

Title	センターだより 大阪大学大型計算機センターニュース No.35
Author(s)	
Citation	大阪大学大型計算機センターニュース. 1979, 35, p. 2-54
Version Type	VoR
URL	https://hdl.handle.net/11094/65428
rights	
Note	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

Osaka University

第43回全国共同利用大型計算機センター 運用会議議事要旨

日 時：昭和54年6月21日（木）
場 所：東北大学大型計算機センター
議 事：

1. 報告事項

- (1) 各センターの現況について
- (2) 各研究会（統計項目、データベース、コンピュータ・ネットワーク）について
- (3) センター長会議報告について
- (4) 負担金検討小委員会について
- (5) 事務長会議報告について

2. 審議事項

- (1) 負担金の一部改訂について

北海道大学、名古屋大学、京都大学、九州大学及び東北大学から、負担金の一部改訂についてそれぞれ提案説明があり、また負担金検討小委員会からこれに関する検討結果について報告があり、いずれも原案どおり了承された。

- (2) 新データ網（DDX）通信回線料金のユーザ課金について

東京大学石田助教授から、「新データ網（DDX）通信回線料金のユーザ課金について」の提案主旨説明があり、種々意見の交換の結果、東京大学、京都大学の関係者及びコンピュータ・ネットワーク研究会で資料を作成のうえ、次回運用会議に諮ることになった。

なお、コンピュータ・ネットワーク研究会からの資料に基づき負担金検討小委員会でも検討することになった。

- (3) 全国共同利用大型計算機センター研究開発連合発表講演会実施要領（案）について

名古屋大学吉田助教授から、研究開発連合発表講演会実施要領（案）について説明があり、種々意見の交換の結果、原案どおり承認された。

3. その他

- (1) 統計項目の利用方法について

統計項目研究会から、今後の統計項目の利用方法について提案があり、種々検討の結果、統計項目研究会で担当し、利用方法を検討することになった。

昭和54年度優秀情報処理システム の表彰を受ける

一般市民に情報化に対する正しい認識と理解を深めてもらおうと、政府が民間諸団体と協力して昭和47年から行っている「情報化週間」が、今年も10月第1週に実施され、同記会式典において当センターシステムが、情報化週間推進会議議長から昭和54年度優秀情報処理システムとして表彰を受けました。

表彰理由は、本格的な図形処理システムや学術データベースを、一般の研究者が簡単に構築できるような体系づけを行ったシステムであり、学術研究におけるコンピュータの利用可能性を拡げ、研究活動の効率化に寄与したというもので、センターとしても非常に喜ばしいことです。

今後も、この表彰に恥じないよう学内外の利用者の方々の御協力を得て、使い易く、そしていろいろに使える計算機システムと利用体制の整備に努力したいと思います。

第1回全国共同利用大型計算機センター 研究開発連合発表講演会の開催について

各大型計算機センターでの研究開発の成果は内容が優れていて、かつ、センター関係者にとつて貴重なものが多数あります。これらをまとめてセンター共通の場で発表・討論し、併せて、その成果を研究開発論文集として刊行することを目的として、次のとおり第1回の研究開発連合発表講演会を開催することになりました。御関心ある方々多数の御来聴を歓迎します。

日 時 昭和54年11月30日(金) 9時20分～17時
場 所 名古屋大学工学部7号館 711講義室

プログラム：

午前(9:30～11:40)

- (1) 「データ作成サービスへのOCRの導入」

北海道大学大型計算機センター 前谷強志

- (2) 「プレビュー・システム」

東北大学大型計算機センター 武田敏夫

- (3) 「写植機を用いたオンライン論文清書システム」

東京大学大型計算機センター 野本征子

午後前半(13:00～14:40)

- (4) 「データセットのカタログ簿へのアクセスの解析及び効率改善」

京都大学大型計算機センター 金澤正憲

- (5) 「大阪大学大型計算機センター稼動状況表からの一考察」

大阪大学大型計算機センター 多喜正城

休 憩

午後後半(14:50～17:00)

