the file or a key value is changed, Enscribe automatically updates the indices to the affected records. The programmer need never worry about assignment or maintenance of indices.

Multiple file structures.

Key-sequenced, relative or entry-sequenced—all under Enscribe and all accessible by the primary key or any of the 255 possible alternate keys. Location of records may be by approximate (range of key values), generic (partial key matches) or exact key value. Enscribe maintains an index of all key values, providing rapid access and update whenever key values are supplied.

Data definition and manipulation.

Under Enscribe, the user can use Tandem's DDL, Data Definition Language, describing the data base as a "schema." Since all programs use the schema to access the data base, a correct view of the data is assured. It also defines which fields are to be used as access paths (keys) to retrieve records from the data base. Changes to data base record layout, additions of types of records or new alternate keys are accomplished with automated ease.

Enform: a major improvement in Queries and Reports.

Operating under Enscribe, and therefore within the framework of Guardian and Guardian/Expand, Tandem's Enform has two major advantages over any other Query/Report Writer. It operates over a distributed data base, invisibly and without special considerations. And, it defines relationships between separate records at the time of inquiry without affecting the data base. File relationships are defined by common codes, keys or fields and can be changed at will. Enform automatically takes advantage of all primary and secondary keys to locate called data in the most efficient way. And the same English-like language is used for both queries and reports, which produces the results in a fraction of the time at a fraction of the cost. Formatting includes all appropriate signs and punctuation; you can even build in calculation of variable formulas, commissions and such. The same data independence, file structure independence and geographical independence available to the interactive users of Enform is also directly available to the application programmer in COBOL, FORTRAN, T/TAL or MUMPS. And of course you can change any aspect you wish—at will. Even convert into French, German or Spanish. It is incredibly flexible. Capable. And economical.

Envoy: multifaceted data communications system.

Operating as an integral part of Guardian, Envoy provides the interface between applications programs and data communications networks. Envoy supports both binary synchronous and asynchronous communications, with single or multi-drop lines on either a local or remote basis.

Data is transferred from terminals directly into main memory, which means processors are not interrupted until a complete message has been received. Binary synchronous terminal polling and character translation between ASCII and other communications codes are hardware executed, minimizing overhead. Asynchronous operations run at rates up to 19.2K baud per line; binary synchronous at rates up to 80K baud per line.

Entry: page mode forms creation, display and access.

One of the easiest-to-use programs of its kind. Users simply design the form on the screen to appear as it will be used. Delimiters, field names and validity checking are automatically recorded in a designated file on disc. In use, invalid data is automatically checked and displayed as a flashing entry on the screen. Individual fields in any form may be accessed by name. This package is available for both Tandem's page mode terminals and IBM 3270 terminals.

XRAY: to balance loads and fine-tune applications.

With less than 1% overhead, Tandem's NonStop XRAY monitors total system performance and resource utilization. You can spot overloads immediately and fine-tune work distribution automatically, even as processing occurs. Bottlenecks in programs or files, processors, controllers or terminals, show up immediately on CRT display or in hard copy printouts and corrective balancing can be imple-

mented without delay. One early application of XRAY allowed enhancement of our own system microcode so that two processors could accomplish what previously required three.

Data Base Integrity.

Probably no one aspect of an on-line system causes more concern, and rightly so. No one in the entire data processing industry offers anywhere near the protection offered by Tandem's NonStop System. And that protection extends not only against catastrophic failures, but also against the far more insidious creeping failure, the kind which doesn't show up as a stoppage in the computer room, but only as mis-billed customers, false inputs on a medical chart, funds transferred to a wrong account, a major purchase of the wrong stock, inventory shipped to a wrong location... who knows how much damage before detection. Because of its unique transaction handling procedures, no transaction is ever lost in process nor is it ever duplicated. The integral redundancy of the Tandem system provides an unprecedented level of protection, not only at the transaction level, but at all levels of the data base.

Remote diagnostics: anywhere in the system.

Tandem's DIAGLINK provides interactive analysis of what's happening anywhere in the system, in hardware or software, from any terminal in the system (provided proper security clearances are handled) or, through a modem connection, from any Tandem Service Center. With this capability, our service personnel arrive at your DP center fully prepared to make the immediate replacement or adjustment required. And of course, the system is up and running during this whole time.

Software tools to work with.

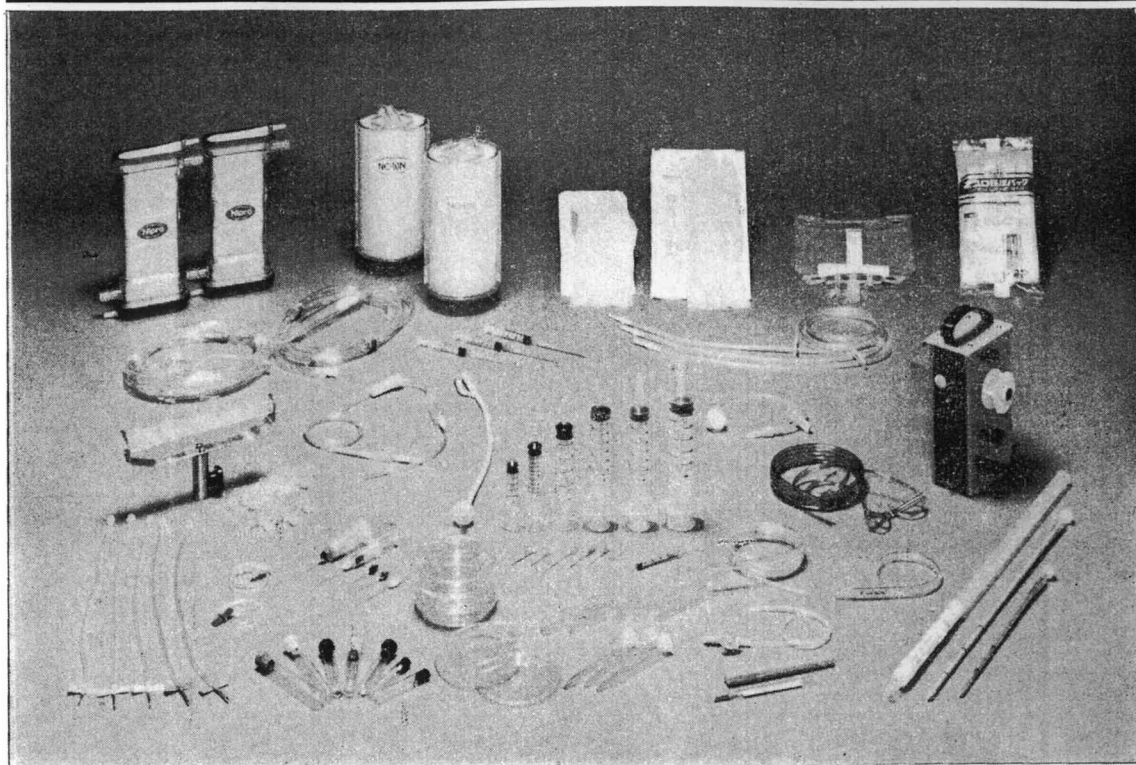
Under the overall supervision of either Guardian/Expand or Guardian, each Tandem NonStop System has available a complete multiple processor and control communications system supporting a host of applications languages including industry standard ANSI 77 FORTRAN, ANSI 74 COBOL and ANSI 77 MUMPS. With our own T/TAL, EDITOR and its associated galley formatter TGAL, SORT/MERGE, DEBUG, ENFORM Query/Report Writer and complete diagnostic capabilities, the Tandem software support system is truly impressive. And best of all, it never requires modification as an individual system, node or the entire network expands or is modified to suit changing requirements. And any network node can communicate with IBM or any other mainframe using industry standard protocols.

Service during On-Line operations.

With any system a hardware failure must be repaired. But only with Tandem can the system keep right on operating, right through the failure and right through the repair too. With approval of all major international safety standards organizations. That's unique. Tandem's customer service representative can remove and replace any failed module in your system without interrupting service. The operators at terminals and the programs in process are unaffected by either the failure or replacement of the failed module. Without this feature, no system can truly be called NonStop.

Nipro

信頼される医療器ニプロ



おとどけする医療器具の一部です

ニプロはいつも すぐれた医療に
役立つよう あらゆる技術を結集して
新しい医療器具を生みだしています。
また、各地にサービス網を持っており
いつでも迅速におとどけします。

ニプロがおとどけするすぐれた医療器具.....
 ディスポーザブル医療器具——●注射針●注射筒●セット類(輸液セット・輸血セット・採血セット等)●グローブ(外科手袋・補助手袋)●チューブカテーテル類
 ●カニューラ類●コネクター類(三方活栓・Y字管等)●バッグ類(栄養バッグ等)●ラボラトリー製品(真空採血器ネオチューブ・シャーレスピッツ等)●その他
 人工腎臓用ディスポーザブル器材——●ダイアライザー(コイル型・ホローファイバー型)●血液回路●AVFニードル
 人工腎臓用器機———透析液供給装置●コンソール●血液ポンプ●注入器●スケールベッド●その他医療用器具器材及び病院用設備器機等

株式会社  ニプロ

本社 大阪市大淀区豊崎3丁目3番13号 〒531
TEL (06) 373-3155(代)

●営業所 札幌・青森・仙台・山形・新潟・千葉・宇都
宮・目黒・東京・横浜・静岡・松本・名古屋
・金沢・京滋・大阪・奈良・和歌山・神戸・
岡山・山陰・広島・高松・松山・福岡・北九
州・熊本・長崎・鹿児島

第6回日本MUG学術大会 昭和54年9月14日(金)

A. 一般演題

A-1. LOG-ON 管理システム

大阪府立成人病センター 情報企画室

服部 敏夫 藤江 昭

MUMPSは、多数のユーザーの同時使用を可能にした手軽なTSS言語であるが、そのわりには、LOG-ONの管理や機密保護の面では決して充分とはいえない。今回、LOG-ONの管理強化および機能の拡張を目的としたLOG-ON管理システムを開発したので報告する。

1. 目的

当センターでは、DEC社のMUMPS-11およびDSM-11を使用して、外来会計・外来予約・入院・サプライ・病棟などの種々の業務を行っているが、業務の拡大に伴いコンピュータにアクセスするエンドユーザーも多数になり、LOG-ON管理や機密保護が重要な問題になってきた。

MUMPS-11やDSM-11では、LOG-ONの管理として、次の2つのパスワードにより、ユーザーのコンピュータへのアクセスをチェックしている。

UCI (User Class Identifier) ---- RUNモード(プログラムの実行、グローバルの使用)へのアクセスをチェック

PAC (Programmer Access Code) ---- COMMANDモード(プログラム、グローバルの作成、変更、削除)へのアクセスをチェック

また、特定の端末から起動すると、あらかじめ指定した特定のプログラムにアクセスできるTERMINAL TIEDモードも可能である。

しかし、LOG-ON管理としては以下のような不十分な点や不便な点もある。

- ・LOG-ON時に入力したUCIやPACがエコーバックで出力されるので、UCIやPACを他人に知られるおそれがある。
- ・LOG-ON情報(時刻、UCI名、端末No.、入力者名、等)が蓄積されていないので、エンドユーザーのコンピュータへのアクセス状況の把握ができない。
- ・TERMINAL TIEDモードにより、特定の端末から特定のプログラムにアクセスできるが、その端末はそのプログラムに固定され、他のUCIやプログラムにアクセスすることができない。

- TERMINAL TIED モードの指定は、システム・ジェネレーション時、または、ユーティリティープログラムで行うが、これらはシステムを止めないと行えない。
 - TERMINAL TIED モード以外は必ず UCI を知らねばならない。多数のエンドユーザーがいる場合には、1つの UCI を共有し、パスワードとしての性格を弱めさせるおそれがある。
- そこで、これらの点を改善した LOG-ON 管理システムを開発し、各端末における LOG-ON の総合的な管理を行なわせることにした。

2. 基本仕様

LOG-ON 管理システムは、MUMPS 自体の機能に加えて、以下の様な機能をもつ。

- パスワードを入力しても、画面上にエコーバックは返さない。(CRT の場合)
- LOG-ON 時に以下の4つのモードによるアクセスを可能にする。
 - (モード1) FAST TIED モード-----パスワード=NULL
MUMPS での TERMINAL TIED モードに相当
 - (モード2) VALID UCI モード-----パスワード=UCI:PAC
MUMPS での COMMAND モードに相当
 - (モード3) LOOSE TIED モード-----パスワード=UCI:(Program 名)
MUMPS での RUN モードに相当
 - (モード4) SPECIAL CODE モード-----パスワード=(SPECIAL CODE)
新たに追加したモード
登録された SPECIAL CODE により、特定の UCI または特定のプログラムをアクセスする。
- FAST TIED モードと SPECIAL CODE モードをグローバル ^LOGON に登録する。これらはシステム・マネージャーが行う。そのフォーマットは以下の様である。
 - FAST TIED モードの場合
^LOGON("TIED", 端末 No.) = (UCI 名):(プログラム名)
 - SPECIAL CODE モードの場合
^LOGON("SPEC", 通し No.)
= (SPECIAL CODE):(UCI 名):(PAC 名またはプログラム名)
- LOG-ON 情報をグローバル ^USE に登録する。そのフォーマットは次の様である。
 - ^USE(日付, UCI No., 通し No.)
= (LOG-ON 時刻)*(端末 No.)*(PAC 名)*
(TRY 回数)*(モード No.)*(パスワード)*(氏名)
 また、システム・シャットダウン時に、蓄積した LOG-ON 情報を出力する。
- LOG-ON に3回以上失敗した場合、不正使用とみなして、その入力端末をシステムより切り放す。切り放された端末は、システム・マネージャーが再接続する。

3. 方法

LOG-ON 管理システムは, LG, LOG, USEの3つのプログラムより構成されている。

プログラムLGでは, エコーバックの制御, 4つのLOG-ONモードの管理, LOG-ON情報の登録, およびLOG-ONエラー時の端末の切り放しを行う。

プログラムLGを, あらかじめ端末にTIEDしておくことにより, その端末を起動するとLOG-ON管理システムがはたらき, 次のようなメッセージを出力する。

```
DSM-11 V1   LINE #8   < LOG-ON SUPERVISOR >
WHAT IS YOUR NAME ? :
ENTER PASSWORD      :
```

LOG-ONする者は, 自分の名前と, 基本仕様で述べた4つのLOG-ONモードに対応するパスワードを入力する。パスワードに対してはエコーバックが返ってこないで, UCI, PAC, SPECIAL CODEなどが他人に知られることはない。入力されたパスワードにより, 各モードが選択され, UCI, PAC, プログラム名, SPECIAL CODEなどがチェックされ, OKならば, LOG-ON情報を登録した後, 各モードにアクセスされる。NOならば, 再トライされ, トライ回数が3回を越えても成功しない場合は, 不正使用とみなされ, LOG-ON情報を登録した後, 端末が切り放される。

*** TERMINAL LOG-ON INFORMATION ***

SYSTEM HAS LOG-ON INFORMATION ABOUT THESE DATE.

79-8-7
79-8-8

WHICH DATE DO YOU WANT ? <CR> IS TODAY : 79-8-8
[LONG] OR [SHORT] FORMAT : L

```
** MODE 1 ---> FAST TIED MODE
MODE 2 ---> VALID UCI
MODE 3 ---> LOOSE TIED MODE
MODE 4 ---> SPECIAL CODE
```

UCI NO.	UCI NAME	TIME	TERM. NO.	PAC CODE	TRY COUNT	INPUT MODE	LOG-ON VALUE	USER NAME
1	MGR	10:52 AM	10		1	2	MGR:	KAWAKAMI
3	OSC	11:49 AM	8		1	2	OSC:	TERAMOTO
4	DOC	9:47 AM	8		1	2	DOC:	TERAMOTO
4	DOC	9:50 AM	5		1	2	DOC:	UCHIDA
4	DOC	11:03 AM	8		1	2	DOC:	TERAMOTO
5	LAB	9:13 AM	4		1	2	LAB:	KAWAKAMI
5	LAB	10:44 AM	8		1	2	LAB:	KITAMURA
5	LAB	11:40 AM	4		1	2	LAB:	OKAZAKI

DO YOU WANT TO KILL THIS DATA ? Y/N <CR>=N :

図1. LOG-ON 情報の出力例 (LONG FORMATの場合)

UCI#	UCI NAME	COUNT
0	FAILURE	1
1	MGR	1
3	OSC	2
4	DOC	9
5	LAB	7

図2. LOG-ON情報の出力例(SHORT FORMATの場合)

プログラムLOGでは、切り放された端末の接続、SPECIAL CODEモードおよびFAST TIEDモードの登録を行う。これらの作業は、必ずシステム・マネージャーが行う。

プログラムUSEでは、プログラムLGで登録されたLOG-ON情報をシステム・シャットダウン時に出力する。出力の際、日付指定、LONG FORMAT/SHORT FORMATの指定が可能である。図1、図2にLONG FORMAT、SHORT FORMATによるLOG-ON情報の出力例を示す。

4. 考察

以上、LOG-ON管理システムの概要について述べた。

LOG-ON管理システムは、現在、病棟システムの一部の端末でテスト稼働中である。LOG-ON操作は、以前と比べて名前を入力する分だけ多くなっているが、ユーザーに対してはそれほど負担とはなっていないようである。ただ、操作ミスにより端末が切り放されることが時々あるので、3回のTRY回数が最直かについては検討の余地がある。システム・シャットダウン時に出力されるLOG-ON情報は、不正使用のチェックなどLOG-ON管理上の資料としても有用であるが、その他に、例えば、病棟における検査結果照会システムの利用状況の把握などにも役立つ。

MUMPSは、だれでもが手軽に使えることを目的として開発された言語であるが、「だれでもが手軽に利用できる」ということと「機密保護」ということをいかに両立させていくかに問題がある。LOG-ON管理システムでは、できるだけ簡単なLOG-ON操作で、しかも、機密保護上最低限必要だと思われる機能を付加したことにより、この問題をある程度は解決したと思われる。

第6回日本MUG学術大会 昭和54年9月14日(金)

A. 一般演題

A-2. マイクロコンピュータ用 MUMPS の開発

日立製作所システム開発研究所
日立メディコ

渡辺 坦, 大沢恒春, 隈 久雄
大家康秀, 塚田湧長

1. まえがき

MUMPSは、データベースの会話型利用という計算機の高度な利用技術を、計算機の専門家の手から解放し、計算機のユーザが直接に使いこなせるようにする言語システムである。これを載せる計算機は、小さければなおさらに簡便さを増やすことができる。従来はこれに主としてミニコンが使われてきた。我々はこれをさらに一回り小さくし、マイクロコンピュータで実現した。

今般開発したMUMPSは、マイクロコンピュータによる医療事務システムHIMEC 10の上で動く。これは、HIMEC 10のプログラム作りを容易にし、ユーザ自身が自分の希望通りに計算機による処理ができるようにすることを目的としている。開発にあたっては、BASICやRPG等、他の言語についてもこの目的にかなうか否か検討したが、

- (1) 会話型で使えること、
 - (2) データベースを利用できること、
 - (3) 使い方が簡易なこと、
 - (4) 文字列処理に便利なこと、
- 等の理由でMUMPSを選定した。

2. 開発方針

(1) 実用性の重視

開発にあたっては、MUMPSをマイクロコンピュータで実現するというにとどまらず、真に実用的な会話型簡易データベースをマイクロコンピュータに装備することを目的とした。そのため、処理性能と操作性を良くすること、異常対策を十分行なうこと、文法チェックは実行時のみでなくプログラム登録時にも行なうこと、ユーティリティを充実させること、などに意を用いた。

(2) 標準MUMPSへの準拠

MUMPSのANSI規格^{1), 2)}で規定されている部分は、できる限りそれに従った。規定されていない部分は、操作性を良くすることを重点にして決めた。ただし、後述するように、マイクロコンピュータでは必ずしも必要とされない機能については、組みこんでいないものもある。

(3) 他システムとの並列処理

ミニコン用に作られたMUMPSの多くは、1つの計算機全体をMUMPSで占有して使う形態である。本MUMPSは、1つのマイクロコンピュータの上で、MUMPS以外のソフトウェアと並列処理できるようにした。医療事務システムHIMEC10では、高速処理の要求される定形業務の処理や、端末の特性を利用する処理には、すでにアセンブリ言語等で書かれたソフトウェアが作られている。本MUMPSは、これらの業務をするかたわら動かすことができる。

(4) 高級言語での記述

MUMPSシステムの作成と保守を容易にするため、記述言語としてはPL/M³⁾を用いた。マイクロコンピュータとしては、現在、インテル8085を使っているが、HMC56800等、他の機種用に作りかえることも比較的容易である。

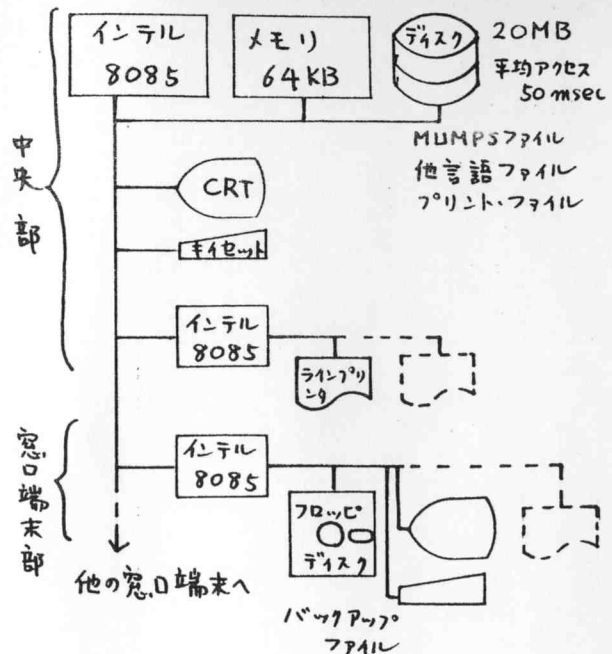


図1. 医療事務システムHIMEC10

3. ハードウェア構成

MUMPSを載せたHIMEC10は、図1のように、中央部といくつかの窓口端末部から構成される。中央部には、64Kバイトのメモリをつけたインテル8085に20Mバイトの磁気ディスクと、96ボタンのキーセット、CRT、ラインプリンタがつく。窓口端末部には、キーセットとCRT、フロッピー・ディスクがつく。中央部でMUMPSを動かしている間、窓口端末部ではそのキーセットの特性を活かしたソフトウェアにより、データの入力や表示ができる。20Mバイトのディスクのうち、半分の10Mバイト分はとりはずし可能である。バックアップ・ファイルを作るには、ディスク・パック自体をとりはずす形もとれるが、グローバル変数やルーチンのうち、指定したものを窓口端末のフロッピー・ディスクに退避する形もとれる。退避したものをフロッピー・ディスクから回復すると、ディスクの内容は、スペースやアクセス時間の無駄が少なくなるように自動的に再編成される。

4. ソフトウェア機能

(1) 使用形態

本MUMPSでは、MUMPS以外の言語で書かれたプログラムと並列処理できるが、同時に走るMUMPSジョブの数は1つにとどめてある。これは、マイクロコンピュータではCPUが安価なため、複数ジョブを走行させる必要性は少ないと判断したためである。一方、MUMPSで使うファイルは図2のようにユーザ・クラスごとに分離して

あり、各ユーザは自分の属するユーザ・クラスのファイルと、ライブラリ・ルーチンが入ったシステム・クラスのファイルを使う。したがって、誤って他のユーザ・クラスのデータを破壊するおそれはない。他のユーザ・クラスのルーチンやグローバル変数を使うときは、バック・アップ・ファイルを紹介して、必要部分を取り出す。

(2) 標準MUMPSの機能

文法はANSI規格の標準MUMPSに合わせたが、マイクロコンピュータ用であることと利用形態を考慮して、以下の点については標準MUMPSより機能を減らし、サブセットとした。

- (a) 浮動小数点演算は、マイクロコンピュータでは著しく時間がかかるので、組みこんでない。数値は小数点以上7桁、小数点以下2桁の固定小数点方式で表わした。
- (b) @記号付きの間接指定と\$TEXTは組みこんでない。これらは定数の選択やプログラムの編集等の目的に使われることが多いが、本MUMPSではそれを他の機能やコマンドで代替可能なことと、間接指定は使い方がむづかしいためである。
- (c) ジョブの多重処理に使う\$JOB, LOCK, OPEN, CLOSEの各機能は、多重処理をしないため、含めていない。
- (d) \$VIEWは含めず、システムの運用に必要な機能はユーティリティとして組みこんだ。
- (e) タイマを持たないため、\$HOROLOGYとREAD命令における時間指定は削った。

(3) 拡張機能

- (a) カタカナを追加し、そのパターン照合コードはKとした。
- (b) アセンブリ言語等、他の言語で書かれたルーチンを呼び出すコマンドZCALLを追加した。パラメータとして渡す値はMUMPSの定数や変数で表わすが、値を受けとる場合は、それ用に設けた特殊変数\$ZAREAを使うことにした。
- (c) 直接モードで働く補助コマンドとしては、以下のものを設けた。

ZFILE	現用ルーチンをディスク・ファイルに入れる。
ZLOAD	ディスク・ファイルからルーチンを取り出し、現用ルーチンとする。
ZERAZE	現用ルーチンの一部または全部を消去する。
ZPRINT	現用ルーチンの一部または全部をプリントする。
ZWRITE	現時点でのローカル変数とその値を出力する。
ZMODIFY	指定された行のある文字列を他の文字列でおきかえる。
ZGO	間接モードへ復帰する。

- (d) 割りこみキイを設け、これが押されるとBREAKを実行したときと同じように処理を中断し、直接モードでコマンドの入力待ちとする。

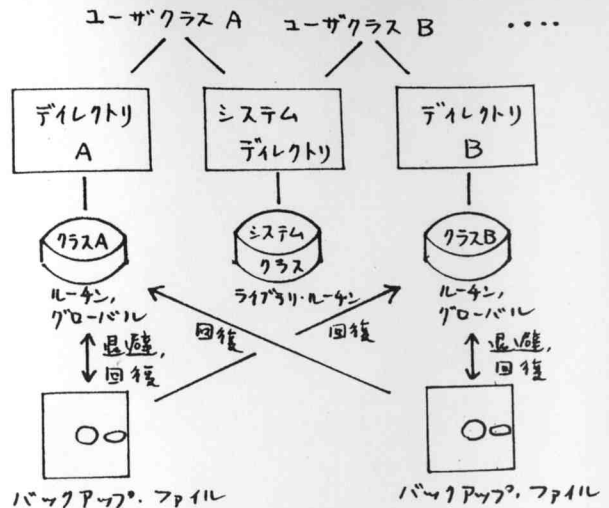


図2. ファイルの使い分け

(4) ユーティリティ

- ZP プログラム・ディレクトリを出力する。
 - ZG グローバル変数ディレクトリを出力する。
 - ZW 指定したグローバル変数の全内容を出力する。
 - ZS 指定したルーチンまたはグローバル変数をフロッピー・ディスクに退避する。ルーチン名やグローバル変数の指定がなければ、該当ユーザクラスの情報と全部退避。
 - ZR フロッピー・ディスクに退避されている内容を回復する。
- 以上のほかに、MUMPSで使うディスク・ファイルの内容を初期設定する命令がある。

5. 作成技法

グローバル変数に対しては、B*トリーの技法を応用して、ディスク上の各ブロックのデータ量がバランスするようにし、インデクス・ブロックの段数が均等化するようにした。また、インデクスの共通部分を縮約して大きさを小さくし、ブロックの区切り点の決定にあたっては局所的最適化をし、できるだけ論理的に大きい区切り点と区切る。

バックアップ領域の管理では、空きバックアップが必要になったとき、ある時間間隔でみて最も使用頻度の低いバックアップの内容を追いついて空き領域を作るようにした。

プログラム構造の点からは、タスク管理と入出力装置制御をするOS部と、MUMPSインタープリタ部、グローバル変数管理部、ユーティリティ部に明確に分離し、保守を容易にしている。テーブル・サイズやバックアップリングの調整パラメータは、システム起動時にユーティリティで設定できる。

6. むすび

MUMPSをマイクロコンピュータ用に作ることにより、データベースの会話型利用を安価な装置で実現できた。当面は、これの評価試験と試用を行ない、MUMPSの新しい使い方を検討してゆきたい。

最後に、本MUMPSの開発にあたり、標準MUMPSの仕様について御教示いただいたマシンステム研究所の若井先生、および、多大の御援助と御討論をいただいた日立メデイコの鈴木部長と津守部長、日立ソフトウェア・エンジニアリングの小野副部長、日立製作所の西山専門コンサルタント、そのほか、関係された多くの方々に深く謝意を表す。

7. 参考文献

- 1) J. T. O'Neill 編: MUMPS Language Standard: NBS Handbook 118 (1976-1)
- 2) 若井一朗, 田中豊穂, 嶋房成 訳: 標準マンパス言語マニュアル: コロナ社 (1977-10)
- 3) Intel Corp.: PL/M-80 Programming Manual: Document Number 98-268B (1976)
- 4) R. Bayer, K. Unberaue: Prefix B-Trees: ACM Transactions on Database Systems (1977-3)

第6回日本MUG学術大会 昭和54年9月14日(金)

A. 一般演題

A-3.

Computer Assisted Talkie Slide (CATS)

林 徹^x, 馬場謙介^{xx}, 木村一元^x, 木村園恵^{xx}, 武田和忠^{xxx}

^x 獨協医科大学 総合研究施設

^{xx} 獨協医科大学 第一病理学教室

^{xxx} 三洋電機株式会社 開発研究所

1 はじめに

疾病の診断に於いて、画像情報、音声情報は重要な位置をしめている。電子計算機によって、医学教育のあり方が変えられようとしている。しかし、通常のCAIが画像情報や音声情報を扱い得ないことが、CAIの欠点であった。又、CAIを日本の学生に用いようとする、どうしても文字、特に漢字の問題がつかまとう。そこで、これらの問題を解決すべく、我々は、各々のスライドマウントに音声を記憶させておいて、そのスライドを電子計算機でランダムにアクセス出来る装置CATS (Computer Assisted Talkie Slide)を開発している。その試作1号機が完成したのでこれを紹介する。

2 CATSの主な仕様

試作1号機の主な性能仕様は、収容最大スライド枚数：100枚、写真スライドの大きさ：35mm版、スライドマウントの大きさ：50×50mm、スライド当りの録音時間：30秒、最大アクセス時間：3秒、プロジェクター光源：ハロゲン電球300Wである。試作1号機の外観ならびにその心臓部である録音再生ヘッド部分は、図1、2に示す如くである。

3 CATSと電子計算機との結合

CATS本体と電子計算機との結合方式には、いろいろなものを用意しておくべきであろう。現在、RT-11モニター下では、Digital I/Oを介して結合されている。一般的にMUMPS下で結合する方法については、技術的検討の範囲を出ていないが、MUMPSシステムに手を加えないですむ、最も実現性の高い方法と現在考えている方法を上げる。MUMPSのプログラムで必要なところに、

S %CATSNumber $\times$ D ↑%CATSGenerator

と書く方式である。ここで、 $\times$ は、ユーザーによって指定された(その時投影したい)スライドの番号であり、%CATSGeneratorは、%CATSNumberを受けて、これをCATSが受け入れられる様に変換するCATS code generationサブルーチンである。

これは、最終的に

W * 4, x' , * 5

を発生する。 x' は x に対応するCATSコードである。CATSコードでは4, 5は各々, START, STOPと解釈される。Terminalとの電氣的な結合方式, CATS本体との電氣的結合方式ならびに x' の形に特別な考慮がはらわれていて, 電子計算機側から見ると, CATS本体は, あたかもTerminalに付属するものとして動作する。

4 むすび

試作1号をいろいろな目的で試用してみて, CATSのスライド収容, 音質, 録音時間の経済値をみつけて行く。又, 電子計算機と結合方式についても, field testを行う。近い将来にCAIの一員としてCATSが有効に働くことを期して, CATSの完成をいそいでいる。

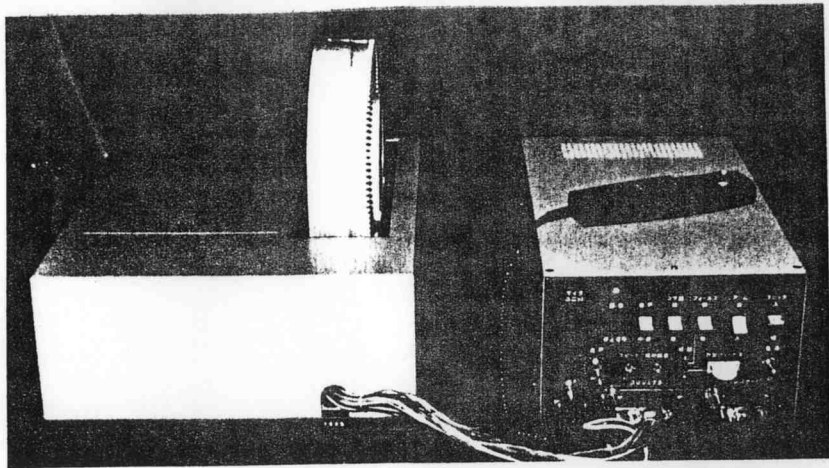


図1 CATSの外観

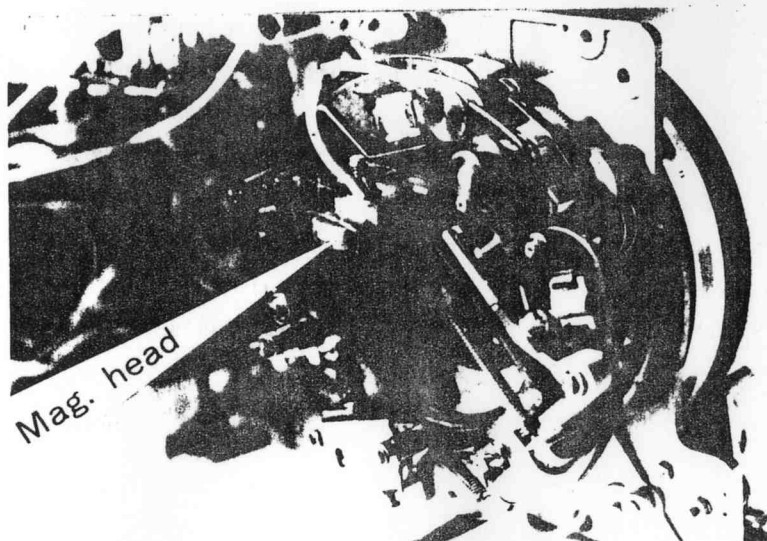


図2 CATSの録音再生部

第6回日本MUG学術大会 昭和54年9月14日(金)

B. パネル 1 MUMPSの速さ 司会 羽曳野病院 今井敏雄

B-1. MUMPSの速さを上げるための技術

日本DEC 笹子 能

1. 速さとは何か?

計算機の速さは、実際に使用する時に本当に速いようにということ念頭に置き、考えられねばなりません。例えば、機械の一命令の実行時間比が倍違っている、Aの計算機はBに比して2倍の命令を必要とするなら、同じ速度と考える必要があります。

例	A		B (PDP-11 etc)
	Load	$R1 \leftarrow A$	
	Add	$R1 \leftarrow R2$	Add $A \leftarrow R2$
	Store	$R1 \rightarrow A$	

一番単純なマシンスピードそのものの速さ、このように評価が難しいのですから、Totalなシステムのスピードの評価は、全体のバランスを考えて行なう必要があります。又、ひとつのシステムのスピードを決定する要因をバランスよく把握しておく必要があります。

2. システムの速さを決定する要因

A ハードウェア

CPU (一命令の速さ及び 命令体系 含 Memory Access)
DISK (Seek Time, Rotational Relay Time, Transfer Time)
割込み及び BUS
Interface
Network Interface
CRT etc 特K Baud Rate & Printer の Speed

ここで Speed 比を一例として考えますと、

1 μ order CPU, 割込み及び BUS

1msec $\times$ 10000 baud の場合の一文字転送

10msec $\times$ Disk (条件のいい時) 1000 baud の一文字転送

このような事から、今ある仕事に 10 秒かかっているとして、CPU を 2 倍の Speed に変えると 5 秒で終わる、という考えが成立しないことがお解りになると思います。

B OS

MUMPS 命令の実行

Data Base の効果的なサポート及び ディスクからの Program Load

入出力処理のためのオーバーヘッド

TSS のスケジュールのためのオーバーヘッド

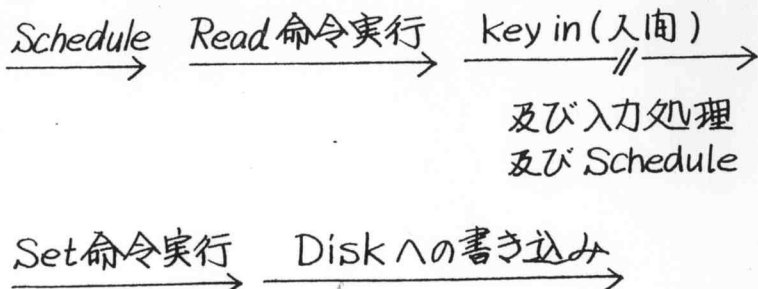
その他の OS 上のオーバーヘッド

ここで各々に要する時間比を考えますと、

Read "TYPE ANYTHING!", X

Set $\uparrow A(X)=1$

上記の実行は、



という概念でとらえられます。

すると、MUMPS の Speed とは、これらの現実的使用条件下での総合力に他ならないことが、おわかり頂けると思います。

3. MUMPSにおける具体的考察

A ハードウェア

特に重要なもの

DiskのSpeed	決定的に重要
CPU ...	MUMPSはCPUにかなりの負荷をかける。
CRT ...	表示が多い時は、高速のものが必要。

今後の方向

- ・数値のString演算(Standardで要求される)のファムウェア化
- ・長さ指定の Long Move 命令
- ・Table処理のファムウェア化
- ・全く新しいタイプの記憶装置
- ・より分散処理を進めるための Network ファムウェア

B ソフトウェア

(i) MUMPS命令の解釈と実行

MUMPSの一命令を実行するには、msecのOrderの時間が必要です。これは、MUMPSの最大の欠点です。普通、MUMPSが遅いというのは、これを指して言います。なぜこのように時間がかかるかというと、まず命令の解釈において次の命令を取り出し、Post conditionを調べ、といったことに幾分かを要し、更けArgumentの解釈に多大の時間がかかります。

例えば、Miniconにおいては、かなり高速の1μsec/命令というマシンスピードで1000命令を実行すれば、既に1/msecを使ってしもうわけです。

仮に、一命令を各命令処理ルーチンに渡す所のみ考えたとしても、下手な事をすれば、


```
MOVE 26 → R0  
MOVE Table → R1
```

```
1$ COMPARE STRING ↔ R1  
   BRANCH  
   EQUAL 2$
```

```
SUBSTRUCT  
ONE AND BRANCH R0, 1$
```

<NOT FOUND>

2\$ <FOUND>

この命令なら、これだけで50マシン命令を使います。
更に、この命令中のどの命令かを調べるのに、もし50マシン
命令が必要とすれば、ただちに100マシン命令を使い
果たすわけですね。

実際には、当然、この部分は Index方式で行っており、
10命令以下で行ないますが、このような例からでも
いかに MUMPS のソフトの改良が重要であるかを、
解ると思います。そして、FORTRAN で書かれた
MUMPS Interpreter などには違いがあるのが当然です。

(iii) Data Base の効率的なサポート及びディスクからの Program Load

この面では、MUMPS はかなり良い効率を上げることが
可能です。

以下に、DSM-II の Data Base について、具体的な例を
挙げてみます。

○ MULTI-WAY TREE

Logical な Tree の表現、↑XYZ (I, J, K) を
physical に Disk 上での Tree として実現するとき
最適化を試みております。それは、Tree の
深さと広がりか 偏らないほど良いという理論
に基づいて行ないます。DSM の特徴は、いか
なる設計をしても、Global Data の Access 時間
には大きな差がないという事です。

資料Aを参照して下さい。

◦ Front key compression

Globalのkey(添字)は、例えば電話帳の氏名のように、多くの場合、前のものとの重複が発生します。この時、その重複部をkeyから取り除くという事です。資料Bを参照下さい。

特に、これはディスクスペースの節約以外に Tree そのものを浅くし、その事によって処理速度をも改善します。

◦ Bufferの管理

全システムレベルで最大32KWのBufferを持つことができます。これらのBufferは、最も参照されなかったものを順次使うという方法で利用すると共に、

MULTI USERのもとでこのBufferの共有のスケジューリングを最低限のオーバーヘッドで行ない、かつ Buffer 待ちのJobが現れなくても、他のJobはその実行を続けます。

又、最後に参照したBufferにあるDataは Full Referenceでも、Pointerをすべてたどる事なく、直接 Buffer から取り出します。

この意味では、Nacked Referenceは、ほとんど不要といえます。

◦ その他、速さを上げる為に、以下のような事もしております。

Write Demon : Dataを変更するたびに Disk に書き込むのではなく、ある程度 Memory 上での変更を貯めてから、Disk へ戻してやる。

例] For I=1:1:10 Set ↑TEST(I)=DATA(I)

この程度は、通常1回(但し、1 Data Block に入るとして)で書きます。

Right Link Pointer: 各物理的レベルは水平方向のPointerを持っており、あるBlockと右側のBlockの間でのDataの移動が可能となっているため、Data追加、変更、削除において、上位のPointerへ影響を最低限にしなから、TreeのBalanceを保つことができます。

又、このPointerは、Garbage Collectionに有効に使用されている他、今後、\$Nextよりも高速のGlobal参照を行なうことも可能です。

Reentrant : Global処理は、複数Userが同時に行なえます。そのため、多くの端末からの要求に、長い待ち行列を作らずに済みます。

Online Garbage Collection:

再度、フリーとなったブロックを正しく、再使用するのがGarbage Collectionです。これをOnline中に実行します。

4. 総合的な評価

MUMPSの処理速度を総合的に評価すると、Data Base中心のOnline業務においては、現在の技術的水準のかなり上位にある。実際にMUMPSを使っている病院とその他の言語を使っている病院を比較して、MUMPSのOnline応答の方が良いことは、ごく普通のことです。

反面、技術計算の複雑なものをMUMPSで行なわせる事は、良い選択とは言えない。但し、これも程度問題で、ごく常識的な統計などは十分行なえる。

MUMPSの処理速度については、しかし、極端KOSKより変化すると考えられる。一般的に、MUMPSが遅いという考えがあるとするれば、それは、幾分かは、いかMUMPSにおいて速度への改善がなされているかの認識不足にもよっていると言える。

5. 今後の方向

今後、様々な方面からの改善が考えられますが、その幾つかを述べてみたいと思います。

1. ハードウェア

VAX-11のような32bitのミニコン。これによって大きなプログラムが可能となり、Compiler形式のMUMPSもOverlay等をせずに動かせるようになる。各種の命令、例えばVAXには\$FINDK似たTEXT編集命令がある。そのようなものの有効利用。

2. ソフトウェア

MULTI-WAY TREEをさらに改善する。具体的には、水平方向Pointerの\$NEXT(ないし別のFunction)への応用。あるいは、Write DemonにおけるDisk Accessの最適化。その他、コンパイラ形(但し、16bitのミニコンには不利)のMUMPSの開発。

ただ、現在においては、単にMUMPSとFORTRANやCOBOLがDataを共有できるというのは、MUMPSを軽視した場合に良い方向と思われるが、DECの最近の実験の結果ではMUMPSが極端に遅くなるという欠点があります。今のハードでInterpreter形でも、なお地味に改善してゆくだけで、まだ2倍位は速くなると考えられます。



第6回日本MUG学術大会 昭和54年9月14日(金)

B. ハネル(1) MUMPSの速さ 司会 羽曳野病院 今井敏雄

B-2. MUMPS ベンチマークテスト

大阪府立羽曳野病院 今井敏雄

1. はじめに

電子計算機システムの性能を知る手軽な方法は、適当なテストプログラムを走らせて、処理時間などを測定、比較検討することである。計算機システムの処理速度を調べる為の標準的なテストプログラムのことを、ベンチマーク(benchmark)プログラムと言う。ベンチマークプログラムとしては、計算機の標準的な使われ方をよく反映したものが良いが、一般的に、何が標準的であるかが難しい。一般に、ベンチマークテストは、システムの性能の幾つかの側面を計るプログラム群から成っているが、テストではシステムの重要な特性を見落とさないようにする必要があり、プログラムの選定が難しい。今回 MUMPS のベンチマークテストプログラムを作成するにあたって、当初は現在の3倍程度の項目、量があり、順次、データ値に相関のあるものからぜい肉を取り、エッセンスを抽出してきたが、それでもまだボリュームが大きく、処理時間にして1日程度を要する。

CPUバウンドのローカル処理は、個々の演算機能、関数機能に処理上の性能差はあるにしても、主にCPUのハードウェアの性能によって影響を受ける部分が大きく、比較的小数のベンチマークテストで、性能は把握できる。問題は、グローバルデータの処理性能テストである。勿論、CPU、ディスク装置のハードウェアの性能にもよるが、むしろ MUMPS OS ソフトウェア、特にグローバルの管理方式、処理速度を上げるための種々の技法や、ユーザプログラム上でのグローバル処理テクニックの上手下手によって、システムの性能、特性が大きく異なる。ユーザの利用技術の雑拙による性能が大きく左右される環境下でのベンチマークでは、ある一定量の処理を、そのシステムのベストのコンディションでどれだけの時間でこなせるのかというテストの他に、そのシステムの特性を考慮して最良の形態でユーザがシステムを治用した場合と比較して、ユーザの使い勝手は種々のバリエーションで使った場合、システムがどういった特性で性能の低下を示すか、言い換えれば、システムが安定な性能を保証するユーザの利用形態の範囲の大きさはどの程度であるかを知るテストプログラムが必要である。特に MUMPS のグローバルファイルについては、近年、B-tree技法応用のファイル編成が導入されており、ユーザの使い勝手は飛躍的に拡大されつつある。ベンチ

マークでも こうした特性を捕える必要がある。

ベンチマークで 特に難しいのは、マルチジョブ環境でのテストである。ジョブの性質、多量度、スケジューリング等の諸条件がからみ合い、いろいろ条件を変えてみる場合でも 再現性が問題となる。今回のベンチマークテストでは、ユニフォームな簡単なジョブのみに絞っている。TSS で ディスクへのアクセスが 頻繁に行われる MUMPS システムでは、総合的でかつ有用なベンチマークテストを作ることは 極めて 難しいというのが 実感である。

2. ベンチマークテストの概要

今回 作成したベンチマークテストの概要を以下に示す。プログラム本数としては 38本であり、各々の項目の 内訳を () 内に明示する。

A ローカル処理 単機能 (3)

(a) ローカル変数の Set, Read, \$D, \$N

(b) オペレーション機能

+, -, *, /, #, \, -, ', &, !, >, <, =,
[,], ?

