コンピュータ・サイエンス誌

昭和49年1月1日発行
毎月1日発行
通巻第63号
昭和44年5月2日
第3種郵便物認可
昭和44年1月18日
国鉄東局特別扱承認雑誌第3065号

bit

1
1974

共立出版

新連載　LISP入門①　マッカーシーの条件式とM式

チューリングとディジタル計算機の起源

IBMの研究開発

コンピュータ・サイエンス誌

bit 1974 1

ヒトは
情報を呼吸する
それは文明のいぶき……
機械と対話しながら
未来の序曲をきく

デザイン・構成／沼田　望

マルティクス近況

―プロジェクトMACに参加して―

1965年の FJCC で発表された Multics は，約7年にわたる開発，使用経験をもとにして種々の改良が加えられた2世代目の新しいハードウェア H6180 を得て，コンピュータ・ユーティリティの理想へと着々と歩を進めています。ここではプロジェクト MAC のなかの一つの研究部門である CSR と，H6180 Multics の近況を簡単に紹介します

小田一博
Kazuhiro Oda
東京芝浦電気（株）

1. はじめに

1973年の3月より約3カ月間，マサチューセッツ州ケンブリッジ，ボストン地区に滞在し，マルティクス(Multics: **Multiplexed Information and Computing Service**）を実地に使用する機会，およびその間の約1カ月間 MIT のプロジェクト MAC の中の CSR（Computer Systems Research）部門に滞在する機会を得ました。そのときの体験をもとに，マルティクスの近況を中心に，最近のプロジェクト MAC の動き，CSR の活動状況，および新しく導入された H6180 などについてそれぞれ簡単に紹介したいと思います。

マサチューセッツ工科大学（MIT）は，日本でもその名が知られているように工科系の大学としては超一流のところで，数々の新技術の開発，研究がなされたり，またなされつつあります。MIT は**写真1**でもわかるように，ボストンとケンブリッジの間を流れているチャールズ河の河辺にそって建物が立ち並び，春先から夏にかけて，この河にヨットが浮かぶ風景を見ることができます。

私のように電子計算機の仕事に従事している者にとって，MIT はかの有名な CTSS，マルティクスというタイム・シェアリング・システムを開発したところとして身近に感じられます。

プロジェクト MAC の研究室のあるところは，大学の建物からは少し離れたテクノロジ・スクェア（Technology Square）とよばれる一角にあります（**写真 2**）。ここは，9階建てのビルが三方をとり囲み，その中の広場が四角形をしているところからこのような名前が付いたものと思われます。ここには，IBM のケンブリッジ・サイエンティフィック・センタ（CP 67/CMS を開発したところ），HIS のケンブリッジ・インフォーメーション・システムズ・ラボラトリ（プロジェクト MAC と一緒に マルティクス を開発したところ），ポラロイド・カメラの研究所などがあり，まさにテクノロジ・スクェアです。

ケンブリッジ，ボストン地区には，ハーバード大学，ボストン大学などがあります。

2. プロジェクト MAC と CSR

テクノロジ・スクェアにある三つの9階建てのビルの一つにプロジェクト MAC の本部および研究室，H 645 のある計算センタがあります。1階には本部，4階には図書室と MAC の論文集を取り扱っている Publication

写真 1　ボストンから MIT を望む

写真 2　テクノロジ・スクェア

写真 3　著者と Saltzer 教授

写真 4　プロジェクト MAC の一室

Office, 5 階には教授，大学院生などの研究室，端末ルーム，会議室，8 階にも研究室の一部，コピー・マシン，9 階には H645 などが置いてあります。

プロジェクト MAC は，いくつかの研究テーマごとにグループあり，CSR もその中の一つです。CSR 以外のグループとしては，

- automatic programming
- MATHLAB
- computation structures
- dynamic modeling, computer graphics and computer network

などの部門があります。各部門の 1 年間の研究成果は，Project MAC Progress Report として年 1 回出版されています。

CSR は，Corbató 授教，Saltzer 教授そして Schroeder 教授のもとに大学院生，学生が約 30 名ほどおり，次のようなテーマで研究，開発活動を行なっています。