- (6) 「磁気テープ・ジョブ・サービスの新方式について」

東京大学大型計算機センター 長谷部紀元

- (7) 「MTスプール・システムの開発」

名古屋大学大型計算機センター 長谷川明生

- (8) 「スプリングシステムによるプロッタ出力」

九州大学大型計算機センター 二村祥一

(庶務掛)

利用負担金の改訂について

すでに速報№65（S. 54. 9. 21付）でお知らせしましたが、利用負担金を昭和54年10月1日から別表のとおり改訂いたしました。今回の負担金一部引き下げは、今後の利用の増加を前提としていますので、これを機会に当センターにおける計算機利用がなお一層活発になることを期待します。改訂部分は次のとおりです。

1. 演算負担額

バッチ処理のCPUタイム1秒につき5円を3.5円に引き下げました。

バッチ処理のCPUタイムが900秒を超える部分については、1秒につき2円を360秒を超える部分については、1秒につき1.5円に引き下げました。

2. ファイルの使用負担額

月ごとに1リンクにつき10円を8円に引き下げました。

別表

区	分	負 担 額
基本負担額	バ ッ チ 処 理	計算依頼1件につき50円とする。ただし、リモートバッチ処理については、計算依頼1件につき20円とする。
	T S S 処 理	計算依頼1件につき20円
演算負担額	バ ッ チ 処 理	CPUタイム1秒につき3.5円とする。ただし、計算依頼1件のうち360秒を超える部分については、1秒につき1.5円とする。
	T S S 処 理	CPUタイム1秒につき3.5円
出力負担額	バ ッ チ 処 理	1記録につき0.02円とする。ただし、センターに出力した場合は、LP用紙1頁につき3円、カード1枚につき1円を加算したものとする。
ファイル使用負担額	バ ッ チ 処 理	月ごとに、1リンクにつき8円とする。ただし、申請に基づく割当量が10リンクまでの部分については、無料とする。
	T S S 処 理	

区	分	負 担 額
使 用 負 担 額	ド ラ フ タ ー	ドラフター使用の計算依頼1件につき400円
	デジタルプロッター	デジタルプロッター使用の計算依頼1件につき150円
	3次元グラフィック	ハードコピー用紙1枚につき200円

備 考

1. TSS処理においては、会話開始（LOG-ON手順）から会話終了（BYEコマンド又はLOG-OFF手順）又はサービス中断の直前のコマンドまでを計算依頼1件とする。
2. CPUタイムに1秒未満の端数が生じたときは、計算依頼1件ごとに、これを1秒に切り上げるものとする。
3. 演算負担額及び出力負担額に1円未満の端数が生じたときは、計算依頼1件ごとに、これを1円に切り上げるものとする。
4. 会話型リモートバッチの場合の基本負担額は、TSS処理及びリモートバッチ処理に係るものを加算した額とする。
5. 大阪大学豊中地区及び中之島地区データステーションからのリモートバッチ処理に関しては、表中出力負担額のただし書を「大阪大学豊中地区及び中之島地区データステーションに出力した場合は、1記録につき0.06円を加算した額とする。」と読み替え、センター処理と同様に扱うものとする。
6. 1リンクとは、3,840語の情報量を表すものとする。