(c) ファンクション機能

\$A, \$C, \$E, \$J, \$L, \$P, \$S, \$T

B ローカル処理 総合 (5)

(a) 数値データのソーティング

(b) 文字データの辞書順ソーティング

(c) 平均値, 分散の計算

(d) e の高精度 (200 ケタ) 計算

(e) π の高精度 (100 ケタ) 計算

C グローバル処理 単機能 (7)

グローバルのレベル数, 添字設計, データ文字長の種々の組合せ構成のファイルに対して, 順 Set, 横順 Read, 縦順 Read, とび順 Read, \$D, \$N と 各 naked syntax 処理を実行し, グローバルデータの処理特性を調べる。

(a) 0レベルの複数グローバルへのアクセス

(b) 1レベル, 1グローバルへのアクセス

(c) 1レベル, 2グローバルへの交互アクセス

(d) 4種の添字設計, 1グローバルへのアクセス

(e) 4種の添字設計, 2グローバルへの交互アクセス

(f) 大量データ(1000, 5000, 10000レコード)へのアクセス

D グローバル処理 総合1 マスターファイル, 検索ファイルへのアクセス(6)
マスターファイルとマスターファイルへのエントリキーを検索する
外部索引ファイル2種を作成し, 種々のマスターファイルへのアクセス
方法について 処理時間を測定する。

E グローバル処理 総合2 グローバルデータのマージ処理(4)
メインファイルとサブファイルを作成し, メインファイルにサブファイルの
データレコードとマッシュするものがあれば, データをマージ, アップ デイトし,
マッシュするものがなければ, データをメインファイルにコピー生成する。
生成, マージ処理時間を測定する。

F マルチジョブ (11)
ユニフォーム な幾つかのグローバルアクセスジョブを同時に走らせて
全体の処理時間を測定する。

その他, 乱数発生時間推定用のユーティリティー本, ベンチマークテストプログラム
自動スタートプログラム一本がある。

A, Bのテストプログラムは ローカル処理のベンチマークである。Cのグローバル
単機能テストでは, グローバルの種々の使い方に対してシステムが どのような性能
を示すかを測定するテストである。特に, MUMPSでは, ディスクバッファリングの技術
が巧みに利用され, 高効率なディスクI/O処理を得ているが, そうしてにバッファリング
の効果や, 或いは 逆に バッファリング効果が 全く期待できない環境でのシステム
の性能を測る目的のテストを用意している。Cについては, いくつかの条件やパラメ
ータを固定として, 処理も単純な内容で, テストの再現性を重視している。

他方, D, Eについては, 実際のMUMPS システムでよく利用される処理形態を
モデル化し, 複雑な処理内容下での総合的な性能を測定しようとする目的のプロ
グラムである。処理内容, 処理量を決定するいくつかのパラメータが 外部より指定す
ることが可能であり, パラメータ値の組合せによって, かなりの範囲のシステム環境を
作り出すことができる。D, Eのプログラムは, ベンチマークテストと言うより,

Synthetic プログラムに近い内容である。特に MUMPSのグローバル処理を
含めた複雑な処理では, トータルの処理時間だけでは 情報として不十分であり,
D, Eのテストプログラムでは, ローカル処理とディスク read, write 処理(と
乱数発生処理)に要した時間と比率も 解析出力するよう工夫が なされている。
それでも おおむね情報不足という感じで, ある条件, ある環境下での総合テストの結果に
ついては云々できても, 条件がちょっと変更された環境での性能について, テスト

結果がどこまで敷衍できるかが疑問である。総合テストの性能は、実際的な経験や、多くの実験によって検定され、より高感度かつ高効率なテストプログラムを作っていく必要がある。マルチジョブでのテストプログラムでは、今回は、全く複雑さへの挑戦は辞め、テスト結果の解読が容易な最も簡単な同一形式のグローバルアクセスジョブを扱っている。テストA~Fでデータの生成のために乱数を用いているが、各メーカーの\$RANDOM関数は特性を異にしている。どのシステムでも同一データを生成するよう、プログラムの中で疑似乱数を発生させている。乱数の発生法は、混合型合同法を用いており、有効ケタ数は6ケタである。今回作成したプログラムで欠落しており、テスト仕様の設定の難しいものに入出力機能のテストがある。

テストCの(b)のプログラム例を図1に示す。
測定データは次のような形で出力される。

```
ORDERED READ
COMMAND ==> F I=1:1:Y F J=1:1:L1 S A="A(J)
TIME = 3 3.00 +- 1.00 M SEC/SINGLE DATA NODE

NAKED READ
COMMAND ==> F I=1:1:Y S A="A(1) F J=2:1:L1 S A="A(J)
TIME = 3 3.00 +- 1.00 M SEC/SINGLE DATA NODE

INVERSED READ
COMMAND ==> F I=1:1:Y F J=L1:-1:1 S A="A(J)
TIME = 3 3.00 +- 1.00 M SEC/SINGLE DATA NODE

INVERSED NAKED READ
COMMAND ==> F I=1:1:Y S A="A(L1) F J=L1-1:-1:1 S A="A(J)
TIME = 3 3.00 +- 1.00 M SEC/SINGLE DATA NODE

SPARSE READ (INTERVAL=10)
COMMAND ==> F I=1:1:Y F J=0:1:9 F K=1:10:L1-9 S A="A(J+K)
TIME = 5 5.00 +- 1.00 M SEC/SINGLE DATA NODE
```

```

B21 ; BENCHMARK TEST PROGRAM 10
B21 ; GLOBAL [2] --- ONE LEVEL AND ONE GLOBAL
; BY T.IMAI 78/11/21, 79/3/13
;
START K W !!,"*** GLOBAL [2] --- ONE LEVEL AND ONE GLOBAL ***",!
W !,"USED WORK GLOBAL ; AA",!
I $D(ABAUT) G ST
R !,"WAIT OR NOT WAIT TO COLLECT GARBEGES AFTER GLOBAL KILL ? YES(Y) OR
K SEC I X="N" S SEC="0"
I '$D(SEC) R !," EX, 2*TIM (TIM;TIME OF GLOBAL SET) : ",SEC
S TIM=0 H @SEC
R !," OK ? YES(Y) OR NO(N) : ",X,! I X!="Y" G EXIT
ST I '$D(SEC) S SEC="TIM" S;$D(ABAUT(1)) SEC=ABAUT(1) ; DEFAULT
S X="1234567890"
S Y=80,L1=10 D ACC
S Y=8,L1=100 D ACC
S Y=2,L1=500 D ACC
S Y=1,L1=1000 D ACC
G KILL
;
KILL K AA H @SEC
EXIT Q
;
;
ACC S:$D(TIM) TIM=0 S N=-1
A S N=N+1,LAB="COM+"_N,CON="CON=$T("_LAB_)",@CON
I CON["END" Q
I CON[" Q" D @LAB G A
I $P(CON," ",2,100)=";" G A
W !," COMMAND ==> ",$P(CON," ",2,100)
S T=$P($H," ",2)
D @LAB
S T=$P($H," ",2)=T W !," TIME = ",T
W ?45,$J(T*1000/LOP,7,2)," +",$J(1000/LOP,6,2)," M SEC/SINGLE DATA NODE"
I CON[" S ^" S TIM=T S;TIM<5 TIM=5
G A
;
;
COM W !!,"*** ONE GLOBAL SET AND READ ***",! Q
W !," X = ",X," $L(X) = ",$L(X) Q
W !," Y = ",Y," L1 = ",L1 Q
;
K AA H @SEC W !!," ORDERED SET (NOT EXIST)" Q
S LOP=L1 Q
F J=1;1;L1 S ^A(J)=X
Q
;
W !!," ORDERED SET (EXIST)" Q
10 S LOP=Y*L1 Q
F I=1;1;Y F J=1;1;L1 S ^A(J)=X
Q
;
W !!," ORDERED READ" Q
F I=1;1;Y F J=1;1;L1 S A=^A(J)
Q
;
W !!," NAKED READ" Q
20 F I=1;1;Y S A=^A(1) F J=2;1;L1 S A=^A(J)
Q
;
W !!," INVERSED READ" Q
F I=1;1;Y F J=L1;-1;1 S A=^A(J)
Q
;
W !!," INVERSED NAKED READ" Q
F I=1;1;Y S A=^A(L1) F J=L1-1;-1;1 S A=^A(J)
Q
;

```

次ページへ続く

```

30  I L1'>10 S N=N+4 Q
    Q
    W !!," SPARSED READ (INTERVAL=10)" Q
    F I=1;1;Y F J=0;1;9 F K=1;10;L1-9 S A=AA(J+K)
    Q
    ;
    I L1'>10 S N=N+4 Q
    Q
    W !!," SPARSED NAKED READ (INTERVAL=10)" Q
    F I=1;1;Y S A=AA(1) F J=0;1;9 F K=1;10;L1-9 S A=AA(J+K)
40  Q
    ;
    I L1'>100 S N=N+4 Q
    Q
    W !!," SPARSED READ (INTERVAL=100)" Q
    F I=1;1;Y F J=0;1;99 F K=1;100;L1-99 S A=AA(J+K)
    Q
    ;
    I L1'>100 S N=N+4 Q
    Q
50  W !!," SPARSED NAKED READ (INTERVAL=100)" Q
    F I=1;1;Y S A=AA(1) F J=0;1;99 F K=1;100;L1-99 S A=AA(J+K)
    Q
    ;
    W !!," $N" Q
    F I=1;1;Y F J=1;1;L1 S A=$N(AA(J))
    Q
    ;
    W !!," NAKED $N" Q
    F I=1;1;Y S A=AA(1) F J=1;1;L1 S A=$N(AA(J))
60  Q
    ;
    W !!," $D (EXIST)" Q
    F I=1;1;Y F J=1;1;L1 S A=$D(AA(J))
    Q
    ;
    W !!," NAKED $D (EXIST)" Q
    F I=1;1;Y S A=AA(1) F J=1;1;L1 S A=$D(AA(J))
    Q
    ;
70  W !!," $D (NOT EXIST)" Q
    F I=1;1;Y F J=L1+1;1;2*L1 S A=$D(AA(J))
    Q
    ;
    W !!," NAKED $D (NOT EXIST)" Q
    F I=1;1;Y S A=AA(1) F J=L1+1;1;2*L1 S A=$D(AA(J))
    Q
    ;
    ;
    K AA H @SEC W !!," INVERSED SET (NOT EXIST)" Q
80  S LOP=L1 Q
    F J=L1;-1;1 S AA(J)=X
    Q
    ;
    K AA H @SEC W !!," NAKED SET (NOT EXIST)" Q
    S AA(1)=X F J=2;1;L1 S AA(J)=X
    Q
    ;
    I L1'>10 S N=N+4 Q
    Q
90  K AA H @SEC W !!," SPARSED SET (INTERVAL=10) (NOT EXIST)" Q
    F J=0;1;9 F K=1;10;L1-9 S AA(J+K)=X
    Q
    ;
    I L1'>100 S N=N+4 Q
    Q
    K AA H @SEC W !!," SPARSED SET (INTERVAL=100) (NOT EXIST)" Q
    F J=0;1;99 F K=1;100;L1-99 S AA(J+K)=X
    Q
    ;
100  END

```

図1 リストCの(b) (グローバル単機能1レベル/グローバルへのアクセス)のプログラム

3. テスト例

いくつかのシステムでベンチマークテストを実行した結果の一部を紹介する。
システム構成を表1に示す。

	A	B	C	D	E	F
メモリー サイクルタイム	600ns	980ns	300ns 980ns	980ns	300ns 980ns	800ns 700ns
ディスク装置 平均アクセスタイム	42.5ms	70ms	70ms	70ms	70ms	57.5ms
ディスクバッファサイズ	7kW	8.5kW	8.5kW	9kW	9kW	4.5kW
グローバル編成法 {伝統的グローバル編成 B-tree技法応用編成}	伝統	伝統	伝統	B-tree	B-tree	B-tree

表1 システム構成

システム構成については、少なくとも以下のような項目を把握しておく必要がある。

ハードウェア---CPU, メモリー容量, メモリーサイクルタイム, ディスク装置,
容量, トラック数, セクター数/トラック, 平均アクセスタイム,
アーム移動時間, 1/2回転時間, 浮動小数点演算オプション,
他(上位, 下位)機種で動くか。

ソフトウェア---OSバージョン, 専用OSか汎用OSか, OSサイズ, パーティションサイズ
とそのオーバーヘッド, パーティションサイズの最大値,
グローバル編成法, ブロックサイズ, ディスクバッファサイズとその
最大値, MUMPS 言語仕様(標準か方言か)

テスト結果は、すべてハードウェア, ソフトウェアの測定環境の条件を十分考慮
して解釈する必要がある。

ローカル処理単機能テストAの結果をくもの巣グラフにまとめたものを図2に
示す。ハードウェア特にCPUの性能によりパターンは相似的に大きく或いは
小さく変形するが、パターン自身の相対的な形状はMUMPSシステムの特徴を
示していると言える。

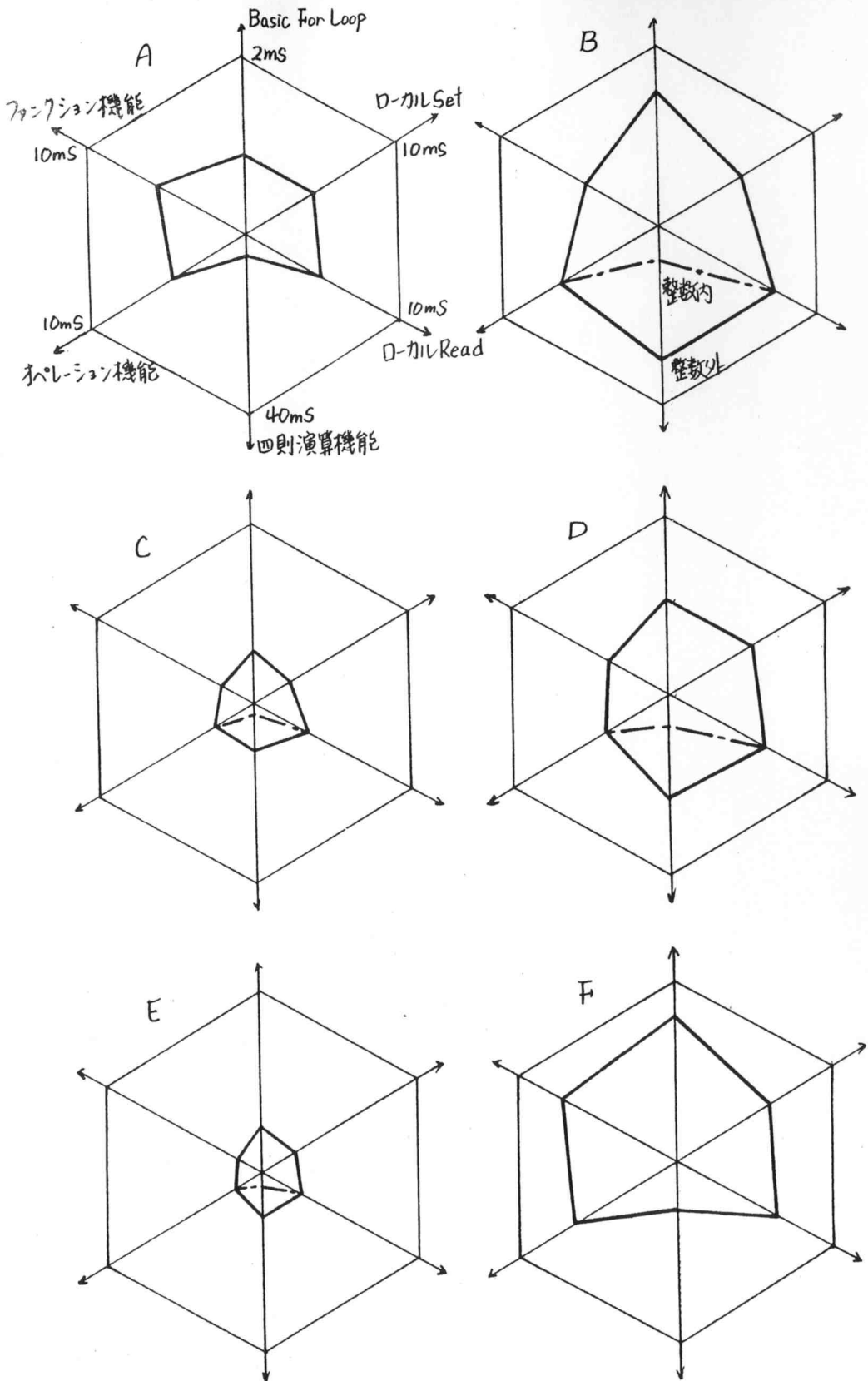


図2 D-カル処理 単機能 (テストA)

図3に 数値データのソーティングテストの例を示す。ソーティングアルゴリズムは最も単純な bubble sort と言われる方法で データ数の2乗のオーダーの処理時間を要する。

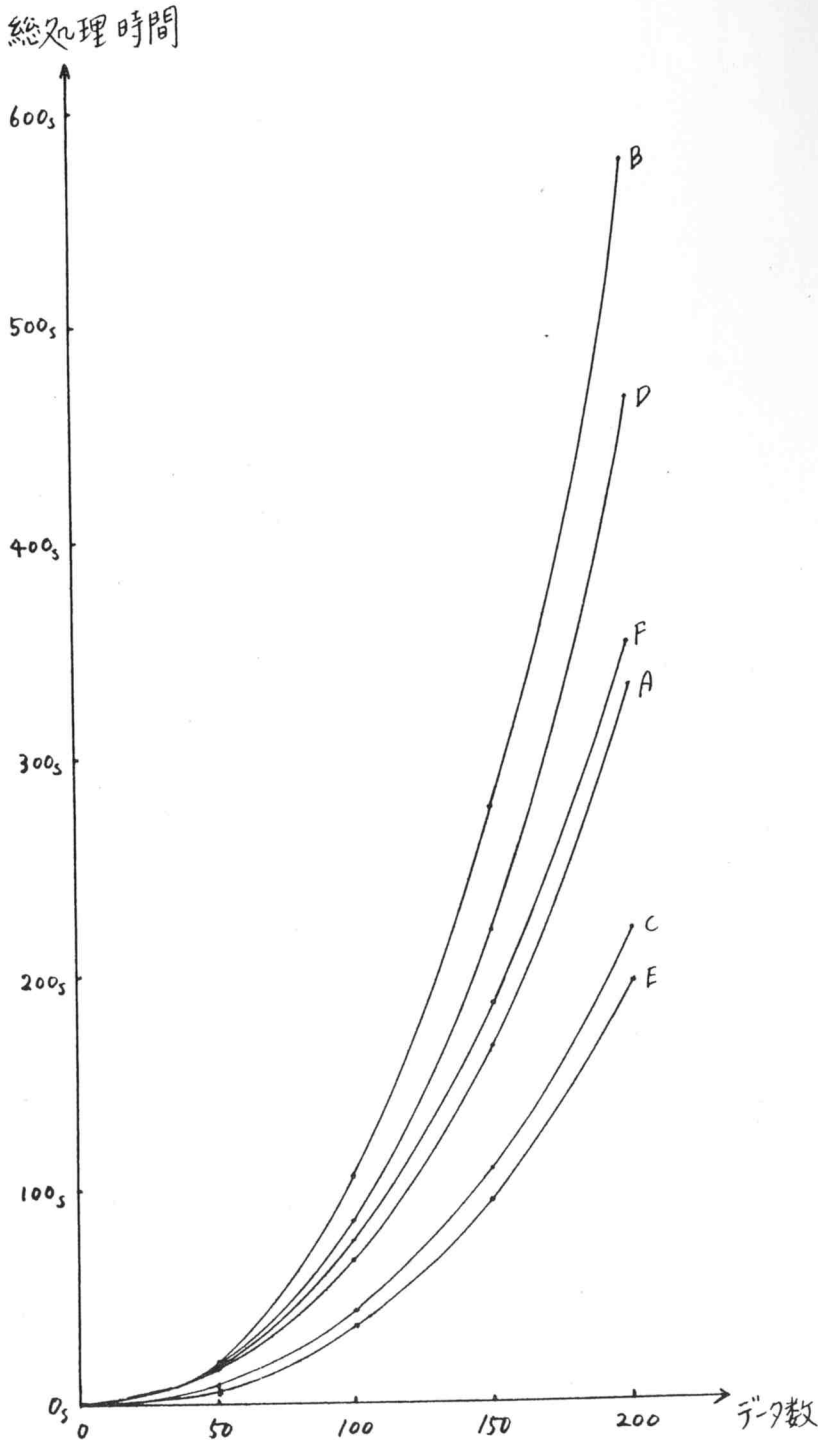


図3 数値データのソーティング (テストBの(a))

図4に 円周率 π の高精度計算の例を示す。プログラムでは 有効ケタ数を得るために 100 進計算を行っており、整数計算、モジュロを多用している。

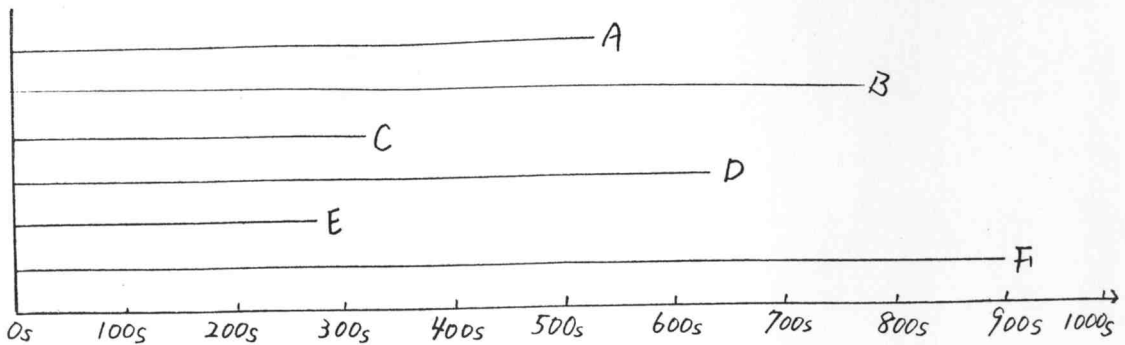


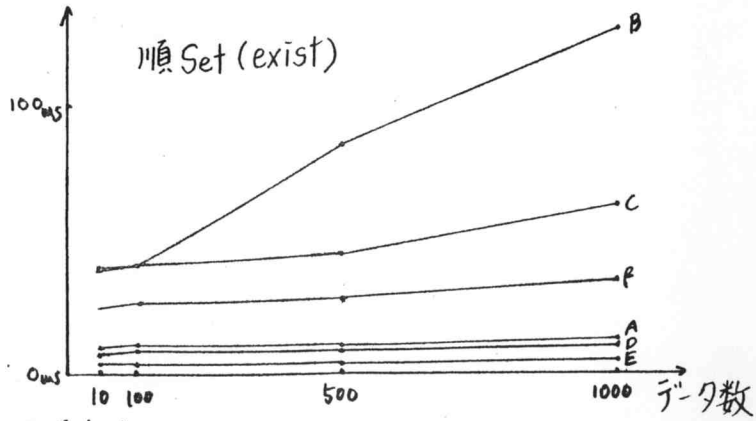
図4 円周率 π の高精度(100ケタ)計算(テストBの(2))

$$\pi \text{ の近似式: } \pi = 16 \left(\frac{1}{5} - \frac{1}{3} \cdot \frac{1}{5^3} + \frac{1}{5} \cdot \frac{1}{5^5} - \dots \right) + 4 \left(-\frac{1}{239} + \frac{1}{3} \cdot \frac{1}{239^3} - \frac{1}{5} \cdot \frac{1}{239^5} + \dots \right)$$

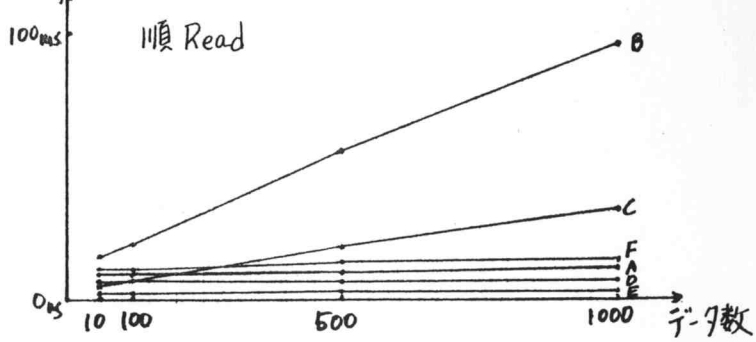
図5に グローバル処理単機能Cの(b) 1レベル設計, 1グローバルデータへのアクセステストの結果の一部を示す。縦軸処理時間は、1データノード当りに正規化している。図より 総処理時間が、データ数の近似的に1乗に近いオーダを示すシステムと2乗のオーダを示すシステムがあり、特性の差は 明白である。B-tree技法応用のグローバル編成採用のシステムは、1次のオーダに近い特性であり、使い勝手の良さを示している。

図6に テストCの(c)の結果の一部を示す。同テストは、図5に示したCの(b)のテストで 1グローバルデータへのアクセスを2グローバルデータへの交互アクセスに代えたもので、*naked syntax* 処理を不能にすると同時に、ディスクバッファに2倍の使用度を与えることにより、ヘビーなグローバルアクセスに対して、ディスクバッファの効果が どの程度 期待できるかを計るテストである。あるシステムでは、全くのディスクバウンドとなっており、またあるシステムでは、こうしたヘビーな環境でも、一樣な特性を示している。

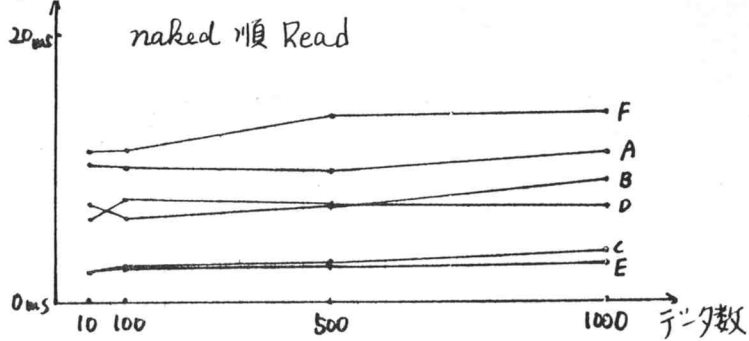
処理時間/1データノード



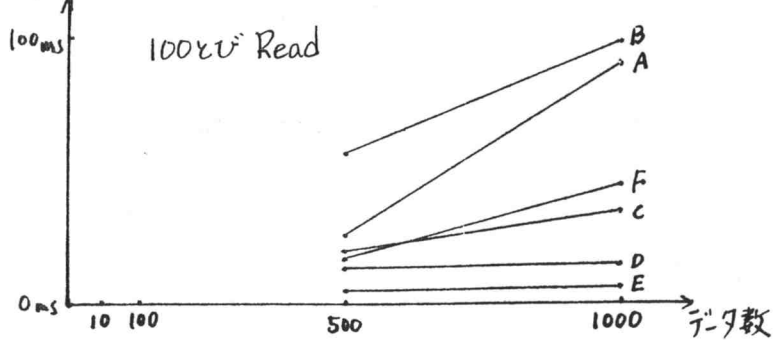
処理時間/1データノード



処理時間/1データノード

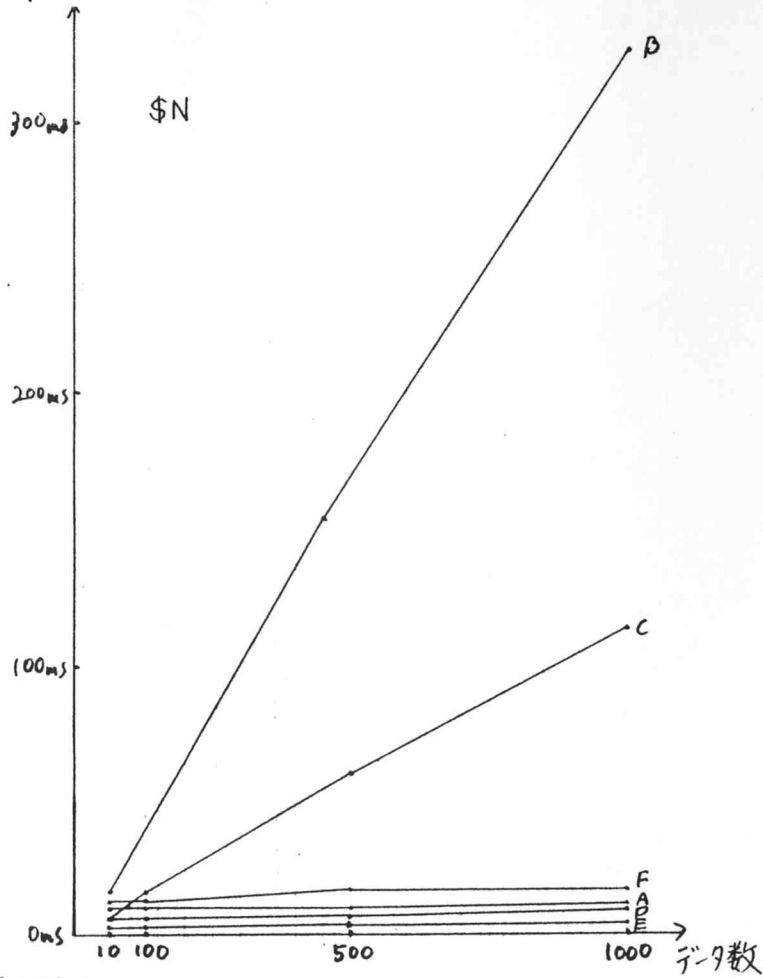


処理時間/1データノード



次ページへ続く

処理時間/1データワード



処理時間/1データワード

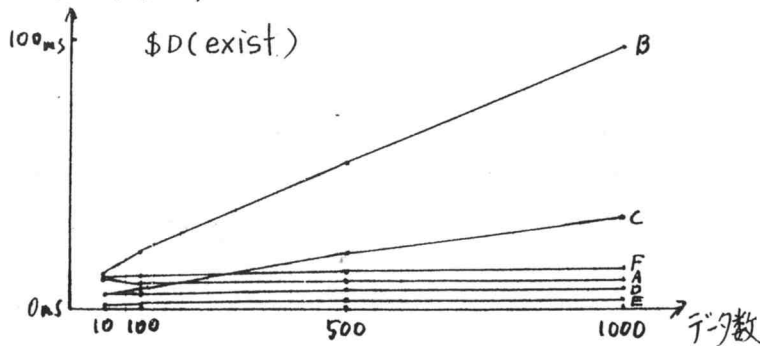


図5 1レベル設計, 1グローバルデータへのアクセス例(テスト(a)(b))
データ文字長=10

処理時間 / 1グローバル / 1データノード

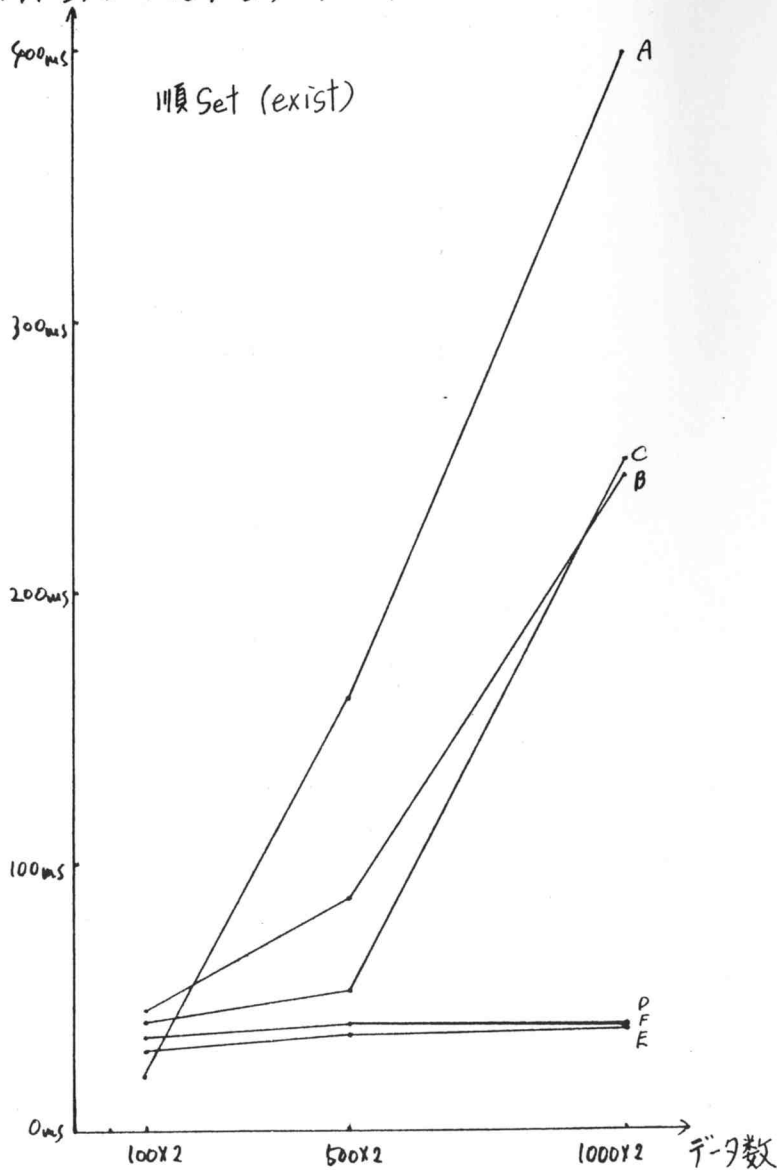


図6 1レベル設計, 2グローバルデータへの交互アクセス例(テストCのC)

テストD, Eは 一応総合テストとして企画したもので, どこまで総合的(?)か疑問であるが, テストDについて紹介する。テストDは マスターファイル(^MAS)とそのエントリーキーである sequence No. をデータとして持つ2種の 検索ファイル(^RF1, ^RF2)を作成し, 種々のマスターファイルへのアクセス方法について処理時間を測定する。ガイドンスを図7に示す。

> B30 ↓

*** MASTER AND REFERENCE FILE (^MAS, ^RF1, ^RF2) ACCESS PROCESS GUIDANCE ***

MASTER FILE : ^MAS(SEQ NO,0)=ID NO;NAME;SEX;BIRTHDAY;ADDRESS;COMMENT
^MAS(SEQ NO,ITEM NO)=DATA FOR ITEM NO

WHERE \$L(ID NO)=6,\$L(NAME)=8,\$L(SEX)=1,\$L(BIRTHDAY)=9,\$L(ADDRESS)=5,\$L
EX. ^MAS(1,0)=000102;ABCD EFG;M;S59.01.01;12345;ABCDEFGH I J (COMMENT)=10
^MAS(1,4)=.000102
^MAS(1,5)=.010303

REFERENCE FILE 1 :

^RF1(ID NO)=SEQ NO/SEQ NO/...SEQ NO/
EX. ^RF1(102)=1/
^RF1(114)=93/

REFERENCE FILE 2 :

^PF2(ITEM NO)=MAXMUM COUNTER
^PF2(ITEM NO,COUNTER)=SEQ NO
EX. ^RF2(1)=20
^RF2(1,1)=5
^RF2(2)=20
^RF2(2,1)=5
^RF2(2,2)=14
^RF2(2,3)=19

OPERATION :

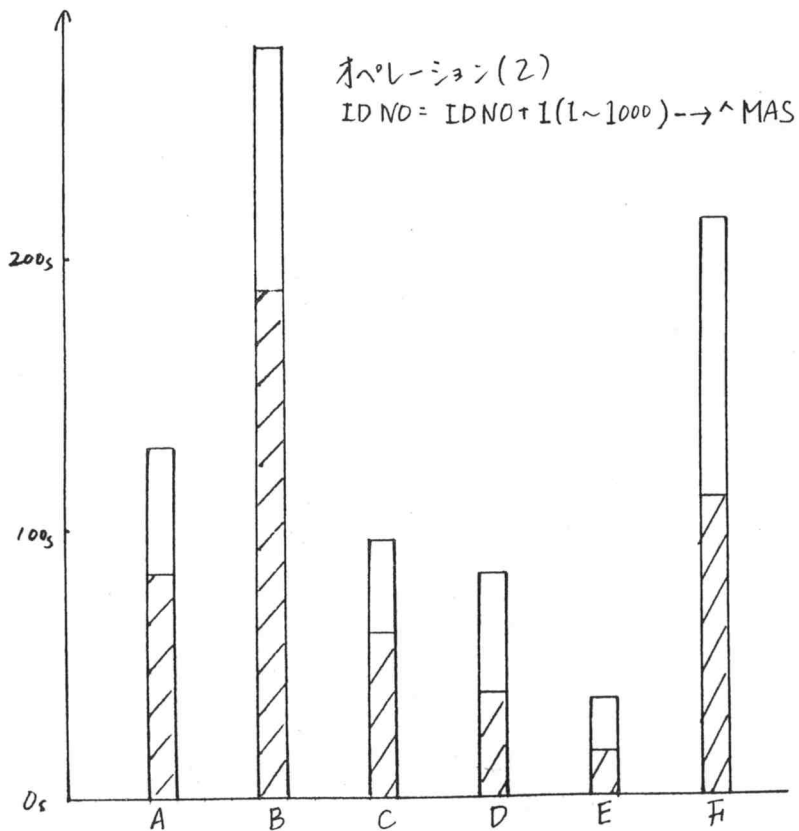
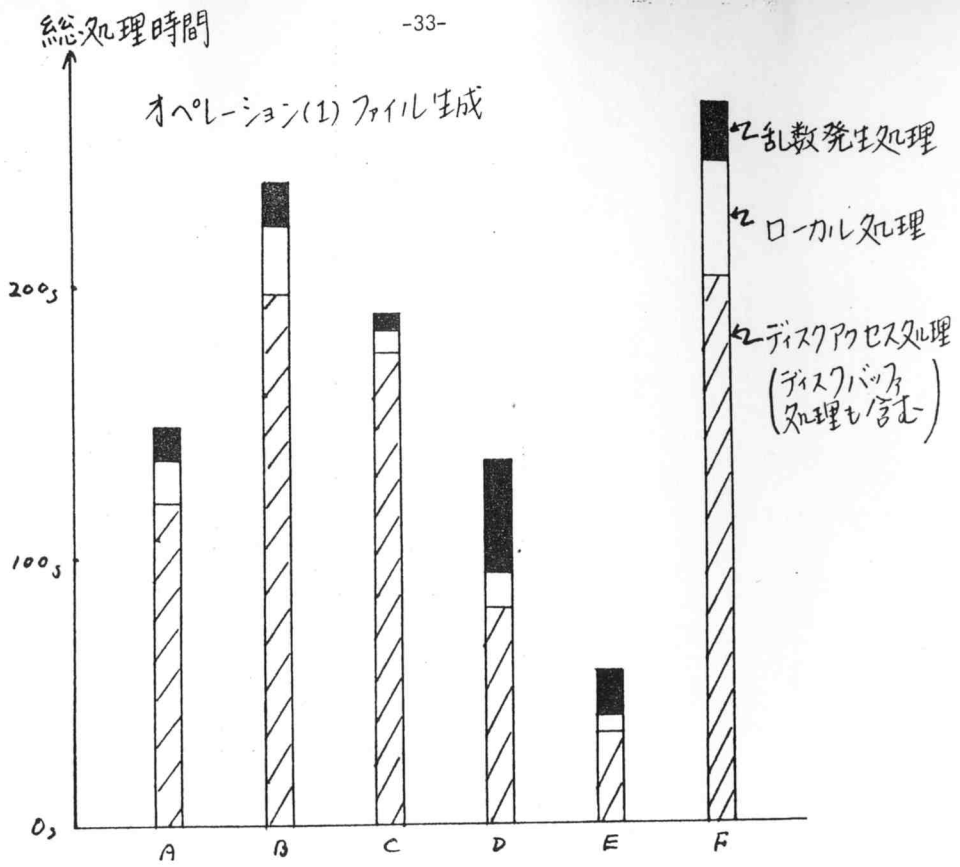
- (1) MASTER AND REFERENCE FILE SET (^MAS, ^RF1, ^RF2 SET)
====> PROGRAM B31
- (2) MASTER FILE DATA SEARCH FROM SEQUENTIAL ID NO (ID NO = ID NO +1 --> ^MAS)
====> PROGRAM B32
- (3) MASTER FILE DATA READ IN ORDER OF SEQ NO (\$N OF ^MAS)
====> PROGRAM B33
- (4) MASTER FILE DATA READ IN ORDER OF ID NO (\$N OF RF1 --> ^MAS)
====> PROGRAM B34
- (5) MASTER FILE DATA READ IN ORDER OF ITEM NO (\$N OF ^RF2 --> ^MAS)
====> PROGRAM B35

PARAMETER :

- (A) TOTAL NO OF MASTER FILE DATA
- (B) TOTAL NO OF ITEMS
- (C) NO OF ITEMS FOR SINGLE MASTER FILE DATA
- (D) MAXMUM LENGTH OF CHARACTER STRING DATA
- (E) RANGE OF ID NO OR SEQ NO TO BE READ

図7 テストD ガイダンス

テストDのモデルとしては、病歴ファイルや検査データファイルを想定しており、Itemとして検査項目などを考えてもらえば分かりやすい。検索ファイル^RF1は患者ID NOからの検索を助けるファイルであり、^RF2は各Item単位にデータをアクセスするための検索ファイルである。マスターファイルへのアクセス方法としては図7のオペレーションに示す(2)~(5)の4種の方法がプログラムされている。システム環境を定めるパラメータとして(A)~(E)が指定可能である。^MAS, ^RF1, ^RF2のファイル構造は論理構造に考えに設計であり、物理的構造を考える必要のあるシステムでは、同設計がほぼ妥当な範囲でパラメータの設定が必要である。



次ページへ続く

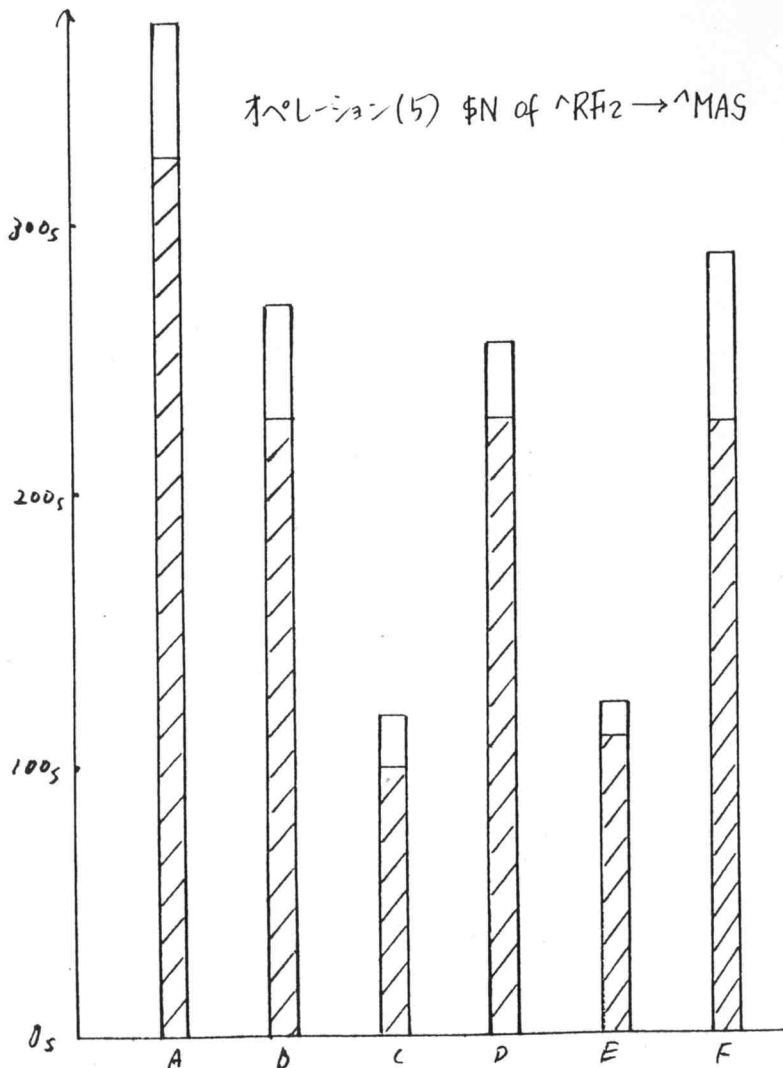
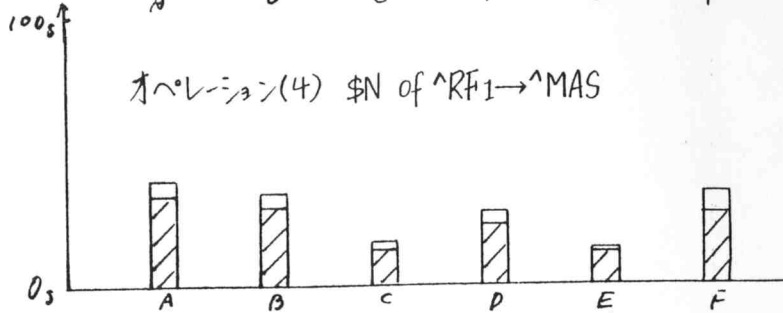
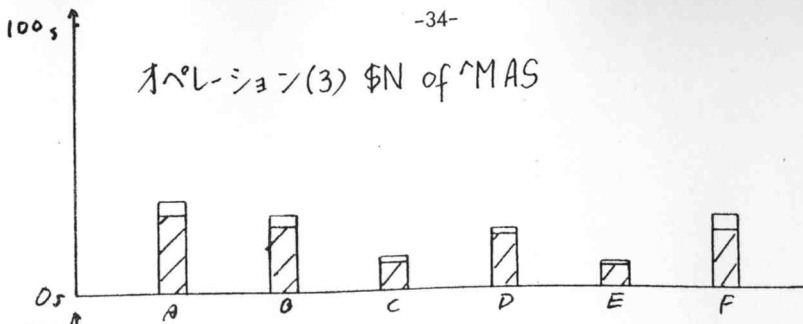


図8 マスターファイル, 検索ファイルアクセス例 (テストD)
パラメータ: (A)=100, (B)=50, (C)=10, (D)=130

図8に テスト例を示す。テストの結果について より突っ込んだ解釈が困難な
ところに 総合テストの難しさがある。

マルチジョブ環境でのベンチマークとして 同一構造1レベル設計の別グローバルを
readするいくつかのジョブを同時に 走らせ、処理時間を測定するテストを行っ
た(テストF)。ガイダンスとスタートプログラムを 図9に示す。

11 B50 ↓

```
*** MULTI-JOB L1J --- GUIDANCE AND GLOBAL GENERATION ***
GLOBAL ACCESS JOB ROUTINE NAME = AAA,BBB,CCC,DDD,...,JJJ
JOBS ARE STARTED BY DIRECT COMMAND. EX. >ZJ AAA,BBB,...
JOBS STOP WHEN PRM(10) STOP FLAG IS DEFINED.
JOB PROCESS = EX. AAA : F I=1:1;Y F J=1:1;L1 S AAA=AAA(I)
Y = PARAMETER SET BY 'S Y=PRM(1)'
L1 = PARAMETER SET BY 'S L1=PRM(2)'
LENGTH OF DATA = PARAMETER
OUTPUT DEVICE = LINE PRINTER
NO. OF JOB BEING RUN SIMULTANEOUSLY = CONTROLLED BY START DIRECT COMMAND (=10)
USED WORK GLOBAL :
AAA,BBB,CCC,DDD,...,JJJ
PRM -- PARAMETER FILE
GLOBAL GENERATOR (1), ALL GLOBAL KILL (2) OR PRM(10) KILL (3) : ↓
```

12 AAA ↓

```
1 ↓
AAA ; BENCHMARK TEST AAA
; MULTI-JOB L1J
; BY T. IMAI 78/11/28
;
START S Y=PRM(1),L1=PRM(2),CT=0
S ST=FP($H,"",2)
ACC S CT=CT+1,T=FP($H,"",2)
F I=1:1;Y F J=1:1;L1 S AAA=AAA(J)
S T=FP($H,"",2)-T
S TT=FP($H,"",2)-ST
O 3 U 3 W CT,T5,"AAA",T15,TT,T25,T,! C 3
I $H(PRM(10)) G EXIT
G ACC
;
EXIT H
```

図9 マルチジョブテストガイダンスとプログラム

ジョブのメインは

$F J = 1:1: L1 S \dots = \wedge \dots (J)$

のグローバル順 read である。L1 = 200, Y = 4, データ文字長10の場合の
テスト例を図10に示す。

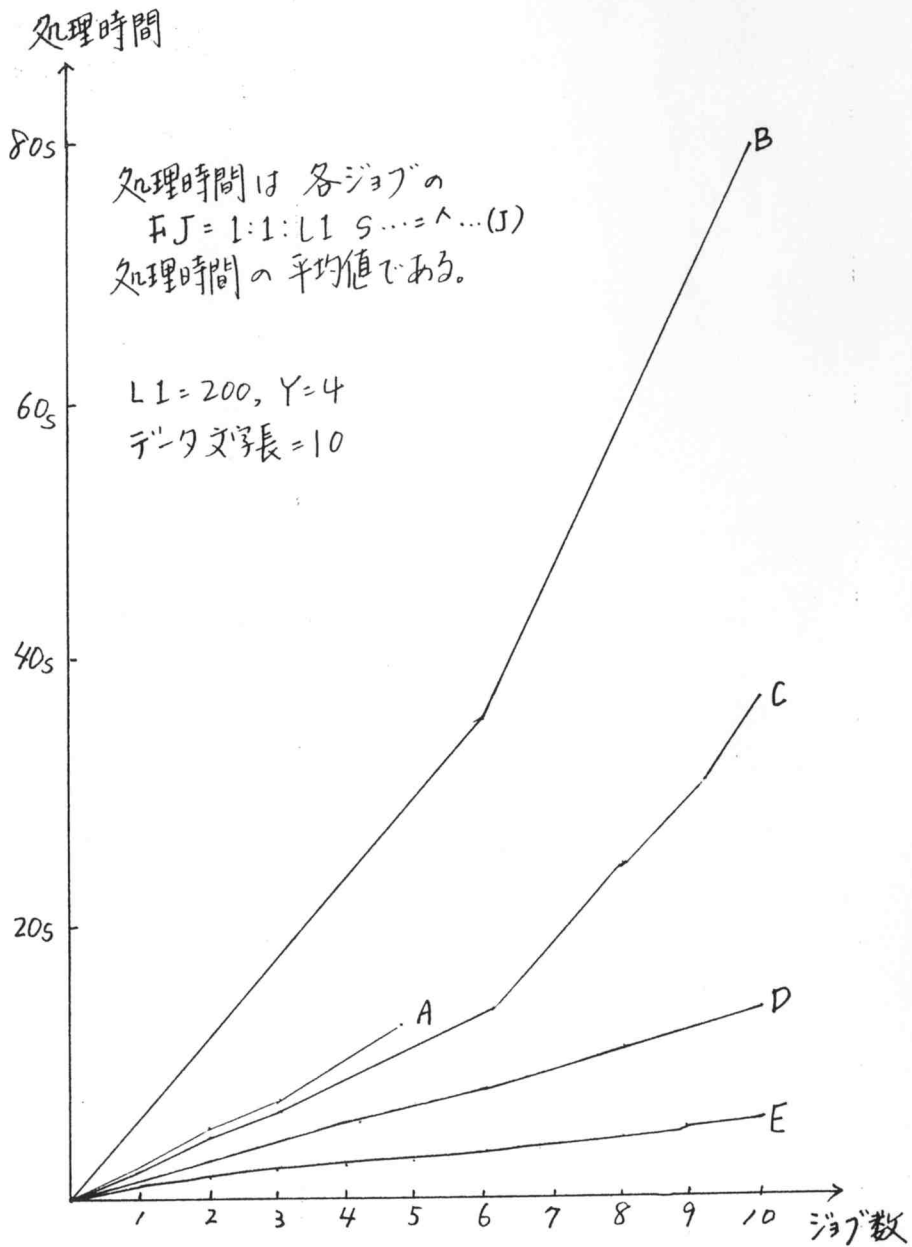


図10 マッチジョブテスト例

図10で 処理時間がジョブ数に比例している領域では、ディスクバッファが有効に働き、主に CPUバウンドの処理と考えられる。

4. おわりに

以上 ベンチマークテストの概要とテスト例を示したが、不十分、不完全であり、テストプログラム作成の第一歩であると思う。MUMPSのベンチマークテストを作成していて、常に脳裏に飛来するのは「群盲象を評す」の感である。あきりしたMUMPSシステムの全体像が見えてくるまでに、今後 更なるプログラムの改良、テストの繰り返しが 必要であろう。



第6回日本MUG学術大会 昭和54年9月14日(金)

B.ハネル I MUMPSの速さ 司会 羽曳野病院 今井敏雄

B-3. MUMPSと他言語との実行速度比較

大阪府立成人病センター 情報企画室

藤江 昭 永井 利廣 服部 敏夫

1. はじめに

MUMPSの速度と他言語と比較するにあたっては、その各々の特徴をよく考慮した上で行う必要がある。ここでは、MUMPSを、代入や数値演算を行う一般のデータ操作機能と、データベースマネージメントシステム(DBMS)としての機能に分離して、BASIC、PL/I、DL/Iとの比較を行う。

MUMPSとBASICは、実行形式は異なるがインタープリタ言語であり、PL/Iはコンパイラ言語である。また、MUMPSは、専用言語方式とも呼ぶべき独自のDB操作機能を持っているが、このテストに用いたBASICには、DB機能はない。PL/Iは、DBの操作を行う時には、親言語方式の汎用DBMSであるDL/Iをユーザする形式をとっている。各テストは、すべてシングルジョブ環境で行っており、使用したハードウェアの性能を、表1にまとめる。

以下では、まず主記憶上での基本データ操作機能について処理速度を比較し、次に、DBMS機能について同じく比較する。最後に、具体例を用いた時の比較について示す。

表1. テストに使用したハードウェア性能一覧表

	(1)	(2)	(3)	(4)
CPU形式	PDP-11/34	PDP-11/40	PDP-11/55	IBM 370/138
主記憶種別	MOS	CORE	Mos + Bipola	MOS
主記憶サイズ	96 KB	160 KB	224 KB	512 KB
サイクルタイム	725 ns/2B	980 ns/2B	300 ns/2B	715 ns/2B
DISK形式	RK 05	RP03	RP03	3340
台数	3	1	1	6
容量/ドライブ	2.4 MB	40 MB	40 MB	70 MB
伝送速度	90 KB/s	267 KB/s	267 KB/s	885 KB/s
平均アクセス時間	50 ms	42 ms	42 ms	25 ms

2. MUMPS, BASIC, PL/Iの基本インストラクション実行速度の比較

MUMPS と他言語と比較するために、インタープリタ言語としてポプユアーな BASIC と、コンパイラ言語としては、恐らく最も高級な部類に属する PL/I を用いる。

(方法) MUMPS-11 V4B (DEC社) のユーティリティープログラムとして配布されている内の "TP2" と同機能のものを各言語で作成し、実行させる。このプログラムは、SET コマンドを中心にした各種のパターンの代入文部分と、数値演算を行う部分、一般的な DISK 操作を行う部分より成っているが、ここでは、前の二者のみについて比較を行う。使用したプログラムを、MUMPS を附録 1 に、BASIC を附録 2, 3, 4 に、PL/I を附録 5 に示す。また出力の例を附録 6, 7, 8, 9 に示す。BASIC は、DEC 社の RT-11 モニターの下で使用し、PL/I は、IBM 社の DOS/VS の下で使用している。

(ハードウェア) MUMPS, BASIC については、表 1(1) の PDP-11/34 を使い、PL/I は表 1(4) の IBM 370/138 を使った。PDP-11/34 と 370/138 とは、サイクルタイムのみに注目すれば、表 2 の(注)に示すように、ほぼ同等であるため、CPU Bound

表 2 MUMPS, BASIC, PL/I の基本インストラクション実行速度比較表

	ハードウェア		PDP-11/34 96KB		IBM 370/138 512KB
			MUMPS-11 V4B	RT-11	DOS/VS
			MUMPS-11 V4B	BASIC	PL/I
基本 操 作	S N=0	TEST No. ②	1 ms	0.7 ms	0.016 ms
	S N=21474836.47	③	3	0.5	0.014
	S A(1)=21474836.47	④	3	1.8	0.019
	S A(I)=I	⑤	21	1.3	0.099
	(S M="STR") S A(I)=M	⑥	21	1.4	0.094
	S A(I)=N	⑦	21	1.4	0.109
	(S M="ABC...XYZ") S A(I)=M	⑧	6	2.4	0.157
	S A=M	⑨	3	2.4	0.033
	S X(I)=H@ " " @ G(30), Y(I)=A+K(1000)+I	⑩	31	12.4	0.549
	小 計		110	24.3	1.090
	スピード比		1	4.5	101
数 値 演 算	MULTIPLY	⑪	3.8 ms	1.5 ms	0.159 ms
	DIVIDE	⑫	4.9	1.7	0.329
	\$INTEGER	⑬	4.0	2.1	0.109
	小 計		12.7	5.3	0.597
	スピード比		1	2.4	21

(注) サイクルタイムの比較 $11/34 : 370/138 = 725 \text{ ns}_{28} : 715 \text{ ns}_{28} = 1 : 1$

Jobの場合、同性能のコンピュータとみなしてさしつかえない。

(結果) 実行結果を表2に示す。当然予想されたように、処理速度は、MUMPS < BASIC < PL/Iとなる。同じインタープリタ言語でありながら、BASICがMUMPSより速いのは、MUMPSがキャラクター単位のインタープリタであるのに対し、BASICはプログラム作成時に、ソースプログラムを一行単位にコンパイルし、実行しやおい中間形式まで落しているためである。PL/Iは、コンパイル言語のため、実行時点では、アセンブラーのレベルで動作している。

基本操作のスピードだが、 $MUMPS : BASIC : PL/I = 1 : 4.5 : 101$ であることは、逆に、PL/Iでの処理時間が、OSによるオーバーヘッドを加味した上で、計算機として粉飾のない最高速度のものであるとするならば、MUMPS, BASICは、各々アセンブラーに対して101倍, 22倍のオーバーヘッド時間を持っていることになる。数値演算では、PL/Iのみが浮動小数点演算機構を使用しており、MUMPSとBASICは、ソフトで演算を行っている。このため、PL/Iと他との比較はいにくいが、MUMPSとBASICでは、やはり中間言語に落して実行しているBASICの方が、 $MUMPS : BASIC = 1 : 2.4$ と2倍強速くなっている。この段階でみれば、MUMPSはやはり速いと言わざるを得ない。