（1） 性能評価とモデル作り
- 可動ヘッドをもつ記憶媒体（ディスクなど）の解析
- 統計解析用プログラム・パッケージ
- コア・メータリング
- H6180 ハードウェア・タイミング解析
- ベンチマーク・プログラムの開発
- マルチプログラミング・レベルの動的制御

（2） プロテクションとセキュリティ
- 保護されたサブシステム
- リンキングを保護された領域から移すこと
- I/O コントロールを保護された領域から移すこと
- 保護された領域を簡単にするためのマルチプロセス構成
- スーパーバイザ／ファイル・システムを簡単にするための大きなセグメント番号

（3） ARPA ネットワーク
- 統合されたユーザ・レベルの protocol
- 全 2 重 IMP 接続のための新しいソフトウェア
- ファイル転送用の protocol の開発

（4） その他
- マルティクスのドキュメント作成
- マルティクス LISP
- Seal（新しい言語）
- APL に対する提言

上記の各研究テーマからも察せられるように，マルティクスに関するテーマはまだ少なからずありますが1973年 1 月に HIS が商用として売り出すことを発表したこ

ともあり，マルティクスそのものの改良，開発はほぼ一段落したというところです。Saltzer 教授の話では，プロテクション／セキュリティについてはまだしばらく研究を続けるとのことですが，今後はソフトウェアの理論のほうへ進みたいということでした。

CSR の中では，そのグループ内だけに，論文またはメモなどの配布が行なわれています。このメモを，何かコメントがあったら下さいという意味で，Request for Comments（RFC）とよんでいます。これは先ほど述べた研究テーマや新しいアイデアなどを各メンバーに知らせ，それに対する意見，批判をあおごうという目的です。学会やシンポジウムに発表予定の論文などが，このような形式で，数カ月以上前からグループの中で意見，批判を受け，その結果はじめて世の中に出ています。このような環境および，マルティクスの開発の例でもわかるように，遠い将来に目標をおき，その上で現実的なアプローチを積み重ねていく態度はわれわれもみならう必要があるかと思います。

CSR では，週1回，2時間程度の会合が行なわれています。連絡事項やその時折りのテーマ（外部から人を呼んで講演をしてもらうこともあるし，グループ内の人が研究テーマの進行状況を発表することもあります）を選んで発表，討議を行なっています。

これ以外に，マルティクスの開発に従事している人たちの会合としてマルティクス・スタッフ・ミーティングが行なわれています。

ちょうど私が滞在した期間は，H645 から H6180 への移行作業の最終段階でした。ただし，まだ一般ユーザ・サービスは始まっておらず H6180 の使用はマルティクス開発スタッフのみに限定されていました。

H645 はプロジェクト MAC のビルにありますが，H6180 は少し離れた IPC(Information Processing Center) にあり，同じ所に IBM /370 モデル 165 が置いてあり，バッチ，TSO などのサービスを行なっています。

マルティクスの端末は，プロジェクト MAC のビルの5階にある端末ルームにテレタイプ，IBM 2741，グラフィック・ディスプレイなどが7台ぐらい置いてあり，空いている端末はいつでも使えるようになっています。これ以外に，各研究室の主な所には専用の端末が置いてあります。Saltzer 教授と Schroeder 教授の部屋の中間に秘書の部屋（ここを通らないと教授の部屋へはいれない）には IBM 1050 という端末が置いてあり，2人の教授のほかにたまに大学院生が使用しています。これ以外にも，たとえばさきほどの RFC のコピー，配布を行なっている部屋では，そう若くない（?）女の人がマルティクス端末を使って資料の管理を行なったりしています。ふつうのタイプライタを使うのと同じ感覚で端末を使っているという感じです。