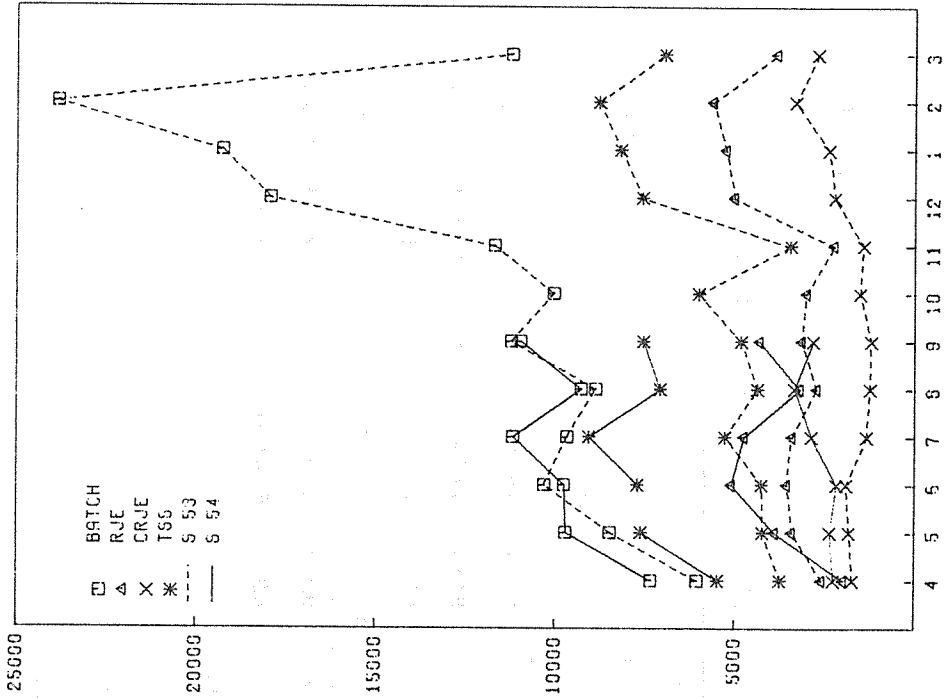
昭和 54 年 度 処 理 状 況

(ACOS 77 NEAC SYSTEM 900)

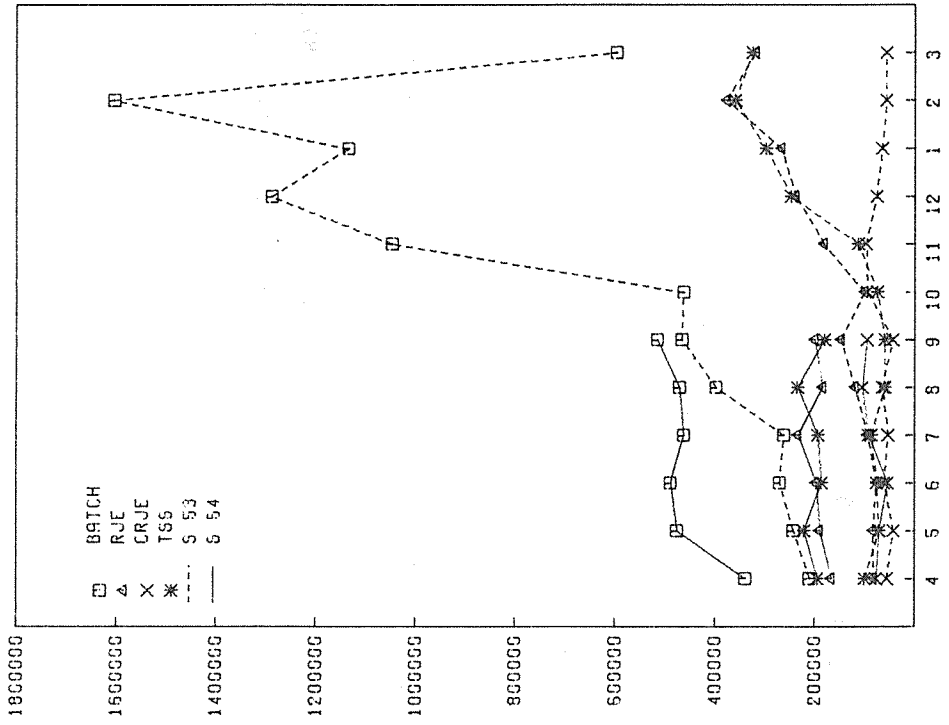
項 目 月	パ ッ チ			処 理			タ イ ム シ ェ ア リ ン グ 処 理			合 計		
	ローカルパッチ		リモートパッチ	会話型リモートパッチ		専用回線	交換回線			件数	演算時間 (秒)	
	件数	演算時間 (秒)	件数	演算時間 (秒)	件数	演算時間 (秒)	件数	演算時間 (秒)				
4	7,320	338,799	1,967	165,005	2,247	77,118	4,254	152,571	1,208	41,999	16,996	775,492
5	9,695	475,432	3,875	189,669	2,357	71,316	6,038	188,596	1,569	32,172	23,534	957,185
6	9,756	488,393	5,091	196,307	2,174	53,559	6,158	157,322	1,538	28,786	24,717	924,367
7	11,160	462,275	4,731	233,341	2,868	92,243	7,282	153,952	1,760	38,670	27,801	980,481
8	9,260	469,955	3,177	184,197	3,333	103,891	5,845	211,644	1,215	22,200	22,830	991,887
9	10,948	514,656	4,299	199,019	2,820	94,708	6,009	156,143	1,526	24,119	25,602	988,645
10												
11												
12												
1												
2												
3												
合計	58,139	2,749,510	23,140	1,167,538	15,799	492,835	35,586	1,020,228	8,816	187,946	141,480	5,618,057