3. MUMPS, BASIC, PL/I, DL/IのDBMSとしての実行速度比較

MUMPS及びDL/Iは、DBMSとしての機能をもっているが、既述のようにRT-11でのBASICにはこの機能はなく、DISK上にDIMENSIONとしてファイルが形成できるだけである。しかも、配列が 50×50 以内の2次元のみと、きびしい制約がある。このため、MUMPS対BASIC, MUMPS対DL/Iと2組に分けて比較する。

3.1 MUMPS 対 BASIC

(方法) 2節と同様に、MUMPS-11 V4Bのユーティリティプログラムとして配布されている内の"TP8"と同機能のものを各言語で作成する。このプログラムは、DISK上に、I, J, K 3レベルの $(1, 1, 1) \sim (n, n, n)$ の形のトリートリー構造のファイルを作成し、各ノードに約20 Byteのデータをセットする。

ここではBASICの制約に合わせるために、I, Jの2レベルのみとし、Nを10より10刻みで40まで変化させた時の所要時間を計測する。用いたプログラムを附録10, 11に示す。

(ハードウェア) 両者とも表1(1)のPDP-11/34, RK05の構成を用いた。

(結果) 結果を表3に示す。Nの変化による、1ノードあたりの処理時間のばらつきについては、ディスク上のブロックのアロケーションによるものが大きいので、平

均値のみと比較する。使用した RK05 の平均アクセスタイムが 50ms であることを考慮すると、MUMPS : BASIC = 31ms : 27ms は、アクセス回数の差とも考えられ、表3のみからは、何ともいえない。

しかし、MUMPS の平均的なコマンドの実行時間は、BASIC より 4.5 倍も遅く、かつ BASIC が単純な 2 次元構造であり、事前に DIMENSION による定義が必要なのに比べ、MUMPS は ^{ツリー} トリー構造の DB であり、スライスアレイが可能、実行時にアロケーションを行うなど、機能的に格段上のものもっている。したがって、これらの点に注目すれば、MUMPS は BASIC に比べ、計算機システムとしてより大きい負荷をもっているにもかかわらず、平均的な処理時間には、あまり差がないと見ることができる。

この段階では、一応ディスク上のデータ処理に関して、MUMPS は BASIC に比べて J は遅くなく、機能的には優れていると言えようである。

3.2 MUMPS 対 PL/I, DL/I

(方法) 3.1 と同様に、"TP8" を用いるが、レベルを I, J, K の 3 レベルとし、各レベルでのロード数 n を、10 より 10 刻みで 40 まで作成した場合の比較を行う。使用したプログラムを附録 12 に示す。PL/I のプログラム中、CALL PLITODLI のステップで DBMS が呼び出されている。DL/I では、同一のセグメントに対してインサートが行われた時、自動的にオーバーレンスとして処理するため、アプリケーションプログラムでの MUMPS でのいう添字の制御の必要はない。

(ハードウェア) 3.1 で使用した構成では、DISK 装置のアクセス速度が遅いため、表1の(2)と(4)である PDP-11/40, RP03 と IBM 370/138, 3340 の構成を用いる。11/40 と 11/34 とは、インターリービングにより実質的に同性能の CPU である。

表3 MUMPS, BASIC の DBMS と 17 の実行速度比較表

ハードウェア		PDP-11/34 (RK05) 96KB			
オペレーティングシステム		MUMPS-11 V4B		RT-11	
言語		MUMPS-11 V4B		BASIC	
I, J=n	ロード数	総計	1ロード当り	総計	1ロード当り
10	110	3 ^s	27 ^{ms}	5 ^s	45 ^{ms}
20	420	11	26	11	26
30	930	30	32	23	25
40	1640	63	38	35	21
50	2550	—	—	51	20
平均		31 ^{ms}		27 ^{ms}	
スピード比		1		0.87	

しかし、逆に汎用DBMSとしての全機能を備えるなら、DL/I程度のプログラムサイズと処理時間が必要であるとも言える。つまりMUMPSは、リアルタイム処理用にDISK上のデータ操作を高速化するために、汎用DBMSとして必要な諸機能（特にプログラムのデータ独立と達成するための、サブスキーマ $\leftrightarrow$ スキーマ $\leftrightarrow$ 物理構造のマッピング）を相当犠牲にして、身軽にしているといえよう。

外来会計での患者別会計明細書を作成するバッチ業務を、従来MUMPSで行っていたが、処理時間短縮のために、図1に示すごとくPL/Iに変更した例について比

ハードウェア		PDP-11/40 (RPO3) 160KB		IBM370/138 (3340) 512KB	
オペレーティングシステム		MUMPS-11 V4B		DOS/VS	
言語		MUMPS-11 V4B		PL/I, DL/I	
I, J, K =	1-ド数	紙計	1/1ボリ	統計	1/1ボリ
10	1110	24 ^S	22 ^{ms}	19 ^S	17 ^{ms}
20	8420	152	18	96	11
30	27930	638	23	301	11
40	65640	1597	24	714	11
平均			22 ^{ms}		12 ^{ms}
スピード比			1		1.8

- 較する。両者とも、まったく同じ形式のソースデータから、同じ様式のリストを作成している。MUMPS例で使用したハードウェアは、表1の(3)のPDP-11/55, RP03の組合せのものである。測定結果を表5に示す。この最終的な実行時間比 MUMPS : PL/I = 1 : 1.2 を比較するにあたって、次の注意が必要である。すなわち、
- ① 11/55でのMUMPS-11 V4Aと、370/138でのPL/Iのインストラクション実行時間の比 MUMPS : PL/I = 37 : 1
 - ② RP03と3340のDISKアクセス時間の比 RP03 : 3340 = 1.7 : 1
 - ③ PL/Iで上記プログラムを実行する時には、患者情報以外のすべての情報(診療点数マスター、保険コードテーブルなど)をINCOREすることにより、ディスクアクセスの回数を減少させている。このことによるMUMPSとPL/IとのDISKアクセス回数の比 MUMPS : PL/I = 7 : 1

表5. 外来会計明細書作成をMUMPSからPL/Iに移した時の諸パラメータの比較表

	MUMPS-11 V4A	PL/I, DL/I
プログラムサイズ	340 step 11.6 KB	1200 step 40.6 KB
コンスタントエリア(ローカル変数) も含めたプログラムサイズ	40 KB	267 KB
シングルJOBでの実行時間	51 分	42 分

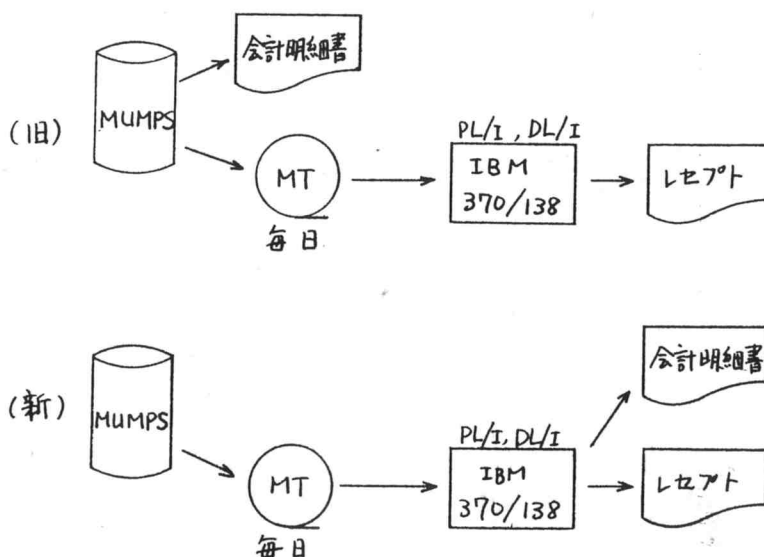


図1. 外来会計明細書作成プログラムの言語変更

これを考慮すれば、PL/I, DL/Iは、もっと速くなければならない。しかし、実際にはMUMPSと大差のない値を示すのは、267KBのプログラムと、リアル域が96KBの仮想記憶域で実行させているために、メインストレージと、ページデータセットとの間で、2KB単位のページングが頻繁に行われていることによる。

このように、一般にMUMPSと同じ業務をコンパイル言語を用いて行う時には、コンパイルすることによるプログラムサイズの増加により、重大な問題が発生する。このケースにおいても、プログラム自体のサイズで約4倍、実行に必要なエリアで6.7倍も大きくなっている。極端な時には、コンパイルによって速くなった処理時間部分が、OSのVirtual Storage機能により、全部スポイルされてしまう場合もあり、使用するOSの機能によっては、必ずしも“コンパイル言語=速い”とはならない。

したがって、速くするためには、いかに小さいプログラムを書くか、またVS機能と使用するなら、いかにしてページングの起りにくいプログラムにするかなど、MUMPSとは違った意味でプログラムに細かい配慮が必要である。

5. まとめ

MUMPSの実行速度と、一般の基本操作機能と、DBMSとしての機能とに分けて各BASIC, PL/I, DL/Iと比較した。

CPU boundのみのジョブでは、当然MUMPSは遅いと言わざるを得ない。しかし、DBMSとしてみた時、ディスク上のデータ処理に関しては、DL/Iよりも3倍速いと言える。但しこれは、DBMSとしての機能を一部削っているためであることと銘記すべきである。したがって、窓口業務のようにオンラインでDBを使用するために、CPU boundとDISK boundの処理が適度に混在している業務では、MUMPSは他の言語よりも十分速いレスポンスを示すといえよう。

```

>W
0.01 TP2 1-JUN-75 V3B
0.02 TEST PROGRAM FOR MUMPS-11 TIMINGS
0.98 COPYRIGHT 1973,1974,1975,1976,1977 DIGITAL EQUIPMENT CORPORATION MAYNARD MASS.

1.01 S XIO=$I C XIO E Q
1.05 T !!$S(.02),!
1.10 T !"PART 6 LOOKS INTO THE SYSTEM TABLE; THIS PROGRAM MUST RUN UNDER UCI # 1"
1.12 T !"THIS PROGRAM USES A GLOBAL NAMED ^A.,!"
1.15 T !"THE SYMBOL TABLE EXERCISOR (PART 3) TAKES ABOUT 5 MINUTES ON AN 11/20.,!"
1.20 T !"THE GLOBAL EXERCISOR (PART 5) TAKES ABOUT 6 MINUTES ON AN 11/20.,!"
1.25 T !"THE ARITHMETIC EXERCISOR (PART 8) TAKES ABOUT 6 MINUTES ON AN 11/20.,!"
1.30 D 1.97 G 2
1.97 H:$(V(65400)/100&.64

2.10 T !!"SEC",?5,"I",?10,"STEP EXECUTED",!
2.20 F STP=3.1!:.05:3.45 D 2.8
2.30 S A=1,B="STRING",G="ARRAY" F I=1:1:50 S G(I)="G"@I@")"
2.40 S C=10000,D="D",E=100,F="STRING",H="H",J=1,K(200)="STRING"
2.50 S K(1000)=1000,K(20)="STRING"
2.60 F STP=3.5,3.55 D 2.9
2.70 K D 1.97 G 4
2.80 S T=$T D STP S T=$T-T T !T,?5,I,?10,$S(STP) K A
2.90 S T=$T D STP S T=$T-T T !T,?5,I,?10,$S(STP)

3.10 F I=1:1 U ($S<100)!(I=1000)
3.15 F I=1:1 U ($S<100)!(I=1000) S N=0
3.20 F I=1:1 U ($S<100)!(I=1000) S N=21474836.47
3.25 F I=1:1 U ($S<100)!(I=1000) S A(1)=21474836.47
3.30 F I=1:1 U ($S<100)!(I=1000) S A(I)=I
3.35 S M="STR" F I=1:1 U ($S<100)!(I=1000) S A(I)=M
3.40 F I=1:1 U ($S<100)!(I=1000) S A(I)=N
3.45 S M="ABCDEFGHJKLMNOPQRSTUVWXYZ" F I=1:1 U $S<100 S A(I)=M
3.50 F I=1:1 U ($S<100)!(I=1000) S A=M
3.55 F I=1:1 U ($S<100)!(I=1000) S X(I)=H@ " @G(30),Y(I)=A+K(1000)+I

4.10 S IOD=$I T !!$ SEC.",?10,"STEP EXECUTED",! F STP=5.1!:.1:5.7 D 4.3
4.20 K ^A K D 1.97 G 8
4.30 A 0:0,IOD S T=$T D STP S T=$T-T A 0:-1,IOD T !T,?10,$S(STP) I STP<5.4 D 6

5.10 F I=1:1:25 S ^A(I)="STRING"
5.20 F I=1:1:3 F J=1:1:10 S ^A(I,J)=I@", "@J
5.30 F I=1:1:3 F J=1:1:10 F K=1:1:25 S ^A(I,J,K)=I+J+K
5.40 F I=1:1:3 F J=1:1:10 S ^A(I,J,1)=I+J+1 F K=2:1:25 S ^A(K)=I+J+K
5.50 S ^A(1)="STRING" F I=2:1:25 S ^A(I)="STRING"
5.60 F I=1:1:3 S ^A(I,1)=I@", "@1 F J=2:1:10 S ^A(J)=I@", "@J
5.70 F I=1:1:3 S ^A(I)="STRING" F J=1:1:10 S ^A(I,J)=I@", "@J F K=1:1:25 S ^A(K)=I+J+K

6.10 K ^A
6.20 I ($V($V(44)+26)/100&2.55) H 3 G $L

8.05 K S LPT=0,S=8.5 T !!$S(S),!
8.10 D 8.35 F I=1:1:10000
8.15 D 8.4 S LPT=T D 8.35 F I=1:1:10000 S X=327.67*327.67
8.20 D 8.4,8.35 F I=1:1:10000 S X=327.67/327.67
8.25 D 8.4,8.35 F I=1:1:10000 S X=$I(327.67+327.67)
8.30 D 8.4,8.35 T !!,$S(S),!! Q
8.35 D 1.97 K T,I,X S S=S+.05,T=$T
8.40 S T=$T-T T $S(S)," ",T-LPT/10," MS",!
8.50 ARITHMETIC TIMING TEST
8.55 BASIC LOOP TIME IS .....
8.60 MULTIPLY TIME IS .....
8.65 DIVIDE TIME IS .....
8.70 $INTEGER TIME IS .....
8.75 TIMING TEST DONE

```

附録 1. 基本インストラクションテストプログラム (MUMPS-11)

LIST

TP2 08-AUG-79 14:31:42

```

10 REM PROGRAM NAME -- TP2
20 REM CODED BY T. HATTORI
100 REM ---MAIN ROUTINE---
105 PRINT
110 PRINT "TEST PROGRAM FOR BASIC TIMING"
120 PRINT
130 PRINT "TEST NO. "; \ INPUT X
135 IF X>3 THEN GOSUB 1500
140 GOSUB 1000
160 STOP
1000 REM ---SUBROUTINE 1---
1010 GOSUB 1200
1020 T=T1
1030 ON X GOSUB 2000,2100,2200,2300,2400
1040 GOSUB 1200
1050 T=T1-T
1060 PRINT "TEST NO. ";TAB(8);X;
1070 PRINT TAB(20);"ELAPSED TIME IS ";TAB(38);T;TAB(42);"SEC";
1075 PRINT TAB(50);"( I =";I;")"
1080 PRINT
1090 RETURN
1200 REM ---TIME-CALCULATING ROUTINE---
1210 T1=CLK$
1220 H$=SEG$(T1$,1,2) \ M$=SEG$(T1$,4,5) \ S$=SEG$(T1$,7,8)
1230 H=VAL(H$) \ M=VAL(M$) \ S=VAL(S$)
1240 T1=3600*M+60*H+S
1250 RETURN
1500 REM ---DEFINE---
1510 DIM A(8000)
1530 RETURN
2000 REM ---TEST NO.1---
2010 FOR I=1 TO 10000
2020 NEXT I
2030 RETURN
2100 REM ---TEST NO.2---
2110 FOR I=1 TO 10000
2120 N=0
2130 NEXT I
2140 RETURN
2200 REM ---TEST NO.3---
2210 FOR I=1 TO 10000
2220 N=2.14748E+07
2230 NEXT I
2240 RETURN
2300 REM ---TEST NO.4---
2310 FOR I=1 TO 10000
2320 A(1)=2.14748E+07
2330 NEXT I
2340 RETURN
2400 REM ---TEST NO.5---
2410 FOR I=1 TO 8000
2420 A(I)=I
2430 NEXT I
2440 RETURN
32767 END

```

READY

```

10 REM      PROGRAM NAME -- TP2
20 REM      CODED BY T. HATTORI
100 REM     ---MAIN ROUTINE---
105 PRINT
110 PRINT "TEST PROGRAM FOR BASIC TIMING"
120 PRINT
125 GOSUB 1500
130 PRINT "TEST NO. "; \ INPUT X
140 IF X>8 THEN GOSUB 1600
150 X1=X-5
160 GOSUB 1000
180 STOP
1000 REM    ---SUBROUTINE 1---
1010 GOSUB 1200
1020 T=T1
1030 ON X1 GOSUB 2500,2600,2700,2800,2900
1040 GOSUB 1200
1050 T=T1-T
1060 PRINT "TEST NO. ";TAB(8);X;
1070 PRINT TAB(20);"ELAPSED TIME IS ";TAB(38);T;TAB(42);"SEC";
1075 PRINT TAB(50);"( I =";I;")"
1080 PRINT
1090 RETURN
1200 REM    ---TIME-CALCULATING ROUTINE---
1210 T1$=CLK$
1220 H$=SEG$(T1$,1,2) \ M$=SEG$(T1$,4,5) \ S$=SEG$(T1$,7,8)
1230 H=VAL(H$) \ M=VAL(M$) \ S=VAL(S$)
1240 T1=3600*M+60*M+S
1250 RETURN
1500 REM    ---DEFINE---
1510 DIM A(1000)
1520 DIM A$(1000)
1530 DIM G$(50)
1540 DIM K(1000)
1550 DIM K$(200)
1560 DIM X$(1000)
1570 DIM Y(1000)
1580 RETURN
1600 A=1 \ H$="H"
1610 FOR J=1 TO 50 \ G$(I)="G$("+STR$(J)+")" \ NEXT J
1630 K$(200)="STRING" \ K(1000)=1000 \ K$(20)="STRING"
1640 RETURN
2500 REM    ---TEST NO.6---
2510 M$="STR"
2520 FOR I=1 TO 1000
2530 A$(I)=M$
2540 NEXT I
2550 RETURN
2600 REM    ---TEST NO.7---
2610 FOR I=1 TO 1000
2620 A(I)=N
2630 NEXT I
2640 RETURN
2700 REM    ---TEST NO.8---
2710 M$="ABCDEFGHJKLMNOPQRSTUVWXYZ"
2720 FOR I=1 TO 500
2730 A$(I)=M$
2740 NEXT I
2750 RETURN
2800 REM    ---TEST NO.9---
2810 FOR I=1 TO 1000
2820 A=M
2830 NEXT I
2840 RETURN
2900 REM    ---TEST NO.10---
2910 A$="00000000000000000000000000000000"
2920 FOR I=1 TO 1000
2930 X$(I)=H$+" "+G$(30)
2940 Y(I)=VAL(A$)+K(1000)+I
2950 NEXT I
2960 RETURN

```

LIST

TS2 08-AUG-79 14:35:17

```
10 REM PROGRAM NAME---TS2
20 REM CODED BY T. HATTORI
100 REM ---MAIN ROUTINE---
110 PRINT \ PRINT TAB(10);"ARITHMETIC TIMING TEST FOR BASIC"
120 PRINT
1000 REM ---BASIC LOOP---
1010 GOSUB 2000
1020 T=T1
1030 FOR I=1 TO 10000
1040 NEXT I
1050 GOSUB 2000
1060 T=T1-T
1070 PRINT "BASIC LOOP TIME IS .....";T/10;"MS"
1080 T=T
1200 REM ---MULTIPLY---
1210 GOSUB 2000
1215 T=T1
1220 FOR I=1 TO 10000
1230 X=32765*32767
1240 NEXT I
1250 GOSUB 2000
1260 T=T1-T
1270 PRINT "MULTIPLY TIME IS .....";(T-T0)/10;"MS"
1400 REM ---DIVIDE---
1410 GOSUB 2000
1420 T=T1
1430 FOR I=1 TO 10000
1440 X=32765/32767
1450 NEXT I
1460 GOSUB 2000
1470 T=T1-T
1480 PRINT "DIVIDE TIME IS .....";(T-T0)/10;"MS"
1600 REM ---INTEGER---
1610 GOSUB 2000
1620 T=T1
1630 FOR I=1 TO 10000
1640 X=INT(327.67+327.67)
1650 NEXT I
1660 GOSUB 2000
1670 T=T1-T
1680 PRINT "INTEGER FUNCTION TIME IS ..";(T-T0)/10;"MS"
2000 REM ---TIMR CALCULATING ROUTINE---
2010 T1$=CLK$
2020 H$=SEG$(T1$,1,2) \ M$=SEG$(T1$,4,5) \ S$=SEG$(T1$,7,8)
2030 H=VAL(H$) \ M=VAL(M$) \ S=VAL(S$)
2040 T1=3600*M+60*S
2050 RETURN
2200 END
```

READY

附録4. 基本インストラクションテストプログラム (BASIC) 303

12	1	0	/* テスト ケース (3) */ TIMES31 = TIME;
13	1	0	DO I = 1 TO 10000;
14	1	1	N = 2147483647;
15	1	1	END;
16	1	0	TIMES32 = TIME;
17	1	0	PUT SKIP EDIT('N = ', N) (A,F(11));
18	1	0	/* テスト ケース (4) */ TIMES41 = TIME;
19	1	0	DO I = 1 TO 10000;
20	1	1	A(1) = 2147483647;
21	1	1	END;
22	1	0	TIMES42 = TIME;
23	1	0	PUT SKIP EDIT('A(1) = ', A(1)) (A,F(11));
24	1	0	/* テスト ケース (5) */ TIMES51 = TIME;
25	1	0	DO I = 1 TO 10000;
26	1	1	A(1) = 1;
27	1	1	END;
28	1	0	TIMES52 = TIME;
29	1	0	PUT SKIP EDIT('A(10000) = ', A(10000)) (A,F(11));
30	1	0	/* テスト ケース (6) */ TIMES61 = TIME;
31	1	0	M = 'STR';
32	1	0	DO I = 1 TO 10000;
33	1	1	B(1) = M;
34	1	1	END;
35	1	0	TIMES62 = TIME;
36	1	0	PUT SKIP EDIT('B(10000) = ', B(10000)) (A,A);
37	1	0	/* テスト ケース (7) */ TIMES71 = TIME;
38	1	0	N = 2147483647;
39	1	0	DO I = 1 TO 10000;
40	1	1	A(1) = N;
41	1	1	END;
42	1	0	TIMES72 = TIME;
43	1	0	PUT SKIP EDIT('A(10000) = ', A(10000)) (A,F(11));
44	1	0	/* テスト ケース (8) */ TIMES81 = TIME;
45	1	0	P = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
46	1	0	DO I = 1 TO 10000;
47	1	1	C(1) = P;
48	1	1	END;
49	1	0	TIMES82 = TIME;
50	1	0	PUT SKIP EDIT('C(10000) = ', C(10000)) (A,A);

```

51 1 0      /* テスト ケース (9) */
           TIMES91 = TIME;

52 1 0      DO I = 1 TO 10000;

53 1 1      D = P;

54 1 1      END;
55 1 0      TIMES92 = TIME;

           /* テスト ケース (10) */
56 1 0      TIMESA1 = TIME;
57 1 0      N = G;

58 1 0      DO I = 1 TO 10000;

59 1 1      X(I) = H || ' ' || G(I);
60 1 1      Y(I) = N + K(I) + I;

61 1 1      END;
62 1 0      TIMESA2 = TIME;
63 1 0      PUT SKIP EDIT('X= ',X(10000),'Y= ',Y(10000))(A,A,A,F(11));

           /* テスト ケース (11) */
64 1 0      TIMESB1 = TIME;

65 1 0      DO I = 1 TO 10000;

66 1 1      R = 327.67 * 327.67;

67 1 1      END;
68 1 0      TIMESB2 = TIME;
69 1 0      PUT SKIP EDIT('R = ', R) (A,F(11,2));

           /* テスト ケース (12) */
70 1 0      TIMESC1 = TIME;

71 1 0      DO I = 1 TO 10000;

72 1 1      R = 327.67 / 327.67;

73 1 1      END;
74 1 0      TIMESC2 = TIME;
75 1 0      PUT SKIP EDIT('R = ', R) (A,F(11,2));

           /* テスト ケース (13) */
76 1 0      TIMESD1 = TIME;

77 1 0      DO I = 1 TO 10000;

78 1 1      R = TRUNC(327.67 + 327.67);

79 1 1      END;
80 1 0      TIMESD2 = TIME;
81 1 0      PUT SKIP EDIT('R = ', R) (A,F(11,2));
82 1 0      PUT SKIP EDIT('*** - (1) START TIME = ', TIMES11)(A);
83 1 0      PUT SKIP EDIT('*** - (1) END TIME = ', TIMES12)(A);
84 1 0      PUT SKIP EDIT('*** - (2) START TIME = ', TIMES21)(A);
85 1 0      PUT SKIP EDIT('*** - (2) END TIME = ', TIMES22)(A);
86 1 0      PUT SKIP EDIT('*** - (3) START TIME = ', TIMES31)(A);
87 1 0      PUT SKIP EDIT('*** - (3) END TIME = ', TIMES32)(A);
88 1 0      PUT SKIP EDIT('*** - (4) START TIME = ', TIMES41)(A);
89 1 0      PUT SKIP EDIT('*** - (4) END TIME = ', TIMES42)(A);
90 1 0      PUT SKIP EDIT('*** - (5) START TIME = ', TIMES51)(A);
91 1 0      PUT SKIP EDIT('*** - (5) END TIME = ', TIMES52)(A);
92 1 0      PUT SKIP EDIT('*** - (6) START TIME = ', TIMES61)(A);
93 1 0      PUT SKIP EDIT('*** - (6) END TIME = ', TIMES62)(A);
94 1 0      PUT SKIP EDIT('*** - (7) START TIME = ', TIMES71)(A);
95 1 0      PUT SKIP EDIT('*** - (7) END TIME = ', TIMES72)(A);
96 1 0      PUT SKIP EDIT('*** - (8) START TIME = ', TIMES81)(A);
97 1 0      PUT SKIP EDIT('*** - (8) END TIME = ', TIMES82)(A);
98 1 0      PUT SKIP EDIT('*** - (9) START TIME = ', TIMES91)(A);
99 1 0      PUT SKIP EDIT('*** - (9) END TIME = ', TIMES92)(A);
100 1 0     PUT SKIP EDIT('*** - (10) START TIME = ', TIMESA1)(A);

```

```

101 1 0 PUT SKIP EDIT('*** - (10) END TIME = ', TIMESAZ)(A);
102 1 0 PUT SKIP EDIT('*** - (11) START TIME = ', TIMESB1)(A);
103 1 0 PUT SKIP EDIT('*** - (11) END TIME = ', TIMESB2)(A);
104 1 0 PUT SKIP EDIT('*** - (12) START TIME = ', TIMESB1)(A);
105 1 0 PUT SKIP EDIT('*** - (12) END TIME = ', TIMESB2)(A);
106 1 0 PUT SKIP EDIT('*** - (13) START TIME = ', TIMESD1)(A);
107 1 0 PUT SKIP EDIT('*** - (13) END TIME = ', TIMESD2)(A);

10d 1 0 END PRUG1;

```

附録5 基本インストラクションテストプログラム (PL/I)

その4

>C TP2

TEST PROGRAM FOR MUMPS-11 TIMINGS

PART 6 LOOKS INTO THE SYSTEM TABLE; THIS PROGRAM MUST RUN UNDER UCI # 1
THIS PROGRAM USES A GLOBAL NAMED ^A.

THE SYMBOL TABLE EXERCISOR (PART 3) TAKES ABOUT 5 MINUTES ON AN 11/20.
THE GLOBAL EXERCISOR (PART 5) TAKES ABOUT 6 MINUTES ON AN 11/20.
THE ARITHMETIC EXERCISOR (PART 8) TAKES ABOUT 6 MINUTES ON AN 11/20.

SEC I STEP EXECUTED

```

5 1000 F I=1:1 U ($S<100)!(I=1000)
7 1000 F I=1:1 U ($S<100)!(I=1000) S N=0
9 1000 F I=1:1 U ($S<100)!(I=1000) S N=21474836.47
9 1000 F I=1:1 U ($S<100)!(I=1000) S A(1)=21474836.47
22 847 F I=1:1 U ($S<100)!(I=1000) S A(I)=I
17 659 S M="STR" F I=1:1 U ($S<100)!(I=1000) S A(I)=M
19 753 F I=1:1 U ($S<100)!(I=1000) S A(I)=N
2 171 S M="ABCDEFGHJKLMNOPQRSTUVWXYZ" F I=1:1 U $S<100 S A(I)=M
9 1000 F I=1:1 U ($S<100)!(I=1000) S A=M
8 244 F I=1:1 U ($S<100)!(I=1000) S X(I)=HE" *@G(30),Y(I)=A+K(1000)+I

```

* SEC. STEP EXECUTED

```

1 F I=1:1:25 S ^A(I)="STRING"
2 F I=1:1:3 F J=1:1:10 S ^A(I,J)=I@", "@J
35 F I=1:1:3 F J=1:1:10 F K=1:1:25 S ^A(I,J,K)=I+J+K
10 F I=1:1:3 F J=1:1:10 S ^A(I,J,1)=I+J+1 F K=2:1:25 S ^A(K)=I+J+K
0 S ^A(1)="STRING" F I=2:1:25 S ^A(I)="STRING"
1 F I=1:1:3 S ^A(I,1)=I@", "@1 F J=2:1:10 S ^A(J)=I@", "@J
7 F I=1:1:3 S ^A(I)="STRING" F J=1:1:10 S ^A(I,J)=I@", "@J F K=1:1:25 S

```

ARITHMETIC TIMING TEST

```

BASIC LOOP TIME IS ..... 1.40 MS
MULTIPLY TIME IS ..... 3.80 MS
DIVIDE TIME IS ..... 4.90 MS
%INTEGER TIME IS ..... 4 MS

```

^A(K)=I+J

TIMING TEST DONE

附録6 基本インストラクションテストプログラム (MUMPS) 出力結果

TP2 08-AUG-79 14:25:28

TEST PROGRAM FOR BASIC TIMING

TEST NO. ? 1
TEST NO. 1 ELAPSED TIME IS 16 SEC (I = 10000)

STOP AT LINE 160

READY
LENGTH

16618 USED, 1358 FREE

READY
CLEAR

READY
RUNNH

TEST PROGRAM FOR BASIC TIMING

TEST NO. ? 2
TEST NO. 2 ELAPSED TIME IS 23 SEC (I = 10000)

STOP AT LINE 160

READY
LENGTH

16618 USED, 1358 FREE

READY
CLEAR

附録7 基本インストラクションテストプログラム (BASIC) 出力結果 (附録2,3に相当)

RUN

TS2 08-AUG-79 10:46:23

ARITHMETIC TIMING TEST FOR BASIC

BASIC LOOP TIME IS 1.6 MS
MULTIPLY TIME IS 1.5 MS
DIVIDE TIME IS 1.7 MS
INTEGER FUNCTION TIME IS .. 2.1 MS

?RETURN WITHOUT GOSUB AT LINE 2050

READY
OLD TP2

附録8 基本インストラクションテストプログラム (BASIC) 出力結果 (附録4に相当)

N = 2147483647
 A(1) = 2147483647
 A(10000) = 10000
 B(10000) = SIR
 A(10000) = 2147483647
 C(10000) = ABCDEFGHIJKLMNOPQRSTUVWXYZ
 X= H G(1)Y= 1234578890
 K = 107367.62
 R = 1.00
 R = 655.00

	ms	左 - 1140	1725.767 (ms)
*** - (1) START TIME = 164226523	1140	0	
*** - (1) END TIME = 164227103			
*** - (2) START TIME = 164227166	1297	157	0.0157
*** - (2) END TIME = 164228403			
*** - (3) START TIME = 164228466	1277	137	0.0137
*** - (3) END TIME = 164229743			
*** - (4) START TIME = 164230133	1330	190	0.0190
*** - (4) END TIME = 164231463			
*** - (5) START TIME = 164231476	2127	987	0.0987
*** - (5) END TIME = 164233603			
*** - (6) START TIME = 164233613	2083	943	0.0943
*** - (6) END TIME = 164235696			
*** - (7) START TIME = 164235706	2233	1093	0.1093
*** - (7) END TIME = 164237939			
*** - (8) START TIME = 164237953	2713	1573	0.1573
*** - (8) END TIME = 164240666			
*** - (9) START TIME = 164240809	1474	334	0.0334
*** - (9) END TIME = 164242283			
*** - (10) START TIME = 164242283	1630	5490	0.5490
*** - (10) END TIME = 164248913			
*** - (11) START TIME = 164248926	2733	1593	0.1593
*** - (11) END TIME = 164251059			
*** - (12) START TIME = 164251669	4434	3294	0.3294
*** - (12) END TIME = 164256103			
*** - (13) START TIME = 164256113	2226	1086	0.1086
*** - (13) END TIME = 164258339			

附録9 基本インストラクションテストプログラム (PL/I) 出力結果

```

1.01 S ZIO=$I C ZIO E Q
1.10 K ^A S ^A="THIS IS A TEST GLOBAL OF THREE LEVELS",^A(0)="LEVEL 1, NODE 0"
1.20 R !"LEVEL 1 NODES? ",II," LEVEL 2 NODES? ",JJ," LEVEL 3 NODES? ",KK,!!
1.25 R !"TIME BETWEEN GLOBAL ACCESSES? ",TIM,!
1.27 S TTT=$T
1.30 F I=1:1:II H TIM S ^I)="LEVEL 1 OF GLOBAL ^A, NODE ^@I D 2 S X=^A(I)
1.35 S TTT=$T-TTT
1.40 T !!"GLOBAL TEST PROGRAM COMPLETED",!!
1.45 T !,"ELAPSED TIME WAD ",TTT," SEC.",!!

2.10 S ^I(0)="*****"
2.20 F J=1:1:JJ S ^J)="GLOBAL ^A, NODE ^@I@",^@J D 3

3.10 S ^J(0)="---+++---+++---+++---+++---"
3.20 F K=1:1:KK S ^K)="NODE ^@I@",^@J@",^@K@ OF ^A"
3.30 S X=^A(I,J)

4.10 S GBL(1)=$V(GH),GBL(2)=$V(GH+2),GBL(3)=$V(GB),GBL(4)=$V(GB+2)
4.20 F N=1:1:4 S ZY=GBL(N) D 6 S GBL(N)=ZY

5.10 S JT=$V($V(44)),PN=$V(JT+38),PT=$V($V(44)+2),PB=$V($V(44)+244)
5.20 G 5.3:$V(PT)/25600>.18 S PN=PN/100&2.55*100,PB=$V(PT+PN)/25600*25600
5.30 S GH=PB+26,GB=GH+4

6.10 S ZI=0;CONVERT DECIMAL TO OCTAL
6.20 S ZI=ZI+1,ZX=ZY/8,ZR(ZI)=ZY-(^I(ZX)*8) G 6.3:ZX<8 S ZY=$I(ZX) G $L
6.30 S ZI=ZI+1,ZR(ZI)=$I(ZX),ZV=" F ZJ=ZI:-1:1 S ZV=ZV@ZR(ZJ)

99.05 S II=15,JJ=5,KK=10 D 1.25
99.10 S TRY=1
99.20 S X=^A(1) D 1.3 S ZIO=$I C ZIO I T "PASS # ",TRY,"COMPLETE",! U 0
99.30 S TRY=TRY+1 G 99.2:($V(65400)/100&.64)
  
```

附録10 データベース操作機能テストプログラム (MUMPS)

TE8 08-AUG-79 13:47:15

```

10 REM PROGRAM NAME -- TP8
20 REM CODED BY T. HATTORI
100 REM ---MAIN ROUTINE---
120 PRINT "LEVEL 1 NODES? "; \ INPUT I1
130 PRINT "LEVEL 2 NODES? "; \ INPUT J1
150 GOSUB 2000
160 T=T1
165 DIM #1,A$(50,50)=40
170 OPEN "TEST1" AS FILE #1
180 A$(0,0)="LEVEL 1. NODE 0"
200 FOR I=1 TO I1
223 A$(I,0)="***=====*****"
230 FOR J=1 TO J1
240 A$(I,J)="GLOBAL ^A. NODE "+STR$(I)+"."+STR$(J)
310 NEXT J
320 NEXT I
330 CLOSE #1
340 GOSUB 2000
350 T=T1-T
360 PRINT
370 PRINT "GLOBAL TEST PROGRAM COMPLETED"
380 PRINT
390 PRINT "ELAPSED TIME WAS ";T;" SEC"
400 KILL "TEST1"
410 STOP
2000 REM ---TIME CALCULATING ROUTINE---
2010 T1$=CLK$
2020 H$=SEG$(T1$,1,2) \ M$=SEG$(T1$,4,5) \ S$=SEG$(T1$,7,8)
2030 H=VAL(H$) \ M=VAL(M$) \ S=VAL(S$)
2040 T1=3600*M+60*M+S
2050 RETURN
32767 END

```

READY

附録11 データベース操作機能テストプログラム (BASIC)

PL/I OPTIMIZING COMPILER DLITPLI:PROC(PCBPTR) OPTIONS(MAIN);

MACRO AND SOURCE LISTING

SIMT LEV NT		
1	0	DLITPLI:PROC(PCBPTR) OPTIONS(MAIN);
2	1	DCL PCBPTR PTR;
3	1	DCL 1 PCB BASED(PCBPTR),
		2 D1 CHAR(10),
		2 STCD CHAR(2);
4	1	DCL SSA1 CHAR(9) INIT('TEST'),
		SSA2 CHAR(9) INIT('CHILD1'),
		SSA3 CHAR(9) INIT('CHILD2'),
		SSA4 CHAR(9) INIT('CHILD3');
5	1	DCL D4 FIXED BIN(31) INIT(4);
6	1	DCL ISRT CHAR(4) INIT('ISRT');
7	1	DCL TIME BUILTIN;
8	1	DCL PLITDLI ENTRY EXTERNAL;
		/* */
9	1	DCL 1 AREA1,
		2 FIL1 CHAR(12) INIT('ROOT_SEGMENT'),
		2 IN1 PIC'99',
		1 AREA2,
		2 FIL2 CHAR(14) INIT('KO_SEG_LEVEL_1'),
		2 IN2 PIC'99',
		1 AREA3,
		2 FIL3 CHAR(14) INIT('KO_SEG_LEVEL_2'),
		2 IN3 PIC'99',
		1 AREA4,
		2 FIL4 CHAR(14) INIT('KO_SEG_LEVEL_3'),
		2 IN4 PIC'99';

附録12 データベース操作機能テストプログラム (PL/I) その1

```

10 1 0 GET EDIT(11)(COL(1),A(2));
11 1 0 TIMES1=TIME;
12 1 0 IN1=1;
13 1 0 CALL PLITDL1(D4,ISRT,PCB,AREA1,SSA1);
14 1 0 IF STCD = ' '
    THEN DO;
15 1 1 PUT SKIP EDIT('I5-1 ',STCD)(COL(1),A,A);
16 1 1 STOP;
17 1 1 END;
18 1 0 IN2=1;
19 1 0 DO B=1 TO 11 BY 1;
20 1 1 CALL PLITDL1(D4,ISRT,PCB,AREA2,SSA2);
21 1 1 IF STCD = ' '
    THEN DO;
22 1 2 PUT SKIP EDIT('I5-1 ',STCD)(COL(1),A,A);
23 1 2 STOP;
24 1 2 END;
25 1 1 IN3=1;
26 1 1 DO C=1 TO 11 BY 1;
27 1 2 CALL PLITDL1(D4,ISRT,PCB,AREA3,SSA3);
28 1 2 IF STCD = ' '
    THEN DO;
29 1 3 PUT SKIP EDIT('I5-2 ',STCD)(COL(1),A,A);
30 1 3 STOP;
31 1 3 END;
32 1 2 IN4=1;
33 1 2 DO D=1 TO 11 BY 1;
34 1 3 CALL PLITDL1(D4,ISRT,PCB,AREA4,SSA4);
35 1 3 IF STCD = ' '
    THEN DO;
36 1 4 PUT SKIP EDIT('I5-3 ',STCD)(COL(1),A,A);
37 1 4 STOP;
38 1 4 END;
39 1 3 IN4=IN4+1;
40 1 3 END;
41 1 2 IN3=IN3+1;
42 1 2 END;
43 1 1 IN2=IN2+1;
44 1 1 END;
45 1 0 TIMES2=TIME;
46 1 0 PUT SKIP EDIT('I5-4 ',STCD)(COL(1),A);
47 1 0 PUT SKIP EDIT('*** START TIME = ',TIMES1)(COL(1),A,A);
48 1 0 PUT SKIP EDIT('*** END TIME = ',TIMES2)(COL(1),A,A);
49 1 0 RETURN;
50 1 0 END DLITPL1;

```

附録12 データベース操作機能テストプログラム (PL/I) その2



第6回日本MUG学術大会 昭和54年9月14日(金)

B. パネル(1) MUMPSの速さ 司会 羽曳野病院 今井敏雄

B-4.

A General Purpose Batch Processing System

Based Upon MFT (MUMPS — FORTRAN Translator)

木村一元^{*}, 馬場謙介^{**}, 木村國恵^{**}, 林 徹^{*}

^{*} 獨協医科大学 総合研究施設

^{**} 獨協医科大学 第一病理学教室

1 はじめに

ジョブを速くするソフトウェア技術の中で、MUMPSのOSをそのままにして目的を果す方法には、アプリケーション作成技術でこれをカバーする方法と他のOSの助けを借りる方法とがある。他のOSの助けを借りる方法には、DECのfast global saveに見られるように他のOSにMUMPS向けユーティリティを持たせる方法と他OS下にある言語をも使用しようとする方法とがある。他言語を使用する方法には、データのみをやり取りする方法(こうなるともはやMUMPSを改良して速度を速くしたとは言えない)と、プログラムとデータとをやり取りする方法が考えられる。後者は(前者と同様にMUMPSを改良したとは必ずしも言えないが)、ユーザにとっては、MUMPS言語を知っているだけで、他のOS、他の言語の恩恵に浴する可能性が生ずる。この意味で、ユーザにとってはジョブを速くすることになる。この方法の1つとして、(多少、文法上の制限を加えて)MUMPS言語で書かれたプログラムをコンパイルして実行させる方法が考えられる。MUMPSコンパイラの実現は、膨大な開発費を余儀なくするばかりでなく、開発における個々の作業内容が機種によって特異であるアセンブラ言語に伴って変える事を余儀なくされる。そこで、我々は先に本学会で報告した如く、オーソライズされたFORTRAN言語に(中間的ではあるが、機種に依存しないように)翻訳移行する事を考えた。こうすると、大部分、機種に依存しないで開発作業を進めることができる。この中間語(FORTRAN)への翻訳移行に引き続いて行なわれる機種に依存する処理部分については、今回の発表ではRT-11で実現している。このように、MUMPSで書かれたプログラムをFORTRANに翻訳移行し、他のOSのもとでジョブを実行さす一連の流れをMUMPS to FORTRAN SYSTEM(MFTS)と我々は呼んでいる。今回は、MFTとその後の処理過程の実行時間等を示しながら、この方法によるバッチ処理でのス

ビード化について述べる。

2 MFTS の概要

MFTS は、汎用目的のバッチ処理ツールで、それは、次の 5 つから構成されている。それらは、プログラミング・システム (PS), MUMPS to FORTRAN トランスレータ (MFT), FORTRAN コンパイラ (FC), FORTRAN を有する OS (FOS), グローバル等のハンドラ (HL) である。MFTS の主要部である MFT は、DEC・MUMPS-11 で書き、HL は RT-11 のアセンブラ言語で書いた。又、PS は MUMPS-11 VO4B を使用し、FC, FOS は RT-11 の FORTRAN コンパイラ及び、RT-11 オペレーティング・システムを用いた。

2.1 MFTS 処理の流れ

ソースプログラムは、後述する多少の制限下で MUMPS-11 で書かれ、最終的にバグのない完全なプログラムと成っていなければならない。MFT は、この MUMPS のソースプログラムを FORTRAN プログラムに変換移行する。変換されたプログラムは、FORTRAN コンパイラによってコンパイルされ、必要な HL をリンクして実行形式のプログラムを作る。この実行形式を得るまでの流れを Fig.1 に示す。

2.2 MFT の概様とその変換機能

MFT は、通常のコンパイル開発の手法に従った。MFT 自身は MUMPS-11 VO4B で書かれており、それは構造化されている。その本幹のプログラムリストを Prog.1 に示した。MFT の主なルーチンは、構文解析ルーチンと FORTRAN コード発生ルーチンとである。この 2 つのルーチンの中心となるプログラムをそれぞれ Prog. 2a, Prog. 2b に示した。また、前者は、コマンド解析や引数解析、ローカル変数・グローバル変数のテーブル作成ルーチン等から成り立っている。その代表として、コマンド解析ルーチンのプログラムを Prog. 3a に示す。これらのルーチンは、多くのサブルーチンを従えており、構文解析ルーチンならびにそのサブルーチンは、60 本のプログラムから成る。FORTRAN コード発生ルーチンは、定義域コード生成、初期値設定 (For, Do, Call 等の)、算術演算コード生成、命令コード生成、グローバル変数指示等のルーチンから成る。その代表として、初期値設定プログラムを Prog. 3b に示す。FORTRAN コード発生ルーチンは、ほぼ 20 のルーチンまたはサブルーチンから成る。このルーチンは、上記の構文解析ルーチンの出力を使って FORTRAN プログラムを作り出す。現バージョンの MFT は、その変換能力にいくつかの制限があり、直接モードのコマンドはその変換の対象外である。つまり、編集コマンド (Modify, Erase, Load, File) やデバックキングコマンド (Break, Go) は、ソースプログラムが完全なものであることを仮定しているので変換されないし、タイムシェアリングコマンド (Lock, Unlock) や Strage 変数は、MFT がバッチ処理を目的としているので必要と見なされない。MUMPS のシステム変数は、変換の対象とはしなかったが、\$X 座標、\$Date、\$Time、\$Address は同様

な機能が RT-11 OS にそなわっているもので、翻訳の対象にした。さらに、制御コマンド (Start, Hang), I / C コマンド (Write), システム I / C コマンド (View) と関数 (\$ Altercase, \$ Next, \$ Step, \$ View) は、変換しない。naked のグローバルの使用には次の制限がある。即ち、サブルーチンの中、ならびに Do に従うライン・パートの中で通常のグローバルの先行使用がない限り、又、サブルーチンや Do に従うライン・パートの後のグローバルの先行使用がない限り naked のグローバルは使えない。関接指定は、その実行内容がループのたびごとに変わる時には使用できない。Do や Goto のコマンドは、その引数に変数であってはいけない。

3 バッチ処理における MFTS の有用性

バッチ処理における MFTS のいろいろな機能をその機能ごとに COBOL, FORTRAN, MUMPS 並びに BASIC と実験的に比較した。

3.1 プログラミング・ターンアラウンド

MFTS では、ソースプログラムは (MUMPS である) PS で書かれる。剖検報告の登録や剖検診断の度数表作成のバッチ処理のプログラム²⁾³⁾は、この PS によって作られた。その時、トランザクションデータのチェックプログラムのミスを避けるために、COBOL で同様なチェックプログラムを作った。それぞれのプログラムのターンアラウンドは、PS で 9 日、COBOL で 4 週間であった。

このプログラミング競走で下記の如き重要な事実が判った。この仕事は、日本病理学会のもとで行なわれたが、学会からの仕様は、その細部でしばしば変り、たび重なる作業の変更と修正を余儀なくされた。たとえば、疾病の分類、入力方式、出力様式等はたびたび変更された。COBOL のプログラマは、このたび重なる要求の修正に追いつけなかった。

3.2 コアメモリでの実行時間

このシステムのコアメモリ上での計算時間の違いを見るために、FORTRAN, BASIC, MUMPS の各システムで次の式

$$\sum_{i=1}^n (x_i - \bar{x})^2 / n, \quad (n = 10)$$

を、1,000 回試算した。MFT の変換時間、コンパイル時間、リンク時間は、それぞれ 390, 16, 25 秒であった。各言語におけるプログラムの実行時間は、FORTRAN, MFTS, BASIC, MUMPS でそれぞれ 5, 14, 44, 278 であった。(Table 1)

さらに、このシステムでの算術演算以外のコア内処理時間を調べる目的で、MUMPS-11 の処理と DSM-11 の処理を比較した。実験には、前述の剖検処理のトランザクションデータのチェックに用いたプログラムを加工して用いた。原プログラムから、グローバル、ファイルにアクセスする部分を除いて上記の剖検処理 1 年分のデータのチェックのみを行なった。この状態で 21 時間を必要とした。また、DSM-11 で書かれた同様の

グローバルファイルへのアクセスなしのチェックプログラムでは、17時間を要した。又同じことをMFTSを用いるとメモリにおける処理時間は3時間であった。(Table 2)

3.3 グローバルアクセス

グローバルアクセスの時間を評価する目的で、先に述べた剖検診断の登録をチェックするルーチンを前とは逆にとり除いて、MUMPS-11, DSM-11, MFTSの比較をした。結果は、それぞれ、7時間、2時間、6時間であった。

3.4 総合実行時間

1年間の剖検例の処理時間(Table 2)は、MUMPS-11の28時間、DSM-11の19時間であるのに対し、MFTSの実行時間は9時間であった。つまり、前2者に比べ、それぞれ0.32, 0.47倍の時間しかかからない。

Table 2のデータから、将来のMFTSの総合実行時間を予測してみる。我々の手で開発検討が始められている新しいMFTSは、標準MUMPS化され、当然Bayer's tree⁴⁾をグローバルとして持つことを前提とすることになるであろう。この場合、メモリ上での実行時間は、Table 2のMFTSのそれとほぼ等しく、グローバルアクセスの時間はこの表のDSM-11のそれとほぼ等しいであろうから、新しいMFTSの年間の剖検例の処理時間は、5時間になると考えられる。従って、MUMPS-11に比べて処理は6倍ほど速くなると予測される。

4 考 按

MFTSは、MUMPSコンパイラに比べはるかに機種依存性が低い。MFTSでは機種依存性を低くするためにすでにオーソライズされたコンパイラ言語にMUMPSを翻訳する道を選んだからである。FORTRANは、文字処理機能や作業機能において劣っていたり、グローバルファイルアクセス機能はないが、しかし、アセンブラで書かれたサブルーチンを比較的容易に呼べ、標準化規格化が進み普及しているのでMFTまで作っておけば、これから先の機種に依存するFOS, HLの設計が最も容易と考えられた。FORTRANを採用したことは、ユーザ、メーカー又は、システムハウスへの負荷を軽くしている。

我々のグローバルに関するHLは、ディスクアクセスの時間に対して最適化が計られてはいない。それにもかかわらず、MFTSのグローバルアクセスはMUMPS-11のそれより速かった(Table 2)。この矛盾した結果は、MFTSにおけるグローバルHLのI/Oバッファ領域が、大きいことに関係していると考えられる。

5 むすび

バッチ処理におけるMFTSの有効性は、緒実験で明白である。しかも、たとえば、多くの施設でかかえているレセプト発行に於けるMUMPSの遅さの問題の解決にすぐ応用

できる。

この様に有用かつ汎用性の高い道具をユーザ側が持つことになったのは、MUGによって初めての経験である。米国MDCが仕様を作り、メーカー側を引っ張って標準MUMPSができた様に、(幸い日本では、MUGの中に開発能力があるので)MUGがMFTを作り、メーカー側を引っ張ってMFTSの後半を作らせることによってMUMPSの永年の1つの夢が適えられることになろう。

6 文 献

- 1) Gries, D.: Compiler Construction for Digital Computers, John Wiley and Sons, Inc., 1971.
- 2) Kimura, S., Baba, K., Kimura, K. & Yamada, T.: Frequency taburation of autopsy findings of 23,085 cases registered in The Annual of the Pathological Autopsy Cases in Japan of one year by computer means. (abstract only) Dokkyo J. Med. Sci. 5(1) 96-97, 1978.
- 3) Kimura, S., Baba, K., Kimura, K., Urano, Y. & Aizawa, S.: Frequency taburation of pathological findings of more than 23,000 autopsy cases of annual collection through Japan. Proc. Conf. of MUG. 8, 1979.
- 4) Bayer, R. & McCreight, E.: Organization and maintenance of large ordered indexes. Acta Informatica 1, 173-189, 1972.