写真 5 ポータブル端末の一種（音声カプラ付き）

写真 6 ポータブル端末を用いてモーテルからマルティクスを使用

3. マルティクス近況

マルティクス開発の過程が，1972年の SJCC で，"Multics: The first seven years" という題目で発表されています。くわしくはその論文を参照していただくとして，ここでは使用の実例などを折りまぜながらマルティクスの近況を報告します。

私が滞在していたときにはまだ H6180 による一般ユーザ・サービスは行なわれていませんでしたが，その後のセンター・ニュース（"the Bulletin" というのが月刊で IPC から発行されています）によると，7月から，"Trial Service" という形でサービスが開始されたとのことです。また，1961年以来約12年間稼動していた CTSS がついに 1973 年7月にそのサービスを停止し，7094 もニューヨーク市の equipment dealer に払い下げられたとのことです。H645 も H6180 が動き始めてから2，3カ月後にはそのサービスを停止する予定です。

IPC に設置されている H6180 の構成図を**図1**に示し

ます。H6180は，H645の経験，および商用として売出されている H6000 シリーズとの互換性などを考慮して作られたもので，ハードウェア面では次のような改良，変更が加えられています。

・リング保護の完全なハードウェア化
・PTW，SDW，連想メモリの拡張
・10進，文字，ビット操作命令の追加
・ページ用デバイスとしてバルク・ストアの採用
・集団ディスク装置の採用
・GIOC に代わって，入出力マルチプレクサ（IOM）とフロント・エンド・プロセッサ DN/355 の採用
・その他

CPU の仮想メモリ，リング保護機構などの部分を除いては，ハードウェアは標準の H6000 シリーズのものが使用されています。H 6180 になってから，一段と効率，信頼性も向上し，さらに多くのマルティクス・ユーザが見込まれています。ソフトウェア上の違いはハードウェアに依存する部分（アセンブラで記述されたモジュール）を除いて変更はなく，OS を PL/I で記述したことがここでも生かされたわけです。

マルティクスの初期の目標の一つにもなっていたように，週7日，1日24時間フル稼動がほぼ達成されており，通常3シフトで，昼よりも夜，夜よりも真夜中というふうに使用料が安くなっていきます。

4. マルティクス使用実例

H 6180 は前にも述べたように，正式には開発スタッフだけにしかその使用を許されていなかったのですが，Saltzer 教授，Schroeder 教授のご好意により使用することができました。私の名前が登録されていないので