昭和54年度処理状況

処理件数の推移



演算時間の推移



「速報」及び「お知らせ」の集録

＜速報№.63(54.7.26)より＞

I. 局線用1200bps交換回線TSSのサービスについて

速報№.58でお知らせしていました局線用の1200bps交換回線TSSサービスが遅れていましたが、6月25日より局線1回線(876-5001)が開設されましたのでお知らせいたします。

1200bps交換回線端末を設置するには下記の申請手続きならびに端末機等の準備が必要となります。

記

(1) 端末設置申請手続

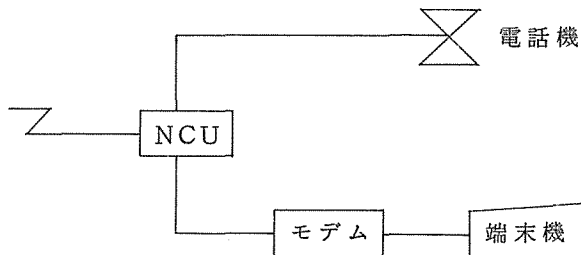
1200bps交換回線TSS端末の設置においては、次の申請書を提出する必要がある。

- (ア) 端局設置申請書(大型計算機センター長宛)
- (イ) 加入電話接続機器自営届書兼自営用品認定申請書(電々公社宛)

(注)吹田構内の電話機に接続する場合には、電々公社宛の申請は不要

(2) 端末機等の設置

300bps交換回線TSSを利用する場合には、端末機と音響カプラを用いることにより利用できましたが、1200bps交換回線TSSの場合には構成図のような機器を構成する必要があります。



1200bps交換回線TSS端末の構成図

NCU………端末側で使用する網制御装置(NCU)として次のいずれかのメーカーの装置を設置。

日本電気 DATA XMA 101 沖電気 BS-2001

モデム……………VADIC社 3405型モデムを使用

端末機……………タイプライタ型またはディスプレイ型の1200 bps で動作する端末

(業務掛)

II. 交換回線端末の展示会開催について

当センターのシステムに接続できる交換回線端末には、各種の機器があり、特に最近低価格の端末機器が出まわっています。

センターでは、今後TSS端末の設置を計画されている利用者のために、下記のとおり端末展示会を行います。当日は各機種の端末を音響カプラを通して、デモンストレーションを行いますので、TSS端末の設置を計画中またはTSS端末に対して興味のある利用者は、ぜひご覧くださいようご案内いたします。

記

日 時 昭和54年9月29日(土) 10時～15時

場 所 センター大会議室(3階)

(業務掛)

III. APL言語について

ACOS-6 APL言語を導入しました。

概要

ACOS-6 APL言語はタイムシェアリングシステムの下で動作するインタプリタである。APL言語は、Dr: Kenneth E. Iversonによって、アルゴリズムの理論の記述のために数学的な記号法として導入された。

APLの特性は、他の多くの言語に比べて異なったいくつかの特徴によって述べることができる。

第一は、それが一方的な指令に基づくものではなく、会話的なものであるということである。すなわち、早く、簡潔であり、有効な情報が得られ、かつ、非常に使用することが面白い。