ACTUAL PROCEDURE TO EXECUTE THE JOB

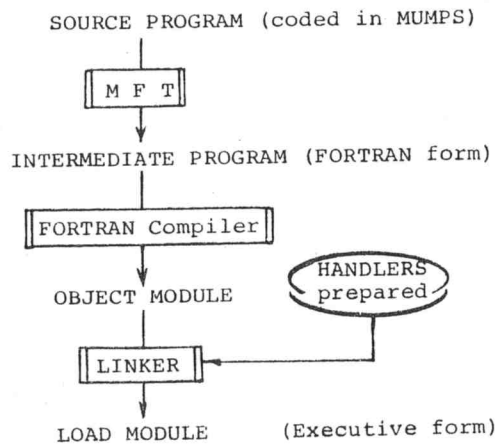


Fig. 1

EFFICIENCY COMPARISON WITH FORTRAN, BASIC & MUMPS
IN EXECUTION AND PREPROCESSINGS

	FORTRAN	MFT	BASIC	MUMPS
Translation	-	390 sec	-	-
Compiling	8 sec	16 sec	-	-
Linkage	15 sec	25 sec	-	-
Execution	5 sec	14 sec	44 sec	278 sec
(rate of above)	(0.36)	(1.00)	(3.14)	(19.9)

Above data are the time requirements of 1,000 times calculation of

$$\sum_{i=1}^n (x_i - \bar{x})^2 / n, \quad (n=10).$$

Table 1

COMPARATIVE TIME REQUIREMENT FOR
ON-CORE PROCESSING AND DISK ACCESSING

	On-core Processing	Disk Accessing	Total
MUMPS-11 V04B	21 hr	7 hr	28 hr
DSM-11	17 hr	2 hr*	19 hr
Present MFT	3 hr**	6 hr	9 hr
MFT $\bar{c}$ B-tree	3 hr	2 hr	5 hr

In order to estimate the time requirement of new MFT with Bayer's tree, * and ** are applied for data of the lowest line.

Table 2

Prog. 1

```

MFT
0.10 ; ***** MFT (MUMPS-FORTRAN TRANSLATER) *****
0.20 ; CSK-HIROKAWA,MFT SYSTEM,1978/9

11.01 K K +MVG, +MVL, +MVC, +MVA, +MIC, +MSB K S MRC=0
11.02 T !"MFT0011 : MFT START"
11.03 D 21
11.04 I MRC=0 C L
11.07 I MRC=0 D 22
11.10 T !"MFT0021 : MFT END"

21.02 S PKC=0 D 31
21.03 I MRC=0 D 32

22.02 D 33
22.03 I MRC=0 D 34

31.02 D 41
31.03 S MPR=MMP C P

32.03 D 42 0:MPR=""
32.05 C P G 32.03

33.01 X MRC,MPR,MMP,MCP
33.02 S MPR=MMP C C

34.03 S MRC="C" D 91 0:MPR=""
34.05 C C G:MRC=0 34.03

41.03 P !"MFT003P : TYPE PROGRAM NAME : ",MMP
41.04 G:4D(PPRG(4C(MMP)))=0 41.03

42.06 S MRC="P" D 91 0:MPR=""
42.12 0:4D(PPRG(4C(MPR)))<>0
42.14 T !"MFT005E : ",MPR," PROGRAM NOT EXIST"
42.16 S MRC=4 G 42.06

91.03 S:MRC="P" MLC=4
91.04 S:MRC="C" MLC=5
91.06 S MPR="",MID=0
91.08 S MID=4H(MSB(MID)) 0:MID=-0.01
91.10 S M01=+(MID) I $E(M01,MLC)=0
91.11 I S MPR=$E(M01,1,3)
91.12 I S M01=$E(M01,1,(MLC-1))@"1"@$E(M01,(MLC+1),132),+(MID)=M01 0
91.14 G 91.08

```

PROGRAM USES APPX. 1075 BYTES 0.54 K WORDS

Prog. 2a

```

P
0.10 ; ***** P (KOBUN-KAISEKI-SHORI) *****
0.20 ;          CSK-HIROKAWA,MFT SYSTEM,1978/9

11.02 K +MDO,+NOT S MCP=$C(MPR) D 21
11.04 I MRC=0 D 22

21.02 S PPN=$C(MPR),PSN=0,PDC=0 T !"MFT0041 ; ",MPR," MUMPS SOURCE LIST"
21.03 S PLN=0,+MVC(MCP,$C("MFA"))="MFA0"
21.06 D 31 Q:PST=""
21.10 D 32
21.18 I MRC=0 D 34
21.22 G 21.06

22.04 C:PDC>0 PDO
22.09 C PQT

31.02 S PST=""
31.04 S PSN=$H(+PRG(PPN,PSN)) Q:PSN<0
31.08 S PST=+(PSN)
31.10 T !,PSN," ",PST

32.02 S PFC=0,PIC=0,PCL=0,PCN=0,AVN=0
32.06 C PGC Q:PCO=""
32.10 I $E(PCO,1)=";" C ACM Q
32.16 C A G 32.06

34.04 C:PIC>0 PIF

```

PROGRAM USES APPX. 737 BYTES 0.37 K WORDS

Prog. 2b

```

C
0.01 ; ***** C (CODE SEISEI) *****
0.02 ;          CSK-HIROKAWA,MFT SYSTEM,1978/9
0.03 ;          PUNCHED BY S.KIMURA,7-OCT-78

11.02 S MCP=$C(MPR),CGN=0
11.04 T !"MFT0281 ; ",MPR," FORTRAN TRANSLATED LIST"
11.06 C CI,CP,CE

```

PROGRAM USES APPX. 433 BYTES 0.22 K WORDS

Prog. 3a

```

A
0.01 : ***** A (COMMAND KAISEKI-SHORU) *****
0.02 :          CSK-HIROKAWA,AFT SYSTEM,1978/9

11.06 I 4E(PCD,1)="A" C AA 0
11.08 I 4E(PCD,1)="B" C AZA 0
11.10 I 4E(PCD,1)="C" C AC 0
11.12 I 4E(PCD,1)="D" C AD 0
11.14 I 4E(PCD,1)="E" C AE 0
11.16 I 4E(PCD,1)="F" C AF 0
11.18 I 4E(PCD,1)="G" C AG 0
11.20 I 4E(PCD,1)="H" C AH 0
11.22 I 4E(PCD,1)="I" C AI 0
11.24 I 4E(PCD,1)="K" C AZA 0
11.26 I 4E(PCD,1)="L" C AZA 0
11.28 I 4E(PCD,1)="M" C AI 0
11.30 I 4E(PCD,1)="O" C AZA 0
11.32 I 4E(PCD,1)="P" C AP 0
11.34 I 4E(PCD,1)="Q" C AQ 0
11.36 I 4E(PCD,1)="R" C AR 0
11.38 I 4E(PCD,1)="S" C AS 0
11.40 I 4E(PCD,1)="T" C AT 0
11.42 I 4E(PCD,1)="U" C AU 0
11.44 I 4E(PCD,1)="V" C AZA 0
11.46 I 4E(PCD,1)="W" C AZA 0
11.48 I 4E(PCD,1)="X" C AZA 0
11.50 D 21

21.04 S MFC=4 T !"NFT008E ; ",PCD," COMMAND IS ILLEGALITY"
21.08 D 31

31.04 C PCC,PSA

```

PROGRAM USES APPX. 975 BYTES 0.49 K WORDS

Prog. 3b

```

CP
0.01 : ***** CP (SHORT-BU SEISEI) *****
0.02 :          CSK-HIROKAWA,AFT SYSTEM,1978/9
0.03 :          PUNCHED BY S.KIMURA,7-OCT-78

11.02 F Y05=100.01:1.01:100.03 S GB=$S(Y05) C GFG
11.04 S CSN=0
11.08 S CSN=4H(4HIC(MCP,CSN)) Q:CSN<0
11.22 D 21.22 G 11.08

21.04 S GB="C
21.06 C GFG

22.04 S CCN=-1
22.08 S CCN=4H(4HIC(MCP,CSN,CCN)) Q:CCN<0
22.12 D 31 G 22.08

31.04 C CP1
31.08 I COP=":" C GN 0
31.10 I COP=":" C GL 0
31.12 I COP="GOTO" C GG 0
31.14 I COP="IFGO" C GI 0
31.16 I COP="DOS" C GDS 0
31.18 I COP="DOR" C GDR 0
31.20 I COP="CALL" C GO 0
31.22 I COP="QUIT" C GO 0
31.24 C GA

50.01 C

100.01 C
100.02 C
100.03 C

***** PROCESS *****

```

PROGRAM USES APPX. 880 BYTES 0.44 K WORDS



第6回日本MUG学術大会 昭和54年9月14日(金)

C. パネル(2) MUMPSの互換性 司会 MUMPSシステム研 嶋 芳成

C-1. 総 論

マンプスシステム研究所 嶋 芳成

§1 互換性の必要性

1967年頃から米国ボストン市のMGH (Massachusetts General Hospital) で開発のはじまったMUMPSは、爆発的に大学病院、医学部に広がり、1972年までに、オリジナルのMGH-MUMPS 以外に次のような7つの方言が派生してしまった。

- ・ MIIS (Meditech Interpretive Information System ; Meditech, Inc.)
- ・ DEC MUMPS-11 (Digital Equipment Corporation)
- ・ DEC MUMPS-15 (")
- ・ AHS MUMPS-11 (Automated Health Systems, Inc.)
- ・ AHS MUMPS-15 (")
- ・ ITA-MUMPS (InTerAx, Inc.)
- ・ MUMPS-PC (Biomedical Computer Laboratory および Artronix)

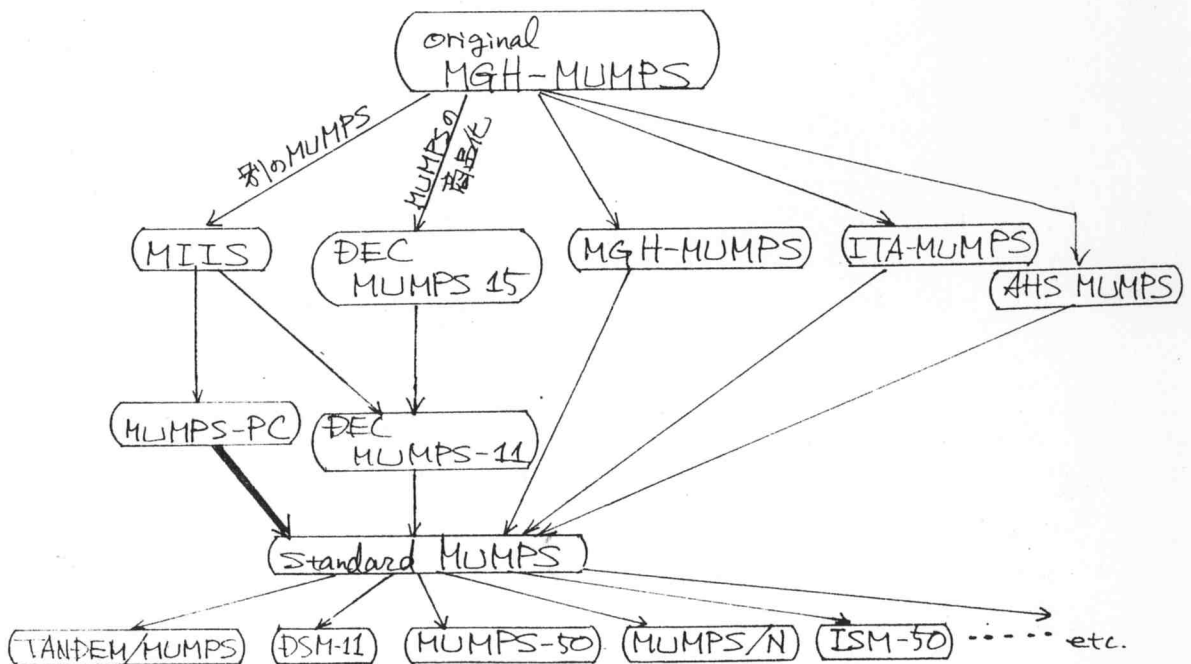
この内で特にMIISの出現は、行ラベル、間接指定および後付け条件を導入し、論理・算術・文字列の各演算の混合演算を許すようになったという点で、MUMPSにとって画期的なことであった。このようないくつかの方言が生まれたということは、その後のMUMPSの発展を示唆するものであったが、医療関係のユーザーにとってめどく深刻な問題を引き起こしてしまった。MGHで開発された細菌学システムをはじめとする有用な医学的アプリケーションを、“MUMPS”を導入した他の医療施設に移殖しようとしても、同じ“MUMPS”でありながら微妙に異なる文法のためにコードを翻訳せねばならず、その作業に膨大な時間がかかることになったのである。

コンピュータのアプリケーションプログラムとは、人間のある組織あるいは作業システムという実体を、コンピュータ言語という特殊な言語で記述したものであるといえる。医療にコンピュータを導入するということは、従って、そのみによって新しい

医療行為を生もうとするのではなく、「合理化」により、不必要な繁雑さを減らし、医療従事者の直感と慣習により行われてきた行爲に、反省の光をより深く当てらるようとして、医療の質を向上せよとする努力のはずである。ゆえに医療の場で開発されたアプリケーションは医療に属し、そのまゝ人類の福祉に継がなくてはならない。もしそのようなアプリケーションがメーカー・ベンダーに独占されることにより、同様なアプリケーションを各医療施設毎に同じように高額の費用を払って開発・購入しなくてはならないとすれば、福祉全体の費用を増大させ、さらに福祉の普及を徒らに遅滞させることとなる。すぐれたアプリケーションを早く普及させるには互換性の保証が不可欠である。また、医療行為そのものを助けようとするアプリケーションがある場合、そこに実現されている医学理論が正しいものであるかどうかという検証を常にこなしてゆかなくてはならない。アプリケーションの交換はこの作業にと、てもまた不可欠である。もし各医療関係者が互に非なる方言で話すことになれば、医療という巨塔はバベルの塔のように崩壊することになる。

§2 実際のプログラムコードの比較

同じMUMPSで実際のどの程度コードが異なるのであろうか？ 先にMUMPS開発の流れとその内の4つのMUMPSのコード例を示しておく。



SAA D \$S=5.46 DELTA D 88/75 8:18

0.03 SAA,DCJ

0.10 STAT: DATA FROM KB

0.20 C BY KB PGS

```

1.01 S XY=0
1.02 I $G(DEC) O SAB
1.05 S %TE="ENTER 'YES' IF WANT X,Y DATA BASE " C SGR I R:"Y"U S XY=1
1.10 S %TE="DBASE #: " C SGR I R:" " O SAP
1.20 S STX="_I" C CAB I R=" " G 1.1
1.25 S K=$V(R)
1.30 I $G(^ZZ[0]($P,K))&$G(^K,B))=2 T " ". $P(^ZZ[0]($P,K)_1) G 1.9
1.35 S %TE="NAME: " C SGR I R=" " G $L
1.37 S R=%Y I XY S R=R."_XY"
1.40 S ^ZZ[0]($P,K)=R
1.50 S I=0 G 2
1.90 S XY=$P(^K)_2,XY=$B(XY="XY",1;1,0),I=^K,B) G 4
=====
2.10 S I=I+1 I $X>55 T !
2.20 S %TE=" " T $C(I), " " C SGR T " " I R=" " S I=I-1 G 4
2.30 I R=" " S I=$B(I=1,1;1,I-1) K ^ZZ[0]($P,K,I) G 2.2
2.35 I XY D 8 G $B(R=" ",2.2;1,2.5)
2.40 D 3 I R=" " G 2.2
2.50 S ^ZZ[0]($P,K,I)=R G 2
=====
3.20 S D=" " I R=" " U S R=$E(R,2,$),D=" "
3.30 I $L(R)>9 T "TOO LONG " S R=" " Q
3.40 I R:1NN!R:N". "1NN S R=D.R Q
3.50 T "NUMBERS ONLY " S R=" "
=====

```

↑ MGH-MUMPS

↓ DEC MUMPS-11

0.10 PROGRAM ABC

0.20 ACID-BASE CALCULATION

0.30 DIRECT PCO2 METHOD

0.40 PROGRAMMED BY YOSHINARI SHIMA

```

1.10 R !,"ACT. PH : ",H,!,"ACT. PCO2 : ",PO,!,"HEMOGLOBIN : ",R
1.20 S X=PO D 82,80 S P=Y,E=-350,D=100
1.30 S E=$M(E+D),B=$M(0.42*R+41.6+E)
1.40 S X=$M(B-11.6) D 80 S C1=$M(Y-0.0453)
1.45 I $M(C1=2.111) T !,"*DIVER>",$L G 12
1.50 S H1=$M(-.315/(C1-2.111)+6.628),P1=$M(C1-H1+7.622)
1.60 S X=$M(E+27.6) D 80 S C2=$M(Y-0.0607)
1.65 I $M(C2=1.799) T !,"*DIVER>",$L G 12
1.70 S H2=$M(-.132/(C2-1.799)+7.084),P2=$M(C2-H2+7.622)
1.75 I $M(P2=P1)!$M(H=H1) T !,"*DIVER>",$L G 12
1.80 S Z=$M(P-P1*(H2-H)/(P2-P1)/(H-H1))
1.90 G 2

2.10 G 2.5:$M(Z>0),3:$M(Z=0)
2.30 I $M(E<-25) T !,"B.E.<-25 ! " G 12
2.40 S E=$M(E-D),D=$M(D/10) G 3:$M(D<0.01),1.3
2.50 G 1.3:$M(E<30) T !,"B.E.>30 ! " G 12

3.01 K Y,%,X0,XY
3.05 I $M(P1=P2) T !,"*DIVER>",$L G 12
3.10 S H4=$M(H2-H1*(1.60206-P1)/(P2-P1)+H1),X=$M(H4-6.02)
3.20 D 92,90 S C4=Y,C5=$M(P+H-7.622),X=C5 D 90 S C0=Y
3.25 K Y,%,X0,XY
3.30 S T=$M(0.03*PO+C0)
3.35 S X1=C0,X2=C4,X3=T,X4=E,X5=B,I=5 C ABF
3.36 S C0=Y(1),C4=Y(2),T=Y(3),E=Y(4),B=Y(5)
3.37 K Y,X1,X2,X3,X4,X5,I,C5,H4,P1,P2
3.40 T !,"ACT. HCO3 : ",CO,!,"STD. HCO3 : ",C4,!,"TOTAL CO2 : ",T
3.50 T !,"BASE EXCESS : ",E,!,"BUFFER BASE : ",B

12.10 T !,"INPUT ERROR" S ERR=1

```

```

EQU      ; 03:33:19 09-AUG-79 ;
B 0: TEXT PRINTER RFB3
I $Y>50 D P
I '$D(J) S J=1
I '$D(MM) S MM=$P(%F,2)
I J=1,MM "*" S MM=$P(%F,2) I $X W !
S SS=-3*$D(MM)+%F,JJ=J+0,"",UU=
A      I $D(TT(J)) G B
D K S UU=J,UU= I _"$D(^".GG."(I,J))" D:UU=JJ XER K J,UU Q
S      S TT(UU)=^(UU) I TT(UU)'="",$S>175 S UU=UU+1 G:$D(^UU)) S S TT(UU)="-"
S      S XX=TT(J),J=J+1 I XX="-" D K K J Q
B      G:XX?'-"U H:%F,R
S      S WW=XX,ZZ=$P(WW,2),UU= D 0 G:$D(TT(J)) N:TT(J)="-",J
I      I _"$D(^".GG."(I,J))",^(J)'="-" G J
N      K MM 0 KEY:F=10,EQU=$P(WW,4),EQU
J      S JJ=JJ,J G N
H      I $Y>56 D P
I      I XX "$" W _$P(XX$1) S XX=$P(XX$2,9)
I      I XX?N") "U,MM' "*" S MM=MM+4."*"
W      W ?MM
R      S YY= I $D(FF)=2,J=2,'$D(WW) S ZZ="( ".I. " " D 1 S UU=
L      S YY=YY+1,ZZ=$P(XX_1),XX=$P(XX_2,99) D _YY#2.
G      G:XX'? L I UU?',%F,$X-MM W " "
G      G A
0      I ZZ " " X ZZ S TT= Q
I      I ZZ?"^"N G: '$D(^EY($I,$P(ZZ^2))) X S ZZ=^($P(ZZ^2)) G Q
I      I _"$D(" ZZ ") " S ZZ=
S      S _"ZZ=" ZZ,ZZ=ZZ.
Q      G:ZZ? X I ZZ?"*U X $P(ZZ$2,99) S TT= Q
1      Q: '%F S UU=$P(ZZ 1) I SS-$X>$L(UU)!(UU?1P) W UU G 2
W      W ! D:$Y>56 P W ?MM "*"&3+MM,UU
2      Q:ZZ' " " W:$X-MM " " S ZZ=$P(ZZ 2,99) G 1
K      K TT,UU,UU,XX,YY Q
P      Q:$A&3=1! '%F W ! G:$A&3=0&$Y#66 P W # Q
X      S XX=$E(XX,UU?&$H(XX)=32&2,99)
Z      Q
Z      F EQU P Z

```

↑ MISC

↓ MUMPS/N

```

%ZEP      ;KO,,EDITOR; 13:17:26 20-JUN-79 ; 'ZP' COMMAND
;
G ^%EERR:$P(%CMD," ",1)'="ZP",EX:'$D(%E) S %ARG=$P(%CMD," ",2)
I %NOL<21 D RS F %I=1:1 D WRITE G EX:%I=%NOL
I %ARG="" S %LF=1 D RS G DRU
D ^%ELINE G ER:%LE S %LF=$S(%L(1)C"#":$E(%L(1),2),1:%L(1)%15+1) D PRS
;
DRU      F %I=%LF*15-14:1:%LF*15 D WRITE Q:%I=%NOL
R      R %Z
I      I Z=13 S %LF=%LF+1 G EX:%LF*15-14>%NOL,DRU
I      I Z=10 S %LF=%LF-1 G:%LF EX D PRS G DRU
I      I 48<Z,Z<58 S %LF=Z-48 G:%LF*15-14>%NOL EX D PRS G DRU
EX      K Z,%K,%LF,%PRS,%CMD,%ARG,%L,%LE,%LEM Q
;
WRITE     W !,$J(%I_": ",4),$P(%E(%I)," ",1),?14 S Z=$P(%E(%I)," ",2,255),%L=$L(Z)
I      I '($I#256=33) F %K=0:64 W $E(Z,%K+1,%K+64) Q:%L-65<%K W "%",!,?13,"%"
E      E F %K=0:120 W $E(Z,%K+1,%K+120) Q:%L-120<%K W "%",!,?13,"%"
Q
;
PRS      W:('$I#256=33) #,"PAGE #",%LF,?55,"<",&RTN,"","#,%NOL,">" Q
RS      W:('$I#256=33) #,?55,"<",&RTN,"","#,%NOL,">" Q
;
ER      W !,%LEM G EX

```

この例を見ると、ルーチン行を識別するために、MGH-MUMPSとMUMPS-11ではステップ番号を用い、MUISとMUMPS/Nでは行ラベルを用いていることが目立っているが、詳細には様々な相異点がある。前2者の出力命令はTYPEであるが後2者ではWRITEである。連結算符は、MGH-MUMPSとMUISでは「.」、MUMPS-11では「@」、MUMPS/Nでは「_」を用いている。MGH-MUMPS以外では、

$G \sqcup 1\phi : X=1, 2\phi : X=2, 3\phi$

という後付け条件を用いることができるが、MGH-MUMPSではこれを、

$G \sqcup \phi B(X=1, 1\phi; X=2, 2\phi; 1, 3\phi)$

と書くてはならない。さらに、MUMPS-11とMUMPS/Nでは、

$\phi P(X, "/, 2, 4)$

と書くところを、他の2者では、

$\phi P(X/2, 4)$

と書くことができる。

このように構文が異なる場合のほか、構文が同じでも解釈法が異なる場合もある。本年2月の第9回日本MED学会専門別研究会MUMPSソフトウェア研究会で発表したように、

$S \sqcup I = 5 \sqcup F \sqcup I = 1 : I : 2\phi \sqcup W \sqcup I$

というコードに対し、これは標準MUMPSでは、161116と出力されるわけではなく、国産MUMPSでは、

161116
1234567891011121314151617181920
124816

という3種類の結果を出している。

大筋が同じ仕様であっても、細かい相異点を重ねることによって互いの変換は困難となった。

§3 標準MUMPSの出現

方言化の事実とその弊害に早くも気付いた米国のMUMPS ユーザー30人が、1972年ボストンに集まり MUMPS Users' Group を結成し、メーカーに対して各方言を交換できる形、標準MUMPS に収斂させるように迫った。このとき、米国の健康教育福祉省 (Department of Health, Education and Welfare) は、1) MUMPS は医療分野に適しているか、2) もし適しているならば如何なる施策をとるべきかという2点について調査を行ない、その結果 1) は肯定され 2) については MLUG と MDC (MUMPS Development Committee) を財政援助することが必要であるとの結論を出した。そして、1973年3月に健康教育福祉省と連邦標準局 (NBS; National Bureau of Standard) の援助の下に MDC が結成され、精力的な活動を開始した。MDC は各方言を分析し、ユーザーの要求とメーカーの妥協の結果、2年半後 1975 年には「MUMPS Language Standard」にまとめ、これは NBS から HANDBOOK 118 として出版された。さらに 1977年9月にはアメリカ国家規格協会 (ANSI; American National Standard Institute, Inc.) が国家規格 (ANSI X11.1-1977) として認可するに至った。これは COBOL, FORTRAN に次いで3番目のコンピュータ言語であった。

その後続々とこの標準MUMPSを“標榜する”表簿が発表され、本年米国 MLUG 発行の MLUG Quarterly に掲載されている“標準MUMPS”を装束している機種は次の通りである(日本のものは除く)。

PDP-11(2種), HARRIS, IBM360/370, Intel 8080, Motorola 6800, NOVA/MICRONOVA, PDP-10, TANDEM/16, Burroughs B6700, ECLIPSE, PRIME, MODCOMP, Philips P856/857, Modulex

(注: これらの表簿が本当に標準MUMPSか否かは未確認である。少なくとも、Philips はまだ標準仕様を満たしていないと言っており、Intel 8080, TANDEM, Burroughs, Modulex などについても誤りがいくつか指摘されている。他の表簿もそのような可能性がある)

日本では 1977 年 ANSI が認可するに先立って、医療情報システム開発センター (MEDIS-DC) が、医用データ・ベース言語の中核として標準MUMPSの採用を求めた。そのプロジェクトとして、中電の OKITAC 50/40 に MUMPS-50 を装束した。これとは別に日本シニコンは ECLIPSE/NOVA に標準MUMPS として MUMPS/N を装束し、東芝は DEC の MUMPS-11 の影響の下に、MUMPS 900 を装束した。その後三菱、パナファコム、日本電産、日立などが次々に参入を上げており、これらはすべて標準MUMPSを“標榜”している。

§4 互換性・移送性に及ぼす種々の因子

- 1) このように標準MUMPS装備が実現することによってどのような効果が現われるであろうか？ 我々の経験では、前記のMGH-MUMPS, MUMPS-11, MIIS から標準MUMPSへの用手的な移送は費用がかなり過ぎ、実験的なもの以外では現実的ではない。それに対して、標準MUMPSであれば機種・メーカーが異な、ても非常に楽に移送できることが経験されている。標準化の威力にはまさに目を見張るものがある。しかし、装備依存的部分以外に手を加えなくともアプリケーションが移送できるというためには、各装備が非常に厳密に標準仕様に従っている必要があることに注意しなくてはならない。
- 2) 例えば米国から送られてきた標準MUMPSプログラムを、ある国産MUMPSに移送しようとした際、その装備はほとんど標準に近かったのであるが、中HOROLOGがなく、未定義変数をKILLするとエラーとなるという2点が標準MUMPSから外れていたため、数十本のルーチンすべてを検索してそれらを修正変更しなくてはならなかった。また別の例では、時間制限がないという点のため、教育用プログラムを完全には移送できないでいる。
- 3) このような経験から標準MUMPSを標榜する装備の傾向をみてみると、仕様が標準から外れる要因として2つのことが言える。ひとつは理解の不足によるもので、ANSI X11.1-1977; MUMPS Language Standardの一部を読み落としていたために誤った解釈をしてしまうということである。もうひとつは、標準MUMPSの仕様が複数の人々の主張と妥協の中から生まれてきたものであるため、見方によっては奇妙であったとか不明確であったと思われる点がいくつかある。それを口実にして、装備者は仕様の細かい点を「より良い」形で装備しようとするということである。例えばHOROLOGは1840年12月31日という年月日から起点となっているがこれは中途半端で使えないので、1899年12月31日を起点にするとか、あるいは拡張算術関数は必ずしも中下の方が納得しやすいのでそれを用いるとか等々である。これらの結果、「標準MUMPS」を標榜してながら微妙に異なる装備が出現する。そのことを知らずにその「標準MUMPS」を導入したユーザーはその微妙な違いに、それが厳密であるが故になおのこと、アプリケーション移送には苦しまなくてはならないという可能性がある。
- 4) 言語仕様のそうした微妙な相異点を検出できるプログラムとして、MUGにはMUMPS検定プログラムがあり、かなり複雑に入り組んだコードまでをその検定対象にしている。ユーザーがこのようなプログラムを用意してはならない大きな理由は、検定プログラムを装備者が作ったとしても、その検定内容は装備者のもつていたMUMPSの概念の範囲を越えることはできず、決して検定に通らないということが起らず、従って検定の意味がなくなるからである。MUGはこのプログラムを慎重に扱い、これを用いて、メーカーの参み寄りを得て標準化の指針としてゆくべきである。
- 5) 検定プログラムによって言語仕様の面では標準化は大きく進められるであろうが、しかし実際移送性を確保するには他にも要因がある。アプリケーションの移送は、多くの場合MT(磁気テープ)を用いることが多いが、同じMTであっても、トラック数

密度、コード体系、ブロック長などが異なれば、移送は不可能ある...は困難になる。
 このような物理的フォーマットが一致しても、さらに論理的フォーマットが異なれば、
 それと合わせた移送用のユーティリティプログラムをユーザーがいちいち書かなくては
 ならない。こういった点まで標準化が進められてこそはじめてMUMPSの互換性と
 移送性が確保されたと言えるであろう。

3.5 日本における特殊性

日本においては、前述のように通産省・厚生省共管のMEDIS-DCが、標準医
 用データベース言語の中核として標準MUMPSを採用したことは高く評価されるべき
 である。この決定が、日本でのMUMPS開発を大きく促進したのである。この医
 用データベース言語では日本的な拡張としてカナ文字をデータおよび識別名として用い
 ることができるようになっている。日本の医療にコンピュータを導入しようとする場
 合、カナ文字をデータとして扱う...得る...ということは不可欠なことであるが、これが識別
 名としても用いることができるれば日本人にと、てさらに使いやすくなるかもしれない。
 日本で現在手に入れたMUMPSはすべてカナ文字をデータとして扱えるが、カ
 ナのパターン照合やカナスペースの取り扱い、カナ文字を識別名に用いるか否かなど、
 種々の点で仕様が異なっている。従ってカナの仕様を標準化することは日本のMUMPS
 共同体にと、て急務であり、現在日本MUGの標準カナMUMPS委員会がこの作業を進
 めている。ユーザー・メーカーの協力と取り組みが望まれる。

さらに日本においては将来的にはコンピュータが漢字を扱うことが一般化するであ
 る。特にMUMPSのように人間とコンピュータの対話の多いシステムでは漢字の
 導入は不可欠であると考えられる。これをどのように標準化して互換性を保証してゆ
 くか...ということが大きな課題であり、そろそろ検討をはじめべき時期に来ている。

[参考文献]

- 1) Johnson, M.E.: MUMPS, A Preliminary Study, MUG, St. Louis (1972)
- 2) 若井 一郎: 医療情報のためのデータベース言語MUMPS, 医用電子と生体工
 学, 第14巻, 第6号 P442-451 (1976)
- 3) 若井 一郎, 嶋 芳成, 久保田 実: 各社MUMPS装備の現況, 日本ME学会専
 門別研究会マニフラスソフトウェア研究会資料 (1979)
- 4) Zapolin, R.E. 編: MUG Quarterly, Number 28 (1979)
- 5) Zapolin, R.E. 編: MUG Quarterly, Vol IX No. 2 (1979)
- 6) Conway, M.E. 著, 若井 一郎ほか訳: 標準MUMPS言語マニュアル, コロナ社, 1977

第6回日本MUG学術大会 昭和54年9月14日(金)

C. パネル(2) MUMPSの互換性 司会 MUMPSシステム研 嶋 芳成

C-2. 標準MUMPSの言語仕様について

日立製作所システム開発研究所 渡辺 坦, 大沢恒春

1. まえがき

プログラムの互換性は、どの言語でも大きい問題とされており、技術の交流や計算機の置き換えに際しては、多大の労力がプログラムの変換に費やされている。MUMPSの諸先達は、いち早くこの点を重視され、ANSIの標準規格を制定された。しかし、現実のプログラムを移行させようとすると、まだ種々の問題が生ずる。

筆者は過去にFORTRAN等の言語で書かれた多くのプログラムの機種変換を行なった。その量は約20万ステップにのぼる。これらの言語では、「計算機のアーキテクチャが同じで、かつ、言語仕様がマニュアルに書かれていない裏仕様まで同じでないならば、プログラムを移行させるには元のプログラムの処理内容を解釈する必要がある」という感じを抱いた。

MUMPSでは、標準規格により、互換性の保てる部分と保てない部分がはっきり区別されているので、ある範囲内のプログラムに対しては互換性を保証できる。以下では、主として、ANSIの標準MUMPSの仕様に合わせて作られたプログラムにおいて互換性の点で何が問題になるかを検討し、互換性を保たせるにはどんな注意をすればよいかを述べる。また、標準にはないがMUMPSシステムを運用するのに必要な機能は何かについても触れる。

2. 互換性に関する問題点

ANSIの規格(MUMPS Language Standard, ANSI X11.1-1977)で定められている標準MUMPSの仕様を付録に示す。これに従って書かれたプログラムは、標準MUMPSを装備している計算機の周では、おおむね互換性を持ちせることができるが、以下のような点については問題を生ずる。

2.1 規定対象外の仕様

(1) 直接モード・コマンド

ANSI規格では、直接モードのコマンドについては規定していない。しかし、プログラムの入力や修正、出力、ならびにグローバル変数の管理をするためには直接モードコマンドが必須である。MUMPSには、これらの直接モード・コマンドを実行させるプログラムを書く能力があるが、そのプログラムについては互換性が保証されない。

(2) カタカナ

ANSI規格にはないが、カタカナは日本では必須である。そのボタン照合用のコードはまだ標準化されていない。欧州等、他の国でもアルファベット26文字では不足な場合があるので、同様の問題があると思われる。

2.2 言語装備者の裁量に任された仕様

以下の事項については、それをどう定めるかは、言語装備者に任されている。

- (1) `ls`, `eol`, `eot` の内部表現。
- (2) `$STORAGE` の値の定め方。
- (3) `$Z` で始まる特殊関数。
- (4) `$Z` で始まる特殊変数。
- (5) `OPEN`, `CLOSE`, `USE` における装置パラメータ。
- (6) `READ`, `WRITE` における*印つき要素。
- (7) `READ` における入力文字列の区切り。
- (8) `WRITE` におけるデータ要素間の区切り。
- (9) `$VIEW` 関数と `VIEW` コマンド。
- (10) `Z` で始まるコマンド。

これらの機能を使えば、他の高級言語では望めないシステム自身の記述をMUMPSで行なうことができる。しかし、そのプログラムについては互換性が保証されない。

2.3 装置の特性に依存する仕様

標準MUMPSで規定されているが、装置の特性を考慮すると、厳密な形では規定を満たせない場合がある。以下の事項はその例であり、これらも互換性の上で問題となる。

(1) データ表現

標準では、すべてのデータは文字として（あるいは文字に変換された形で）表現形式を定められているが、現在の計算機では、数値は内部で2進数として表現するものが多い。浮動小数点数のときは、内部表現形式の差異によって、機種を変えると結果の異なる場合が生ずる。これは、どんな言語でも問題となる点である。

(2) プログラムの表示形式

標準では、1行の長さは255文字までとなっているが、現実の装置では、255文字までを1行として扱うものはほとんどない。MUMPSでいう1行のコマンド列を何行かに分割して表示するとすれば、継続行をどう表現するかが問題となる。

(3) 桁位置と行番号

1行の文字数と、1ページまたは1画面の行数は、装置によって異なるので、`#$X` と `#$Y` の値を利用する場合、装置を変えると所期の目的と一致しない場合が生ずる。

(4) USE指定

`USE` コマンドは、入力と出力を1台で兼ね備えたタイフライタ型の装置に対しては非常に便利な指定であるが、テープやディスク等、入力と出力を一時に混合しては使えない型の装置であると、あまりうまく使えない。また、CRTに表示すると同時にライ

ンプリンタにも出力したい場合なども、特別な工夫をすることになり、互換性の保証がなくなる。

2.4 その他の問題点

標準に従って作り、互換性の点で問題となる使い方を避けたプログラムであっても、他で利用しようとする、さらに次のような問題が発生する。

(1) 名称の重複

MUMPSでは、変数名やルーチン名は一旦作られるとKILLされるまで有効である。したがって、他で作られたルーチンを自分のプログラムにとりこんで利用しようとする場合、そのルーチン内で使っている変数名やルーチン名を自分のプログラムの中で別の意味に使っていないかどうか調べる必要がある。もし、意味の異なるものの間で名称の重複があれば、どちらかの名称を変えるかKILLする必要がある。また逆に、とりこむルーチンとの間での受け渡しデータは、ローカル変数またはグローバル変数として受け渡ししなければならぬので、それに対する修正の必要が生ずることがある。

(2) データ・ファイル、プログラム・ファイルの形式

プログラムを移すとき、それを改めてキー・インするならば問題はないが、磁気テープや紙テープ、フロッピー・ディスク等に入れて移すのであれば、その記録形式が問題となる。すべてを文字列として扱うにしても、磁気テープやフロッピー・ディスクでは制御情報が必要なので、それをどんな形にするかで扱いを変えなければならない。

入力データやグローバル変数の値を移す場合には、さらに、各データの識別しかたや区切り方が問題となる。

3. 共通プログラムの書き方

3.1 標準仕様により共通化できる範囲

前章の議論から、どんな書き方をすれば共通に使えるプログラムとなるかがわかる。すなわち、次の要件を満たすプログラムならば、標準MUMPSを装備している機種間で共通に使えることになる。

- (1) 間接モードのコマンドのみを使う。直接モードのコマンドを入力として受けつけたり、@印つき間接指定や\$TEXTでとり出したり、EXECUTEコマンドで実行させたりしない。
- (2) カタカナは、標準カナMUMPS委員会が標準が定められる迄は、(単なる特殊文字の一種としてしか)使わない。
- (3) コマンドや行、ルーチンを区切る1s, eol, eorは、それぞれ、SP, CR LF, CR FFとして扱う。
- (4) #STORAGEの値は、99%の誤差があっても結果に影響がないようにする。
- (5) OPEN, CLOSE, USEにおいて、装置パラメータを利用しない。
- (6) #Zで始まる特殊変数と関数、コマンドを使わない。
- (7) READ, WRITEにおいて、*印つき要素を使わない。
- (8) READコマンドによえる入力文字列を、出力として作り出すことをしない。

- (9) WRITE コマンドで書かれた出力文を列を入力として読むことをしない。
- (10) 常VIEW 関数とVIEW コマンドを使わない。
- (11) 入力、出力、演算結果として、有効数字9桁を越えるもの、範囲 $10^{25} \sim 10^{-25}$ を越えるものを扱わない。
- (12) 小数点以下の最終桁に演算に伴う誤差があっても有意性に変化を来たさないようにする。
- (13) 継続行を示す印は使わない。
- (14) 1行の文字数、1ページの行数が使用装置によって変わっても、有意性に変化を来たさないようにする。

MUMPS の用途としては、エンド・ユーザが応用プログラムを作るのに使うのと、言語装備者またはMUMPSを熟知したユーザが直接モード・コマンドやシステム管理用ルーチンを作るのに使う2つの場合がある。前者の応用プログラムは、ほぼ上記制約を満たす形で書くことができ、かなり高い互換性を持たせることができる。後者のプログラムに対しては、その性質上、互換性はほとんど期待できない。

3.2 互換性を高めるプログラミング技法

互換性のあるプログラムであっても、それを作成者以外の人を使うには、使用法がはっきりしてなくてはならない。互換性に欠けるプログラムであれば、移送のためには修正をできる限り容易にしておかなくてはならない。そのためには、以下の配慮が必要と思われる。

- (1) プログラムについては、その機能、および、入出力の形式と内容、所要ファイル容量、構成ルーチンなどを説明する。例題も必要である。
- (2) 移送するルーチンについては、その機能、および、受け渡しデータを入れるローカル変数またはグローバル変数の名称と形式、意味を説明する。一時的に使う作業用変数があれば、その名称と使用スペースの見積りも書く。呼び出すルーチンがあればその名称も明記する。説明は注釈としてつけると散逸しない。
- (3) 移送するプログラムまたはルーチンの中で、互換性のない書き方が必要であれば、その部分はできる限り機能的に小さくまとめ、コーディング上で明確に分離しておく。その部分については、機能、使用法、移送上の注意などを細かく説明し、変換が容易なようにする。
- (4) 他の装置では実現がむづかしいようなプログラムの書き方をしない。
- (5) 一時的な作業用変数は、使い終わったらKILLする。
- (6) プログラムをわかりやすく書く。そのため、個々のルーチンの機能を単機能化し、処理の流れも単純化する。間接指定の使用はできる限り避ける。

4. MUMPSの拡張仕様について

MUMPSのシステムをスムーズに運用するには、次ページの図に示す機能が必須と思われる。これらの指定形式をどう定めるかで、MUMPSの使い易さが大きく左右されるので、言語装備者にとっては、種々のくふうを要するところである。標準MUMPSでは、これらを規定していない。これらは、プログラムの作成時と使用時に必要ではあ

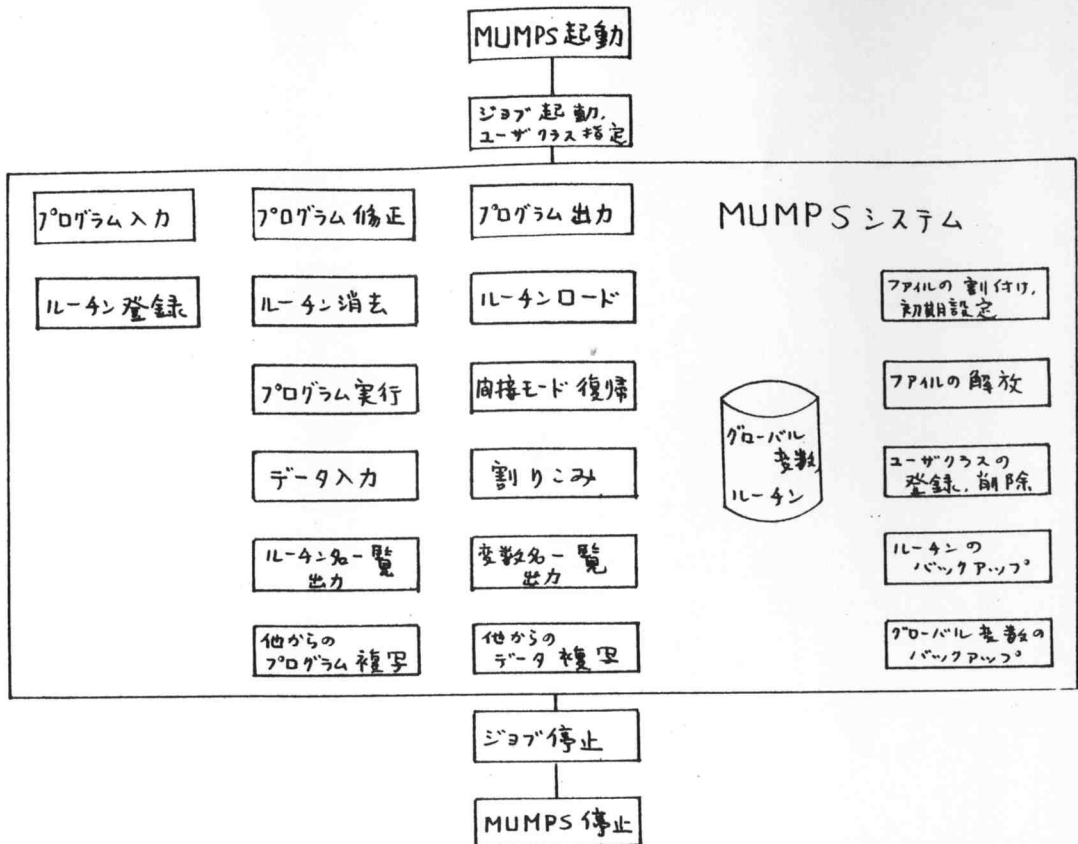


図1. MUMPSの運用に必要な機能

るが、作られたプログラムの記述中にはほとんど出現しない。したがって、規格を定めなかったのも一つの筋の通った行き方といえよう。規格を定めるとすれば、図で MUMPS システムという大きい四角で囲んだ部分については、ある程度定め得るであろうが、やはり装置に依存する部分や言語装飾者のくふうに任せたい所が残るであろう。

5. おまけ

標準 MUMPS では、応用プログラムに対しては、作るさいに少し注意すれば、高い互換性を保つことができ、他システムへの移送が容易に行なえる。MUMPS には、MUMPS のシステム管理用のプログラムを自分自身で記述できる能力があるが、当然のことながら、そのようなプログラムには互換性を期待できない。標準 MUMPS では、移送可能な部分とそうでない部分が明記されているので、無理に互換性を持たせようと苦勞しなくても、互換性のある部分とない部分を分離してプログラミングするのが現実的な行き方と思われる。

最後に、本報告の機会をよみて下さった日本 MUG の平川会長とマンプロシステム研究所の嶋芳成氏、および、MUMPS の標準仕様について御教示いただいた若井書記長と日立ソフトウェア・エンジニアリングの小野間副部長に深く謝意を表する。

付録. 標準MUMPS言語仕様要約

文字セット

95個のASCII文字 (SPも含む), ls , eol , eor
 $ni = \text{ゾ}$, $ls = SP$, $eol = CR LF$, $eor = CR FF$ とする。

ルーチン構成

ルーチン = ルーチン名 ls 行 [行]... eor
行 = [ラベル] ls [コマンド]... [コマンド] eol
コマンド = コマンド名 [:後行条件] ls [引数 [, 引数]...] ls

名前

upper caseの英数字 z . 先頭の文字は英字または%。
(先頭から8文字を有効とする。)

値の範囲

文字式 : 255文字以内
実数 : $-10^{25} \sim -10^{-25}$, 0 , $10^{-25} \sim 10^{25}$. 有効桁数は9桁。
整数 : 実数に準ずる。
添字値 : 正の整数 (0も含む)

ストレージ・スペース

ルーチン・サイズ + ローカル変数サイズ + 仮引数・サイズ ≤ 4000 文字

ネスティング

DO, FOR, XECUTE, 間接指定の合計は15レベル以下.