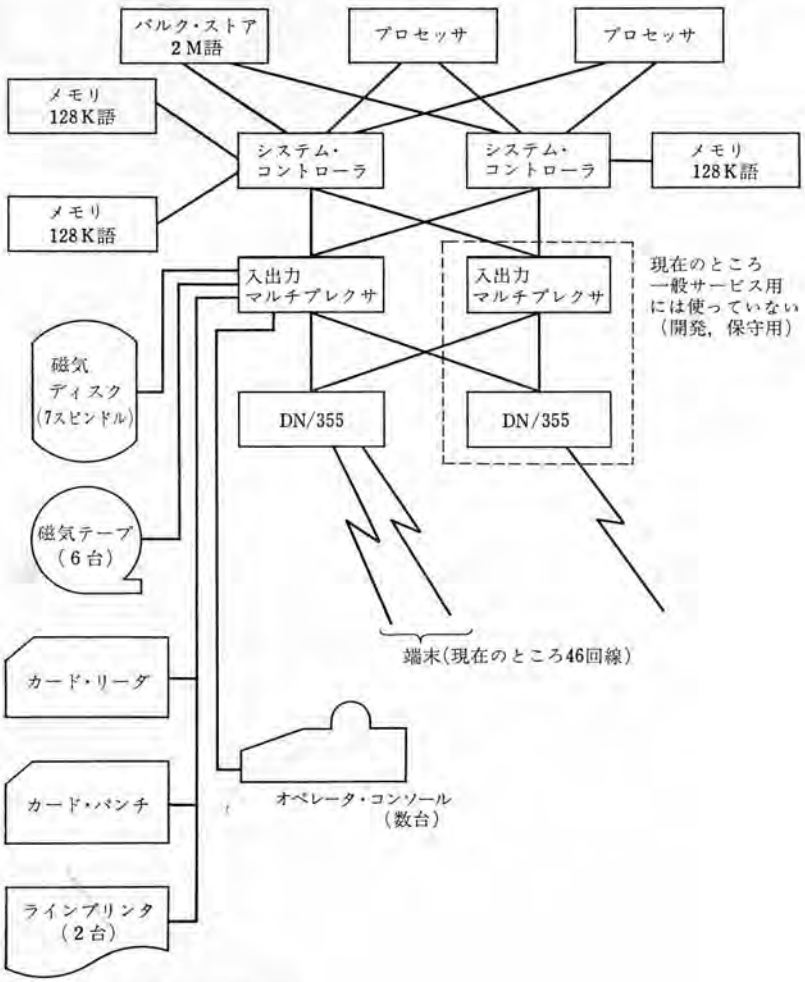

図1 MIT，IPC にある H6180

Schroeder 教授にログインしてもらい，そのあと使うというふうにしてもらいました。

H645 のマルティクスに比べ，システム・メッセージの一部の出力形式や内容などに少し違いはありますが，ほとんど同じです。

図2はマルティクスへのログイン（サービスの開始を要求する）手順です。l (login の省略形) コマンドのあと，パスワードをタイプインします。図2の例では，印字機構を一時オフにできる端末を使用したのでパスワードは印字されていません。この機構のない端末の場合は，文字の重ね打ちでぬりつぶされた行の上にパスワードをタイプします。

図3は，list コマンドでディレクトリの内容を印刷したものです。途中で QUIT という文字が印刷されているのは，これ以上出力が不要と判断し端末のクイット・ボタンを押したからです。

図4はログアウト（サービスの終了を要求する）の例です。

図5はテキスト・エディタ (edm) を使って，新しくセグメント test.ec を生成し，その中に情報を記憶する場合の例です。 Input. というメッセージのあとに記憶したい情報をタイプし，その終了をピリオド (.) で示します。ここで入力モードから編集モード (Edit.) になり，今までにタイプしたものに誤りや，追加したいものがあればこのモードのときに修正，追加ができます。その後，w (write, 書込み) 要求でセグメントに書き込み，q (quit, 停止) 要求でコマンド・レベル（このレベルになったらコマンドがタイプでき，また r (ready) メッセージが印刷されます）に戻します。

図6は，マルティクスを使って作り上げたプログラムの一例で，PL/I で記述されています。先頭の pr(print の省略形) は，テキスト・セグメントの内容を印刷する

```
list -a

Segments= 14, Records= 13.

re      1  count_ch
r wa    1  count_ch.pl1
rewa    1  mipt
rewa    1  loop.pl1
rewa    1  mipt_ada30
rewa    1  loop
r wa    1  Schroeder.con_msgs
re      1  tcall
r wa    1  tcall.pl1
re      1  call_test
r wa    1  call_test.pl1
re      1  int_trace
r wa    1  int_trace.pl1
r wa    0  mailbox
QUIT
r 1716  .119  .306  24
```

図3 ディレクトリの内容の印刷

```
edm test.ec
Segment not found.
Input.
&ready on
fo test.out
rename count_ch count_ch.pl1
pl1 count_ch -tm
count_ch ggg a
count_ch ggg b
count_ch gggc
count_ch ggg d
count_ch ggg e
delete count_ch
delete xyz
print count_ch.pl1
create xyz
pl1 count_ch -ot -tm
count_ch ggg a
count_ch ggg b
count_ch ggg c
count_ch ggg d
count_ch ggg e
addname ggg xxx
deletename xxx
sa count_ch rewa *.*.*
delete count_ch
rename count_ch.pl1 count_ch
co
aprint end exec_com
.
Edit.
1 gggc
count_ch gggc
c /gggc/ c##ggg c
count_ch ggg c
w
q
r 1659  1.534  7.538  431
```