次にそれは、簡単な上に強力であるということである。すなわち、学習が簡単で、平易であり、それでいて難解な問題に取り組むことができる。APL演算子が強力であるので、他の言語では扱いにくくかつ長くなるような多くの問題もAPLでは、直接的でかつ短くなる。

APLを呼び出した後、利用者は評価してほしい式をタイプする、APLインタプリタは計算を行い、結果を印字し、新しい行の入力を持つ、式の結果はまた変数に代入することもでき、後でその値を引用することができる。さらに、入力行を蓄積しておき、名前を与えて、後に、それ

らがあたかも端末から再投入されたかのように、呼び出し、解釈することも可能である。

参照 ACOS-6 APL説明書

(システム管理掛)

IV. クラスタ分析について

ACOS-6アプリケーションシステム CLUSTER-6を導入しました。

概要

CLUSTER-6は、個体(対象)の集合を個体間の測度(相関または距離)によって階層構造にクラスタ(群)化するためのアプリケーションプログラムである。

クラスタ分析とは、個体の集合をいくつかのクラスタ(群)に分類した場合に使われる統計的手法である。

特徴

1. クラスタ分析の入力となるデータは、連続型変量、離散型変量のいずれでも処理できる。
2. 観測値からの分類の測度計算として、19種類準備されている。
3. 分類構造を構成する樹形図を出力できる。
4. 測度行列からの分析ができる
5. 統計解析システム(STAPAC-6)のファイルからデータの入力ができる。

JCLの説明

```
1      8          16
$      SNUMB
$      JOB
$      CPROC      CLUST[, , CPU, mmmk, -nk, ppp]
          コントロール カード
          データ
$      ENDJOB
***EOF
```

[, , CPU, mmmk, -nk, ppp] ---省略可能
指定するときは\$LIMIT カードに準ずる。

(システム管理掛)

V. PASCAL言語について

ACOS-6 PASCAL言語を導入しました。

概 要

ACOS-6 PASCAL コンパイラは、タイムシェアリング、バッチの両モードで動作するコンパイラであり、その目的プログラムもまたタイムシェアリング、バッチの両モードで動作する、言語仕様はK・JensenとN・Wirthの共著による「PASCAL User Manual and Report」(第2版)に準拠し、ACOS-6のサービスが有効に受けられるように、いくつかの拡張が加えられている、言語仕様の拡張に当たっては原著者によるPASCAL言語の設計理念を踏襲しながらも、オペレーティングシステムと直接インタフェースをもてるよう設計することによってACOS-6との親和性を高めるよう配慮した、すなわち、PASCALの目的プログラムファイル、利用者サブルーチンライブラリファイル、実行形式ファイル等は、FORTRANやCOBOLのそれらと同一の形式をもっていて、ACOS-6のJCLやTSSと直接インタフェースをもつことができる、ACOS-6 PASCALで拡張されている主な機能は次のとおりである。

- 外部手続きおよび関数の呼び出し、すなわち、PASCALで記述されたルーチンのライブラリ化機能、これによってPASCALで記述された共通性の高いルーチンをライブラリとして登録しておくことができ、高度なモジュラプログラミングが可能となる。
- 動的ファイルアクセス機能、これによってプログラムの外に存在する物理的ファイルとPASCALで宣言した論理的ファイルの結合、作成およびその解除がプログラムで動的に行える。
- FORTRANで記述されているルーチンとの結合機能
- 他のタイムシェアリングサブシステムの呼び出し機能

これらの拡張機能に加えて、PASCAL言語が本来もっている性質によって、信頼性の高い、良い構造をもつプログラムをより短い期間で作成することができる。

参 照

ACOS-6 プログラム管理 PASCAL説明書

(システム管理掛)

VI. プログラム・ライブラリーの移行登録について

次に示す16個のサブプログラムのNEAC2200-700からACOS900への書き換え作業が完了しました。これらのプログラムは8月1日からセンター・ライブラリーに登録され、使用可能となります。

各プログラムの使用法は従来と同一ですから、センター・ニュース№19, PP. 16~52を御覧下さい。ただし、センター・ニュース№19, P. 29のサブルーチンUBNCGDの引数にISPが脱落しています。正しくはUBNCGSと同一ですから、御注意下さい。なお、これらのプログラムのランクはすべて2です。