演算子

+	単項プラス (unary plus)	
-	単項マイナス (unary minus)	
⊖	論理否定 (logical NOT)	
+	加算 (algebraic sum)	} arithmetic binary operators
-	減算 (difference)	
*	乗算 (product)	
/	除算 (quotient)	
\	整数除算 (integer division)	
#	剰余 (modulo)	
—	文字連結 (concatenation)	} relational operators
=	等しい (equivalence)	
<	より小 (Less than)	
>	より大 (Greater than)	
[	包含 (contains)	} logical operators
]	追従 (follows)	
&	論理積 (logical AND)	
!	論理和 (logical OR)	
?	パターン照合 (pattern match)	

C

33個の制御文字 (DELも含む)

N

10個の数字

P

33個の特殊文字 (SPも含む)

A

52個の英字

L

26個の lower case の英字

U

26個の upper case の英字

E

任意の文字

特殊変数	\$HÖRÖLOG	現在の日付と時間を文字列として持つ。
	\$IÖ	現用装置のデバイス番号
	\$JÖB	現パーティションの job 番号
	\$STÖRAGE	未使用エリアの大きさ
	\$TEST	IF コマンドの引数値
	\$X	現用装置の水平カーソル位置
	\$Y	垂直カーソル位置
	\$Z	非標準変数
特殊関数	\$ASCII (C [,i])	文字列 C の i 番目の文字の ASCII コード。
	\$CHAR (i [,i]...)	i は ASCII コードとする文字を連結した文字列。
	\$DATA (変数)	変数が定義されているが、下位変数を持つかとは関係する。
	\$EXTRACT (C, i ₁ [, i ₂])	文字列 C の i ₁ 番目から i ₂ 番目までの文字列。
	\$FIND (C ₁ , C ₂ [, i])	文字列 C ₁ の i 番目以降に C ₂ の文字列を含むか探す。
	\$JUSTIFY (C, i)	文字列 C を長さ i に補正した文字列とする。
	\$JUSTIFY (e, i ₁ , i ₂)	e の値を桁数 i ₁ 、小数点以下 i ₂ 桁の文字列に変換する。
	\$LENGTH (C)	文字列 C の長さ。
	\$NEXT (変数)	指定された変数と同レベルで次に来るものの最下位添字値。
	\$PIECE (C ₁ , C ₂ , i ₁ [, i ₂])	文字列 C ₁ の i ₁ -1 番目と i ₂ 番目の文字列 C ₂ で囲まれる文字列を取り出す。
	\$RANDOM (i)	値が 0 から i-1 の擬似乱数値 (整数)。
	\$SELECT (b: e [, b: e]...)	b が true とする所の e の値を取り出す。
	\$TEXT (行指定)	指定された行の内容を文字列として取り出す。
	\$VIEW	} 非標準関数 (引数は任意)
	\$Z	

コマンド	BREAK	□ [任意]	現用装置から信号を受け取るまで実行を中止する。
	CLOSE	□ 装置番号 [: 装置パラメータ]	device, file, data set の所有権を放棄する。
	DO	□ $\left \begin{array}{l} \text{ラベル} \\ \text{[ラベル]} \wedge \text{ルチン名} \end{array} \right $ [: 後付け条件]	サブルーチンの呼び出し。
	ELSE	□	TEST=0 以下以降のコマンドを実行する。
	FOR	□ 0-加変数 = 初値 [: 増分] [: 最終値]	値を代入し、以降のコマンドを繰り返して実行する。
	GOTO	□ $\left \begin{array}{l} \text{ラベル} \\ \text{[ラベル]} \wedge \text{ルチン名} \end{array} \right $ [: 後付け条件]	制御を移す。
	HALT	□	LOCKを解除し、処理を終了する。
	HANG	□ 時間指定	指定した時間 実行を停止する。
	IF	□ [論理式]	TESTを更新し、TEST=1 以下以降のコマンドを実行する。
	KILL	□ $\left[\begin{array}{l} \text{変数, ...} \\ \text{(0-加変数, ...)} \end{array} \right]$	変数を消去する。
	LOCK	□ $\left[\begin{array}{l} \text{変数} \\ \text{(変数, ...)} \end{array} \right $ [: 時間制限]	変数の排他的所有権の取得または解除。
	OPEN	□ 装置番号 [: [装置パラメータ] [: 時間制限]]	device, file, data set を排他的に所有する。
	QUIT	□	ループからの return, FOR ループからの脱出。
	READ	□ $\left \begin{array}{l} \text{書式} \\ \text{文字定数} \\ \text{[*] 0-加変数 [: 時間制限]} \end{array} \right $	現用装置から文字列を入力し、0-加変数に代入する。
	SET	□ $\left \begin{array}{l} \text{変数} \\ \text{(変数, ...)} \end{array} \right = \text{式}$	変数に式を代入する。
	USE	□ 装置番号 [: 装置パラメータ]	指定した device, file, data set を現用装置とする。
	VIEW	□ 任意	非標準コマンド
	WRITE	□ $\left \begin{array}{l} \text{書式} \\ \text{式} \\ \text{*整数値} \end{array} \right $	現用装置に文字列と出力する。
	XECUTE	□ 式 [: 後付け条件]	式数とコマンド列と見做して実行する。
	Z.....	□ 任意	非標準コマンド

(注) IF, ELSE, FOR 以外のコマンドは、コマンドの後付け条件を記述できる。

FOR コマンドは、引数の並記、同様指定の記述はできない。

第6回日本MUG学術大会 昭和54年9月14日(金)

C. パネル(2) MUMPSの互換性 司会 MUMPSシステム研 嶋 芳成

C-3. 実際の移送の経験から(ユーザーの立場から)

大阪府立羽曳野病院 今井 敏雄

1. 移送の経験から

プログラム及びデータを異機種システムに移送する際に、特にプログラムについては言語、文法上の互換性が必要である。機種間で言語上の規約が異なる場合、いずれかのシステムでコードの変換をしてやらねばならない。ここでは、こうした言語上の問題とは別に、実際に、紙テープあるいは磁気テープ等の媒体を用いて、プログラム、データを移送するときに遭遇する困難、問題点についてまとめている。

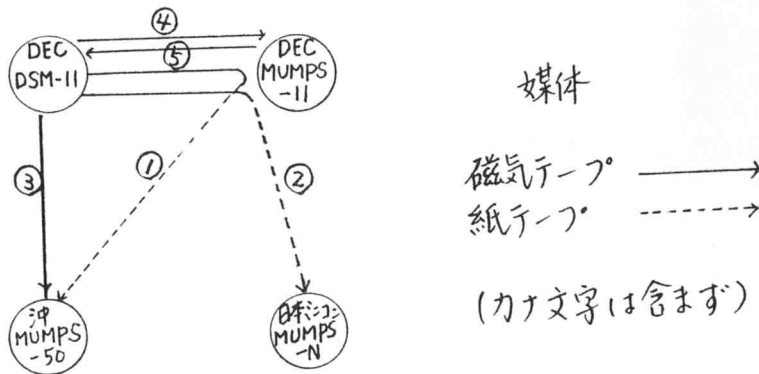


図1 移送経験例

これまでに経験し、一応移送に成功した例を図1に示す。目的は主に DSM-II で作成したプログラムとデータを、日本ミニコン MUMPS-N、沖電 MUMPS-50 へ移送することである。まずは、移送媒体として、記録形式が明瞭な紙テープを選んで移送を試みた。(ルート ①, ②)

DSM-11は、現在 紙テープ装置がサポートされていないので、一旦 磁気テープを介してMUMPS-11にデータを移し、MUMPS-11から 紙テープ出力を行った。DSM-11とMUMPS-11は 同-DEC社製であるので、磁気テープを介したデータの移送は容易である。ただし、MUMPS-11の データ文字長が 最大132であるので、長いデータは、DSM-11 から書き出す際に 適当に分割する必要があった。MUMPS-11で 磁気テープから MUMPS-N, MUMPS-50 フォーマットの紙テープと変換出力する。紙テープについては、移送先のMUMPS システムが要求する論理フォーマットさえ考慮しておれば 比較的簡単に 変換出力ができる(ただし、カナ文字に関しては SI/SO 方式か 8ビットカナコードかで異なれば やっかい)。その後 紙テープで 安定に移送が可能になり、段階で 順次 磁気テープベースでの移送を試みているが、磁気テープを媒体とする移送は、記録物理フォーマットの違いにより、色々 やっかいである。

2. 紙テープによる移送

媒体(8単位紙テープ)上で 済むべき要件は 以下の2点である。

① コード体系

{	ASCII	{	JIS 7ビット(SI/SOカナ)	{	パリティ付	奇 MUMPS-N
			(JIS C6220 7単位)		偶	MUMPS-N, MUMPS-50
		{		パリティなし	MUMPS-N	
{	EBCDIC	{		JIS 8ビット(8ビットカナ)	MUMPS-11,	
		{		(JIS C6220 8単位)	MUMPS-N	

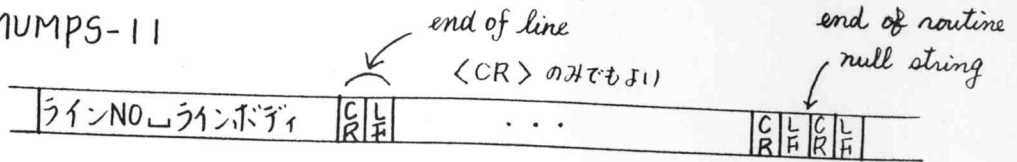
② end of record コード

MUMPS-11	<CR><LF> or <CR>	) <LF>は無視
MUMPS-N	<CR><LF> or <CR>	
MUMPS-50	! は <LF> , は <HT>	) <CR>はデータとして トリ入られる。

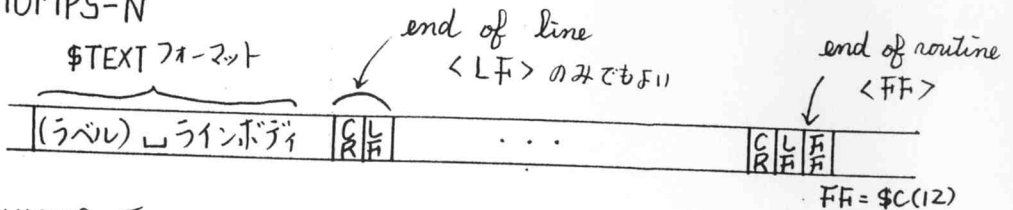
<HT>=\$C(9), <LF>=\$C(10), <CR>=\$C(13)

ZLOADコマンドによって、プログラムを媒体よりパーティションのプログラムエリアに格納できる論理フォーマットは以下の通り。

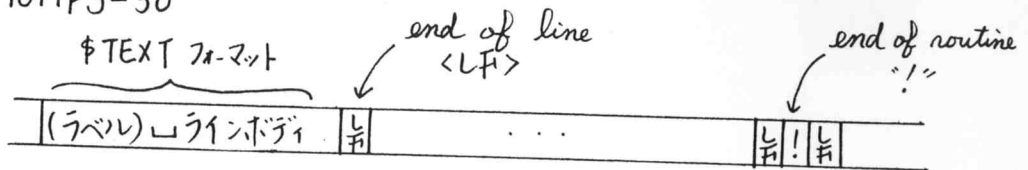
MUMPS-11



MUMPS-N



MUMPS-50



DSM-11から MUMPS-N, MUMPS-50へのプログラム移送用の変換ルーチンを図2-5に示す。

```

MTPW (DSM-11)
MTPW ; PROGRAM --> MT(MUMPS-11) --> OKI ETC.
; BY I,IMAI 79/4/10
;
A R !!,"MT DEVICE NO = ",DN,! I DN="" Q
R !!,"PROGRAM SD,,, YES(Y) OR NO(N) : ",X I X="" Q
O DN:("DSU":0:512)
U DN W *2,*5
I X="" Y" G C
B U O R !!,"PROGRAM = ",X I X="" C DN Q
X "ZL @X U DN P" G R
;
C U O R !!,"PROGRAM NAME GLOBAL = ",X I X="" S X="SDPROGRAM"
I 'SD@("A"-X,"(0)") W " GLOBAL NOT FOUND !!",! Q
S PRG=A(0)
W !!,"PROGRAM = ",PRG,!
D TST
R !!,"OK ? (<CR>) : ",X I X="" Q
F I=1:1 S X=$P(PRG,"",I) Q:X="" U O W !!,"PROGRAM = ",X X "ZL @X U DN P"
C DN Q
;
;
TST S COM=$P($T(COM)," ",2,999) F I=1:1 S X=$P(PRG,"",I) Q:X="" X COM
;
COM ZL @X I $P($T(+1)," ",1)'=X U O W !!,"PROGRAM = ",X," TOP LABEL NOT MATCH !!"
    
```

変換プログラムリストが
^SD...(0)=プログラム名1, プログラム名2,...の
形式でグローバルに与えられている。

図2 DSM-11で MUMPS-11で読める磁気テープを作るルーチン

(RT-11(DEC) Macro Assembler)

```
; *** PTPARI,MAC ***
; PAPER TAPE CONVERSION 7 BIT ASCII--> 8 BIT ASCII WITH EVEN PARITY
; BY T,IMAI 78/12/5
```

```
PTPARI: .TITLE PTPARI
        .MCALL ..V2,,,REGDEF,.EXIT
        ..V2..
        .REGDEF
```

```
PRS=177550
PRB=177552
PPS=177554
PPB=177556
```

```
PTR:    INC    PRS
1$:      TST    PRS
        BMI    ERR
        TSTB   PRS
        BPL    1$
        MOV    PRB,R0

        MOV    R0,R1
        CLR    R2
        MOV    #10,R3
2$:      DEC    R3
        BEQ    PAY
        ROR    R1
        BHIS   2$
        INC    R2
        BR     2$

PAY:     BIT    #1,R2
        BEQ    PTP
        ADD    #200,R0

PTP:     BIT    #100200,PPS
        BEQ    PTP
        BMI    ERR
        MOV    R0,PPB

        BR     PTR

ERR:     .EXIT
        .END    PTPARI
```

図5 7ビットコードに偶数パリティをつけるプログラム

DSM-11では、現在紙テープ装置をサポートしていないので、一旦磁気テープにプログラムを移し(図2)、MUMPS-11で紙テープ出力を行っている(図3、図4)。DSM-11で PRINT コマンドを利用していることと、MUMPS-11の最大文字長132文字の制約からプログラムの1ラインは132文字以下に限っている。プログラムはカナを含まないので、MUMPS-11からの出力紙テープは JIS 7ビットパリティなしコードである。MUMPS-Nは、いずれのコード体系でも端末の特性の変更が可能であるので、出力紙テープはそのまま入力可能である。MUMPS-50の場合は偶数パリティ付でなければならぬので、MUMPS-11からの出力紙テープに1パリティビットを付けておかなければならない。この種のビットハンドリングはアセンブラ言語の方が容易である(図5)。

以上述べた方法は、これよりも もっとスマートで効率のよい方法があると思うが、とにかく ほんとか紙テープでも移送を可能にしようという第一歩であつた。紙テープの場合は 以上のようにコード体系と end of record さえ考慮すれば、確実に移送はできる。しかし、大量のデータ量と比べると 磁気テープベースでの移送が望まれる。

3. 磁気テープによる移送

磁気テープを媒体として移送する場合、磁気テープ装置のハードウェアの特性、記録フォーマット、ハンドラーレベルでの処理形態等が うまく折り合わなければならぬ。以下に、物理的に移送可能であるための考慮すべき要件を示す。

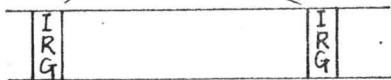
① トラック数, 記録密度, 記録形式

{ 9トラック	{ 800 BPI	{ NRZ
{ 7トラック	{ 1600 BPI	{ NRZI
		{ PE

② コード体系, BOT, EOT, EOF 等

{ ASCII	JIS 7ビット	MUMPS-N
	JIS 8ビット	MUMPS-11, DSM-11, MUMPS-N, MUMPS-50
EBCDIC		MUMPS-11, DSM-11

③ 物理的ブロックサイズ ブロックサイズ(固定)



ブロックサイズ(標準)

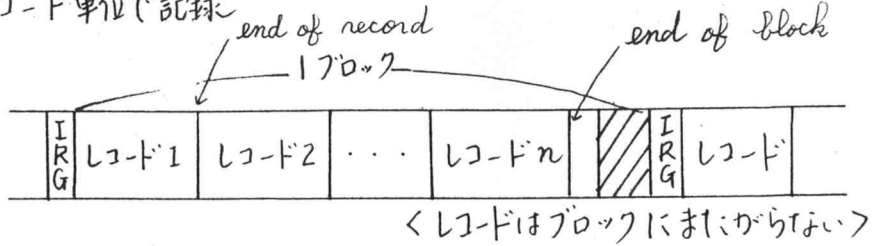
510 byte	MUMPS-N
512 byte	MUMPS-11, MUMPS-50
1024 byte	DSM-11

ブロックサイズ	固定	MUMPS-50
		MUMPS-N (非慣行的にサイズ変更可能)
指定可能	MUMPS-11	140~512 byte
	DSM-11	20~1024 byte

④ 記録フォーマット

ラベル { あり { ANSI スタANDARD }
 { DOS-11 スタANDARD }
 { ... }
 なし } DSM-11, MUMPS-11
 DSM-11, MUMPS-11
 MUMPS-N, MUMPS-50

レコード単位で記録



end of record コード

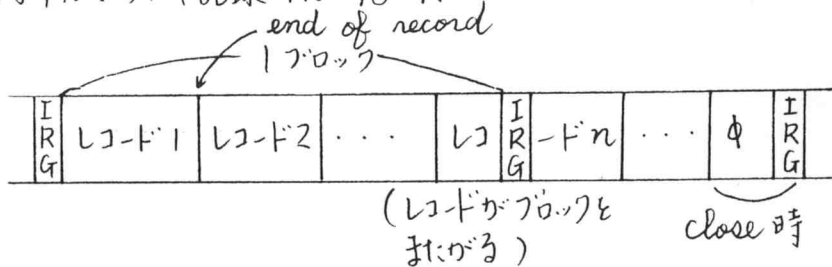
MUMPS-11, DSM-11 <CR><LF> or <LF> (<CR>は無視)
 MUMPS-50 ! は <LF>) (<CR>はデータとして取り入れられる)
 , は <HT>

end of block コード

MUMPS-11, DSM-11 ϕ
 MUMPS-50 <VT>

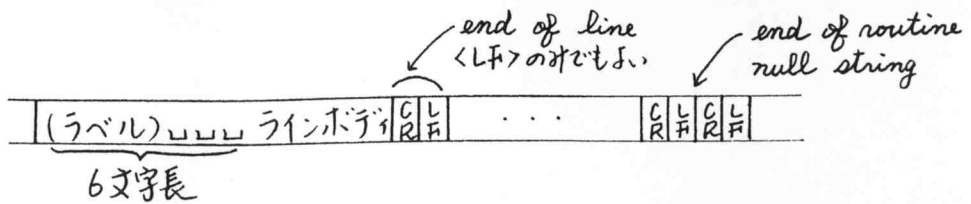
$\phi = \$C(0)$, <VT> = $\$C(11)$

1文字単位でつめて記録 MUMPS-N



end of record <CR> or <CR><LF> (<LF>は無視)

ZLOADできる論理フォーマットは DSM-11以外は紙テープの場合と同じである。DSM-11の場合、PRINTコマンドにより、磁気テープ上に書かれる論理フォーマットは以下の通り。



必ず行頭が6文字長となるようにスペースがつけ加えられて記録される。ZLOAD時に、行頭の連続したスペースがスペースに置き換えられる。従ってラインボディの始めにスペースがある場合、磁気テープにPRINTし、再びZLOADすると、ラインボディの先頭にあるスペース群が削除されてしまうので注意が必要である。

```

(DSM-II)
>P
A      W $P($T(TEXT), " ", 2, 999)
Q
TEXT   *** TEXT ***

>D 47 U 47 P C 47
      LMT device no.
      ZREMOVE (プログラムのI/Oを空にする)
>D 47 U 47 W *5
      <<スペースがなくなっている>>
>U 47 ZL rewind

>P
A      W $P($T(TEXT), " ", 2, 999)
Q
TEXT   *** TEXT ***
  
```

DSM-IIの場合、PRINTコマンドを用いれば、磁気テープ上にスペースがつけ加えられる形式で記録されるが、

ラベルラインボディ
↑ 1スペース

のいわゆる\$TEXTフォーマットの形で記録されている磁気テープでも、ZLOADコマンドで正しくプログラムの読みこまれる。

以上のように、磁気テープ上での記録形態は各社各様である。移送に際して、送る側と受ける側とで、いずれの条件項目について、一致あるいは疑似的に一致させないと磁気テープによる移送は不可能である。

DSM-11からMUMPS-50へ磁気テープを介して移送した例での移送ルーチンを図6, 7に示す。プログラムの移送では、DSM-11の方で、MUMPS-50で ZLOAD できるフォーマットに書き込む方法をとっている。end of recordは MUMPS-50が<LF>のみであるので WLF のかわりに WLC(10)を使っている。end of block については、DSM-11が<CR>コード、MUMPS-50では<VT>と全く異なっている。DSM-11のユーザプログラムで<UT>を出力してやらねばならない。従って1ブロックに入るレコード数、長さをユーザプログラムでチェックする必要がある。ここでは面倒なので記録効率の落ちるサブブロックレコードとして(DSM-11の1ブロックライト WLC*4)を利用している。

図7はグローバルデータの移送ルーチンで、レベルグローバルのみを扱っている。任意レベルのグローバルデータの移送もさして難しくない。

MTOKIP (DSM-11)

MTOKIP ; PROGRAM --> OKI PROGRAM FORMAT MT
; BY T, IMAI 79/4/16, 79/5/9

```

;
A  U O R !!," OUTPUT MT DEV. NO : ",MT,! I NT="" Q
   Q MT:("DSU";0;512)
   U MT W *5
   F I=1;1;2 S C(1)=$P($T(COM+I)," ",2,999)
   U O R !!,"PROGRAM GLOBAL NAME = ",X I X="" S X="SDPROGRAM" W X
   I '$D(A("_X_"(0))) W " GLOBAL NOT FOUND !!",! Q
   S PRG=A(0)
   W !!,"PROGRAM = ",PRG
   R !!,"OK ? (<CR>) : ",X I X="" Q
   F K=1;1 S X=$P(PRG,"",K) D R Q;X=""
   Q
;
B  U O W !," PROGRAM NAME = ",X I X="" U MT W "/"*",$C(10),$C(11),*4 C MT Q
   U MT W X,$C(10),$C(11),*4
   X C(1)
   U MT W "!", $C(10),$C(11),*4 Q
COM
;
ZL @X F I=1;1 S T=$T(+I) Q;T="" X C(2)
S Z=$P(T," ",1)_"_"$P(T," ",2,999) U MT W Z,$C(10),$C(11),*4

```

(MUMPS-50)

>O L16:(^SOS:"B") L V L16 L F I=1;1 R U A,! L Q:A=" L L Z E L L Z L L L Z F L @(^A-A)

図6 DSM-11からMUMPS-50へのプログラム移送ルーチン(磁気テープ)

(DSM-11)

```

MTOKIG
MTOKIG ; GLOBAL(1 LEVEL) --> OKI GLOBAL FORMAT MT
; MT FORMAT : GLOBAL NAME,LF,S=SUBSCRIPT,LF,D=DATA,S=SUBSCRIPT,LF,D=DATA,,,,LF,LF,GLOBAL NAME,,,
; BY T,IMAI 79/4/16, 79/5/9
;
R !,"MT DEVICE NO = ",DN I DN="" Q
O DN:("DSU":0:512) U DN W *5
A U O R !,"GLOBAL NAME = ",GBL I GBL="" U DN W SC(9),SC(10),SC(11),*4 C DN Q
I $D@("A",GBL) W " GLOBAL NOT FOUND !!" G A
S @("S"=SN("A",GBL,"(-1)")) W !,"GBL = ",GBL," S = ",S I S<0 Q
U DN W GBL,SC(10),SC(11),*4
S S=-1
C S S=SN(A(S)) I S<0 W SC(9),SC(10),SC(11),*4 G A
S X=A(S) W "S="S$SC(10),"D="X$SC(10),SC(11),*4 G C

```

(MUMPS-50)

```

SDMTGW
SDMTGW ; DSM-11 GLOBAL(MT) SAVED BY 'MTOKIG' --> OKI MUMPS GLOBAL
; BY T,IMAI 79/4/16, 79/5/9
;
START K S U="U "C="W " W I,$P($I(SDMTGW)," ",2,999),I R !," MT(UNIT 1) SET OK ? (<CR>) : ",X I X!=" Q
A O 16:(ASOS:"B") U 16
H R NAM,!
I NAM="" G EXIT
X U W !," GLOBAL NAME = ",NAM R !," OK ? YES(<CR>) , NO(N) OR RENAME(REN) : ",X
I X="" G EXIT
I X="REN" R !," RENAME GLOBAL NAME = ",NAM G B
I X="" W " ERROR !!" G B
I $D@("A",NAM) W " GLOBAL EXIST !!" G E
U 16
F K S ! G H:S="" I $(S,1,2)="S" X U W !,"S= ERROR !!" G EXIT
C S S=$(S,3,999)
D R O ! I $(O,1,2)="D" X U W !,"D= ERROR !!" G EXIT
S Z=$(O,3,999)
S @("A",NAM,"(S)=Z")
G C
;
R "EXIT(1), KILL(2), MERGE(3) : ",Y
I Y="2" R !,"KILL OK (Y) : ",Y K:Y="Y" @("A",NAM) G B
I Y="3" G F
Q
EXIT C 16
Q

```

プログラムを移送する際、以上の例では、移送先ですぐにZLOADコマンドを用いてパーティションのプログラムエリアに格納できる形式で、プログラムを媒体に記録する方法をとっている。ZLOADできるフォーマットで書き込むことが困難な場合とか、より一般的に移送方法を考える場合などには、ZLOADフォーマットで書き込まず、あくまで、ラインデータデータレコードとして媒体に記録し、移送先でユーティリティを用いてラインデータをプログラムとして格納する方法がある。データとして読みなおしにラインデータをプログラムとして格納する方法は、コマンドレベルで処理する方法と別媒体を用いる方法が考えられる。コマンドレベルで処理できる例として、DSM-11ではZINSERTコマンドが利用できる。

(DSM-11)

```
>P
A      A
      B
      C
D      D
```

) プログラム

```
>O 47 U 47 F I=1:1 S X=$T(+I) Q:X=" " W X,!
```

```
>U 47 W !
```

MTへデータとして
ラインデータを書き込み

```
>C 47
```

```
>O 47 U 47 W *5
```

```
>ZR
```

```
>P
```

ZINSERTコマンド

```
>O 47 U 47 F I=0:1 R X Q:X=" " (ZI X:+I
```

```
>P
```

```
A      A
      B
      C
D      D
```

データとしてラインデータを読みなおし
プログラムエリアに格納

> (コマンドによってデータをプログラムエリアに格納する例)

MUMPS-N, MUMPS-50では、この種のコマンドがないので、磁気テープやシーケンシャルディスクファイル等の別媒体に、ZLOADできるフォーマットにラインデータを再編集して、あらかじめその媒体よりZLOADし、プログラムとして格納する方法がとられる。MUMPS-Nの場合はプログラムはシーケンシャルファイル上に貯えられているので、直接シーケンシャルファイルにデータとして書き込む方がよい。

(MUMPS-N) MT device NO. シーケンシャルファイル NO.

```
> O ①⑥: ("R") ⑥: ("AAA": W)
> F ①⑥: 1: 1 ⑥: 16 ⑥: R ⑥: X ⑥: Q: end条件 ⑥: ⑥: 6 ⑥: W ⑥: X, !
> U ⑥: 6 ⑥: W ⑥: *12
> C ⑥: 6
```

(シーケンシャルファイルにZLOADできる形式で書き込む例)

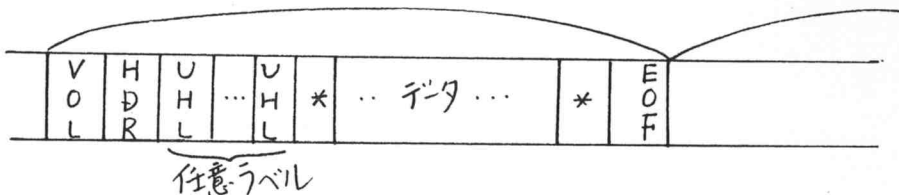
このように、レコードデータが物理的に移送可能であれば、プログラムに対しても対応可能である。磁気テープを介してデータが移送できる為には、先に挙げる①～④の項目を、何らかの方法で一致解決しなければならない。①については、ハードウェアのかかわりで全く妥協を許さない。②については、JIS 8ビット系で統一できればカナについても全く問題がなくなる。③のブロックサイズについては、ハードウェアとバッファサイズなどのハードウェアレベルのかかわりがある。一般に、ブロックサイズの小さいシステムから大きいシステムへは移送可能であるが、逆の大きいブロックサイズのシステムから小さいシステムへは、ブロックサイズの指定が可能でないと不可能である。④の記録フォーマットに関しては、end of recordの違いは出力側で考慮すれば可能。入力システム側で end of record の違いの対応は、一文字 read 処理でないと不可能で、対応は難しい。end of block のコードが異なれば、やっかいであるが、図6, 7に示すルーチンのように、プログラムレベルで end of block コードが書き込めれば対応可能である。しかし、end of block のコード自身が磁気テープ装置へのコントロールコードとして、いつも使われていないという保証はない。レコードがブロックにまたがるシステムと、またからないシステム間での移送は、いつも取り道があるとお限らない。

以上①～④が、ほとんど切り合いがつかず、現在でも移送は可能である。しかし、一般にメーカーは、自社のシステムのことしか、熟知していないので、移送ユーザーが移送用の両システムについて、以上のような特性を全て調べ、熟知していないと移送はできない。

4. 今後の問題

現状では、ユーザが移送しようとする機種の特性間に、記録媒体を通して
 なんとか あい路を見つけ出し、移送を行っている。柔軟性のあるシステムならよいが
 あるパターンしか受け入れないシステムでは災難である。今後の方向として、一定の
 磁気テープ記録物理フォーマットを定め、各メーカーが同フォーマットについて、
 MUMPSで read, write をサポートすることを望む。さらに、プログラム 及び
 グローバルデータについて移送用の論理フォーマットを定め、各メーカーは MUMPS の
 ユーティリティレベルで、プログラム、データの磁気テープへの書き込み 及び読み出し
 ディスクへの格納を可能にする。望むらくは、標準 MUMPS と異なる言語仕様に対し
 ては、自動コード変換、変換不能のときは コメント出力などを行なってくれば
 有難い。

磁気テープの記録フォーマットについては、JIS C6245「情報交換用磁気テープ
 のラベルとファイル構成」(JIS ハンドブック 情報処理 78 RP 127~135)に
 準拠する形式が考えられる。JIS C6245は、MUMPSに限らず、より一般的な
 規格であり FORTRAN 等でも読むことが可能である。MUMPSの場合は、同規
 格の8ビットJISコード体系で、可変長レコード構成を採用するのがよいと思える。
 JIS C6245の概要を以下に示す。



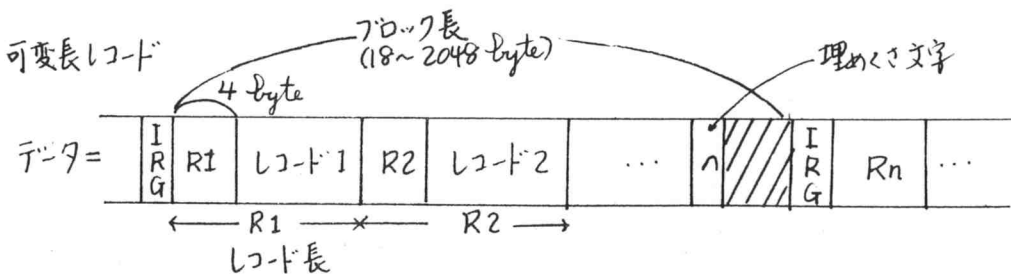
3.2 ラベルの大きさ ラベルは80キャラクタのブロックとする

VOL .. ボリューム見出しラベル

HDR .. ファイル見出しラベル

UHL .. 使用者ラベル

星印(*)は テープマークを示す



5.1 レコードのブロック化 レコードは次の規則に従ってブロックする。

- (1) レコード間の境界は明示しない。
- (2) ブロックは、整数個のレコードによって構成する。
- (3) 埋めくさ文字を使用する場合は、当事者の合意を必要とする。
埋めくさ文字には JIS C 6220 (情報交換用符号) の 5/4 のキャラクタ
ハットを使用する。

5.1.2 可変長コード

- (1) ファイル中のレコードのすべてが同じ長さでないとき、各レコードの
長さ(そのレコードを含むキャラクタ数)を各レコードの最初の欄に記
録する。その欄は、各レコードの最初の4キャラクタ分とし、レコードの
長さは十進数で表わす。この欄もレコードの長さの一部とする。
(ハンドブックより)

JIS規格では、当事者間の任意性によりかかれている部分が幾つあるの
で、実際に移送媒体として使用可能にするためには、より厳密な移送規格を定めておく必
要がある。特に、任意ラベルの個数、ラベルの扱い、ブロックサイズ、埋めくさ文字ハ
ットを使用するかどうか等は、合意を必要とする。

論理フォーマットについては、第11, 12回 ME学会MUMPS ソフトウェア研究会
での 嶋 芳成 氏からの提案が参考となる。同案では、<CR><LF>のセットが
end of record として扱われており、<CR><LF>がデータとして存在してい
ても移送可能な形式である。問題は end of record のコードである。現状では
<CR><LF>セットで認識しているシステム、いずれか一方で認識し他方を無視
するシステム、一方で認識するが他方はデータとして取り入れられるシステムなど
各社様々である。<CR><LF>のセットの代が end of record とすると、<CR>
<LF>単体のコードはデータとして取り入れなければならない。<CR><LF>の一方
のみを有効とするシステムでは、<CR><LF>のセットが来なのか、その一方だけが
来なのか不明となる。W₀! によって<CR>, <LF>のいずれか一方しか書き出さない
システムではデータコードとして \$C(13,10) が書き出せる必要がある。いずれに
しても、うまく折り合わないときは、一文字 read, write (R₀*X, W₀*X) を
利用するしかない。しかし DSM-11 のように、磁気テープを媒体とした場合、一文字
read (R₀*X) で連続的に記録された一文字続きでなく、各レコードの先頭の
一文字しか順次に返してこないシステムもある。DSM-11 の場合、連続的に一文字
read を行おうとすれば、read one block し、\$VIEW コマンドにより、
ダイレクトにメモリの磁気テープバッファを読むしか方法はない。いずれにせよ
一文字 read が可能であれば、すべてに対して対応ができるが、効率面ではやはり
end of record を一定に定めて、一レコード read ができる方がよい。
<CR> 或いは <LF> (その他のコントロールコード) もデータとして移送可能にする

にめには、<CR><LF>のセットを end of record とするしかない。<CR><LF>等をデータとして含めないとする。end of record の条件はゆるくして、<CR><LF>の<LF> (<CR>無視) などでもよい。さらに、同提案では、\$C(30)、\$C(29) が特殊制御コードとして使われているが、コードデータとして システムから書き出し、読み取れることを確認しておく必要がある。

以上のような形式に、磁気テープの移送フォーマットを定める場合、あるメーカーでは、磁気テープハンドラーレベルの改修が必要となるし、また、あるメーカーでは、文字 read などのテクニックによって、現在でも MUMPS のプログラムレベルで処理可能である。メーカーの方々の一考を期待する。



第6回日本MUG学術大会 昭和54年9月14日(金)

C. パネル2: MUMPSの互換性 司会 MUMPSシステム研 嶋 芳成

C-4. 実際の移送の経験から(メカの立場から)

日本ミニコンピュータ 木下博義

ここでは、あるシステムから別のシステムへのプログラムを含めたデータの移送の方法について述べています。

1. 移送手段
2. 論理的フォーマット
3. 物理的フォーマット
4. 移送用ユーティリティ
5. まとめ

2~4についてはMUMPS-Nについて述べます。

1. 移送手段

1) プログラム・リスト/データ・リスト

ライン・プリンタ等の印字装置により出力し、人間が目を見てキーボードから入力する。印字装置は価格的に広範囲であり、それは主に印字速度に依存している。入力に人間の手を介するため非常に遅く、また、誤入力により信頼度は低くなる。印字フォーマットについては、人間が目を見て理解するので、特別に制限を設ける必要はない。本方式は少量のデータ、特にプログラムの移送が可能であろう。

しかしながらOCR(光学読取装置)が普及すれば、かなり大量のデータを移送することが可能である。その場合には印字フォーマットを統一しておく必要であろう。

2) 紙テープ

紙テープ・パンチにより出力し、紙テープ・リーダーにより入力する。両装置とも安価であるが、速度が比較的遅く、操作性にも難がある。少量のデータ及びプログラムの移送に適し、よりコンパクトなシステムに利用されよう。

本方式では、出力フォーマットを統一しておく必要がある。

3) バンチ・カード

カード・パンチにより出力し、カード・リーダーにより入力する。オンライン・カード・パンチは高価であり、ミニコンに接続する例はあまりない。バンチ・カードはデータの移送に用いるよりもむしろ、マーク・カードと共にプログラムの登録やデータの登録に適している。

本方式の場合、カード・コードとASCII(JIS)コードの対応を統一しておく必要がある。

4) フロッピー・ディスク

入出力はフロッピー・ディスク装置により行なう。データ量としてもフロッピー・メディア1枚当たり数100KBから数MBもあり、かつ操作性も良い。しかしながら、各社により物理的仕様(片面/両面記録, 記録密度, セクタ/トラック, バイト/セクタ)が異なるため、その標準化が待たれる。さらに、論理的仕様(プログラム及びデータの入出力フォーマットをフロッピー・ディスク上にどのように実現するか)の統一が必要になる。

5) 磁気テープ

入出力は磁気テープ装置より行なう。データ量としては1リール当たり数10MBもあり、操作性も良い。データ・ベースを指向するMUMPSシステムにとっては必要不可欠であり、現在、データの移送において最もよく使用されている。

本媒体においても4)フロッピー・ディスクと同様に、ブロック・サイズが統一されていない。

6) 回線(特定/公衆)

入出力にはモデムや音響カプラー等を使用する。移送するデータの量に制限はないが、転送速度は回線により抑えられる。

MUMPSシステムのネットワークやMUMPSシステムと他のシステムとの通信に使用される。

この方式では両システム間で通信手順を統一しておく必要がある。

2. 論理的フォーマット

プログラムやデータをシステムの間で移送する場合、MUMPS-Nでは次の論理的フォーマット(ダンプ・フォーマットと呼ぶ)を用いている。

ダンプ・フォーマット ::= ファイル・ヘッダ | ルーチン・ダンプ | グローバル・ダンプ | ... eor

ここで

ファイル・ヘッダ ::= リビジョン番号 ダンゴ日付 ダンゴ時刻 eōl

ルーチン・ダンゴ ::= ルーチン・ヘッダ ルーチン・ボディ eōr

ルーチン・ヘッダ ::= ルーチン名 ; セキュリティ・コード, 作成者ID, バイト・カウント;
作成日付, 作成時刻 eōl

ルーチン・ボディ ::= ライン { ライン } . . .

グローバル・ダンゴ ::= グローバル・ヘッダ グローバル・ボディ eōt

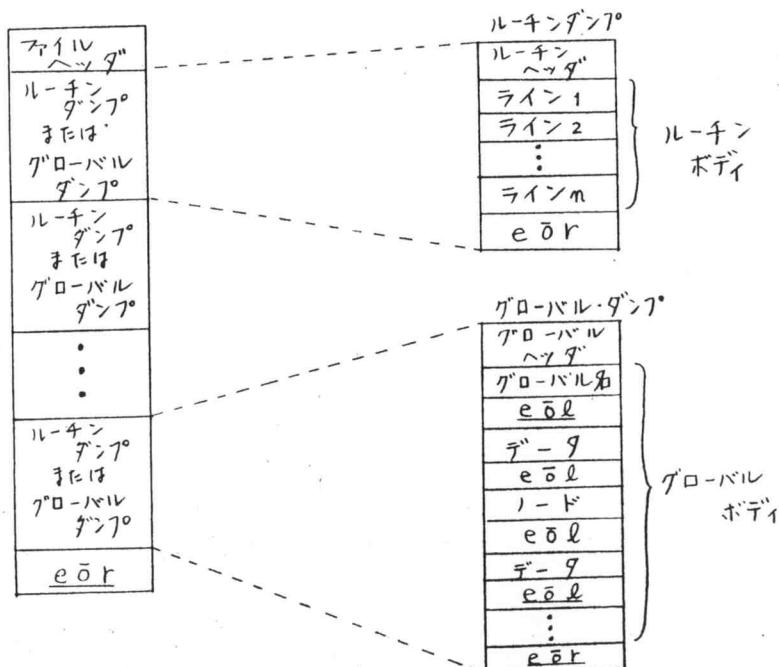
グローバル・ヘッダ ::= グローバル名 ; セキュリティ・コード, 作成者ID, 0 ;
作成日付, 作成時刻 eōl

グローバル・ボディ ::= グローバル名 eōl { デ-タ eōl }
{ ロード eōl { デ-タ eōl } } . . .

eōl ::= \$C (13, 10) CR LF

eōr ::= \$C (13, 12) CR FF

1) ダンゴ・フォーマットの略図



2) ダンプ・フォーマットの例

・ルーチン・ダンプ

```
AA ;0,4782,146;50182,11 eol
AA ;SAMPLE PROGRAM eol
W !,"YOUR ROUTINE AA IS EXECUTING" eol
CD S VAR="STRING VALUE",NUM=37 Q eol
X ;END OF ROUTINE "AA" eol
Z ZF AA W !,"AA HAS BEEN FILED" Q eol
eor
```

・グローバル・ダンプ

```
^SAMPLE ;0,4782,0;50182,10 eol
^SAMPLE eol
SAMPLE GLOBAL eol
^SAMPLE(1) eol
1 eol
^SAMPLE(1,1) eol
11 eol
^(2) eol
12 eol
^(3) eol
13 eol
^SAMPLE(2,1) eol
21 eol
^SAMPLE(2,1,1) eol
211 eol
^(2) eol
212 eol
^(3) eol
213 eol
^SAMPLE(2,2) eol
22 eol
^SAMPLE("SUB3") eol
STRING SUBSCRIPT 3 eol
^SAMPLE("SUB3","SUB31") eol
STRING SUBSCRIPT 31 eol
^("SUB32") eol
STRING SUBSCRIPT 32 eol
^("SUB33") eol
STRING SUBSCRIPT 33 eol
^SAMPLE("SUB4","SUB41") eol
STRING SUBSCRIPT 41 eol
^("SUB42") eol
STRING SUBSCRIPT 42 eol
eor
```

3. 物理的フォーマット

紙テープを用いた移送においては、2で述べた論理的フォーマット（ダンプフォーマット）をそのまま実現すればよく、特に問題はない。

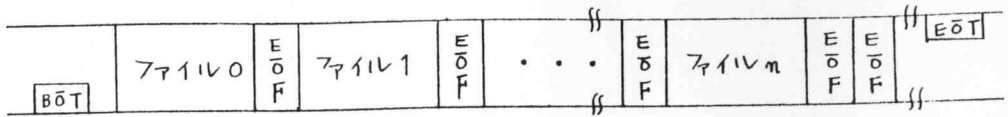
回線を用いた移送においても同様に、ダンプ・フォーマットでそのまま転送すればよいが、たとえばどのような手続きに従うかはシステム間で統一していなければならない。

これに対し、フロッピー・ディスクや磁気テープの場合は、論理的フォーマット

(ダンプ・フォーマット) のデータを物理的に格納するとき、データはブロッキングされる。

ここでは、これを磁気テープについて述べる。

磁気テープは1つのリールに複数のファイル(ここではダンプ・フォーマットのデータ・ファイル)を格納することができ、それぞれを番号により管理している。



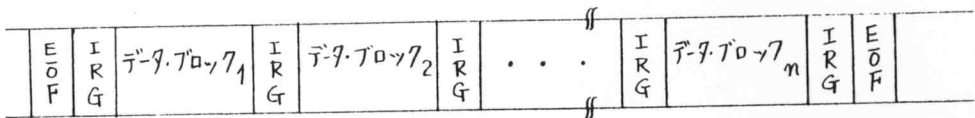
ここで

BOT ... Beginning of Tape Mark

EOF ... End of File Mark (Tape Mark)

EOT ... End of Tape

磁気テープのアクセスはこのファイル番号を用いる。次に各ファイルの物理構造を示す。



ここで

IRG ... Inter Record Gap

MUMPS-Nでは各データ・ブロックは510バイト固定長であり、最後のブロックのデータが510バイトに満たないとき、ナル・コードが詰められる。

4. 移送用ユーティリティ

移送用のユーティリティとしては、2で述べた論理的フォーマット(ダンプ・フォーマット)を作り出すものと、そのダンプ・フォーマットから本来のプログラムやグローバルを取り込むものが必要である。

通常、オペレーティング・システムはデバイス・インディペンデンスな入出力を許すので、移送のためにどの媒体を用いても移送用ユーティリティは汎用的である。

しかしながら、システムによっては次の機能を持つておく必要があるかもしれない。

- ・ EBCDIC - ASCII (JIS) 変換
- ・ 異なるブロック・サイズに対する手段

また、使い易さという観点から、移送用ユーティリティは以下の機能を持つておくべきである。

- ・ 移送ファイルの指定

- ディレクトリ内のすべてのファイルのダンプ
 - 移送媒体上のすべてのファイルのロード
 - 選択的指定 (- , *)
 - 排他的指定

- ・ ロード方式

- 移送媒体上のファイルの名前だけの確認
 - ディレクトリに存在しないファイルだけのロード
 - 同じファイルが存在したら、それを消去してからのロード
 - 同じファイルが存在したら、新しいときのみのロード

- ・ グローバル変数のみのダンプ/ロード

- APPEND ロード

- UPDATE ロード

- 赤字を指定したダンプ/ロード

5. まとめ

異なったシステム間でデータを移送するには、移送用媒体の物理的フォーマットおよび移送ファイルの論理的フォーマットがシステム間で統一されていなければならない。

磁気テープの物理的仕様は、ほとんどがIBMと同じであり、操作性もよく、大量のデータにも耐え、移送ファイルの論理的フォーマットが統一されれば、現時点では最適な移送媒体であろう。

第6回日本MUG学術大会 昭和54年9月14日(金)

C. パネル(2) MUMPSの互換性 司会 MUMPSシステム研 嶋 芳成

C-5. MUMPS におけるカナ文字の現状と展望

大阪府立羽曳野病院

野口 弘 , 大槲 陽一

§1 はじめに

MUMPSにおけるカナ文字の、国産各メーカーの現状については、お13回マン
プスソフトウェア研究会やMUMPSニュースなどで、報告してきたので、御存知の
方も多いと思う。ここでは、重複することも承知の上で、これまで、回収され
たアンケートの結果の全貌(今回新たに得られたデータもある。)と問題
点、および展望について述べたい。

§2 カナ文字の現状

アンケートを依頼したMUMPS提供メーカーは、以下の8社である。

1 DEC	2 沖	3 日本三ニオン	4 NEC
5 三菱	6 東芝	7 パナファコム	8 日立

このうち、NEC、日立については、現在のところ確固とした仕様の決定
がなされていないので、MUGで、早期に標準化されれば、できるだけそれに
従うとのことであった。

「互換性」の問題から見ると、カナ文字がどの範囲まで使用できるか
ということが、移送の際に問題となってくる。例えば、行ラベルにカナが使
用できるシステムで作られたカナラベルを含むルーチンは、カナラベル使用不可の
システムには、移送できない。したがって、互換性、他システムへの移送を、

考えるとき、各システムでのカナ文字の使用範囲を知っておかねばならない。

国産8社が、一様にカナ文字を使用できる範囲は、データ、定文字列、注釈文(Comment)であり、これ以外に関しては、各社まちまちである。

表2-1にデータ、定文字列、注釈文以外にカナ文字を使用できる範囲とそのメーカーを挙げておく。

ルーテン名	沖, NEC, 三菱, 東芝
行ラベル名	沖, NEC
変数名	沖, NEC, 三菱
文字添字 グローバル ローカル	DEC, NEC, 日本三ニコン, パナファコム NEC, 日本三ニコン

表 2 - 1

使用コード体系は、互換性、移送の観点からすると、大きな問題であり、システム間の使用コードの体系が異なると、移送には、非常に困難が伴う。各社のコード体系は、現在のところパナファコムが、EBCDICを、使用している以外は、JIS 8単位コード(JIS-C-6220)に従ったものであった。

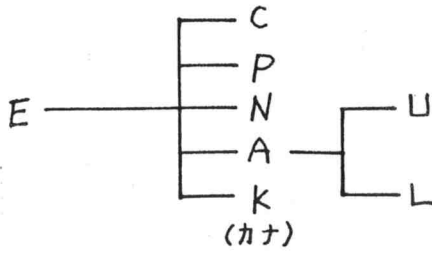
各キャラクターのパターンの分類、いわゆるパターンコードに関しては、ANSIでは、当然のことながら、カナに関しては、何もふれられていない。国内各社の、パターンコードの分類を、表2-2にまとめておく。

表2-2を見ると、カナのパターンコードとしては、各社例外なく、「K」を使用しており混乱は見られない。しかし、カナ文字をさらに「大文字」、「小文字」、「特殊文字」などに区別しているところもみられ、細かいところでは、各社まちまちである。

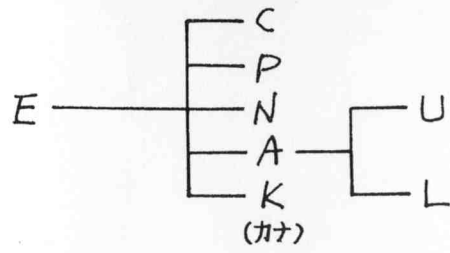
「カナ文字」というものの範囲もあいまいで、「(始カっこ)」、「(終カっこ)」、「(句点)」、「(読点)」、「(中点)」などをカナ文字に含んでいる社(DEC, 日本三ニコン, 三菱)と Punctuationとして取扱っている社(沖, 東芝)とがある。その他「(長音)」、「(濁点)」、「(半濁点)」は、大文字(東芝)か、小文字(日本三ニコン)か、カナスペース(160₁₀₀)はカナ(DEC)かカナでないか互いの問題がある。特にカナスペースは、JIS 8単位符号でも明確には定義されておらず、各社もどう取扱ってよいのか迷っているのが現状と思われる。

カナ文字関係の図形キャラクターとそのパターンコードの一覧を表2-3に載せておく。パナファコムに関しては、内部コードがEBCDICで、他社とは異なるので、表2-3には載せなかった。ただしカナ文字「K」の範囲は、アヘンの文字ということでカナ小文字「r」、「j」、「o」、「h」は存在せず、カナ文字と英小文字は、同時に使用不可能である。

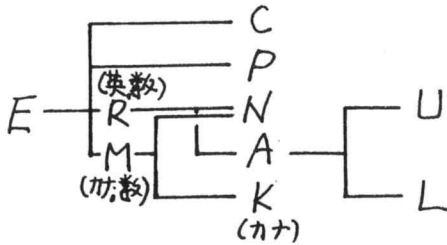
DEC



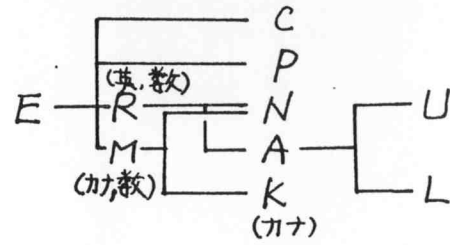
三菱



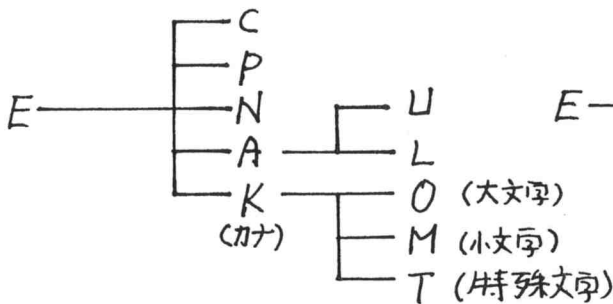
シ中



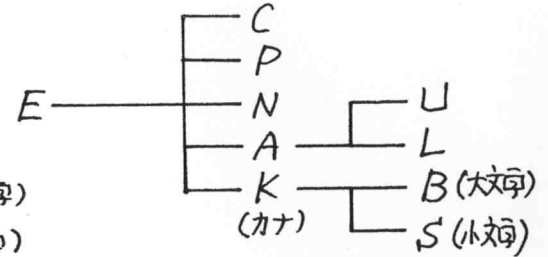
パナソニック



日本三ニコン



東芝



NEC ('79 6月時点での仕様)

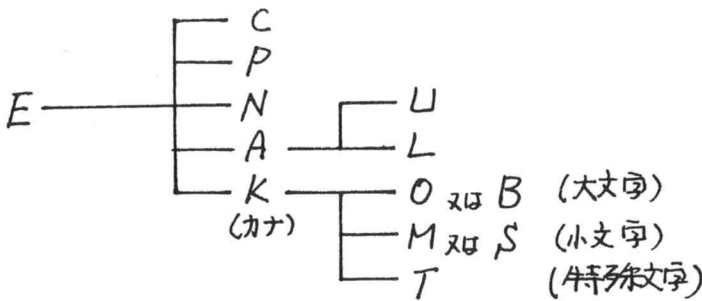


表 2-2 パターンコード分類

デラミル	オワフル	コード	Dec	シ中	日本 三三三	三菱	東芝
160	240	(SP)	EK	EP	EP	EP	
161	241	・	EK	EP	EKT	EK	EP
162	242	・	EK	EP	EKT	EK	EP
163	243	・	EK	EP	EKT	EK	EP
164	244	・	EK	EP	EKT	EK	EP
165	245	・	EK	EP	EKT	EK	EP
166	246	ラ	EK	EKM	EKO	EK	EKB
167	247	ア	EK	EKM	EKM	EK	EKS
168	250	イ	EK	EKM	EKM	EK	EKS
169	251	ウ	EK	EKM	EKM	EK	EKS
170	252	エ	EK	EKM	EKM	EK	EKS
171	253	オ	EK	EKM	EKM	EK	EKS
172	254	カ	EK	EKM	EKM	EK	EKS
173	255	キ	EK	EKM	EKM	EK	EKS
174	256	ク	EK	EKM	EKM	EK	EKS
175	257	ケ	EK	EKM	EKM	EK	EKS
176	260	コ	EK	EKM	EKM	EK	EKB
177	261	サ	EK	EKM	EKO	EK	EKB
178	262	シ	EK	EKM	EKO	EK	EKB
179	263	ス	EK	EKM	EKO	EK	EKB
180	264	セ	EK	EKM	EKO	EK	EKB
181	265	ソ	EK	EKM	EKO	EK	EKB
182	266	タ	EK	EKM	EKO	EK	EKB
183	267	チ	EK	EKM	EKO	EK	EKB
184	270	ツ	EK	EKM	EKO	EK	EKB
185	271	テ	EK	EKM	EKO	EK	EKB
186	272	ト	EK	EKM	EKO	EK	EKB
187	273	ナ	EK	EKM	EKO	EK	EKB
188	274	ニ	EK	EKM	EKO	EK	EKB
189	275	ヌ	EK	EKM	EKO	EK	EKB
190	276	ネ	EK	EKM	EKO	EK	EKB
191	277	ノ	EK	EKM	EKO	EK	EKB
192	300	ハ	EK	EKM	EKO	EK	EKB
193	301	ヒ	EK	EKM	EKO	EK	EKB
194	302	フ	EK	EKM	EKO	EK	EKB
195	303	ヘ	EK	EKM	EKO	EK	EKB
196	304	ホ	EK	EKM	EKO	EK	EKB
197	305	マ	EK	EKM	EKO	EK	EKB
198	306	ミ	EK	EKM	EKO	EK	EKB
199	307	ム	EK	EKM	EKO	EK	EKB
200	310	メ	EK	EKM	EKO	EK	EKB
201	311	モ	EK	EKM	EKO	EK	EKB
202	312	ヤ	EK	EKM	EKO	EK	EKB
203	313	ユ	EK	EKM	EKO	EK	EKB
204	314	ヨ	EK	EKM	EKO	EK	EKB
205	315	ラ	EK	EKM	EKO	EK	EKB
206	316	リ	EK	EKM	EKO	EK	EKB
207	317	ル	EK	EKM	EKO	EK	EKB
208	320	レ	EK	EKM	EKO	EK	EKB
209	321	ロ	EK	EKM	EKO	EK	EKB
210	322	ワ	EK	EKM	EKO	EK	EKB
211	323	ン	EK	EKM	EKO	EK	EKB
212	324	・	EK	EKM	EKM	EK	EKB
213	325	・	EK	EKM	EKM	EK	EKB
214	326	・	EK	EKM	EKM	EK	EKB
215	327	・	EK	EKM	EKM	EK	EKB
216	330	・	EK	EKM	EKM	EK	EKB
217	331	・	EK	EKM	EKM	EK	EKB
218	332	・	EK	EKM	EKM	EK	EKB
219	333	・	EK	EKM	EKM	EK	EKB
220	334	・	EK	EKM	EKM	EK	EKB
221	335	・	EK	EKM	EKM	EK	EKB
222	336	・	EK	EKM	EKM	EK	EKB
223	337	・	EK	EKM	EKM	EK	EKB

表 2-3 カナ文字ハターンコード

カナ文字に関する関数及び文字演算については、各メーカーとも特殊あつかいをして
いるということであった。

関数としては、\$ASCII, \$CHAR, \$DATA, \$EXTRACT, \$FI
ND, \$JUSTIFY, \$LENGTH, \$PIECE, \$SELECT, 文字演
算としては、-(結合), =(合同), [(包含),](追従)が考えられる。

これらの関数とパターンバリフィケーションを組み合わせた簡単なプログラムを、
各メーカーに実行してもらった。プログラムの内容を表2-4に、各メーカーの実行の結果
を表2-5にまとめておく。

カナファンクション(表2-5の1~19), カナ演算(20~28), 数値化, 文
字化(29~36)については、各社とも本質的互差は見られず、三菱、東芝の真偽値
が、-0.01と0であること、\$DATAの値が三菱、東芝で標準MU
MPSと異なることを除けば、英数文字と異なった取扱いはなかった。

各社間に差が見られたのは、カナスペースの取扱いで、37のカナモードで、“ア”イ
を入力したとき“”の\$ASCIIの値が、沖が160(10)その他の社では、32(10)
と取っている。その他メーカーによって差が見られたものを列挙すると、

- 43 DEC のみ 真他社は偽
- 44 東芝, DEC が偽 他社は真
- 48 東芝, DEC が偽 他社は真
- 51 沖 のみ 偽他社は真

である。

以上カナ文字に関して標準化が必要と思われるものは、

- 1 カナ文字のパターンコード
- 2 さらにカナ文字を大文字, 小文字などに分類するか?
- 3 カナ文字を大文字, 小文字などに分類したときこれらのコードは.?
- 4 カナ文字の範囲は?
- 5 4.と関連して、160(10)~165(10), 176(10), 222(10), 223(10)
の取扱い。

の以上5点にしばらく思われます。

現在上記5点に関する意見を各ユーザに、アンケートの形として、提出して
あります。

K A, ^A S A="アイエ", A(1,2)="サシセ", ^A=A, ^A(1,2)=A(1,2)
S W="FAPE", X="FAPBFCDFEFAFGPH", Y="FAFAFAFAFAFAFAFAFAFAFA"
S Z="FAIBWCIDIEKFGHIEJ", AB="ABCDEFGHIJKLMNQRST", KN="アイエオカククコサシセソタチツテ"

の実行後

```

1 W $E(Y,3,5),!
2 W $E(X,3,5),!
3 W $F(Y,"I"),!
4 W $F(X,"I"),!
5 W $L(Y),!
6 W $L(X),!
7 W $P(Y,"A",4),!
8 W $P(X,"A",4),!
9 W $A(Y),!
10 W $A(X),!
11 W $J("A1",6),!
12 W $J("A2",6),!
13 W $S("A1"="A":0,"A"="A1":1,"A1"="A1":2,1:"3"),!
14 W $D(A),!
15 W $D(A(1)),!
16 W $D(A(1,2)),!
17 W $D("A"),!
18 W $D("A(1)),!
19 W $D("A(1,2)),!