図5 テキスト・エディタの使用例

```
Multics 20.5: MIT, Cambridge, Mass.
Load = 17.0 out of 80.0 units: users = 17
l Schroeder
Password:
You are protected from preemption.
Schroeder CompSys logged in 05/29/73  1708.5 edt Tue from 1050 terminal ".".
Last login 05/24/73  1605.4 edt Thu from 1050 terminal ".".
new/updated help segs: pending_changes, memo_changes, new_dprint, problems.
                       user, user_info_, db_changes
Please report any problems to Consultant.Consult (c.c) via mail.
r 1708  1.360  3.782  57
```

図2 ログイン

```
logout

Schroeder CompSys logged out 05/29/73  1727.7 edt Tue
CPU usage 31 sec, memory usage 162.0 units.
hangup
-
```

図4 ログアウト

コマンドです。count-ch. pl1 がこのプログラムのソース・テキストが記憶されているセグメントです。

5. おわりに

マルティクスの技術的な面については，1965年の FJCC で発表された一連の論文やその後発表された種々の論文，Saltzer 教授が来日されたときの講演集などに述べられています。ここでは紙面の都合もあり，表面的な紹介だけにとどめました。技術的な面の詳細については別の機会にゆずりたいと思います。

百聞は一見にしかずということわざにもあるように，実際にマルティクスを使ってみてその偉大さを感じたとともに身近かにも感じました。

```
pr count_ch.pl1

                    count_ch.pl1        05/29/73  1724.7 edt Tue

count_ch:proc;
dcl cu_$arg_count ext entry (fixed bin);
dcl com_err_ ext entry options(variable);
dcl cu_$arg_ptr ext entry(fixed bin,ptr,fixed bin,fixed bin);
dcl expand_path_ ext entry (ptr,fixed bin,ptr,ptr,fixed bin);
dcl hcs_$initiate_count ext entry (char(*),char(*),char(*),fixed bin(24),
    fixed bin(12),ptr,fixed bin);
dcl ioa_ ext entry options(variable);
dcl ios_$write_ptr ext entry (ptr,fixed bin,fixed bin);
dcl hcs_$terminate_noname entry(ptr,fixed bin);
dcl (null,index,divide,addr,substr) builtin;
dcl (pathptr,segptr) ptr;
dcl pathlen fixed bin;
dcl code fixed bin;
dcl patharea char(256) based;
dcl (first,count,i,j,k,kk,l,ii,jj)fixed bin;
dcl fcount fixed bin init(0);
dcl dir char(168);
dcl ent char(32);
dcl bitcount fixed bin (24);
dcl path char(256);
dcl segarea char(chcount) based;
dcl (chcount,chc) fixed bin;
dcl charg char(1) based;
dcl chargptr ptr;
dcl chlen fixed bin;
          call cu_$arg_ptr (1,pathptr,pathlen,code);
          if code¬= 0 then do;
error:    call com_err_(code,"count_ch","¬/error in path name !");
          return;
          end;
call cu_$arg_ptr(2,chargptr,chlen,code);
if code ¬=0 then do;
          call com_err_(code,"count_ch");
          return;
end;
call expand_path_(pathptr,pathlen,addr(dir),addr(ent),code);
if code ¬=0 then go to error;
initiate:call hcs_$initiate_count(dir,ent,"",bitcount,1,segptr,code);
if segptr=null then do;
          call com_err_(code,"count_ch",ent);
          return;
end;
chcount=divide(bitcount,9,17,0);
kk=1;
chc=chcount;
loop2:    k=index(substr(segptr->segarea,kk),chargptr->charg);
          if k=0 then go to finish;

          fcount=fcount+1;
          kk=kk+k;
          go to loop2;
finish:   call ioa_("¬/there are ¬d occurrences of ""¬a"" in : ¬/",fcount,chargptr->charg);
          call ios_$write_ptr(segptr,0,chc);
          call hcs_$terminate_noname(segptr,code);
end;

r 1727  .967  .672  43
```

図 6 プログラムの例