```

20	W	X_Y, !
21	W	Y_KN, !
22	W	KN_JY, !
23	W	X_Z, !
24	W	Z_X, !
25	W	A_KN, !
26	W	K_NA, !
27	W	W_X, !
28	W	X_W, !

```

29 W +X; !
30 W +Y; !
31 W "FAY"=0; !
32 W "FAY"=+0; !
33 W + "FAY"=0; !
34 W "1FAY"=1; !
35 W "1FAY"=+1; !
36 W + "1FAY"=1; !

```

87 R !, "カ" 'ア' ? = ",K,! F I=1:1:3 W \$A(\$E(K,I)),!
38 R !, "AL" 'A B' ? = ",A,! F I=1:1:3 W \$A(\$E(A,I)),!

5 KS=\$E(K,2),AS=\$E(A,2),KANA=\$C(160),ALPH=\$C(32)

の 実行 後

39 W KS?, K, !
40 W KS?, F, !
41 W AS?, K, !
42 W AS?, F, !
43 W KANA?, K, !
44 W KANA?, F, !
45 W ALPH?, K, !
46 W ALPH?, F, !
47 W \$C(160)?, K, !
48 W \$C(160)?, F, !
49 W \$C(32)?, K, !
50 W \$C(32)?, F, !
51 W AS=KS, !
52 W KANA=ALPH, !
53 W \$C(32)=\$C(160), !

カナモードで"アイ"を入力

英数モードで
"A+B"を入力

表 2-4 アンケート内容

29	F	
30	F	
31	F	
32	F	
33	T	
34	F	
35	F	
36	T	
37	A 177	
※	177 32 178 → 沖のみ 160	
38	A 65	
	32 66	

	三菱	三井	東芝	DEC	日本三洋
39	F	F	F	F	F
40	T	T	T	T	T
41	F	F	F	F	F
42	T	T	T	T	T
※ 43	F	F	F	T	F
※ 44	T	T	F	F	T
45	F	F	F	F	F
46	T	T	T	T	T
47	F	F	F	F	F
※ 48	T	T	F	F	T
49	F	F	F	F	F
50	T	T	T	T	T
※ 51	T	F	T	T	T
52	F	F	F	F	F
53	F	F	F	F	F

※ 各社間で差が見られたもの。

表 2-5 実行結果

§3 互換性と標準化

§2のカナ文字の現状を標準化することは、互換性の向上にもつながらるものと思われる。カナ文字の仕様の標準化については、やはりMUMPSを実際に使用しているユーザの声が反映されるべきだと思われる。

互換性の問題は、カナ文字の標準化よりはさらに次元の高い問題であるが、各メーカーともカナ文字を表すのに、内部コードとして、8単位符号を用いているので、紙テープなど7単位で出力される可能性のある媒体では、カナ文字をどう取扱うかが、問題とってくる。(S1, S0コードの使用など)

ルーチンの移送を考えると最も原始的な方法はルーチンのリストであるが大きなパッケージでは、この様な方法は、現実的ではなく、紙テープ、磁気テープなどで移送を行うのが、一般的である。

各社の移送に使用される入出力媒体の現状を表るー1にまとめておく。現在のところ最も利用価値、可能性の高いものは、磁気テープと考えられる。しかし磁気テープでは、トラック数、記録方式、ビット密度の問題、ブロック長の問題、End of recordの問題さらにヘッダーラベルの問題があり、MUMPSアプリケーションでは、解決できないハンドラーレベルの問題とってくる。

これらの問題の解決には、やはりメーカーの協力が必要となり、力添いの討議を行なう必要がある。この問題に関しては、MUMPS向のみの移送だけでなく他言語とのデータの移送を包括して、考える必要がある。(この辺の詳細い内容については、羽根野病院・今井の発表を参考にして下さい。)

しかし一定の磁気テープの出力フォーマットが確立されれば、カナ文字に関しては、ルーチン、グローバルなどの移送が活発に行なわれるものと期待されますので、関係各位の御協力をお願いしたいと思います。

	DEC	シ中	日本ミニコン	NEC	三菱	東芝	ホナフアエム
PTR	NO SUPPORT (換言中)		{ 8単位JIS 7単位JIS } 110175 無 偶 奇 オンラインで変更可	NO SUPPORT	カナデータの先頭に S0コード 7ビット(N0175有) 8ビットは容易 に交換できる		NO SUPPORT
MT	EBCDIC 以外		JIS 8単位		JIS 8単位 対応バイナリ- データ	JIS 8単位	EBCDIC (8単位)
その他	ターミナル (DL, DZ, DH) ラインプリンタ その他について は、有料で サポート	DOS 及び ハードの仕様 に従っている	カード(パナミック) 入力のみ IBM 29コード (ZREADで) (バイナリも可)		カード JIS 8単位対 応バイナリ-データ ラインプリンタ- S0 + カナデータ		カード 入力のみ 29圧縮コード

表 3-1 各社入出力媒体の現状

第6回日本MUG学術大会 昭和54年9月14日(金)

D. パネル(3) MUMPSの医療への応用 司会 大阪医大 山本和子

D-1.

MUMPSによる健康管理システム

防衛医科大学校

西田 正之 渡辺 隆郁 風間 富栄 他ME同好会員

砂川 裕之 高谷 治

昨今、健康づくりの運動が盛んである。実際、いろいろの地域社会で、いわゆる成人病を予防する上で、悪性新生物は勿論、脳血管障害、心臓病等の疾病は、若いうちから十分な健康管理が考えられなければならない。我々は、MUMPSを利用して、我々自体の健康管理システムを作ろうと試みた。

目的； 我々防衛医大生を対象として、身体情報をMUMPSを用いたシステムにより管理し、様々な角度から検索を行い、集団としての統計処理を行う。一方、個人的にデータを編集して、個人データ表を作成する。このデータ表には、データより健康状態をチェックし、若干のアドバイスも加えて表示するようにして、健康づくりのための情報を提供する。また、学生自体、或は各方面の依頼に答えて、例えば学生の身長順リストを作成し、或いは身長分布ヒストグラムを出力し、その他可能な限りの要望に応えることのできるよう将来にそなえてシステムに拡張性を持たせることを目的にした。

方法； 使用しているミニコンの主メモリは64KB、データは、ME同好会員(学生)が分担して入力している。出力はTWをもちいている。ハードウェア構成を図1に示す。

入力ソースは身体歴及び体力測定実施報告であり、これらは、学生部より提供を受けた。

身体歴より

①ID・氏名・出身地

②身体計測値

③各検査値等

体力測定実施報告より

①2,000m走

②100m走

③立幅跳

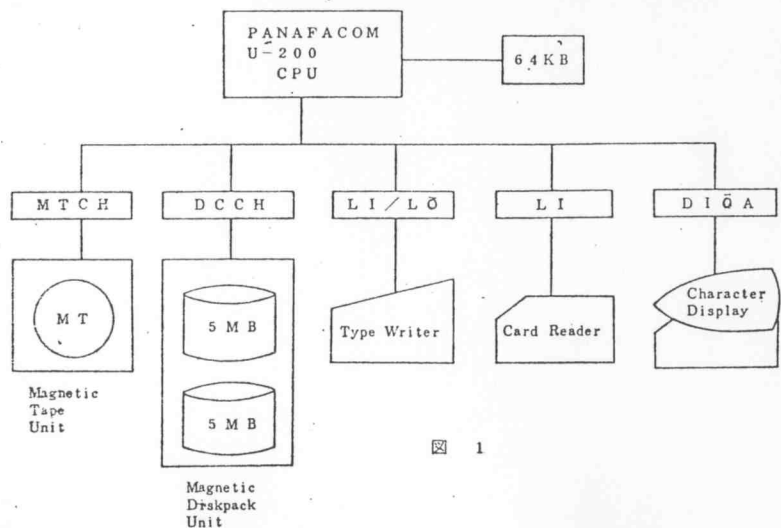


図 1

④ソフトボール投げ

⑤けんすい

の各データが得られる。登録システムとして、二次登録の防止、個人のプライバシー保護の対策等を一系列とした一群のプログラムシステムを用いて、これらのデータから、全情報を管理するマスターファイル、又目的に応じて、例えば、身長をKEYとするインバーテッドファイルが作成される。身長をKEYとすることにより、SORT機能が実現される。更にアプリケーションプログラムとして、身長順リスト及びヒストグラムを作表するプログラム、指定するデータの平均・標準偏差・最大・最小を求める汎用統計プログラム、出身地のヒストグラムを作成するプログラム、血液型のヒストグラムを作成するプログラム等が現在用意されている。

プログラム構造はパーティションエリアの制約・プログラムの拡張及び簡潔化のために、各機能ごとに分割し、サブルーチンコールする形式をとった。

ソフトウェア構成の概念図を図2に示す。

マンブスシステムの
特徴を考慮し、マスター
となるデータベース
は浅いレベルで作成し
メモリの節約をはかる
とともに、アクセス回
数を減らして容易にア
プリケーションプログ
ラムの作成及びシステ
ムの拡張ができるよう
単純構造化するように
努めた。MUMPSで
は1ノードにつき1KB
のエリアが確保され
るので、1ノードにつ
き10人分のデータを
収めてある。1人のデ
ータ長が90文字程度
であるので、エリア利
用率は90%弱と考え
られる。

マスターファイルの
構造を図3に示す。

インバーテッドファ

イルは、身長を6ブロックに分け、ブロック番号を第一添字、身長(mm)を第2添字として、マスターファイルと同様、浅いレベルで抑えた。

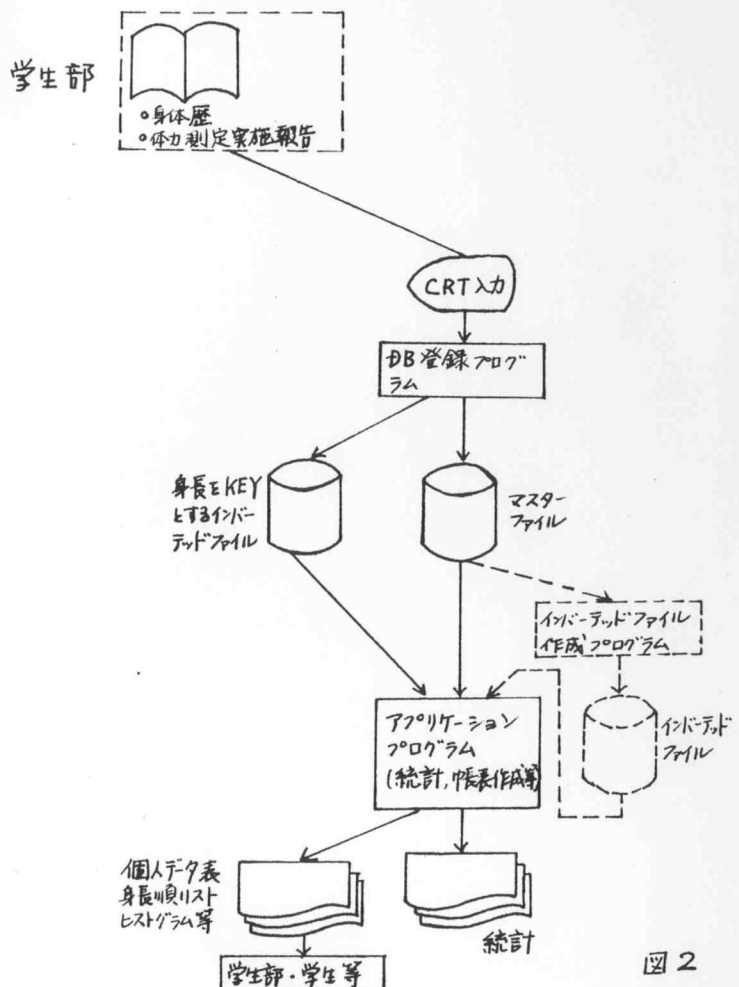


図 2

結果及

考察； 我々の大学は、学生数が500名弱であり、身体検査及び体力測定は年一回行われる。データが割合容易に入手でき、形式が一定しているためにデータベースが作成し易い。

年一回の業務であるから、ディスプレイ入力でも不便を感じない。また、処理速度も遅いが十分といえよう。

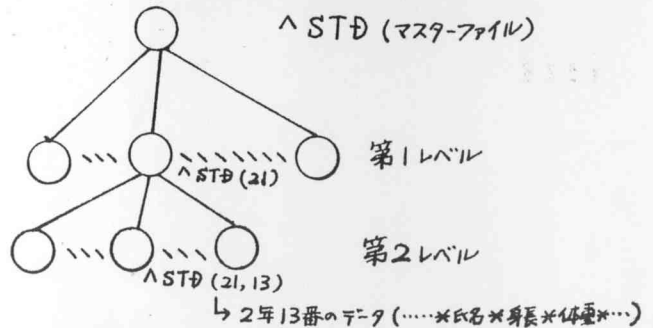


図3

システムを作るにあたって注意したことは、マスターファイルのエリア利用率を高めレベルを浅くして入出力・変更が容易にできるようにしたことである。マスターファイルの構造が単純であるから、マスターファイルからインバーテッドファイルが簡単に作成でき、アプリケーションプログラムの開発が容易である。一学生のデータ長が90文字弱と一定していることは、データベース構造の決定には好都合であり、ディスクメモリを節約することができた。

データはなるべくコア上で扱うようにして、処理時間の短縮をはかっているが、まだ速い出力を考える余地がある。現在は各学年毎に統計を出しているが、そのうちに、経年的なデータも入力されるので、更に応用が拡大することは明らかである。最近MUMPSと同様に同じOSのもとで使用できるFORTRANが提供されたので、この利用を検討中である。

統計処理においては、各データ間の相関関係をみたり、各種の検定を行うなど様々な可能性がある。また、健康アドバイスを決定するにあたっての判断論理を充実させ、運動処方箋にまでもつてゆきたいとも考えている。実際、判断論理はかなり幼稚なものであり、必要な各データをすべて考慮して判断させるように改良したい。システムの充実のためには入力ソースを拡大してゆく必要を感じる。例えば、年間の通院・受診歴などはぜひ加えたいところである。

本年度からスタートしたシステムであるので、経時の変化をトレースすることは来年度以降の課題であり、ME同好会員のMUMPSプログラム演習も兼ねてひきついでいきたい。

第6回日本MUG学術大会 昭和54年9月14日(金)

D. ハネル(3) MUMPSの医療への応用 司会 大阪医大 山本和子

D-2.

全国剖検症例の検索

木村園恵^x, 木村一元^{xx}, 馬場謙介^x, 浦野順文^{xxx}, 藍沢茂雄^{xxxx}

^x 獨協医科大学 第一病理学教室

^{xx} 獨協医科大学 総合研究施設

^{xxx} 東京大学 医学部 病理学教室

^{xxxx} 慈恵会医科大学 病理学教室

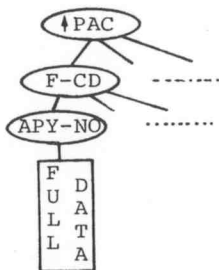
1 はじめに

初め、統計表を電子計算機で作成する目的で、全国の病理剖検症例に関するデータを蓄積していたが、4年を経過すると、この大量のデータに目をつけて、その中から目的とする症例を検索したいという要望が起こってきた。主として、疾病の疫学研究があるいは稀有な症例の報告者からの要望である。これに答える目的で今までのSYSTEMを条件付の症例検索システムに改良したので、これを報告する。

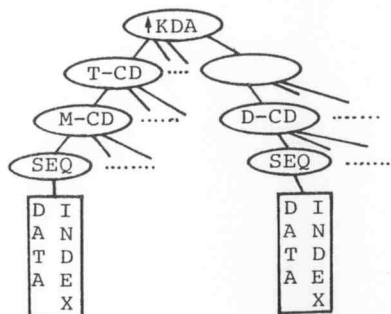
2 GLOBAL FILE

今までのSYSTEMで築き上げたMAINGLOBAL FILEのはかに、検索を考慮して2つのGLOBAL FILEを作った。したがって、現在我々のSYSTEMが持っているGLOBAL FILEは次のとおりである。

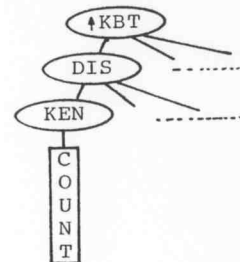
MAIN FILE



疾病別 FILE



県別 FILE



F - C D : 施設コード
A P Y - N O : 剖検番号
T - C D : M O T N A Cコードに準ずる悪性腫瘍原発部位コード
M - C D : M O T N A Cコードに準ずる悪性腫瘍組織型コード
D - C D : I C Dコードに準ずる悪性腫瘍以外の疾病コード
D I S : I C Dコードに準ずる疾病コード
K E N : 自治省自治体コードに準ずる都道府県コード

3 検索SYSTEMの使用頻度と利用方法

本SYSTEMへは、年間約50件の単純な検索依頼及び、数件の複雑な検索依頼が寄せられる。一般の情報、特に商業ベースの検索サービスでは、検索KEYの設定の仕方の良し悪しは、依頼の責任で行われている。これに対し我々は、充分コンサルテーションを行っている。このコンサルテーションの内容について述べる。現在のところ、このSYSTEMは、実質負担で、病理学会員に開放されている。

4 むすび

我々の検索SYSTEMは十分に需要に答えられていると考えられる。これまでのDATA BASEには、入っていない腫瘍からの転移・浸潤臓器、転移リンパ節等の情報も扱えるようにすると、利用価値はさらに増大すると思われる。これらの副データを扱うことによって生じるSYSTEMへの負荷、DATA保持、検索プログラムの拡張を考慮しながら、試験的に副データの入力を始めている。

第6回日本MUG学術大会 昭和54年9月14日(金)

D. パネル(3) MUMPSの医療への応用 司会 大阪医大 山本和子

D-3. 血液透析用病態解析システムについて

(財)鷹揚郷腎研究所

平山 順朗
磯野 正子
内山 公一
石戸 谷 豊

山内 裕子
八木 槁栄子
森 喜代志
舟生 富寿

慢性腎不全に対する慢性血液透析療法は現在では広く普及し、腎疾患治療の中で重要な位置を占めるようになっていきます。しかし、慢性腎不全は全身の諸臓器にさまざまな影響をおよぼすことや、本療法が週3回、1回 5~6時間で終生継続されるなどの特殊性から、これらの患者の長期にわたる症状の把握、多彩な合併症の原因の究明などはそのデータ量があまりにも膨大なため、ほとんどおこなわれていないのが現状です。したがって一般には臨床検査データ等を参考にしながら、さわめて大まかに、経験的に治療にあたっているわけです。そこで私達は、人工腎治療における管理基準の判定や多彩な症状を示す人工腎患者の病態を十分把握することなどを目的として、血液透析病態解析システムの導入を試みにわけです。

以上のような点からまず、データの収録を完全に行うことを最終目的であるデータの統計・解析に主眼点をおいてシステムの構成をすすめてきました。現在はまだ未完成ですが、本システムの特徴や、今後の計画等について報告いたします。

なお、ハードウェア構成は CPU 128KB・DISK 200MB 2台、その他周辺機器で使用する語は、MUMPS-900を用いています。

1. データベースの特徴

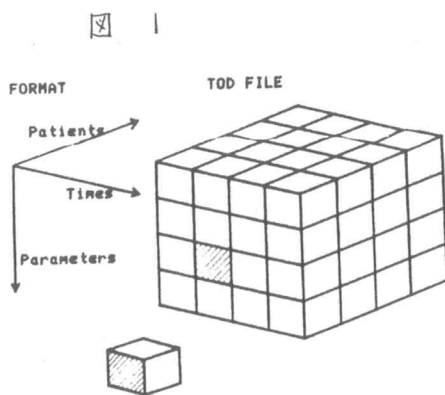
まず、前述した血液透析療法の特殊性を十分考慮して、Dataの収録する範囲は治療に直接関連のある血液透析記録(図3)は勿論のこと臨床検査結果や投薬内容等も含んだ日常の診療記録に記載されたものを全てを含むこと、さらに看護記録、ケースワーカー記録までを対象としています。これらの内容は表1に示したとおりです。したがってこれらのDataの特性上、このデータベースは基本的に図1に示したようなTime Oriented Databankの形態をとっています。

2. グローバルファイル(↑KDD)について

本システムに用いられるグローバルファイルは、KIDNEY DIALYSIS DATABANKと名づけられ、↑KDDと略称されています。

↑KDDのTree構造の概要は図2に示したとおりです。本ファイルは基本的には、前述したようなTOD方式をとっており、透析時系列項目（TOR）ではこの時系列体系をさらに一段階下げた0時0分の記録までを収録できるようになっています。図4は↑KDDのGlobal Listingですが、これからDisk内の構成を想定していただけるものと思います。なお、↑KDD（74055, 7406, 10）は図3に示した血液透析記録の内容のほとんどを収録していますが、血液透析療法は年間1人当り150~160回と頻回に行うものであることを考えますと、↑KDDの大きさも想像していただけるものと思います。また図2の右に示した↑IMFは↑KDDファイルを効率よくサーケする為のサブファイルです。添字J, Kの情報により時系列項目（DIG, TOR, EXG, TRE）と非時系列項目（IDG, PHG）項目との相違、又検査項目（EXG）の小分類等を表わしています

表1 データベースの内容



1. 患者基本項目（IDG）

ID番号・氏名・生年月日・住所等21項目

2. 透析基本項目（DIG）

透析日、人工腎の種類、灌流液の種類等23項目

3. 透析時系列項目（TOR）

時間、血圧、血流量等10項目

4. 検査項目（EXG）

血液生化学、ガス分析、内分泌、血液像、胸部X線、心電図等15項目

5. 治療項目（TRE）

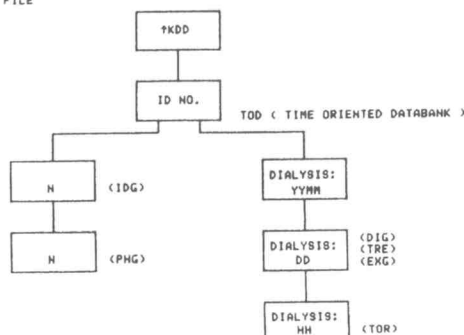
投薬内容、入院経過表、食事記録等14項目

6. 既往歴項目（PHG）

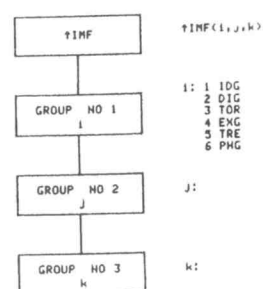
腎疾患発症時期、治療歴、家族歴等21項目

図 2

*KDD FILE



*IMF FILE



[illegible]

图 4

PAGE 1

```

74005 検査の回数 (15 種の検査)
- 1 透査 506/0/0/0/0/0/0/0/0/175/0/0/0/0/0
- 1.1 74005^0000 00^B0^M^S/21/03/30^A アモリ^^74U608^CRF
- 1.2 CGN-B^HD^^ヒロタヱ ヒニョウキカ } 患者基本情報 (IDG)
- 1.3 ^^A/+^^^^^Y^790307
7406 YMM
- 10 DD 1/0/0/0/0/0/0/0/0/1/0/0/0/0/0
- 1 S^0905^38^12^200^500^^^
- 2 N^0920^38^12^200^500^152/90^^^
- 3 N^1000^38^12^200^500^^^0193
- 4 N^1030^38^12^200^500^152/80^72^^
- 5 N^1100^37^12^212^500^150/90^90^0193
- 6 1日分 N^1200^37^12^212^500^130/70^82^36.8^
- 7 の情報 N^1300^37^13^180^500^160/92^88^0193
- 8 N^1400^38^14^184^500^142/90^72^37.2^
- 9 N^1500^38^14^184^500^140/90^72^^
- 10 E^1520^^^^^^ A1
- 11 A^1540^^^^^90/72^^^0022
- 10.01 740610^01^^^00^35^6.0^62.5/60.0^0^0 } 透析基本情報 (DIG)
- 10.02 ^^200^^^2700^0850^150/80^^^
- 10.1 6.6/6.6^1.1/1.2^^^6.5/5.7^22/^21/^81/39^18.1/10.9^ } 検査項目
- 10.25 146^15.5^5.1^4300^^^^^^^^ } データ
- 10.62 12.4/5.5^134/133^5.8/3.4^107/104^7.8/9.7^8.6/5.1^^^^^ } (EXG)
- 12 2/0/0/0/0/0/0/0/0/1/0/0/0/0/0 BLCH
- 1 S^0900^39^12^200^500^^^0193
- 2 N^0910^39^12^200^500^140/60^82^^
- 3 N^1020^39^10^200^500^120/50^80^36.8^0193

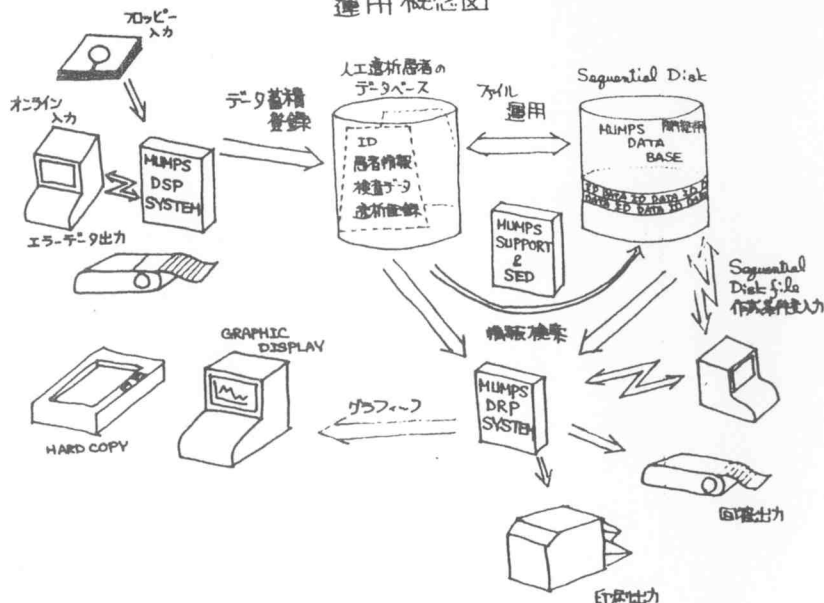
```

3. システムの運用

図5に本システムの運用概念図を示しました。現在は主に、1) データ登録・修正ルーチン 2) データ検索ルーチン 3) データ保全・運用ルーチン 4) グラフックディスプレイ運用ルーチンの4つの部位から成り立っています。

図 5

運用概念図



1) データ登録ルーチン データ登録ルーチンはフロッピー (FLOP) 入力と CRT のオンライン入力によってなされています。この時にエラーデータの掃き出し等は CRT の場合は即座に行えますが、Flop 入力に関しては入力時にエラー修正を行えないことが、現状のネックになっています。しかし入力時間に関しては 2~3 倍のオーダーで Flop 入力の方が早く入力できています。最終的には、File の容量は 300,000 Block 以上のファイルになるはずです。現在も入力しつづけています。

2) データ検索ルーチン これには 2 つの検索方法が考えられています。1 つは MUMPS データベースの ↑KDD ファイルからの直接検索する方法です。

もう 1 つは ↑KDD ファイルより指定したデータを抽出し、これからシーケンシャルファイルを作成し、それを使用した間接的なデータ検索の方法です。これは現在開発中ですが、シーケンシャルファイルは RRA 方式 (Random Record Access) を使用し、それによって MUMPS データベース検索の時間を速める方法をとっています。

3) データ保全・運用ルーチン 我々のシステムにおいては、特に収録データがすべての基本となっています。それ故、不慮のトラブルによるマスターファイルの破壊対策として、MAIN DISK の内容を SUB DISK (または MT) Save する必要があります。これらを図滑に行うためにこのルーチンを持っています。

4) グラフィックディスプレイ運用ルーチン グラフィックディスプレイは、検索結果等のグラフ表示に使っています。現在はオフラインで入力されていますが、将来は CPU と接続してオンラインとして使用することを考えています。

図 6

*** オオヨウキョウ KDD システム ***

79/03/19 PAGE

<< FDD ニュウリョク >> エラー リスト

NO. = 11 スキップ* データ ケンサ データ (2) ID = 74041 DATE = 771220
セクタNO スキップ* データ

277* ---^---^---^---^---^ALB^ 63.0^ ^A1G^ 3.0^ ^A2G^ 1
0.1^ ^BG ^ 9.0^ ^GG ^ 14.6^ ^ ^ ^

NO. = 12 スキップ* データ ケンサ データ (2) ID = 74039 DATE = 771219
セクタNO スキップ* データ

320* ---^---^---^---^---^ALB^ 71.9^ ^A1G^ 2.3^ ^A2G^
8.4^ ^BG ^ 2.5^ ^GG ^ 14.6^ ^ ^ ^

図 7

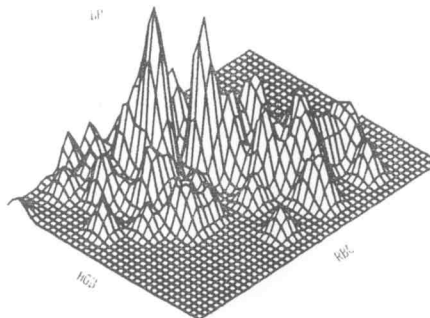
*** オオヨウキョウ KDD システム エラー リスト ***

ID= 75029 < シメイ > XXXXXXXXXX < キカシ > 740101-781231

DATE	BW=1	DWIG	ULT	S=RP	D=BP	TP	ALP	LDH	GOT	GPT	BUN	CRN	HA	NA	KN
751202	59.0	7	4980	170	110	7.1	5.3	225	14	17	81	15.5	13.3	133	5.0
751204	56.5	0.5	1570	160	100	7.6	6.6				77	14.2	11.6	133	4.3
751206	57.0	1	1620	190	110	7.2	5.6	352	47	35	71	13.4	14.2	134	
751209	57.5	1.5	3750			7.0	5.5	249	20	22	89	16.6	13.2	133	4.6
751211	57.0	1	1430	170	100	7.2	5.2				78	15.0	10.7	135	4.8
751213	56.8	0.8	1900			7.2	5.3	464	17	13	70	13.3	9.4	127	
751216	57.0	1.8	1850			7.9	5.6	226	9	17	79	14.8	10.6	135	4.8
751218	57.0	1	1670			7.2	5.2	301	8	20	64	12.7	10.3	133	4.2
751219	56.8	0.8	1500	130	90	7.1	5.6				61	13.0	9.8	135	4.2
751222	57.5	1.9		210	120	7.3	5.5				80	13.3	11.3	135	4.6
751223	56.5		1300			7.7	6.7	232	13	16	63	12.1	9.6	133	4.7
751225	56.5	1	1780	150	110	7.5	6.3				65	11.6	9.7	133	4.2
751227	56.5	1.5	1230			7.2	6.1				67	11.3	11.6	134	4.3
751229	57.0	1.2	2865	180	120	6.8	5.8				69	13.3	10.2	135	4.3
751231	57.0	1.5	1600	150	110	6.9	5.1				75	13.3	10.1	132	4.7
ツキハイキ	57.0	1.6	2075	168	108	7.3	5.7	293	18	20	73	13.6	11.0	133	4.5

図 8

*** ODOYOKYO KDD SYSTEM ***



4. 現在の問題点と今後の計画

1) データベースの問題点 現在データベースに登録されているものは、血液透析記録が主体であり、投薬内容やX線所見、心電図記録等が未登録のままとなっています。これは、十分な仕様の検討がなされていないため今後早い時期に登録を開始する予定でいます。

2) シーケンシャルファイルとグラフィックディスプレイ 現在開発中のシーケンシャルファイルは、RRA方式をとりますがこの方式は、Record番号指定又Block番号指定が可能なのでMUMPSのノードサーチの方法と同じ考え方でサーチできるメリットがあります。またグラフィックディスプレイとMUMPSをオンライン化して接続する時にもシーケンシャルファイルで構成しておけば、プログラムの簡易化ができるのではないかと考えています。グラフィックディスプレイにどのような機能をもたせるかなどは今後に残される問題といえます。

以上、本システムについての現状と計画等について簡単に紹介させていただきました。
なお、本システムは日本自転車振興会の補助事業によるものであり、東芝メデカル・東芝電算機事業部各位の多大な御協力を頂きました。関係各位に深謝致します。

第6回日本MUG学術大会 昭和54年9月14日(金)

D. ハネル(3) MUMPSの医療への応用 司会 大阪医大 山本和子

D-4. 産科研究データと産科臨床データとの統合データベースの作成

京大産婦人科・国立京都病院 吉村 誠之

京大病院中央情報処理部 平川 顕名

—緒言—

臨床研究は、一般に問題となる研究データと病歴から得られる臨床データとの対応を明らかにすることが基本とされている。今回、ヒト羊膜上皮細胞層の妊娠経過に伴う変化の生物学的ならびに産科学的意義を明らかにするために、ヒト羊膜上皮細胞層の形態計量所見のデータベースと産科病歴のデータベースとをMUMPS言語を用いて作成し、両者を統合してデータ間の対応をみるという臨床研究を行なったところ、有意義な成果を得た。そこで、研究データと臨床データとの統合データベース作成におけるMUMPS言語のソフトウェア論的検討を行なった。

—目的—

MUMPS言語を用いて、羊膜研究データと産科臨床データの統合データベースを作成すること。データベースの作成に際しては、以下の5項目を満足させるように努めた。

- (1) 産科臨床研究の立場から、MUMPS言語の初心者でも短期間にしかも最終的には自力でプログラミングを行なうこと。
- (2) 入力を柔軟に行なうこと。すなわち入力時の文字と数字の使用規則を少なくして臨床および研究データの適時の分割入力方式ができる等の臨床研究上の特色を配慮すること。また入力データの許容量を大きくすること。
- (3) 記憶ファイルの設計を柔軟に行なうこと。すなわち入力修正と同時にファイル修正も可能にし、また入力後のデータを用いて新しいファイル設計が次々と目的に応じてできるようにすること。特に症例の臨床分類を多方面にわたってできるようにすること。たとえば児の娩出期別の臨床分類、児娩出時の体重別の臨床分類、正常例と異常例の分類などのファイルが自

動的に作られ、また随時新ファイルが設定できること。

- (4) 統計計算を臨床研究に必要なレベルで行なうこと。少なくとも相関係数や有意差検定ぐらいはできること。
- (5) 出力は多目的に行なうこと。

—方法—

ハードウェアはDEC-PDP11/40のミニコン(基本データ長:16ビット/言語, 記憶装置タイプ:磁気コア, 記憶容量:24K語, ディスクブロックの大きさ:265語)

ソフトウェアはDEC-MUMPS11(データ長:8ビット, ASCII, 文字列長:132文字, 添字:0~20971.51, 数値データ:±21474836.47, ステップ番号:0.01~327.67, データベースのデータの位置:ディスク上にある。)

入力データは正常および異常経過を有し、妊娠初期に流産した例から分娩予定日超過の分娩例に至る妊娠各期娩出の約150症例の母体、児(胎児と新生児との総称)、胎児付属物に関する臨床データ、およびその症例から採取した羊膜の採取部位別の細胞形態計測値に関する研究データである。

—成績—

目的の項で述べた5項目に関する満足度を成績とした。

- (1) MUMPS言語を用いて、羊膜研究データとその症例の臨床データとの統合データベースを作成するためのプログラミングは、産婦人科研究者の立場から、自力で行ないえたと考える。MUMPS言語の学習は、京大病院中央情報処理部主催のMUMPS講習会で行ない、講師の個人指導を受けながら学習を始めて約90日間で、当初の目的に沿ったプログラミングを終え、実行にも成功した。その後約10日間で全データを入力し、必要な出力を終えた。なお90日未滿というのは、京大病院産婦人科医員としての日常業務を遂行しながら、講習会でMUMPS言語に初めて接して学習しつつ、プログラミングを行なったのべ日数である。

- (2) 入力は柔軟に行ないえたと考える。すなわち、入力データは数字と文字を組み合わせて、最適の入力が可能であった。たとえば、破水後経過時間とその間に行なった投薬治療などの注釈を入力する場合には、破水後経過時間は数字(時間は整数、分は小数で表示)で入力し、コロン(:)に引き続き注釈を132文字(数字も含む)で表わすこととすれば、破水後の臨床経過は、ほぼ完全に入力できることになる。次に入力データの修正は、入力後直ちに電動タイプライタのペーパー上で入力データを確認するさいに、修正が必要なら直ちに行ない得た。さらに、臨床研究を行なうに際して、臨床データと研究データとが全てそろふことはまれなので、手元のデータを逐次入力できるようにした結果、各種データの整理統合をノートへの転記などによって行なうことも

なく、最終的に臨床データと研究データの全てのデータをコンピュータ内でファイルすることに成功した。つまり、データの一括入力方式ではなく、適時の分割入力方式を実現させた。

次に入力データの許容量は大きくできたと考える。研究に必要なデータは全て入力できたと考える。症例の同定に関するデータ(8項目、演算後の自動入力を含めると9項目、うち3項目は132文字以内の記述式入力が可能)、月経に関するデータ(3項目、演算後の自動入力を含めると8項目)、遺伝、家族、生活環境歴に関するデータ(2項目、うち1項目は130文字以内の記述式の入力が可能)、妊娠歴に関するデータ(7項目、うち1項目は記述可)、今回の妊娠経過に関するデータ(6項目、うち3項目は記述可)、分娩経過に関するデータ(13項目、うち5項目は記述可)、児付属物に関するデータ(3項目、うち1項目は記述可)、児の娩出時点での転帰に関するデータ(3項目、うち1項目は記述可)、児の娩出様式に関するデータ(6項目、うち3項目は記述可)、児の生後6日間の経過に関するデータ(20項目、演算後の自動入力も含めると22項目、うち1項目は記述可)、児の生後7日以後の経過に関するデータ(4項目、うち1項目は記述可)、症例が新規ファイルに自動入力された時の分類番号に関するデータ(5項目)、羊膜標本作成過程に関するデータ(2項目、うち1項目は記述可)、胎盤実質部を覆う羊膜に関するデータ(11項目、演算後の自動入力も含めると34項目)、卵膜反転部を覆う羊膜に関するデータ(同上)、胎盤辺縁部を覆う羊膜に関するデータ(同上)、卵膜裂口部を覆う羊膜に関するデータ(同上)、が入力できた。すなわち、一症例につき直接入力データの最大許容量は131項目であり、さらに入力後に自動演算して自動入力されるデータも含めると最大許容量は236項目である。またそのうち記述式の直接入力が可能なのは21項目である。

- (3) 記憶ファイルの設計は柔軟に行ないえたと考える。すなわち、研究症例の同定番号を結節点にして、同一症例の各種ファイルの設計を行なった結果、同定番号さえ判明すれば、既に入力されディスクファイルされたデータでも、入力修正と同時にファイル修正も可能に示えた。また、入力後のデータを基に同定番号順のファイル、児娩出時点の妊娠週数順のファイル、児娩出時点の妊娠月数順のファイル、児娩出時点の妊娠月の2ヶ月単位順のファイル、児娩出時の体重-g単位順のファイル、児娩出時の体重-500g単位順のファイル、児娩出時の妊娠週数に対応する体重別の臨床分類(SFD, AFD, LFD)のファイルなどの新しいファイルの設計が、随時可能であった。

次に正常例と異常例を分類したファイルに関しては、上記のいくつかのファイルの中に分類体系を導入することによって、実質的なファイル作成を行なった。正常・異常の分類体系は、今回は3つ設定した。1つは児の娩出時点の生死およびいわゆる臨床的正常例・異常例の臨床分類に基づくもの、2つめは、児娩出時の妊娠週数に対応する体重別の臨床分類に基づくもの、3つめは、羊膜研究上のもので、羊膜標本作成過程の問題の有無別、

羊水量の正常・異常別、胎盤変性に関する問題の有無別に基づくものである。

(4) 統計計算は臨床研究に必要なレベルで行なうことができたと考える。たとえば、児娩出時の妊娠期別の正常例および異常例毎に、例数、平均値、標準偏差値、平均値プラス標準偏差値、平均値マイナス標準偏差値、分散値、最大値、最小値、範囲をそれぞれ得られ、また各分類群における各種入力データ間のあらゆる組み合わせの可能な相関係数、一次回帰式をそれぞれ得られた。しかしただ各分類群における各種入力データ間の有意差検定(尤検定)に関しては、MUMPS 11のVersion 3では\$Mの使用が不可能であったため、検定をしようとする二群の例数、総和、平方和を表示化して出力するところまではMUMPSで行ない、その後出力リストをみながら入力して有意差検定が可能なプログラムをBASIC言語で作成して、同じハードウェア(PDP 11/40)で実行させて検定することに成功した。

(5) 出力データを多目的に、ほぼ出しえたと考える。入力直後にディスクファイルされているかのチェックのために症例毎の臨床データと研究データとを分けて、別々に出力したり一緒に出力したとか、症例の全データを同定番号順に出力したり、症例番号のチェックのために同一娩出年月日の症例数とそれぞれのチェック番号および母親の姓を出力させたり、暦で表わした児娩出日の順に同一症例の数および個々の症例番号を出力させたり、母親の姓のアイウエオ順に症例番号を出力させたり、児娩出日の妊娠週数順に同妊娠週の症例の数および症例番号を出力させたり、児娩出日の妊娠月数順に同妊娠月の症例の数および症例番号を出力させたり、児の娩出時体重(g単位)の順に同一体重児の数および個々の症例番号を出力させたり、児の娩出時体重(500g単位)の順に同一体重児の数および個々の症例番号を出力させたりした。上記の後半の4つの出力では、正常例と異常例とを分けて出力することができた。次に(4)で述べた各種統計計算の結果も、出力できた。

—考察— (省 略)

—結論—

MUMPS言語を用いて、羊膜研究データと産科臨床データの統合データベースを作成することに成功した。その際、データベース作成上のソフトウェア論的観点から5項目の目標を設定したが、いずれも満足させることができた。

資料2 症例番号を同定した後、臨床データを入力することなく、いきなり研究データ(胎盤部を覆う部分より採取した羊膜の上皮細胞形態計測値)を入力した具体例。下の方1/2の(0)から(103)までのデータは、既に入力してある臨床データ(0)から(10)まで)に加えて研究データ(101)から(103)まで)が記憶ファイル上に入力されていることを示している。

入力開始

C B10

同定

BIRTH DATE(X/YY/MM/DD) : 5/50/10/8 --- (症例の生年月日)

(1)81-627-013 YOSIMURA SEIKOBERI M 5/50/10/8 KYODAI BYUIN YASUMOTO

(2)81-627-021 YOSIMURA MEIKOBERI F 5/50/10/8 KYODAI HIRAKAWA

COULD YOU FIND YOUR ID#? Y --- (症例の同定)

WHICH ONE IS IT ? 12

DO YOU WANT TO INPUT CLINICALS ? N --- (臨床データは入力しない)

DO YOU WANT TO INPUT AMNION-DATA ? Y --- (羊膜研究データを入力する)

^(1)=35/45/25^280^40^11^B^OVER-TERM

^(10)=7510.08^140^113.50

AMNION-DATA NOT YET !

ABNORMAL SAMPLE(0-5) : 2; FIXATION-SLI-LONGER=3DAYS

WHAT DATA DO YOU WANT (FC,BC,FM,BM OR N) ? FC --- (羊膜の胎盤実質部を覆う部分のデータを入力)

1)N-NUMBER(PIECE/10000MICRON-SQ) = 70^57

2)DISAPPEARED-N CELL(F/FIELD) = 3

3)FI-N CELL(F/F) = 4

4)N-AREA:FIELD(Z) = 30

5)C-AREA:FIELD(Z) = 20

6)N-AREA(MICRON-SQ/CELL) = 30

7)N-DIAM(MICRON/CELL) = 10

8)ALL S-NUMBER(F/F)/(F/10CELLS) = 12

9)ALL S-AREA(M-SQ/F) = 12

10)S-F-AREA(M-SQ/S-F) = 3

11)S-F-DIAM(M/S-F) = 1.5

ARE THESE DATAS WRIGHT ? Y

WHAT DATA DO YOU WANT (FC,BC,FM,BM OR N) ? N --- (これ以上研究データの入力は無い)

DO YOU WANT LIST-ALL,-DATA,-CLEVEL#,OR N ? ALL --- (この症例の全ての入力を入力する)

(0) YOSIMURA MEIKOBERI F 5/50/10/8 KYODAI HIRAKAWA S 16/7/8 S 50/1/1 S 50/10/7

(1) 35/45/25^280^40^11^B^OVER-TERM

(2) 2;DIABETIS-MELLITUSCH01;HYPERTENSIONEFA1

(3) 34^2/1/2/0;LAST DERIV-CS^2;30Y0-2ND-DERIV=FLA;PREVIA

(4) 2/MAR.1-2=BLEED;PROLUTON-DEP01125MG^2;APR.2-7=LEGS-EDEMA^1

(5) 0.30^10.30^0.15^2;LREDE-PRESS^3;MECONIUM-STAIN^2;CER-LACE-SUTURE/400/350

(6) 500/470/^20/18/2^50/2/1.8^500;ALMOST^1

(7) 0^0^1;NO-EXTRACT&PRESS-BECAUSE-PRECS^F;MAMILLA-MICKY-DISCHARGE^2;HAIR^+

(8) 3.5/18/20^50/35/12/10^33/12/10^8^1^1AFD^FULLT-INF

(9) 3.7/5/^2/FEVER38-0C1.20-22

(10) 7510.08^140^113.50

(101) 2;FIXATION-SLI-LONGER=3DAYS

(102) 57^3^4^30^90^30^10^12^12^3^1.5

(103) 56^5.35^7.14^7.01^17.10^10^53.57^160.71^33.33^0.21^21.42^2.52^13.33^40^1.75^0.

56^8.50^78.50^0.38^0.68^1.76^1.70^12.17

DO YOU WANT LIST-ALL,-DATA,-CLEVEL#,OR N ? N

BIRTH DATE(X/YY/MM/DD) : _

羊膜研究データの入力

臨床研究データの構造と出力

第6回日本MUG学術大会 昭和54年9月14日(金)

D. パネル(3) MUMPSの医療への応用 司会 大阪医大 山本和子

D-5. 標準マンプスによる退院病歴の管理

——病名の入出力について——

大阪医大 衛生・公衆衛生 山本和子

退院病歴を診療、研究、医療評価に、より有効に利用できるようにするために、病歴管理のシステムを開発したので紹介する。

使用機種はPDP11/34, メモリー32 KW, ディスク1.2 MW×2, タイプライター2台で言語は標準マンプスを用いた。

病歴管理の全システム・フローは図1に、主要ファイルの内容を表1に示す。病院医事課の退院カードより退院患者の氏名等をカルテ回収ファイルに入力し、各科にカルテ請求書を出し、カルテを回収する。次にカルテの内容を患者登録票(図2)に転記し、患者マスターファイルに入力している。データの出力は全入力項目について必要時に必要なものだけを作表できるようなプログラムを作成した。

1) 患者マスターファイルの構造(表1)

Aレベル: 科コードと退院年度

Bレベル: ID番号(生れた月と日)

Cレベル: ID番号(生れた年…明治元年より計算した年…と3ケタの順位番号…女は501より)

③ 患者の基本情報

Dレベル: 退院した月と日

④ 入院中の患者情報…主治医名はカタカナで、その他の項目はコード化して数字で入力

(D+0.01) 退院時診断名

(D+0.02) コメント(250字以内)

(D+0.03)

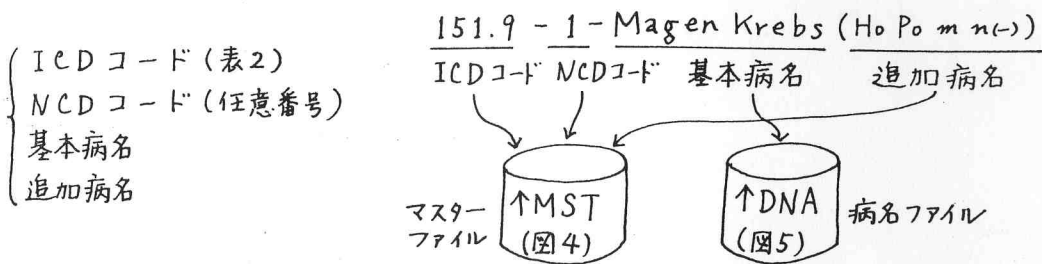
現在入力していないが将来各種情報を追加の予定

Eレベル: 新生児順位番号

⑤ 産科母のファイルに新生児記録を入力(新生児のカルテが独立していないので)

2) 病名の入力 (図3)

例えば、カルテに Magen Krebs (Ho Po m n(-)) と記入してあったとき、ICDコード、NCDコードをつけて下図のようにそれぞれのファイルに入力する



3) 病名のコード化

病名をそのまま入力して、自動的に必要な病名のみコード化している

①はじめに病名を基本病名と追加病名に分けICDコードをつけて

ICDコード, NCDコード, 基本病名, 追加病名を入力する

②↑DNAに基本病名のデータを貯える (ICDコード不明のとき↑DNAより参照)

③↑DNAの病名を整理 (病名の表現法, 種類等)

④↑DNAより患者登録票 (図2) にのせる病名を選定して科別患者登録票を作成

⑤患者登録票 (図2-2, 3) より入力

主要疾患は ICDコード, NCDコード, 追加病名のみ入力すればよい

利 点 1. ICDコードだけでは満足できない微妙な表現もそのまま入力できる

2. 同一病名は再度入力しなくてもよい. 入力時間の短縮

3. ICDコード, NCDコード不明のとき↑DNAより参照できる

4. 最終的には患者登録票に主要病名をセットしておくことによりカルテから病名の転記, ICDコードをつける時間の短縮

問題点 1. 基本病名を何にするかという決定がむづかしい

2. 同じ病名で、日本語, 英語, 独語, ラテン語等の表現をどう整理するか?

3. ↑DNAが大きくなりすぎるおそれがあるため, 使用頻度の少ない病名は追加病名として扱う等の工夫が必要, あとで整理しなければならなくなる可能性あり

表2 ICDコード表

国際疾病分類を基本にしている

病 名	ICDコード	病 名	ICDコード
I. 感染症, 寄生虫	1~139	XII. 皮膚, 皮下組織	680~709
II. 新生物	140~239	XIII. 筋骨格系, 結合組織	710~739
III. 内分泌, 栄養, 代謝, 免疫	240~279	XIV. 先天異常	740~759
IV. 血液, 造血器	280~289	XV. 周産期	760~779
V. 精神障害	290~319	XVI. 診断名不明確	780~799
VI. 神経系, 感覚器	320~389	XVII. 損傷, 中毒	800~999
VII. 循環系	390~459	E. 損傷, 中毒の外因	4800~4999
VIII. 呼吸系	460~519	V. 健康, 保健サービス	5001~5082
IX. 消化系	520~579	† のついた二重コード	1001~1999
X. 泌尿生殖系	580~629	* のついた二重コード	2001~2999
XI. 妊娠, 分娩, 産褥	630~676	オニ内科循環器病名コード	3001~3109

4) 病名の出力

患者のサマリー: ID番号にて↑MSTより 図7

病名検索: ICDコードにて↑DISより 図6

病名統計: ICDコードにて↑MSTより 図8

おわりに、システムの設計にあたり御指導いただきました 羽曳野病院 大節陽一先生に
感謝いたします

図7 患者のサマリー

カンシヤ / キロク

ショウワ 54 ネン 8 カツ 2 ニチ

ID NO.=311-38001 ナマエ=ウ ■■■■■ セイ=M セイネンカッヒ°=M.38.3.11
 カ=34 ネンレイ(ニューインシ)=73 ニュウインネンカッヒ°=53.8.22 タイインネンカッヒ°=53.9.22
 トクター=ミヤモト テンキ=2 カルテNO.=53-388 シュウジョ=320 ホウケン=2 ヲケツ=2
 マヤク= シンフセツ=1
 ヒョウメイ

1. 151.9-1 Magen Krebs (H0 P0 m n(-))

オワリ

カンシヤ / キロク

ショウワ 54 ネン 8 カツ 2 ニチ

ID NO.=1015-73001 ナマエ=キ ■■■■■ セイ=M セイネンカッヒ°=S.15.10.15
 カ=13 ネンレイ(ニューインシ)=38 ニュウインネンカッヒ°=53.12.22 タイインネンカッヒ°=54.1.31
 トクター=スズキ テンキ=2 カルテNO.=78-354 シュウジョ=301 ホウケン=2 ヲケツ=2
 マヤク=2 シンフセツ=1
 ヒョウメイ

1. 3043-1 ウツツカタ シンキンショウ

コメント:

R and L cardiac catheterization. R Aand L biopsy. Cardioversion.

オワリ

カンシヤ / キロク

ショウワ 54 ネン 8 カツ 2 ニチ

ID NO.=423-84501 ナマエ=タ ■■■■■ セイ=F セイネンカッヒ°=S.26.4.23
 カ=51 ネンレイ(ニューインシ)=27 ニュウインネンカッヒ°=54.2.12 タイインネンカッヒ°=54.2.19
 トクター=ノムラ テンキ=2 カルテNO.=54-81 シュウジョ=320 ホウケン=2 ヲケツ=2
 マスイ=2 PP,MP=2 ニンシン シュウ=40 セイシ(ハヒ-) =1 イチ(ハヒ-) =1 フンヘンホウホウ=1
 ヒョウメイ

1. 658.1-1 Vorzeitiger Blasensprung
2. 642.4-3 Schwangerschaftstoxikosen
3. 650-2 Hinterhauptslase 2 ste
4. 663.1-1 Nabelschnur Umschlingung Neck 2 mal

ハヒ- / キロク

ID NO.= ナマエ= セイ=F セイネンカッヒ°=S.54.2.12 カ=50
 ニュウインネンカッヒ°=54.2.12 タイインネンカッヒ°=54.2.19 テンキ=1 カルテNO.=54-81
 ホウケン=2 ヲケツ=2 タイシュウ=3020

オワリ

図1 退院病歴管理のシステム・フロー

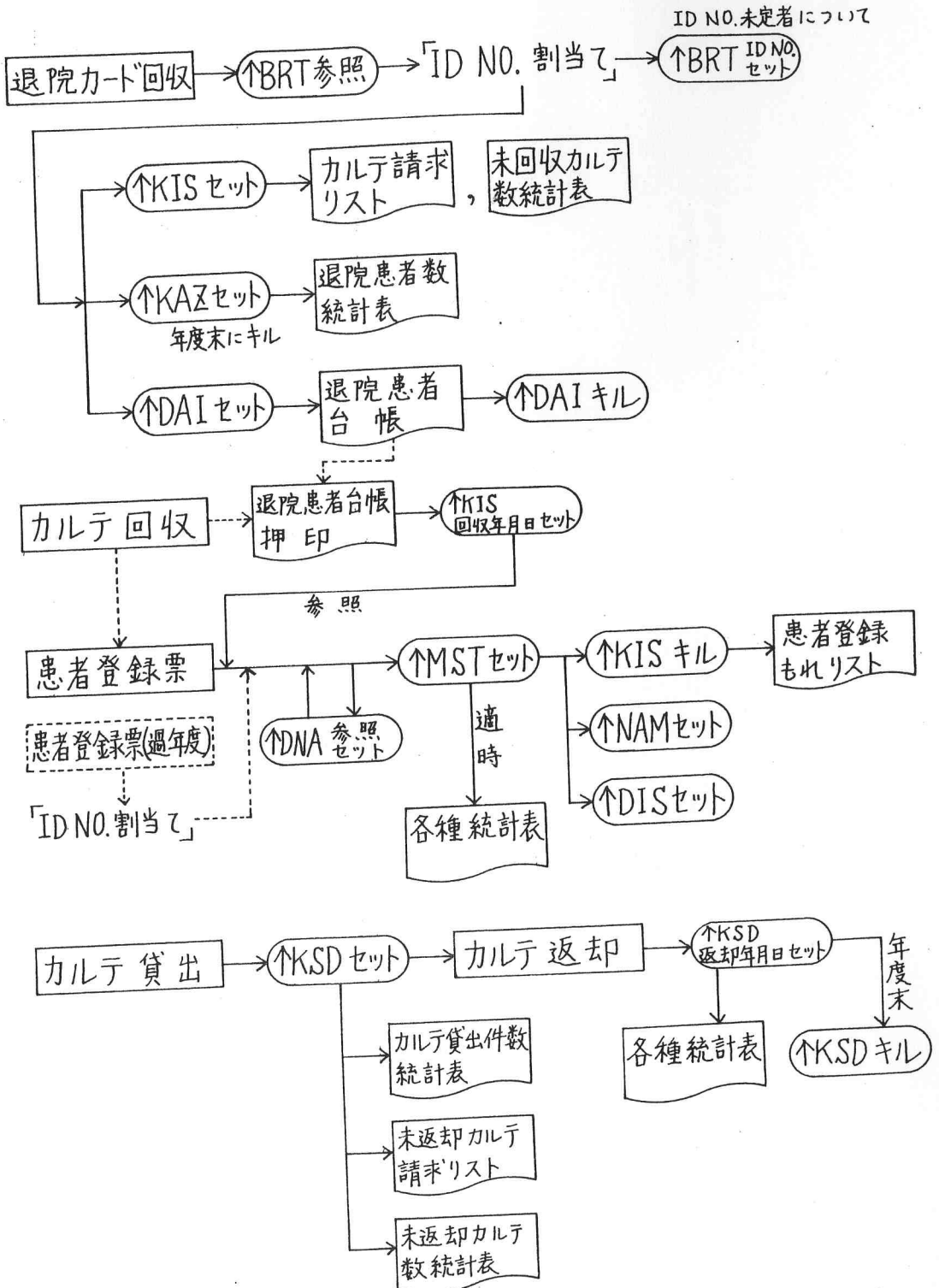


表1 主要ファイルの内容

	↑MST 患者マスター ファイル			↑NAM 氏名検索 ファイル	↑DIS 病名検索 ファイル	↑BRT 生年月日 ファイル	↑KIS カルテ回収 ファイル	↑DAI 台帳 ファイル
	一般の科	産科(母)	産科(ベビー)					
レベル	A 科NO.退院年	科NO.退院年	科NO.退院年	氏	ICDコード	生 月 日	科NO.	科NO.
	B 生 月 日	生 月 日	生 月 日	名	疾患・順位	生年・順位	退院年月	順位番号
	C 生年・順位	生年・順位	生年・順位	順位番号	順位番号		退院日・順位	
	D 退院 月 日	退院 月 日	退院 月 日					
	E		ベビー番号					
項目	C	C	E	C	C	B	C	B
	1. ID NO.	1. ID NO.	1. ID NO.	1. ID NO.	1. NCDコード	1. ID NO.	1. ID NO.	1. ID NO.
	2. 氏 名	2. 氏 名	2. 氏 名	*母のID NO.	2. ID NO.	2. 氏 名	2. 氏 名	2. 氏 名
	3. 性	3. 性	3. 性	2. 氏 名	*母のID NO.	3. 性	3. 性	3. 性
	4. 生年月日	4. 生年月日	4. 生年月日	3. 科 NO.	3. 氏 名	4. 生年月日	4. 生年月日	4. 生年月日
	D	D	5. 科 NO.	4. 退院年月日	4. 科 NO.		5. 科 NO.	5. 科 NO.
	1. 科 NO.	1. 科 NO.	6. 入院年月日	5. カルテNO.	5. 退院年月日		6. 入院時年齢	6. 入院時年齢
	2. 入院時年齢	2. 入院時年齢	7. 退院年月日		6. カルテNO.		7. 入院年月日	7. 入院年月日
	3. 入院年月日	3. 入院年月日	8. 転 帰				8. 退院年月日	8. 退院年月日
	4. 退院年月日	4. 退院年月日	9. カルテNO.				9. 病 棟	9. 病 棟
	5. 主 治 医	5. 主 治 医	10. 剖 検	*産科ベビーのみ	*産科ベビーのみ 氏名不明のベビーは母の氏名, ID NO.を用いる		10. 主 治 医	10. 主 治 医
	6. 転 帰	6. 転 帰	11. 輸 血				11. 転 帰	11. 転 帰
	7. カルテNO.	7. カルテNO.	12. 体 重				12. 入院番号	12. 入院番号
	8. 住所NO.	8. 住所NO.	E. 01				13. 順 位	13. 順 位
	9. 剖 検	9. 剖 検	1. 疾患・順位				14. \$ (受領年月)	14. \$
	10. 輸 血	10. 輸 血	2. ICDコード					
	11. 麻 薬	11. 麻 酔	3. NCDコード					
	12. 心不全	12. 初産経産	4. 病名追加					
	D. 01	13. 妊娠週	E. 02					
	1. 疾患・順位	14. 児 生 死	1. コメント					
	2. ICDコード	15. 児 位 置						
	3. NCDコード	16. 分娩方法						
	4. 病名追加	D. 01						
	D. 02	1. 疾患・順位						
	1. コメント	2. ICDコード						
		3. NCDコード						
		4. 病名追加						
		D. 02						
		1. コメント						

注: ID NO. はカルテ番号である

カルテ NO. とは各科毎に任意につけられた番号

図2-1 患者登録票(一般)

-141-

5.546

311-3800/

患者登録票

科	病歴番号	性別	氏名
34	53 388	男①女2	マツカ オカ
生年	月	日	年
53	8	22	38
入院	日	月	年
53	8	22	38
転科	理由	転科	理由
1. 全治	2. 軽快	3. 不変	4. 増悪
5. 死亡	6. 検査終了	7. 外来継続	8. 転科
9. 転出	10. 希望	11. 中止	12. その他
1. ミヤコ	2.	3.	4.
1. 15.9.1	M. K. (H. P. m. n. G.)		
2.			
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			

*印は転科しなすもの

処置及び 備考	1	
	2	
	3	
	4	
	5	

COMMENTS

1	
2	
3	
4	
5	

図2-2 患者登録票 (才三内科)

1-1015-73001															患者登録票														
科		病歴番号				性別		氏名																					
13		78-354				男①女2		キ																					
生年		M		年		月		住所		No																			
⑤		15		10		15		3,8		3,0,1 オオサカシ イクノクイクノニシ																			
入院		年		月		日		在院数		剖検		有1無②		輸血		有1無②		麻薬		有1無②		心不全		有①無2					
53		12		22		54		13		1		3		1		1		1		1		1		1					
転		1. 全治		② 軽快		3. 不変		4. 増悪		5. 死亡		6. 検査終了		7. 片米継続		8. 転科													
帰		9. 転医		10. 希望		11. 中止		12. その他																					
主治医		1		スズキ		2		3		4		5																	

循環器疾患名

〔Ⅰ〕先天性疾患 1. 心房中隔欠損症 2. 肺動脈流異常 3. 心内膜欠損症 4. 心室中隔欠損症 5. 動脈管閉鎖症 6. Eisenmenger 症候群 7. 肺動脈弁狭窄症 8. フローニ四位症 9. 大動脈弁狭窄症 10. 心臓の位置異常 11. Ebstein 奇形 12. 三尖弁閉鎖症 13. 大血管転位症 14. 総動脈幹痿遺症 15. 大動脈幹動脈中隔欠損症 16. 大動脈縮窄症 17. 大動脈弓異常 18. Valsalva 洞動脈瘤 19. 冠動脈奇形 20. 単心室 21. 左心形成不全症候群 22. 先天性僧帽弁疾患 23. その他の先天性心疾患	33. 肺動脈弁閉鎖不全症 連合弁疾患 34. 僧帽弁＋大動脈弁疾患 35. その他の連合弁疾患 〔Ⅲ〕虚血性心疾患 36. 狭心症 37. 中間型(症候群) 38. 急性心筋梗塞 39. 陈旧性心筋梗塞 40. 慢性虚血性心疾患(ASHD) 〔Ⅳ〕心筋疾患 41. 非閉塞性肥大型心筋症 42. 閉塞性肥大型心筋症(HOCM) 43. うっ血性心筋症 44. 絞扼性心筋症 45. 心臓腫瘍 〔Ⅴ〕心膜疾患 46. 急性心膜炎 47. 慢性心膜炎(含収縮性心膜炎) 48. 心膜血腫(含心タンポナーデ) 49. その他の心膜疾患 〔Ⅵ〕細菌性心膜炎 50. 急性型(ABE) 51. 亜急性型(SBE) 〔Ⅶ〕大動脈疾患 52. 解離性大動脈瘤 53. 動脈硬化性大動脈瘤	54. 梅毒性大動脈瘤 55. 大動脈閉塞 56. 大動脈炎候群 57. その他の大動脈炎 〔Ⅷ〕不整脈 刺激生成異常 58. 洞調律異常 59. 補充収縮・補充調律 60. 補収縮 61. 房室解離 62. 心房性不整脈 63. 心房性不整脈 64. 房室性不整脈 65. 発作性不整脈 興奮伝導障害 66. 洞房ブロック 67. 洞室伝導遅延 68. 洞室不全症候群(SSS) 69. 1度房室ブロック 70. 2度 71. 3度 72. 右胸ブロック 73. 左胸ブロック 74. 二枝ブロック 75. 三枝ブロック 76. その他の興奮伝導障害 早期興奮症候群 77. WPW 症候群 78. その他の早期興奮症候群 〔Ⅸ〕肺性心 79. 急性肺性心 80. 慢性 81. 尿毒性肺高血圧症	(X) 高血圧症 82. 本態性高血圧症 83. 腎血管性高血圧症 84. 腎实质性 85. その他の腎性 86. 内分泌性 87. 原発性アルドステロン症 88. 褐色細胞腫 89. クッシング症候群 90. その他の内分泌性高血圧症 91. その他の高血圧症 92. 高血圧性心疾患 (XI) 末梢動脈および静脈疾患 93. 急性動脈閉塞症 94. 慢性動脈閉塞性疾患 95. 血栓性静脈炎 96. 静脈血栓・血栓性静脈炎 97. 静脈炎後症候群 98. 下肢静脈瘤 (XII) その他の循環器系疾患 99. 粘着性心膜炎 100. 甲状腺機能亢進性心疾患 101. その他の内分泌異常に伴う 102. リウマチ熱 103. その他の膠原病に伴う心疾患 104. Marfan 症候群 105. 肺気腫 106. 貧血性 107. 低血圧症 108. 神経循環障害症 109. その他分類不能の心疾患
--	--	--	--

その他の病名

その他の別名

Comments (200字以内のカナ文字, 英文字, 数字)

Comments (200字以内のカナ文字, 英文字, 数字)

R. and L. cardiac catheterization. R. and L. Biopsy.
Cardioversion.

NO	51	54-81	性別	♀	住所	320	生年	⑤	26	4	23	日	令	27	②	mp	
入部	54年2月12日	退部	54年2月19日	主理	147	W	1. 生屋 2. 死屋 3. 未分岐 4. 女生後死元(少)	明)	① 頭便	5. 小地							
部員	1. 有 ② 無	部員	1. 有 ② 無	分岐	40	5. 流産 6. ♀の地			2. 横位								
分娩	① 自然 (2. 誘発) 3. 吸引 4. 鉗子 5. 骨盤 6. 帝王切	分娩	① 軽快 8. 軽科 () 5. 死元	分娩	① 軽快 8. 軽科 () 5. 死元				3. 股位								
分娩	(7. 全切) 8. ♀の地	分娩	① 1. 有 2. 無 3. 有 4. 有 5. 有 6. 有 7. 有 8. 有 9. 有 10. 有 11. 有 12. 有 13. 有 14. 有 15. 有 16. 有 17. 有 18. 有 19. 有 20. 有 21. 有 22. 有 23. 有 24. 有 25. 有 26. 有 27. 有 28. 有 29. 有 30. 有 31. 有 32. 有 33. 有 34. 有 35. 有 36. 有 37. 有 38. 有 39. 有 40. 有 41. 有 42. 有 43. 有 44. 有 45. 有 46. 有 47. 有 48. 有 49. 有 50. 有 51. 有 52. 有 53. 有 54. 有 55. 有 56. 有 57. 有 58. 有 59. 有 60. 有 61. 有 62. 有 63. 有 64. 有 65. 有 66. 有 67. 有 68. 有 69. 有 70. 有 71. 有 72. 有 73. 有 74. 有 75. 有 76. 有 77. 有 78. 有 79. 有 80. 有 81. 有 82. 有 83. 有 84. 有 85. 有 86. 有 87. 有 88. 有 89. 有 90. 有 91. 有 92. 有 93. 有 94. 有 95. 有 96. 有 97. 有 98. 有 99. 有 100. 有 101. 有 102. 有 103. 有 104. 有 105. 有 106. 有 107. 有 108. 有 109. 有 110. 有 111. 有 112. 有 113. 有 114. 有 115. 有 116. 有 117. 有 118. 有 119. 有 120. 有 121. 有 122. 有 123. 有 124. 有 125. 有 126. 有 127. 有 128. 有 129. 有 130. 有 131. 有 132. 有 133. 有 134. 有 135. 有 136. 有 137. 有 138. 有 139. 有 140. 有 141. 有 142. 有 143. 有 144. 有 145. 有 146. 有 147. 有 148. 有 149. 有 150. 有 151. 有 152. 有 153. 有 154. 有 155. 有 156. 有 157. 有 158. 有 159. 有 160. 有 161. 有 162. 有 163. 有 164. 有 165. 有 166. 有 167. 有 168. 有 169. 有 170. 有 171. 有 172. 有 173. 有 174. 有 175. 有 176. 有 177. 有 178. 有 179. 有 180. 有 181. 有 182. 有 183. 有 184. 有 185. 有 186. 有 187. 有 188. 有 189. 有 190. 有 191. 有 192. 有 193. 有 194. 有 195. 有 196. 有 197. 有 198. 有 199. 有 200. 有 201. 有 202. 有 203. 有 204. 有 205. 有 206. 有 207. 有 208. 有 209. 有 210. 有 211. 有 212. 有 213. 有 214. 有 215. 有 216. 有 217. 有 218. 有 219. 有 220. 有 221. 有 222. 有 223. 有 224. 有 225. 有 226. 有 227. 有 228. 有 229. 有 230. 有 231. 有 232. 有 233. 有 234. 有 235. 有 236. 有 237. 有 238. 有 239. 有 240. 有 241. 有 242. 有 243. 有 244. 有 245. 有 246. 有 247. 有 248. 有 249. 有 250. 有 251. 有 252. 有 253. 有 254. 有 255. 有 256. 有 257. 有 258. 有 259. 有 260. 有 261. 有 262. 有 263. 有 264. 有 265. 有 266. 有 267. 有 268. 有 269. 有 270. 有 271. 有 272. 有 273. 有 274. 有 275. 有 276. 有 277. 有 278. 有 279. 有 280. 有 281. 有 282. 有 283. 有 284. 有 285. 有 286. 有 287. 有 288. 有 289. 有 290. 有 291. 有 292. 有 293. 有 294. 有 295. 有 296. 有 297. 有 298. 有 299. 有 300. 有 301. 有 302. 有 303. 有 304. 有 305. 有 306. 有 307. 有 308. 有 309. 有 310. 有 311. 有 312. 有 313. 有 314. 有 315. 有 316. 有 317. 有 318. 有 319. 有 320. 有 321. 有 322. 有 323. 有 324. 有 325. 有 326. 有 327. 有 328. 有 329. 有 330. 有 331. 有 332. 有 333. 有 334. 有 335. 有 336. 有 337. 有 338. 有 339. 有 340. 有 341. 有 342. 有 343. 有 344. 有 345. 有 346. 有 347. 有 348. 有 349. 有 350. 有 351. 有 352. 有 353. 有 354. 有 355. 有 356. 有 357. 有 358. 有 359. 有 360. 有 361. 有 362. 有 363. 有 364. 有 365. 有 366. 有 367. 有 368. 有 369. 有 370. 有 371. 有 372. 有 373. 有 374. 有 375. 有 376. 有 377. 有 378. 有 379. 有 380. 有 381. 有 382. 有 383. 有 384. 有 385. 有 386. 有 387. 有 388. 有 389. 有 390. 有 391. 有 392. 有 393. 有 394. 有 395. 有 396. 有 397. 有 398. 有 399. 有 400. 有 401. 有 402. 有 403. 有 404. 有 405. 有 406. 有 407. 有 408. 有 409. 有 410. 有 411. 有 412. 有 413. 有 414. 有 415. 有 416. 有 417. 有 418. 有 419. 有 420. 有 421. 有 422. 有 423. 有 424. 有 425. 有 426. 有 427. 有 428. 有 429. 有 430. 有 431. 有 432. 有 433. 有 434. 有 435. 有 436. 有 437. 有 438. 有 439. 有 440. 有 441. 有 442. 有 443. 有 444. 有 445. 有 446. 有 447. 有 448. 有 449. 有 450. 有 451. 有 452. 有 453. 有 454. 有 455. 有 456. 有 457. 有 458. 有 459. 有 460. 有 461. 有 462. 有 463. 有 464. 有 465. 有 466. 有 467. 有 468. 有 469. 有 470. 有 471. 有 472. 有 473. 有 474. 有 475. 有 476. 有 477. 有 478. 有 479. 有 480. 有 481. 有 482. 有 483. 有 484. 有 485. 有 486. 有 487. 有 488. 有 489. 有 490. 有 491. 有 492. 有 493. 有 494. 有 495. 有 496. 有 497. 有 498. 有 499. 有 500. 有 501. 有 502.														

分娩 100	妊娠に終わったもの(630-639)	59.5	1	VAP	65.9.5	2	APP
630	1	Blasenmole	65.9.8	1	VP		NP
632	1	Retentio abortion	65.4	2	Uterus bicornis unicollis		
632	2	Protrahiert Abort	65.4.1	1	Myoma uteri		
632	3	Verhaltung Abort	65.4.5	1	Zervikal insufficiency		
633	1	Extrauterine Schwangerschaft	65.4.5	2	" incompetency		
633.1	1	Tubal abortion	65.4.7	1	Septem vaginae		
633.8	1	Zervikal Schwangerschafts	65.4.8	1	Damm Paralysis		
634	1	Spontan Abort	65.6.1	1	Rh(-)		
634	2	Vollkommener "	65.6.2	1	Hyperbilirubinaemia neonatorum		
634	3	Unvollkommener "	65.6.3	1	Fetaldistress		
主として妊娠に関連した合併症(640-648)		65.6.3	2	Asphixia			
640	1	Drohender Abort	65.6.4	1	Intrauterin Sterbling		
641.1	1	Blutung(分娩前)	65.6.5	1	SFD(small for date)		
641	1	Plazents praevia	65.6.6	1	LFD(large for date)		
641.1	1	" " (出血)	65.6.6	2	Riesen Kind		
641.2	1	Vorzeitige Ablösung	65.7	1	Hydroamnion		
642.1	1	Nephritis(高血圧ともなる)	65.8	1	Ligoamnion		
642.1	2	Nephrose(高血圧ともなる)	65.8.1	1	Vorzeitiger Blasensprung		
642.4	1	Schwangerschaftsniere	65.8.1	2	Frühzeitiger Blasensprung		
642.4	2	Schwangerschaftshypertonie	65.8.1	3	Verspäteter Blasensprung		
642.4	3	Schwangerschaftstoxikosen	65.8.1	4	Hoher "		
642.6	1	Eklampsie	分娩の経過に主として発生する合併症(660-669)				
643	1	Perniciös Erbrechen	66.1	1	Wehenschwäche		
644	1	Fehlgeburt	66.1.2	2	Insufficiency of uterine contraction		
644	2	Frühgeburt	4	66.3.1	1	Nabelschnur Umschlingung Neck	
644	3	Drohender Frühgeburt	66.4	1	Dammriss		
646.2	1	Nephritis(高血圧のもの)	66.4	2	Scheidenriss		
646.2	2	Nephrose(高血圧のもの)	66.4.5	1	Haematoma(会陰)		
646.3	1	Habituell Abort	66.5	1	Uterusruptur(分娩前)		
646.6	1	Pyelonephritis	66.5.1	1	" (分娩中, 分娩後)		
646.6	2	Pyelitis	66.5.3	1	Uterusriss		
646.6	3	Cystitis	66.5.3	2	Zervixriss(Zervikalriss)		
646.7	1	Hepatitis	66.5.7	1	Haematoma(骨盤内)		
648	1	Diabetes mellitus	666	1	Retentio Plazenta		
648.1	1	Hyperthyreoidismus	666	2	Placenta incarcerata		分娩後
648.2	1	Anämie	666	3	Placenta accreta		出血ともなる
648.2	2	Megaloblastic Anämie	666	4	Placenta increta		
648.6	1	Herz Krankheit(Arhythmie)	666	5	Placenta Percreta		
648.6	2	Aortic Insuffizienz	666.1	1	Atomisch Blutung		分娩後
正常分娩, および妊娠・分娩における治療のその他の適応症(650-659)		666.1	2	Zervical Blutung			
650	1	Kopflage	667	2	Placenta incarcerata		
650	2	Hinterhauptslage IIste	667	3	Placenta accreta		
651	1	Zwillinge(1ein, 2.Zwei)	667	4	Placenta increta		
652	1	Beckenendlage	667	5	Placenta percreta		
652	2	Steisslage	667	1	Retentio Plazenta		
652	3	Fusslage	667.1	1	Retention Echoute		
652	4	Knielage	669.1	1	Verblutung Schock		
652	5	Querlage	669.7	1	Kaiser Schnitt (Nachr. 659.2-1)		
652	6	Schräglage	産じょく<溝>の合併症(670-676)				
652	7	Schulterlage	672	1	Wochenbett Fieber		
652	8	Deflexionslage	670	1	Endometritis(産褥性)		
652	9	Vorderhauptslage	673.8	1	Anämie(Wochenbett)		
652	10	Stirnlage					
652	11	Gesichtslage					
653.1	1	Verengtes Becken					
653.4	1	Cepharo pelvic disproportion					
654	1	Uterus bicornis bicollis					

[illegible]

図4 患者マスタファイルのグローバル

[illegible]

図5 基本病名ファイルのグローバル

~DNA(150,9,1) = Esophageal Krebs
 ~DNA(151,1) = Cardia Krebs
 ~DNA(151,9,1) = Magen Krebs
 ~DNA(153,3,1) = Sigmoïd Colon Cancer
 ~DNA(154,1,1) = Rectum Krebs
 ~DNA(3043,1) = 胃がん 胃 胃がん
 ~DNA(3053,1) = カリフォルニア州 胃がん
 ~DNA(3058,1) = ノースカロライナ州 胃がん
 ~DNA(3065,1) = 胃がん 胃がん

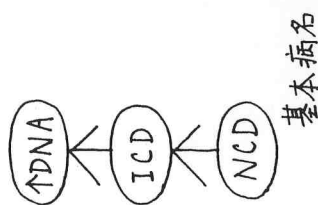


圖 6 病名檢索

ICD コード	ICD	NCD	ID NO.	ナリ	カ	タイイ	カ	ナリNO.
1	151.9	1	905-68002	757	34	54.5.10	34	54-157
1	151.9	1	730-41501	757	34	54.3.3	34	54-34
1	151.9	1	501-63501	757	34	54.3.17	34	54-60

第6回日本MUG学術大会 昭和54年9月14日(金)

D. ハネル(3) MUMPSの医療への応用 司会 大阪医大 山本和子

D-6. 病院管理システム

東京都三鷹市・長谷川病院

医事課
管理室

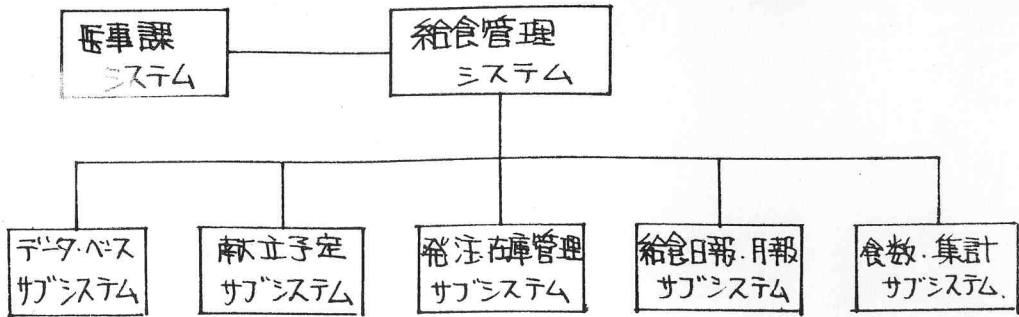
佐藤 弘美
川島 英樹
山浦 寿子

長谷川病院に於ける PDP11/10を中心とした EDPS化は昭和51年8月に着手して以来、医療事務管理部門でのEDPS化を中心に3年の歳月を経、小規模ながら一応の満足すべき成果を得ている。

今回、更に日用品管理、給食管理、臨床検査部及び病床管理等のEDPS化に伴い、現行システムの最終段階を迎え、又これらのEDPS化に対処する為のハード面でのシステム全体の拡大も検討中である。尚、当院の基本となっている情報システムの概要及び新システムについては別紙の通りである。

目的 当院は精神科を主とした病院である関係上、入院患者の購入
及び日用品に対する管理を必要とする部分があり、その為に在庫管理
概略面での事務処理が、会計部門との継がりと併わせて煩雑化している。日用品管理のEDPS化は此の点での簡素化を目的としており、更に物のフローの明確化を言ったものである。

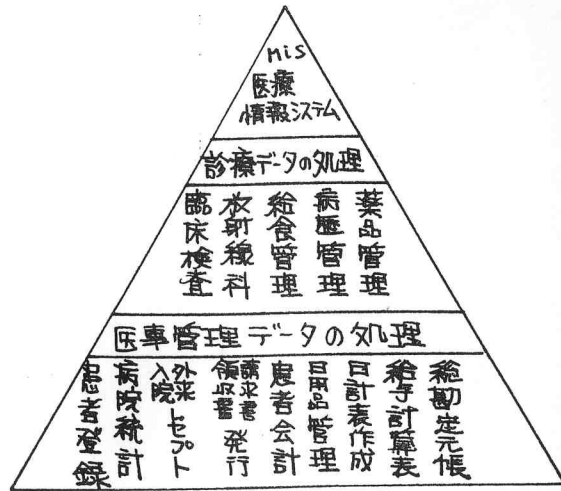
給食管理システムは栄養士が事務計算に費す少くない時間を短縮する目的をもって計画されたものであり、現在開発中である。此のシステムは3つのデータベース(食品材料データベース、献立データベース、食費課用患者データベース)を基本に次の様なサブシステムによって構成されている。



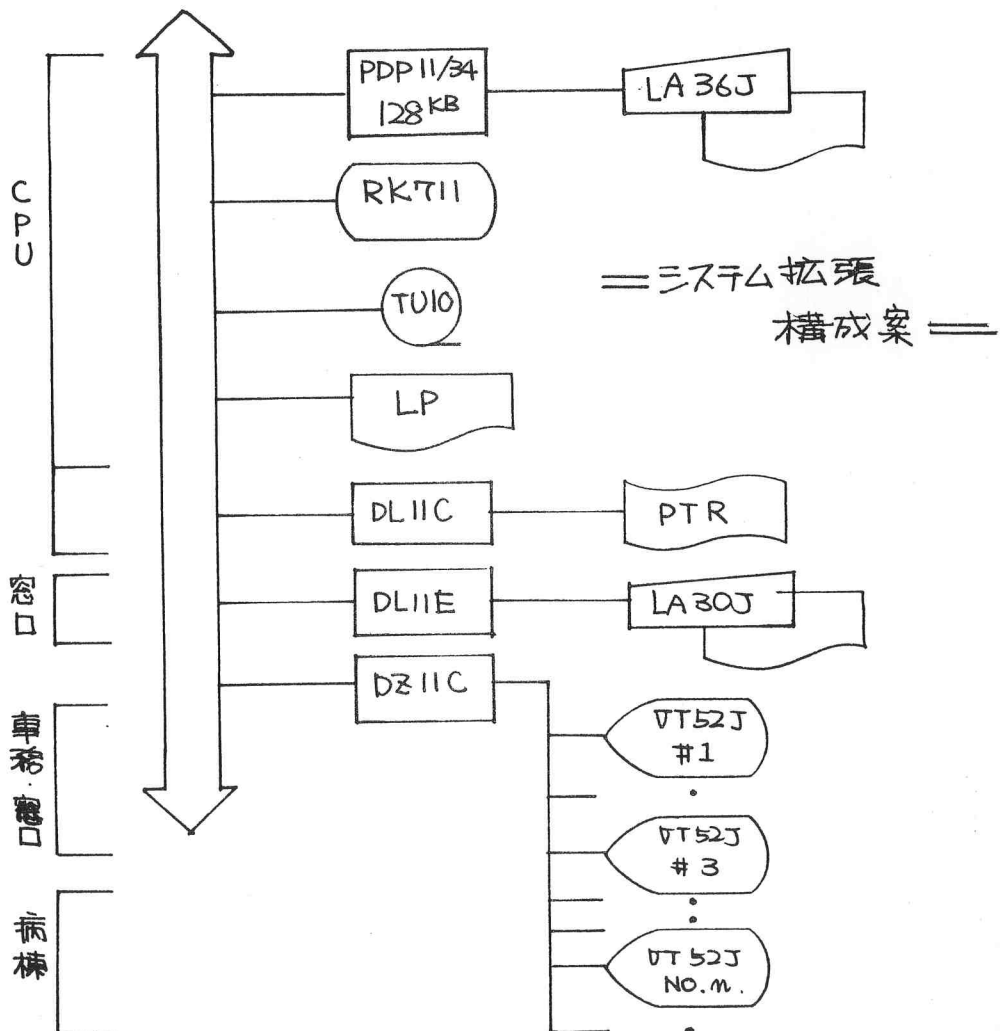
臨床検査部門に於けるEDPS化は最終的に自動分析装置とのオンライン化を計画しているが、今回は臨床検査技師のワグシート作成や転記等に費す時間の短縮化を計ったものである。外注が多いという内部事情により、外注先とのCPU-CPU間の媒介を紙テープに決定する迄に手間取り、現在着手したばかりである。

病床管理システムは、システムの拡大による端末の病棟への導入を前提に、看護課及び医事部門での病床状態の把握を容易にする目的で計画されたものである。病棟への端末導入という新しい段階は、現行システムではほとんど行われていない端末管理の問題を生じ、プログラムの保護機能と共にデータの機密保持の点で更に検討しなければならない問題である。これらの機能には、MUMPSシステムからの保護とアプリケーションからの保護との2種類が考えられると思うが、此の2種類を如何に効率良く活用するか、又他に何様な方法があるのか御指導いただければ幸である。

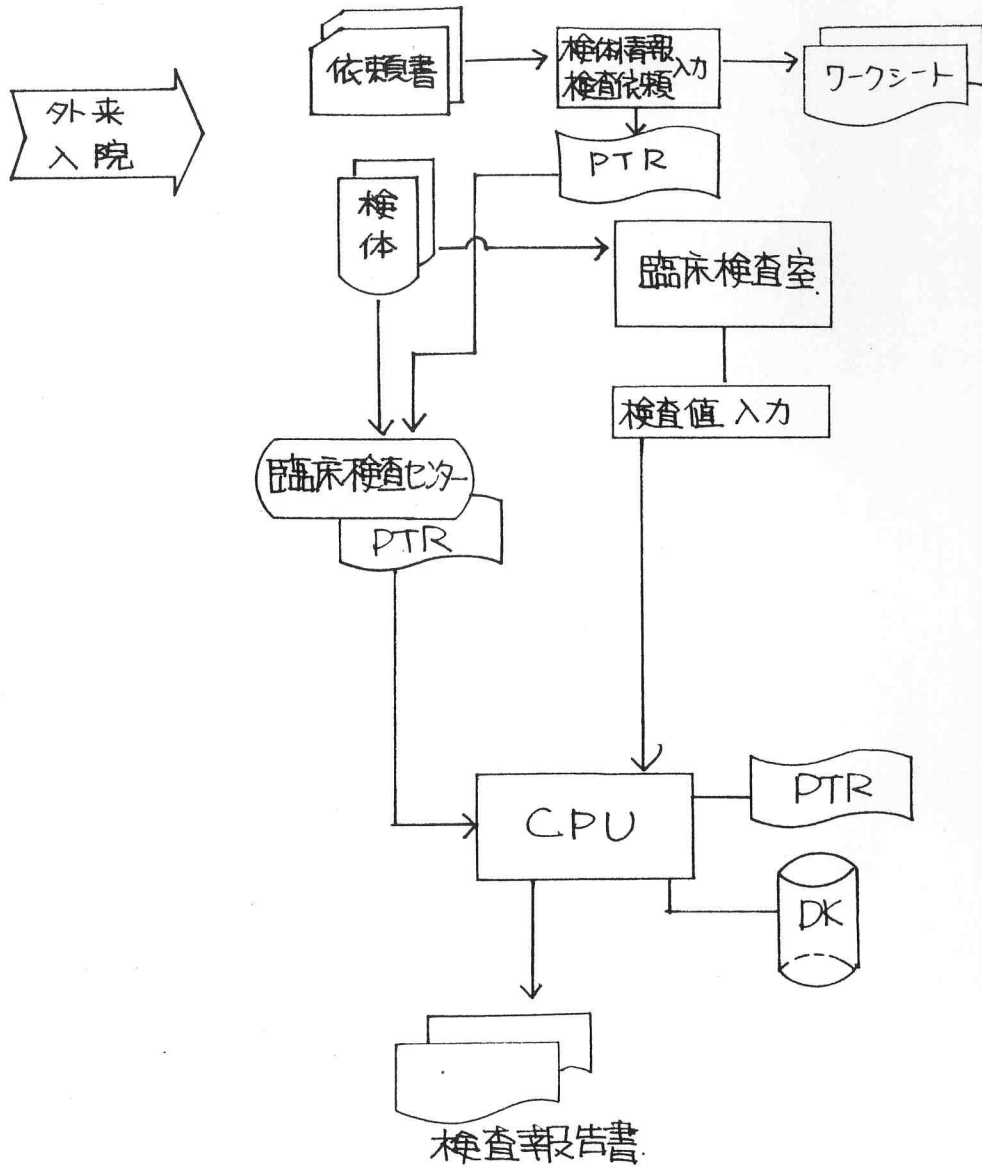
尚、未筆ながら、給食管理システム、臨床検査部門に関して春日外科病院の佐藤氏、大阪府立成人病センターの厨房部門の方々及び大阪府立平野野病院の大橋氏から様々の御指導をいただいた事とこの紙面をかりて感謝する次第である。



== 病院情報システム ==



臨床検査システム



第6回日本MUG学術大会 昭和54年9月14日(金)

D. ハネル(3) MUMPSの医療への応用 司会 大阪医大 山本和子

D-7. 検査結果 問合わせシステムの試み

大阪府立成人病センター 情報企画室 北川 護 上田 清治
大阪府立成人病センター 臨床検査部 森井卓郎

1. 開発の目的

病棟において、医師が入院患者の検査結果を一刻も早く知りたいというニーズは、非常に強い。しかし、入院患者が多く検査件数も多い病院においては、迅速かつ正確に結果が判明したものから逐次報告するには、困難な問題が多い。

成人病センターでは、検体検査の処理にPDP11/40を用いた「臨床検査システム」を採用しており、検査の受付から報告書作成までの一貫した処理を行っている。このシステムで作成される報告書は、カルテへの貼布の手間やコンピュータへの負荷などを軽減するために、多くの検査項目が4枚に集約されている。したがって、結果が判明するまでの日数が最も長い検査の結果が入力されるまで、一括出力が待たされることになっている。このため、検査指示を出して報告書を受取るまでの間、すでに結果が判明しているはずの検査について、病棟から検査部へ電話による問合わせが頻ぱんに行われていた。

しかしこれは、双方にとって繁雑なばかりでなく、聞きまちがいや転記ミスなどによる重大な誤りの原因ともなり得る。

これらの問題を解決し、病棟側と検査側双方の要求を満たすために、我々は病棟の端末装置を使って検査結果が問合わせできるシステムを開発した。

2. 概要

臨床検査システムにおいて、結果が入力された入院患者のデータを無条件に、毎日磁気テープに出力する。このテープを既存の「ドック・病棟システム」へ入力し、患者のデータファイルを作成する。医師・看護婦は、病棟に設置されているCRTディスプレイ装置、出力端末装置を使って自由に結果の問合わせを行う。

この方法によって当初の問題点が解決したばかりでなく、①全入院患者の検査データが病棟で検索・参照できるようになった、②そのデータは必要に応じてハードコピーがとられ仮報告書として使用できる、③患者の入院中の全検査結果を時系列的に経過一覽表(フローシート)として出力できるなど、診療にもたらされるメリットは大きいものがある。

3. 開発の方法

〔設計方針〕

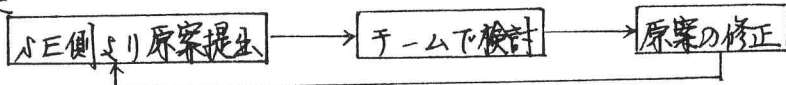
このシステムの設計には、そこで取扱うデータの性格と「臨床検査システム」を利用するのが最も好ましいと考えられる。しかしこのためには、新たに検査システムのCPUより各病棟諸所へ端末用信号線の配線工事を行う必要があるほか、種々の問題があった。一方、各諸所にはすでに「ドック・病棟システム」（病院内の全病棟〈現在は2病棟〉を対象に、各種の病棟サポート業務を行う別のサブシステム）より信号線が設置されている。したがって、検査結果問合わせシステムを2つの機能に分け、各々を検査システムと病棟システムで分担することによって、できるだけ少ない費用と時間で実現化をはかった。すなわち、検査データの収集、加工、品質管理は従来どおり検査システム側で行い、データ表示、問合わせのためのデータ分類、統合は病棟システム側で行うことにした。また、両システム間のデータリンクは、当初は磁気テープで行うことにした。

このことにより、各病棟では端末装置を設置する以外は何らハードウェアの追加や変更を行うことなく、新しいシステムを利用することができた。また、本来は検査システムの延長であるべきこのシステムを、2台の独立したコンピュータを用いて実現したことは、検査システムに新たな負荷がかかるのを防ぐと同時に、病棟側と検査部側の運用時間帯が異なることによって生じる問題を吸収するのに役立っている。

〔開発体制〕

本システムは、病棟において医師や看護婦が直接使用するシステムであり、また検査部にとっても、迅速に見易いデータを提供することによるメリットは大きい。このため、システム設計当初から、医師5名、看護婦3名、検査技師1名、BEM4名の使用者側に入ったチームを編成し、週一度の割合で定例ミーティングを行った。

基本的に



のサイクルで作業を進め、仕様書も、医師・看護婦・検査技師向けのオペレーションや機能の説明を中心にしたもの、オペレータ・システムマネージャ向けの詳細なレベルのものとの2種類を作成することにより、チームで検討する際に不必要な情報で作業が滞ることのないように配慮した。また、オーパスのシステム完成後も実際に使用するにあたって約2ヶ月間のブラッシュアップのための期間を設け、細かい仕様変更を行った。

〔システム上の工夫〕

・ソフトウェア；ひとつのシステムが2つのコンピュータシステムにまたがっていることにより、各種仕様変更の際の両システム間の同期合わせをより簡単にするために、病棟システムでは、検査項目や問合わせのための項目の分類などのアウトフレームのみを一致させ、検査結果自作の単位、桁数など細かいフォーマットについては関与しないようにした。このため、結果の内容については検査システムで全てを制御することができ、このレベルでの変更は、病棟システムにとっては仕様変更と見なすに済んだ。

この問合わせ業務に用いるCRTは、使い易くするためにタイドモードで動作させ、また各画面ごとに制御情報（Y or N）を入力する際に操作の手間を省くためにデフォルトの機能を持たせた。これにより、全くの初心者でも比較的容易に問合わせを行うことが

可能となっている。

・ハードウェア；各病棟に設置されているCRTおよびプリンターの電源は、運用の統一化・簡略化をはかるために、全てCPU側よりリモートコントロールされている。現在使用しているハード構成を図1に示す。

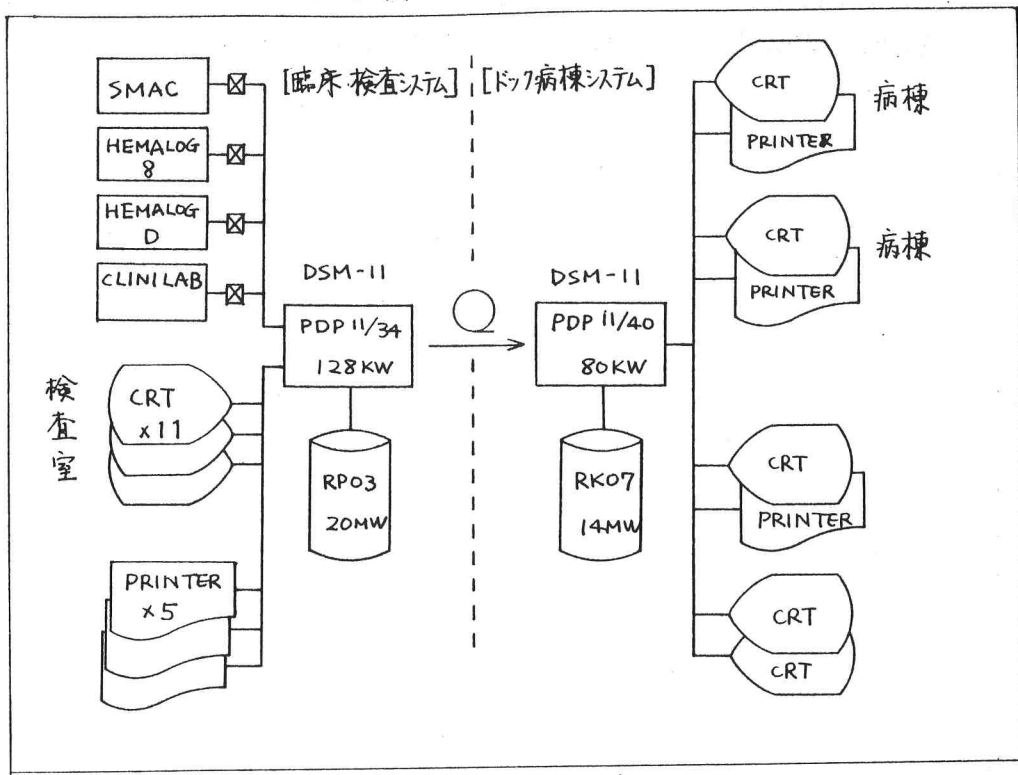


図1. 検査結果問合せシステムの構成

4. 処理内容

本システムにおいては、すべての入院患者（500床）に関する検体検査結果のほとんどを、CRT画面より照会することができる。

〔対象検査項目〕

本システムで現在扱っている項目は、図2～5に示す4種類の報告書に記載されている項目すべてである。これらの報告書は、本来検査指示1回分のデータをコンピュータから出力してカルテに貼布するためのものである。

〔処理件数〕

このシステムでは、150人/日 前後の入院患者の検査結果データすなわち、指示件数にすると2600件/日を検査システムより受け取っている。但し、この件数はPMA C（20項目同時測定）やHEMALOG-8（8項目同時測定）などは1件として数えた場合のものである。

臨床検査報告書(A)

No.

カルテNo.

氏名

外来 ☐ 入院 ☐

受付月日

報告月日

医師名

医師所属科名

@ 別紙報告 ※ 異常値

バック検査

血液化学 I バック検査名 ()

血清蛋白

自動分析 ()

SGOT		≤40 U/L
SGPT		≤40 U/L
ALK-P		25 ~ 120 U/L
γ-GTP		≤60 U/L
LDH		100 ~ 250 U/L
CPK		≤160 U/L
T-Protein		6.5 ~ 8.2 %
Albumin		3.5 ~ 5.0 %
Cholesterol		130 ~ 250 mg/dl
Triglyceride		30 ~ 150 mg/dl

T-Bilirubin		0.2 ~ 1.2 mg/dl
Glucose		65 ~ 110 mg/dl
BUN		8 ~ 20 mg/dl
Creatinine		0.5 ~ 1.3 mg/dl
Uric Acid		2.0 ~ 8.0 mg/dl
Na		134 ~ 147 mEq/L
K		3.4 ~ 5.0 mEq/L
Cl		96 ~ 110 mEq/L
Ca		8.6 ~ 10.8 mg/dl
IP		2.4 ~ 4.2 mg/dl

T protein		6.8 ~ 8.2 %
Serum Electrophoresis		
Alb		45.2 ~ 62.0 %
α ₁ -gl		3.3 ~ 5.7 %
α ₂ -gl		6.4 ~ 10.8 %
β-gl		10.1 ~ 15.7 %
γ-gl		12.8 ~ 25.2 %

A/G

Ca/P

血液化学 II

Amylase		
Serum		60 ~ 150 U/L
Urine F		≤800 U/L
Urine 2H		≤190 U/L
1W Urine 2H		≤190 U/L
1		
2		
3		
4		
5		
6		
7		

LAP		20 ~ 50 U/L
Aldolase		0.5 ~ 3.1 U/L
Ch-esterase		2.2 ~ 4.9 × 10 ³ U/L
MAO		24 ~ 48 U

Acid Phosphatase

Total		2.4 ~ 4.6 U/L
Prostatic		0 ~ 30 %

血液化学 III

Glucose		
空腹時		<110 mg/dl
食後2H		
日内変動		
1		
2		
3		
4		
5		
GTT ()		
Blood (mg/dl) : Urine (mg/dl)		
空腹時		
30'		
1H		
2H		
3H		

血清

ASLO

定性		
定量		≤166 T.U
CRP		
RA		
RAHA		≤40 dils
COLD HA		≤64 dils
Paul-B		≤128 dils

HBs-Antigen		dils
α-Fetoprotein		
Syphilis - Serum		
定/RPRCT		
性/TPHA		
定/RPRCT		
量/TPHA		

Coombs

Direct		
Indirect		

Thyroid		<10 ² dils
Micrösosome		<10 ² dils
CH 50		38 ~ 52 U/ml
Sugar w		

血液化学 IV

ZTT		4 ~ 12 Unit
TTT		0 ~ 3 Unit
ICG-15分		<10 %
β-Lipo		250 ~ 500 mg/dl
Bilirubin 分画		
Total		0.2 ~ 1.3 mg/dl
Direct		0 ~ 0.2 mg/dl
Indirect		
Fe		50 ~ 150 μg
UIBC		70 ~ 120 μg
		190 ~ 250 μg

Comment

図2. 血液化学・血清検査用報告書

内分泌検査報告書

カルテNo.	氏名	No.
受付月日	報告月日	医師名
① 別紙報告 ※ 異常値		医師所属科名
血 液		尿
Thyroxin 7 ~ 14 $\mu\text{g}/\text{dl}$	血中Cortisol 日内変動 朝 夕	尿量 ml/day
Triosorb 23 ~ 35 %	Rapid ACTH test 前 後	17KS $\frac{7.3}{2} \sim \frac{15}{10} \text{ mg}/\text{day}$
T ₃ 90 ~ 190 ng/dl	デキサメサゾンテスト 	17OHCS $\frac{3}{2} \sim \frac{9}{6} \text{ mg}/\text{day}$
TSH 10 mU/mL		17KGS $\frac{7.4}{2} \sim \frac{12}{11} \text{ mg}/\text{day}$
Renin 活性 (ラシックス負荷) 1 ~ 11 $\text{ng}/\text{ml}/\text{hr}$		Metanephrine $9,900 \sim 1600 \text{ ng}/\text{day}$
Comment		VMA 2 ~ 7 mg/day
		5HIAA 2 ~ 8 mg/day

図 5. 内分泌検査用報告書.

〔データ保存期間〕

これらのデータは入院患者単位に保存期間をシステムで管理されている。すなわち、基本的にはどの患者についても最近の30日間の検査データをディスク内に保持しているが、ある患者について30日間にわたってデータが入力されないとき、すなわち、検査指示の間隔が30日を越えた場合、その患者は退院したものと見なされ、その患者のデータはすべて自動的に消去される。しかし、医師の要求があった場合などのため、特別な指定をすることにより保存しておくことも可能である。

〔検査結果データの入力〕

本システムでは、臨床検査システムで当日あるいは前日に行われた検査結果が編集された磁気テープを介してデータ入力を行う。この処理の時間帯は必然的に業務時間終了間隙となるので、実質的にその日の検査結果を問合わせできるのは、翌朝からになることが多い。

〔検査結果の問合わせの操作〕

病棟前にはCRTディスプレイとシリアルプリンターを設置しており、医師や看護婦は簡単な操作で問合わせができる。問合わせの手順を図6に示す。

次にCRTディスプレイの画面を通じて具体的に説明する。まず、問い合わせを行う者は図7に示す画面にしたがってカルテNo.を入力する。カルテNo.は、テンキーあるいは磁気カード付き診療券を用いて、磁気カード読取り部から入力することができる。

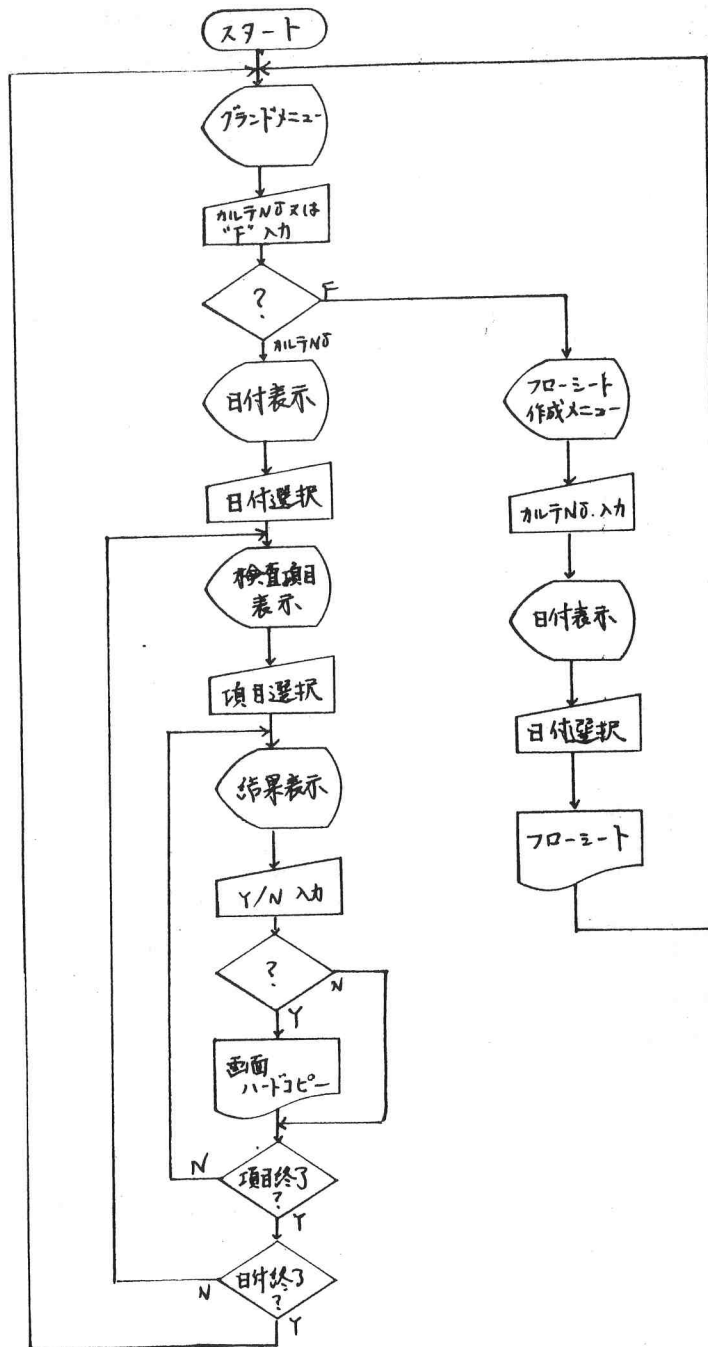


図 6. 検査結果問合せ操作手順

システム ショウカイ システム

カルテNO. (カード) = ■

* システム ショウカイ ノ トキハ カルテNO. ヲ ニュウリョク シテクダサイ
(システム ショウカイ トキハ <CR> ヲ シテクダサイ)

* フォーシート ノ システム ノ トキハ "F" ヲ ニュウリョク シテクダサイ

図7. 磁気カード(診察券)によるカルテNO.入力画面.

カルテNO. がいすれかの方法で入力されると、次にCRT画面には、図8に示すように、その患者に関して検査が実施された日がすべて表示される。したがって、問い合わせを行いたい日を単独または複数選択し、テンキーからその番号を入力する。

システム ショウカイ システム

カルテNO. (テンキー) : 21-■■■■-002
シメイ = ■■■■

** 1) 04/25 2) 04/27 3) 05/01 4) 05/02 5) 05/04 6) 05/07
7) 05/08 8) 05/09 9) 05/10 10) 05/11 11) 05/12 12) 05/14
13) 05/17 14) 05/21 15) 05/22 16) 05/23 17) 05/24 18) 05/28
19) 05/29 20) 05/30 21) 05/31 22) 06/01 23) 06/04 24) 06/05
25) 06/07 26) 06/08 27) 06/11 28) 06/12 29) 06/13 30) 06/14
31) 06/15 32) 06/19 33) 06/20 34) 06/21 35) 06/22 36) 06/25
37) 06/27 38) 06/29 39) 07/02

リポート NO. (1 OR 1.2.3...) : ■

図8. 検査実施日の選択画面.

日を指定し入力すると次に、指定された日に行われた検査が、報告書の単位に分類されるとめて表示される。図9に、検査項目の画面を示す。

21-002-002 08N DR. 719 717 4-JUL-79

719717 : 06/25
 <<719717 717717 717717>>

0) ALL
 719717717 A 719717717 B 719717 717717 719717717717

1) SMAC 7) 719717717717
 8) *719717717 717717
 9) *719717717 717717

5) 719717717717
 6) 719717

719717 717717 717717 : ■

図 9. 検査項目選択画面

ここで、問合わせを行いたい項目を単独又は複数入力を行う。表示されているすべての検査について結果を表示したいときは、0 (ゼロ) を入力する。図9の 0), 9) は、検査指示は出ているが、結果がまだシステムに入力されていない場合で、「*」が表示されている。

次に、項目の指定によって画面に表示される検査結果の例を図10～13に示す。

21-002-002 08N DR. 719 717 4-JUL-79

719717 : 06/25 コメント :

719717 717717
 SERUM ELECTROPHORESIS
 T-PROTEIN : 7.2
 AIB : 45.5
 A1-GL : 4.0
 A2-GL : 10.5
 B-GL : 14.1
 G-GL : 25.9

719717 717717 - V, N (=CR) : ■

図 10. 血清蛋白分画検査結果表示画面。

21-002-002 08N DR. 4-JUL-79
 06/25 BL: 06/25

AUTO() DIFF() BSG
 PLAT : 223 BLAS : BASO : 1.4 1H :
 WBC : 4.7 PROM : EOSI : 2.1 2H :
 RBC : 4.26 MYEL : LYMP : 10.5
 HGB : 13.0 NEUT : 71.8 AT-LY : PLAT :
 HCT : 37.4 STAB : PLAS : WBC :
 MCV : 88 SEGM : LUC : 3.2 RET :
 MCH : 30.6 PROM : HPEX : 1.3 LE :
 MCHC : 34.8 MONO : 10.7 N-RBC :

0-10 CR - Y.N(CR) :

図 11. 血液検査結果表示画面

21-002-002 08N DR. 4-JUL-79
 06/19 UR: 06/19

COULTE ()
 PH : 6.5
 BILIRUBIN : -
 KETON : -
 UROBILI :
 GLUCOSE :
 24H :
 24H :
 24H :

0-10 CR - Y.N(CR) :

図 12. 尿定性検査結果表示画面

21-002-002 08N DR. 4-JUL-79
 06/19 UR: 06/19

SEDIMENT ()
 RBC : 0 WBC : 0 GLITTER : PALE : DARK :
 HYALINE : GRANULAR : SPERMA :
 -EP : CO-GRAN : YEANT :
 -RBC : FATTY :
 -WBC :
 -GR : TRANSITIONAL :
 -FAT : SQUAMOUS :
 RBC : DEGENERATED :
 WBC : ROUND :
 EPITHEL :
 0-10 CR - Y.N(CR) :

図 13. 尿沈渣検査結果表示画面

*** FLOW SHEET ***

カルテNO. : 10-0000-000

シメイ : イセ

09N

29-MAY-79

【ケンサ・コード】*ケンサメイ】

	05/01	05/08	05/15	05/22	/
【3005*SMAC-2】					
GOT	17	14	14	21	
GPT	9	6	5	5	
ALP	117	102	105	107	
GGTP	7	21	15	16	
LDH	158	149	128	131	
CPK	16	10	9	4	
TP	6.5	6.1	6.1	6.9	
ALB	3.5	3.4	3.4	3.7	
CHOL	190	171	176	175	
TRIG	99	82	77	84	
T.BL	0.6	0.7	0.6	0.8	
GLU	80	73	72	85	
BUN	7	8	10	22	
CREA	1.0	1.1	1.2	1.4	
UA	5.9	6.2	8.5	11.9	
NA	140	139	140	134	
K	4.1	4.2	3.5	3.6	
CL	103	105	105	94	
CA	8.8	8.6	8.5	9.0	
P	3.4	3.5	3.7	3.7	
A/G	2.2	2.3	2.3	2.2	
CA/P	2.6	2.5	2.3	2.4	
【4000*CRP】					
	2+				
【1000*ケツエキケンサ】					
PLATELET	241	225	229	244	
WBC	4.6	4.7	5.2	6.0	
RBC	4.63	4.17	3.89	4.62	
HGB	13.5	12.4	12.3	14.1	
HCT	44.3	40.1	39.4	45.9	
MCV	96	97	102	100	
MCH	29.1	29.7	31.5	30.5	
MCHC	30.4	30.9	31.1	30.7	
【2000*ニョウハ°ン】					
PH	-		-	-	
PROT-GU	-		-	-	
GLUC-GU	-		-	-	
ゼンケツ	-		-	-	
ヒ°リルビ°ン	-		+-	-	
ケトン	7.5		5.0	6.0	
ウロビ°リノー	-		-	-	
【2015*チンサ】					
WBC	+-		+-	+-	
【2060*センケツター1】					
GLITTER	<2				
PALE					
DARK					
RBC	0				
セ°ンシヤ	-				
SQUAMOUS	F				
ネ°エキシ	1-10				
ソノタエンルイ	10-/T				
【1520*PT】					
	92%				
【2060*センケツター1】					
オルトトリシ°ン	-	+		+-	

図.15. フローシートの一部

5. 考察

〔仕様変更について〕

前にも述べたように、このシステムの設計にあたっては開発メンバーによる打ち合わせを充分行い、種々の意見に基づいて仕様を決定した。通常は仕様に対する頻回の変更はあまり好ましくない。プログラムの変更だけでなく、場合によってはファイルの構造まで影響する可能性があるからである。

しかし、今回のようなシステムについては、ある程度の仕様変更は止むをえないし、またそうする必要もあるように思われる。なぜなら、従来からコンピュータ処理にあまりなじみのない多数の医師・看護婦が効率よく使いこなすためには、当初の仕様を基にして実際によづ使ってもらい、更に意見を取り入れて改良して行くことで本当に使い易いシステムにしてゆかねばならないからである。

〔問題点〕

(1) 運用時間帯；現在このシステムの稼働時間は、他の業務との関連もあり、午前11:00～午後5:00までとしているが、病棟業務の性格上夜間における問い合わせの要求も多い。しかしコンピュータ管理の問題があり、今後検討を要することである。

(2) 設置場所；既存の病棟詰所の限られたスペースへ端末を設置する場合、病棟側の都合のよい場所へ置けるよう配線工事をしたり、特殊な設置台を用意することが必要であるが、経験上各詰所の特殊事情に逐一マッチさせるには難しい問題がある。前もって利用者側と十分な話し合いに頼る以外に方法はなさそうである。

(3) 騒音対策；端末に対する操作を極力簡単にすると、CRT、プリンターの電源はシステム稼働中は常時ONにしておくことが好ましい。この際モーター音が病棟業務の妨げとなるため、通常はモーターを停止させ出力時のみ自動的にモーターが作動するようハードウェアを改良した。今後、病棟に設置する端末はノンインパクト型プリンターの採用など、「音」に対する十分な考慮を払う必要がある。

〔将来計画〕

問い合わせシステムへのデータ転送は、毎日1回磁気テープを介して行われていたため、データ入力が遅刻になり、当日のデータが翌朝にならないと照会できない場合が生じる。したがって、さらに即時性を持たせるために臨床検査システムと病棟システムとをオンライン結合する予定である。

また今回の開発は、パイロット的な要素を持っており病院全体で14詰所のうち2詰所にしか端末装置を設置していない。したがって他の詰所の医師・看護婦はこの2病棟のどちらかに行くねばならず多少不便な点がある。

いずれにせよシステム化が困難と言われがちな医療の場でも1考を踏み出したわけであり、我々はこれを実用的なものにしてゆくと共に全病棟に端末を設置することを計画している。更にまた、同じような方法で外来診療にも利用されないか、検討を行ってゆくつもりである。

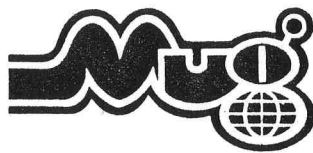
6. まとめ

現段階ではこのシステムは、端末装置の設置場所はまた特定の病棟に限られ、検索対象データも検体検査のみに限られ、システムの運用についてもいくつかの制約があるが、集約すると以下のような特徴を有していると言えよう。

- ① 単なるテストでなく、日常業務として稼働している。
- ② 病棟という診療の場で利用されている。
- ③ 医師・看護婦自身により簡単な操作でデータを直接アクセスできる。
- ④ 診療方針の設定に不可欠なデータを取扱っている。
- ⑤ 診療上重要な「データの通時性」を満足させている。

このシステムの日常診療への適用は緒についたばかりで、まだ本格的な評価を行う時期には至っていないが、当初の開発目的はひとまず達せられたと思われる。今後は前章で述べたような計画を鋭意推進してゆきたいと考えている。

このようなシステムが、診療の現場に自然な形で抵抗なく導入されるならば、医師や看護婦のコンピュータに対する正しい認識の浸透を助長するとともに、さらに積極的なコンピュータ利用への関心が醸成されてくることが期待される。



第6回日本MUG学術大会 昭和54年9月14日(金)

D. パネル(3) MUMPSの医療への応用 司会 大阪医大 山本和子

D-8. MUMPS-systemを利用した臨床診断のためのLaboratory aid

和歌山県立医科大学 中央検査部

大阪府立羽曳野病院 消化器内科
ME研究室

前田 次郎
宮本 一雄
西村 雅隆
山本 實彦
大槌 陽一
今井 敏雄
野口 弘

1 はじめに

当県においては1977年以来、地域医療情報システム開発研究プロジェクトを作り、包括的な医療の地域への展開を最終目的とし、医療技術、医療情報の供給、相互連携の強化、医療の効率化等を目標に、地域における臨床検査並びに医療技術情報の供給を中心としたサポートシステムの研究開発を行ってきた。

医療の中でも、臨床検査の進歩は目覚ましいものがあり、実地医家にとってはなくてはならない存在になっている。しかしながら検査を行うには多くの有能な検査技師や、高価な装置が必要であり、たとえその検査データを供給されても、生かすことができず多くの情報が実地医家に送りされているのが現状であり、情報を臨床病理学的に supervised され、一定の解析がなされておらず、専門医以外の実地医家にとって、データの判読更には臨床での意思決定の資料として利用する事は必ずしも容易でなく、最近の様に同時多項目分析が進められている現状で情報公著り感すら認められている。

当然ながら臨床検査室サイドで、多くの情報の有効利用、更には臨床診断に際して検査項目の効率化を検討する必要がある。

これからの現状を踏えて、現在迄に臨床検査本来の目的である実地医家の診療をサポートするためのシステム開発を実施してきたが、本システムの一つのサブシステムである MUMPS を使用した研究開発用システムを利用して、臨床診断のための Laboratory aid を目的として、2-3の試みを行なわれたので報告する。

2 システムの概要

本システムは市医師会病院内に設置されている。当医師会病院は臨床検査センターを主体として、当モデル地区内の400の医療機関を対象として、検査を実施し情報の供給をおこなっている。

センター内には電算機システムが設置され、日常検査業務が運用されている。このシステムに使用しているコンピュータは、OKITAC 50 model 40である。本システムの一つの特徴は院外に公衆電話回線により接続された医院用及び病院用端末に入出力が出来る。更に当大学検査部にプログラム開発のための研究用システムが専用回線により接続されている事である。

このシステムの開発用言語として国産MUMPS(MUMPS-50)を採用している。

3 臨床診断のための Laboratory aid

先に報告した如く、寄席サポートするためのプログラムは多く用意されているが、現在開発中である検体検査依頼に対して、検査結果に必要なコメントを付け、実地診療の一助にすることを目的としたプログラム等について報告する。

このコメント付けは現在3種類に分けられる。

即ち 1)検査を依頼する上で検査診断を体系化するための指針

2)スクリーニング検査に対するコメント付け及び検査指針

3)特定の疾患群に対するセツト検査結果からの検査診断

以上3グループに分け、臨床専門医が開発研究が可能な様、doctor supportableなプログラムが作製されており、updateされる様になっている。

3-1 検査を遂行し、体系化するための指針用プログラム

実地医家の外来診療において、検査を依頼する場合、種々な条件により単項目又は2-3の項目の依頼する事が多く、内科専門医以外では相互に関連性が少ない項目の依頼が比較的多い傾向にある。この検査が異常値をとった場合、種々な疾患を想定して、診断確定のために検査を体系づける必要がある。この様な時に検査体系を立てることを容易にすると共に、検査の効率化も考慮して、一定のrecommendationを検査等に出力するプログラムを開発した。

単項目又は2-3項目の依頼があつた場合、checkする検査項目は血液検査、化学検査、合わせて24項目で、現在の所51のrecommendationが作られている。

単項目又は2-3の項目の依頼があつた場合、その検査値が設定されたlevel(正常値)より、上、下にはなれた場合、2-3の疾患を想定して、51項目の中で更に深く検索するための追加検査コメントを選択して、検査等に出力する。

• プログラムメニュー (D⁵DPR)

*** RECOMMENDATION TEST (TAN KOMOKU IRAI) ***

- 1 RECOMMENDATION TEST
- 2 TABLE MAINTENANCE (TABLE DATA MODIFY)
- 3 TABLE LIST UP
- 4 TABLE STRUCTURE CHANGE (TABLE DELETE, COPY, KENSA KOMOKU EXCHANGE)

• テーブルメニュー (SDRCM) の一部

コメントとして出力される追加指示は検査名のテーブル

*** RECOMMENDATION TEST KOMOKU *** TABLE = SDRCM

1 URINE	2 BP.	3 CHEST X
4 BOWEL X	5 CHOLECYST	6 HAEMATO.
7 ECG	8 RA	9 LE, DNA
10 CRP (ESR)	11 OGTT	12 PSP
13 ICG	14 TRH	15 PT
16 PREGNANCY	17 THYROID T.	18 BMR
19 TP	20 A/G	21 ELP
22 TTT, ZNTT	23 BUN	24 CREATININE
25 URIC ACID	26 FE	27 TIBC (UIBC)
28 CU	29 NA, K, CL	30 CA, IP
31 GOT, GPT	32 LDH	33 ALP
34 CPK	35 LAP	36 GAMMA-GT
37 CES	38 LDH ISO	39 TB, DB
40 AMYL (S., U.)	41 RT3-U	42 T3
43 T4	44 TSH	45 TRIGLY.
46 TC	47 HBS	48 ALPHA-FETO
49 PBC	50 VB12	51 COOMB'S

当然ながらこの項目の追加、変更は自由に出来る。

• 単項目チェックと水素化還元検査項目のテーブル

*** KENSA KOMOKU *** TABLE= SDRCM

1 BUN	2 URIC A	3 CREATININE	4 CREATINE
5 TRIGLY	6 CHOL	7 BILI	8 FE
9 CU	10 CA	11 GOT/GPT	12 LDH
13 LAP	14 AMYL	15 CPK	16 BS
17 ALP	18 G-GT	19 CHE	20 COR
21 RT3-U			

• 出力項目を選択するためのテーブル (BUNが高い時及び低い時一例)

* アスタリスクの項目が出力される。

BUN (NO=1) HIGH

TABLE = SDRCM

1 URINE	**	2 BP.	**	3 CHEST X	*
4 BOWEL X		5 CHOLECYST		6 HAEMATO.	*
7 ECG	*	8 RA		9 LE, DNA	
10 CRP(ESR)		11 OGTT		12 PSP	*
13 ICG		14 TRH		15 PT	
16 PREGNANCY		17 THYROID T.		18 BMR	
19 TP	*	20 A/G	*	21 ELP	
22 TTT, ZNTT		23 BUN		24 CREATININE	**
25 URIC ACID		26 FE		27 TIBC(UIBC)	
28 CU		29 NA, K, CL	**	30 CA, IP	*
31 GOT, GPT		32 LDH		33 ALP	
34 CPK		35 LAP		36 GAMMA-GT	
37 CES		38 LDH ISO		39 TB, DB	
40 AMYL(S., U.)		41 RT3-U		42 T3	
43 T4		44 TSH		45 TRIGLY.	
46 TC		47 HBS		48 ALPHA-FETO	
49 PBC		50 VB12		51 COOMB'S	

BUN (NO=1) LOW

TABLE = SDRCM

1 URINE	*	2 BP.		3 CHEST X	
4 BOWEL X		5 CHOLECYST		6 HAEMATO.	*
7 ECG		8 RA		9 LE, DNA	
10 CRP(ESR)		11 OGTT		12 PSP	
13 ICG		14 TRH		15 PT	
16 PREGNANCY	*	17 THYROID T.		18 BMR	
19 TP	**	20 A/G	**	21 ELP	
22 TTT, ZNTT		23 BUN		24 CREATININE	
25 URIC ACID		26 FE		27 TIBC(UIBC)	
28 CU		29 NA, K, CL		30 CA, IP	
31 GOT, GPT	**	32 LDH	*	33 ALP	*
34 CPK		35 LAP		36 GAMMA-GT	
37 CES		38 LDH ISO		39 TB, DB	
40 AMYL(S., U.)		41 RT3-U		42 T3	
43 T4		44 TSH		45 TRIGLY.	
46 TC		47 HBS		48 ALPHA-FETO	
49 PBC		50 VB12		51 COOMB'S	

- 血液検査における check 項目と追加検査コメント。
現在開発中であるが RBC, Hb, Ht を check 検査項目とし、MCV, MCH
で貧血を分類した後、それぞれ追加検査、コメント指示のプログラムである。

Recommendation;	ANEMIA (RBC↓ Hb↓ Ht↓)					
	MCV→	MCH→	MCV↓	MCH↓	MCV↑	MCH↑
Blood smear	*		*		*	
Urinalysis	*		*			
Cocult blood	*		*			
Chest-X ray	*					
Stomach X-ray	*		*			
TP	*		*			*
A/G ratio	*		*			*
Ti & DB					*	*
SGOT					*	*
SGPT					*	*
SLDH					*	*
Serum Iron	*		*			*
TIBC(UIBC)	*		*			*
T4					*	*
VB12					*	*
CRP	*		*		*	*
War					*	*
	Normocytic Anemia		Hypochromic Microcytic Anemia		Macrocytic Anemia	

以下実際に打出した例である。

- BUNが30mg/dlであるとする。下段の様な recommendation が出力される。
尚*アスタリスクは特に重要であることを示す。

ENTRY : 'H' FOR HIGH, 'L' FOR LOW OR ' ' (ISP) FOR NORMAL

1 BUN	H	2 URIC A	3 CREATININE
4 CREATINE		5 TRIGLY	6 CHOL
7 BILI		8 FE	9 CU
10 CA		11 GOT/GPT	12 LDH
13 LAP		14 AMYL	15 CPK
16 BS		17 ALP	18 G-GT
19 CHE		20 COR	21 RT3-U

Recommendation TEST *

URINE *	1
BP. *	2
CHEST X	3
HAEMATOL.	6
ECG	7
PSP	12
TP	19
A/G	20
CREATININE *	24
NA, K, CL *	29
CA, IP	30

このルーチンは Table によって簡単に発展、変更が自由に出来る。又他の目的例いは data check のためのプログラムにも変更可能である。

3-2 screening test の結果より臨床診断のための Laboratory aid

screening test は診断の確定のため、医師の診察所見と共に重要な所見となる。古くより screening test はなんらかの形で実施されてきたが、検査技術の発達と共にその項目も多くなり、最近では multichannel の自動分析器が普及するにつれ検査技術が先行する形で検査項目数も多くなり、一つの組合せを作り上げていった。

日本においては紫田等の血液スペクトル、Preston, Diano 等による biochemical profiling、又は biopsy の概念を導入した。

これらより得られた結果をより有効利用するために、又臨床医に理解しやすい形で提供するため多くの試みがなされて来た。

これ等の考へは更に発展し、情報処理技術の発達と共にコンピュータ処理がなされ、多変量解析、主成分分析等多くの研究がなされている。

一般に病院における admission profile は SMA 12/60, SMA 4, VDR L, Urinalysis 等を screening test とされ日常運用されている。これ等の検査結果は臨床病理医がエックして、異常値と臨床診断との対比がなされ報告される。これを CPU システムにより、置換したシステムは Grams 等が報告している。

我々はこのような機能をこの検査システムに持たすべく、その一つの機能である検査結果を判断する機能と臨床医の診断を助けとなる Laboratory aid, コメント付けの機能を開発中である。ここでは検査の screening test から得られる検査診断及びコメント付けのプログラムを紹介する。

計量診断学では、多くの確定診断のついた症例の検査データを用いて種々の統計学的処理によって判別関数、多変量解析などから、病名推定を行っている例が多く、多くの確定データが必要であり、このデータも時間軸に多くは関係なく利用されており、推定モデルに種々の制約、条件を必要とする等多くの問題点がある。今回我々は採用している方法はパターンマッチ方法である。即ち臨床医の判断論理から各検査項目の各疾患時の変化を正常値を中心として高・低にわけ、各疾患の pattern を作成し、10段階に評価をあたへ、未知疾患の検査値のこの pattern と比較し、一致度を数値で評価する。

尚多くの場合、検査項目は化学検査のサデおこなわれているが、この screening test に使用する検査項目は血液学検査、血清学検査、化学検査、18項目、パラメータ-21 を合せて判断している。

・プログラムメニュー (D^{ASDPS})

*** COMPUTER-AIDED DIAGNOSIS V3 (SCREENING) -- MENU ***

- 1 COMPUTER-AIDED DIAGNOSIS
- 2 TABLE MAINTENANCE (TABLE DATA MODIFY)
- 3 TABLE LIST UP
- 4 TABLE STRUCTURE CHANGE (TABLE DELETE, COPY, KENSA KOMOKU EXCHANGE)

• テーブルメニュー (SDSCR) の一部

使用する検査項目及びパラメータ

*** KENSA KOMOKU MEI ***

TABLE = SDSCR

1 RBC	2 HB	3 HT	4 WBC
5 MCV	6 MCH	7 TP	8 ALB
9 BUN	10 URIC A	11 ALP	12 GOT
13 GPT	14 LDH	15 CHOL	16 CA
17 FE	18 TB	19 A/G	20 CRP
21 RA			

• 評点(塗み付け)のテーブル

*** HYOTEN TYPE ***

TABLE = SDSCR

0=0,0,0
2=20,0,0
4=5,5,0
6=0,5,5
8=0,0,20
A=-9999,-9999,,30

1=30,-9999,-9999

3=10,0,0

5=0,10,0

7=0,0,10

9=-9999,-9999,30

評点911

H.

N.

L.

評点

• 出力される病名テーブル

*** BYOMEI KOMOKU ***

TABLE = SDSCR

1 - ACUTE INFECTION	20 - NEPHROSIS
2 - CHRONIC INFECTION	21 - DIABETES MELLITUS
3 - PERICARDITIS	22 - HYPERLIPEMIA
4 - MYOCARDIAL INFARCTION	23 - MULTIPLE MYELOMA
5 - CHRONIC HEART FAILURE	24 - GOUT
6 - POSTGASTECTOMY SYNDROME	25 - DEHYDRATION
7 - PROTEIN LOSSING ENTEROPATHY	26 - SLE
8 - MALEABSORPTION SYNDROME	27 - DERMATOMYODITIS
9 - ULCERATIVE COLITIS	28 - RHEUMATOID ARTHRITIS
10 - ACUTE CHOLECYSTITIS	29 - IRON DEFICIENCY
11 - ACUTE PANCREATITIS	30 - MEGALOCYTIC ANEMIA
12 - CHRONIC PANCREATITIS	31 - HEAMOLYTIC ANEMIA
13 - ACUTE HEPATITIS	32 - ACUTE LEUKEMIA
14 - CHRONIC HEPATITIS	33 - CHRONIC LEUKEMIA
15 - CHOLESTATIC HEPATITIS	34 - HYPERTHYROIDISM
16 - LIVER CIRRHOSIS	35 - HYPOTHYROIDISM
17 - LIVER CARCINOMA	36 - HASIMOTO'S DISEASE
18 - ACUTE NEPHRITIS	37 - HYPERPARATHYROIDISM
19 - RENAL FAILURE	38 - HYPOPARATHYROIDISM
	39 - CUSHING'S DISEASE
	40 - SARCOIDOSIS
41 - TOXICEMIA OF PREGNANCY	
42 - COLLAGEN DISEASE	
43 - RHEUMATIC FEVER	
44 - FATTY LIVER(OBESITY)	
45 - CORTICOID INDUCED	
46 - THIAZID INDUCED	

• 判断用テーブル

*** BYOMEI HYOTEN TYPE TABLE *** TABLE = SDSCR

1- ACUTE INFECTION

1	RBC	5
2	HB	5
3	HT	5
4	WBC	1
5	MCV	5
6	MCH	5
7	TP	357
8	ALB	37
9	BUN	5
10	URIC A	5
11	ALP	5
12	GOT	5
13	GPT	5
14	LDH	5
15	CHOL	7
16	CA	5
17	FE	7

2- CHRONIC INFECTION

1	RBC	7
2	HB	7
3	HT	7
4	WBC	3
5	MCV	7
6	MCH	7
7	TP	7
8	ALB	7
9	BUN	5
10	URIC A	5
11	ALP	5
12	GOT	5
13	GPT	5
14	LDH	5
15	CHOL	5
16	CA	5
17	FE	57

NEXT PAGE ? YES((CR)) :

18	TB	35
19	A/G	57
20	CRP	1
21	RA	5

18	TB	5
19	A/G	7
20	CRP	2
21	RA	5

↑
記入911°

*** COMMENT ***

URINALYSIS, X-RAY, ESR

URINALYSIS, X-RAY

*** BYOMEI HYOTEN TYPE TABLE *** TABLE = SDSCR

3- PERICARDITIS

1	RBC	5
2	HB	5
3	HT	5
4	WBC	2
5	MCV	5
6	MCH	5
7	TP	5
8	ALB	5
9	BUN	5
10	URIC A	5
11	ALP	5
12	GOT	4
13	GPT	5
14	LDH	4
15	CHOL	5
16	CA	5
17	FE	5

4- MYOCARDIAL INFARCTION

1	RBC	5
2	HB	5
3	HT	5
4	WBC	2
5	MCV	5
6	MCH	5
7	TP	5
8	ALB	5
9	BUN	5
10	URIC A	5
11	ALP	5
12	GOT	2
13	GPT	3
14	LDH	1
15	CHOL	5
16	CA	5
17	FE	5

NEXT PAGE ? YES((CR)) :

18	TB	5
19	A/G	5
20	CRP	1
21	RA	3

18	TB	5
19	A/G	5
20	CRP	2
21	RA	5

*** COMMENT ***

URINALYSIS, X-RAY, ECG

X-RAY, ECG

以下実例を示すが、プログラム開発中のため、キーボードから検査値を低高判断し、H、L、N、を入力する必要があるが、この判断機能はルーチン運用時には必要である。尚病名数は6と多いので推定にはかなりの時間を要し、処理時間は約5秒である。

・実例 25才男子 急性白血病

*** COMPUTER-AIDED DIAGNOSIS (V3-PM) ***

TABLE NAME (<CR> FOR SDSCR) OR END(Z) = SDSCR

KENSA KOMOKU SU = 21

BYOMEI SUITEI MAX SU = 5

HYOTEN MIN VALUE = 100

*** INPUT DATA ***

1 RBC	L	12 GGT	H
2 HB	L	13 GPT	N
3 HT	L	14 LDH	H
4 WBC	H	15 CHOL	N
5 MCV	N	16 CA	N
6 MCH	N	17 FE	L
7 TP	N	18 TB	N
8 ALB	N	19 A/G	N
9 BUN	N	20 CRP	H
10 URIC A	H	21 RA	N
11 ALP	N		

* DATA *

HYOTEN--- BYOMEI NO

240	---	32
200	---	4, 31
185	---	17
175	---	24
170	---	1, 3, 26, 27
150	---	18
145	---	43
140	---	2, 10, 28, 30, 36
130	---	6, 9, 11, 35
125	---	44, 46
120	---	7, 12, 16, 25, 38, 39
115	---	42
110	---	8, 22, 45
100	---	5, 37
90	---	14, 15, 34, 41
80	---	21, 40

* DIAGNOSIS *

240

ACUTE LEUKEMIA COMMENT : BLOOD SMEAR

200

MYOCARDIAL INFARCTION COMMENT : X-RAY, ECG

HEAMOLYTIC ANEMIA COMMENT : URINALYSIS, COMB'S TEST
185

LIVER CARCINOMA COMMENT : LIVER SCINTI. ELECTROPHORESIS, ALFA-FETO
175

GOUT COMMENT : X-RAY
170

ACUTE INFECTION COMMENT : URINALYSIS, X-RAY, ESR

PERICARDITIS COMMENT : URINALYSIS, X-RAY, ECG

3-3 臓器別セット検査による肝疾患の推定病名付けのためのプログラム。

セット検査の結果を判断して、検査診断コメントを出力するためのプログラムである。

臨床検査データからの検査診断は、代謝性疾患の病態解析には有力な手がかりになることは言ふまでもないが、多くの研究者により、判別可能な手法が報告されている。しかし、その検査項目の種類、処理方法は多岐にわたるが、理想的には肝の排泄と代謝の機能を測定する一連の検査を用ひるべきである。いくつかの独立した検査を同時にこなすことによつて最大の情報が得られる。しかし、特異的に肝臓を直接的に評価する検査はない。他の因子の影響され原発性肝疾患以外病態によつても変化をうける。更に肝疾患は時間的要素が非常に大きい事等、肝疾患の計量診断に重要な問題点である。

従つて肝機能検査は肝機能を評価しているわけではなく、肝疾患の活動性を評価している事が多く、これからの考へを取り入れたパターン認識による検査診断を考へられる。

この様に多様性のもつた肝機能検査の内から一般的に臨床医がよく使用する検査項目を出来るだけ利用して、対象とする疾患名を明示付けし研究開発中である。

方法として、各病名、各検査項目毎に経験的知識や実際の検査値の中を考慮して、ある範囲を設定し、実際の検査値と各病名の推定範囲パターンとの一致度を計算して、病名や追加検査名を推定する。

最近、多次元解析を応用し、多項目検査データの相互関連性にもとづいた健康及び病態の状態の像を総合的に把握、評価を行なうとする種々の試みが報告されている。

我々はこのような統計学的処理によるパラメータは出来るだけ避け、出来るだけ臨床サイドで実施している診断作法に基づいて、経験的知識を吸収していくモデルを用ひた。

パターンとの一致度を計算するモデルを出来るだけ精度を上げるため、モデルを複雑化している。

- 即ち Δ level 1. すべての検査が範囲内にある病名を出力する
- Δ level 2. 範囲内にある検査項目の数の多い順に出力する
一部、枝分かれ論理を併用
- Δ level 3. 範囲外、範囲内、欠損検査に対して、各病名、各検査項目毎に
評点を割り、総評点の大きい順で出力する。
- Δ level 4. 範囲を多層化する。事前確立の高い順に出力する。

D^SDPK

*** COMPUTER-AIDED DIAGNOSIS V2 (ZOKI BETU) -- MENU ***

- 1 COMPUTER-AIDED DIAGNOSIS level 1.
- 2 TABLE MAINTENANCE (TABLE DATA MODIFY)
- 3 TABLE LIST UP
- 4 TABLE PATTERN RANGE PRINT OUT (LINE PRINTER)
- 5 BYOMEI DATA COMMON RANGE CHECK
- 6 BYOMEI COMBINATION CHECK OF COMMON RANGE
- 7 TABLE STRUCTURE CHANGE (TABLE DELETE, COPY, KENSA KIMOKU EXCHANGE)
- 8 COMPUTER-AIDED DIAGNOSIS (MATCH ORDER) level 2.
- 9 COMPUTER-AIDED DIAGNOSIS (HYOKA TEN) level 3.

●肝セツトテーブル(検査項目, 評失病名, 検査値の範囲, 評失917°) (SD LIV)

*** KENSA KOMOKU MEI *** TABLE = SDLIV

1 TP	2 AL	3 GL	4 TTT
5 T-C	6 TB	7 DB	8 DB/TB
9 GPT	10 GOT	11 GOT/GPT	12 LDH
13 G-GT	14 LAP	15 ALP	16 CES
17 FE	18 ALL (ALの弱体化)	19 TBL (TB)	20 GPTL (GPT)
21 LDHL (LDH)			

*** HYOTEN TYPE *** TABLE = SDLIV

1=0, 5, 0	2=0, 10, 0	3=0, 20, 0	4=0, 0, 0	5=0, -99, 0
6=-99, 0, 0	↑ 評失多 ↑ 範囲外	↑ 範囲内 ↑ 欠損		

*** BYOMEI KOMOKU *** TABLE = SDLIV

1 -	ACUTE HEPATITIS, NON ICTERIC STAGE
2 -	ACUTE HEPATITIS, ICTERIC STAGE
3 -	CHRONIC ACTIVE HEPATITIS
4 -	CHRONIC INACTIVE HEPATITIS
5 -	PERISTANT HEPATITIS
6 -	CHOLESTATIC HEPATITIS
7 -	ACTIVE LIVER CIRRHOSIS
8 -	INACTIVE LIVER CIRRHOSIS
9 -	FATTY LIVER WITHOUT HEPATITIS
10 -	OBSTRUCTIVE JAUNDICE
11 -	LIVER CARCINOMA
12 -	HAEMOLYTIC JAUNDICE
13 -	ROTOR, DUBIN-JOHNSON SYNDROME
14 -	NOT PARTICULAR

*** BYOMEI PATTERN TABLE *** TABLE = SDLIV

1- ACUTE HEPATITIS, NON ICTERIC STAGE

1 TP	1	6.5	--	8.5
2 AL	1	3.5	--	4.5
3 GL	2	2.5	--	4
4 TTT	1	4	--	15
5 T-C	1	150	--	260
6 TB	3	0.5	--	1.2
7 DB	4	0	--	99
8 DB/TB	1	0	--	99
9 GPT	1	100	--	1500
10 GOT	1	100	--	1500
11 GOT/GPT	1	0.1	--	0.6
12 LDH	1	100	--	400
13 G-GT	1	25	--	200
14 LAP	1	20	--	100
15 ALP	1	40	--	120
16 CES	1	1000	--	2500
17 FE	1	100	--	350
18 ALL	5	0	--	2
19 TBL	5	0	--	1.2
20 GPTL	5	0	--	200
21 LDHL	5	1000	--	9999

2- ACUTE HEPATITIS, ICTERIC STAGE

TP	1	6.5	--	8.5
AL	1	3.5	--	8.5
GL	2	2.5	--	4
TTT	1	4	--	20
T-C	1	150	--	250
TB	3	1.5	--	30
DB	4	1	--	30
DB/TB	1	0.4	--	0.8
GPT	1	300	--	3000
GOT	1	250	--	3000
GOT/GPT	1	0.1	--	0.6
LDH	1	100	--	500
G-GT	1	30	--	300
LAP	1	30	--	100
ALP	1	40	--	150
CES	1	1000	--	2000
FE	1	120	--	400
ALL	5	0	--	2
TBL	5	0	--	1.2
GPTL	5	0	--	200
LDHL	5	1000	--	9999

↑

評失917°

*** COMMENT *** 追加検査

・实际例

54才男子, 主訴 全身倦怠, liver biopsy i hepatoma 疑念,
処理時間は level 2, level 3, 共 12秒であった

*** INPUT DATA ***

1 TP	7.1	12 LDH	495
2 AL	3.2	13 G-GT	317
3 GL	3.9	14 LAP	123
4 TTT	9.1	15 ALP	273
5 T-C	199	16 CES	803
6 TB	2.1	17 FE	102
7 DB	1.6	18 ALL	3.2
8 DB/TB	0.76190476190476	19 TBL	2.1
9 GPT	122	20 GPTL	122
10 GOT	165	21 LDHL	495
11 GOT/GPT	1.35245901639344		

level 2.

* DIAGNOSIS *

LIVER CARCINOMA

COMMENT : LIVER SCINT. LIVER BIOPSY, SONOGRAPHY, ALPHA-FT
(15/21) GOT/GPT, FE, ALL, TBL, GPTL, LDHL

CHRONIC ACTIVE HEPATITIS

COMMENT : HBS-AG OR HBS-AT
(13/21) DB, GOT/GPT, LDH, ALP, ALL, TBL, GPTL, LDHL

CHOLESTATIC HEPATITIS

COMMENT : PLEASE CHECK USING DRUGS
(12/21) AL, T-C, GOT/GPT, LDH, CES, ALL, TBL, GPTL, LDHL

ACTIVE LIVER CIRRHOSIS

COMMENT : LIVER SCINT. LIVER BIOPSY, ALP ISOENZYME
(12/21) AL, T-C, GOT/GPT, LDH, LAP, ALL, TBL, GPTL, LDHL

level 3.

HYOTEN--- BYOMEI NO

90 --- 11
85 --- 3
75 --- 6, 7
45 --- 5
0 --- 1, 2, 4, 8, 9, 10, 12, 13, 14

* DIAGNOSIS *

90

LIVER CARCINOMA

COMMENT : LIVER SCINT. LIVER BIOPSY, SONOGRAPHY, ALPHA-FT

85

CHRONIC ACTIVE HEPATITIS

COMMENT : HBS-AG OR HBS-AT

75

CHOLESTATIC HEPATITIS

COMMENT : PLEASE CHECK USING DRUGS

ACTIVE LIVER CIRRHOSIS

COMMENT : LIVER SCINT. LIVER BIOPSY, ALP ISOENZYME

4 もすび

検査データに臨床医の判断論理に近い手法で処理を加えて、測定された検査データに、検査診断コメントを付加することにより、実地医家の診療をサポートする事を目的として、現在、上述のプログラムが研究開発されている。開発段階で評価をおこないながら、改良を加へ、一定の level になり、実地医家に受け入れやすい形になった時、現在稼働している検査システムの data base とドッキングする予定である。

尚この研究は医療情報開発センターの医療情報システムの研究開発の一連の研究活動にある地域医療情報システム研究(和歌山県)の一翼といたなされたものである。

第6回日本 MUG 学術大会
抄 録 集

昭和 54 年 9 月 1 日 発行 頒価 2000 円

編輯人 606 京都市左京区聖護院川原町54
京都大学医学部附属病院中央情報処理部
平 川 顕 名 TEL 075-751-3646

発行人 日本 MUMPS ユーザーズ グループ
460 名古屋市中区新栄町2-1-9 雲竜ビル東館
マンブス システム 研究所 TEL 052-251-7408
振替口座 名古屋 56255 番

よりよい医療とくらしのために

医薬品総合商社



昭和薬品株式会社

名古屋市名東区本郷2-173 〒465

TEL 大代表「052」773-7373

沼津支店	富士支店	静岡支店	藤枝支店	浜松支店	豊橋支店	岡崎支店
半田支店	名古屋東支店	名古屋西支店	名古屋南支店	名古屋北支店	一宮支店	岐阜支店
岐阜病専店	大垣支店	美濃加茂支店	多治見支店	高山支店	桑名支店	四日市支店
津支店	松阪支店	伊勢支店	尾鷲支店			

脂質代謝異常の改善に…

酪酸リボフラビン製剤

アイカス[®] 顆粒・錠

【作用】

- 生体内に長時間貯留し、持続的なビタミンB₂作用をあらわす。
- 組織へのとり込みがすぐれ、高率にFADに転化する。
- 脂肪酸のβ酸化を促進し、脂質代謝を改善する。
- 生体内過酸化脂質に直接作用し、これを分解する。



昭和新薬株式会社

名古屋市千種区松軒町2ノ12

本社 052 722 6657
東京営業所 03 362 2701
福岡営業所 092 473 0417



いま確かな1歩。

耐性菌に パニマイシン

PANIMYCIN FOR INJECTION ● PANIMYCIN INJECTION

特長

- グラム陽性菌・陰性菌に殺菌的に作用します。
- 緑膿菌、変形菌による感染症にすぐれた効果を示します。
- 多剤耐性の肺炎桿菌、大腸菌および黄色ブドウ球菌による感染症にも有効です。
- 筋注ですみやかに血中に移行し、腎、肺に高濃度で認められます。
- 活性のまま高濃度で尿中に排泄されます。



■適応症

緑膿菌、変形菌による下記感染症および肺炎桿菌、大腸菌、黄色ブドウ球菌のうち、カナマイシンを含む多剤耐性菌で、ジベカシン感受性菌による下記感染症
●敗血症 ●膿瘍、疳、痼腫症、蜂窩織炎 ●扁桃炎、肺炎、気管支炎 ●腹膜炎、腎盂腎炎、膀胱炎、尿道炎 ●中耳炎 ●術後感染症

■用法・用量

成人：1日量100mg(力価)を1～2回に分けて筋肉内注射
小児：1日量1～2mg(力価)/kgを1～2回に分けて筋肉内注射

なお、年齢、症状により適宜増減してください。

■包装・薬価基準

注射用パニマイシン

50mg(力価)10バイアル……(1,480.00円/バイアル)

100mg(力価)10バイアル……(2,730.00円/バイアル)

パニマイシン注射液

50mg(力価)/1ml 10アンプル(1,480.00円/アンプル)

100mg(力価)/2ml 10アンプル(2,730.00円/アンプル)

■使用上の注意

製品添付文書を熟読してください。

※医師等の処方せん・指示により使用すること。

健保適用

明治製薬株式会社

東京都中央区京橋2-8

I(0976)



新しい医療システムを創る。

OKIME Terminal-100 F

OKIME Terminal-100

OKITAC system 50 病院情報システム

MUMPS-50 (医用データベース言語)

臨床検査システム

肺機能検査システム

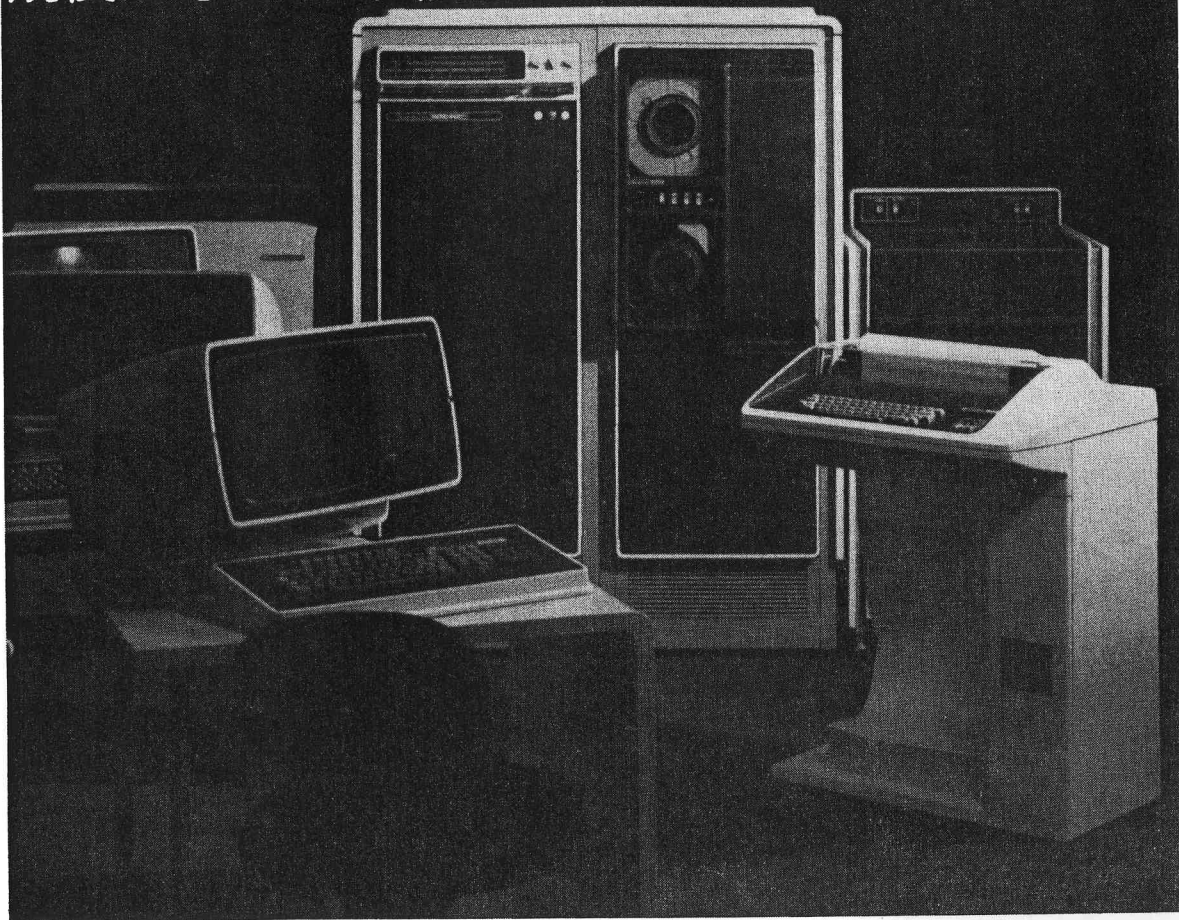
豊かな情報化社会をひらく
エレクトロニクスの
沖電気

◇ 沖電気工業株式会社

●お問合せは——電子計算機部・医療システム課 ☎ (03)454-2111 (大代) まで

MUMPS-N

規模に応じたシステムを、標準MUMPSで



MUMPS-Nは、小規模システムから大規模システムまで、様々な要求にお応えできます。	いえましょう。	MUMPS-Nは、標準MUMPS言語です。
	日本ミニコンのMUMPS-Nシステムは、ソフトウェアの機能だけではなく、ハードウェアも一貫した開発思想で統一することにより、NOVAを使用する小規模システムから、ECLIPSEの最上位機種であるM/600を使用する大規模システムまで、様々なレベルの業務に対応できる柔軟なシステムです。	MUMPS-Nは、標準MUMPS言語仕様(ANSIX 11.1, 1977)を完全に包括強化していますので、システムを管理使用する人々が世界的レベルで共通の言語仕様で利用でき、標準としての大きなメリットをお約束します。



日本ミニコンピュータ

本社 東京都渋谷区神宮前6 12 20 〒150 Tel (03)406 6451(代)
大阪(06)365 7051・名古屋(052)531 4636・福岡(092)472 5917・横浜(0298)24 1609