移行登録されたサブプログラム
(共役勾配法による連立一次方程式)

SUBROUTINE 名		計 算 内 容	作成, 改訂者
定) 単精度	倍精度		
SQACGS	SQACGD	連立一次方程式 (正値対称正方行列)	長岡技術科学大学 林 正
USQCGS	USQCGD	" (非対称正方行列)	"
BANCGS	BANCGD	" (正値対称帯行列)	"
UBNCGS	UBNCGD	" (非対称帯行列)	"
CODCGS	CODCGD	" (正値対称行列, コードマッチング法)	"
UCOCGS	UCOCGD	" (非対称行列, コードマッチング法)	"
BLOCGS	BLOCGD	" (正値対称行列, ブロックコードマッチング法)	"
UBLCGS	UBLCGD	" (非対称行列, ブロックコードマッチング法)	"

注) 収束判定常数EPSが大きい場合 (たとえば 10^{-3})を除いて、単精度のものよりも倍精度の方が実行時間が短いことが多い。

(研究開発部)

Ⅶ. センターニュース№33の訂正について

P. 61とP. 71の「BROYDENの方法による非線形連立一次方程式」を「BROYDENの方法による非線形連立方程式」に訂正いたします。

P. 73の引数ILLの説明で「3000」を「30000」に訂正いたします。

(研究開発部)

Ⅷ. 利用の手引き〔基本編〕の発行について(再掲載)……省略

Ⅸ. ファイル使用説明書・オンライン関係利用説明書について(再掲載)……省略

Ⅹ. 出力編集サブシステムについて(再掲載)……省略

<速報№64(54.8.21)より>

I. 端末展示会開催について(再掲載)……省略

Ⅱ. 三次元グラフィック・ディスプレイのサービスについて

三次元グラフィック・ディスプレイのサービスを8月1日から始めました。

1) 運用方法

三次元グラフィック・ディスプレイは、S-900にオンラインで接続された汎用周辺処理装置によって制御されるため、一つの周辺装置と考えて下さい。

利用形態はドラフター、X-Yプロッター等同様、オープン利用方式を採用しています。利用者は図形処理装置予約簿に予め使用時間帯を確保した後、図形処理装置室に設置されているカード・リーダーからジョブを投入して下さい。三次元グラフィック・ディスプレイとX-Yプロッターは同時動作ができませんので注意して下さい。

2) ジョブ制御言語の説明

1	8	16
\$	SNUMB	
\$	JOB	課題番号 \$ パスワード, G
\$	LIBRARY	GB
\$	FORTRAN	BIN

ソース プログラム

```

$          CPROC      { GDB 0 }
                      { GDB 1 }
                      { GDB 2 }

$          LIMIT
          データ

$          ENDJOB

*** EOF

```

(注意事項)

1. JOB区分は必ずGを指定します。
2. LIBRARYはGBのパラメータを指定します。
3. FORTRANはBINのパラメータを指定します。
4. CPROC GDB 0は三次元グラフィック・ディスプレイに初期画面を表示し、実行可能な状態にします。

CPROC GDB 1は三次元グラフィック・ディスプレイに初期画面を表示しないで実行します。

CPROC GDB 2は三次元グラフィック・ディスプレイを稼働させないで、コンパイルエラー、リンクエラー等のチェックを行う。ジョブ区分「G」以外で実行すること。

3) 操作方法

ジョブ投入後、数分以内に初期画面として、SNUMB名と実行開始時刻を表示しファンクション・スイッチ0と15が点灯、0を押下げると実行、15を押し下げると強制終了します。

(注意事項)

ジョブがコンパイルエラー、リンクエラー等のとき、三次元グラフィック・ディスプレイは稼働しません、MDC、VIDEO等で確認して下さい。

異常終了した場合はジョブ受付に連絡して下さい。

三次元グラフィック・ディスプレイの利用にあたっては、原則として説明会を受講して下さい。

(システム管理掛)

Ⅲ．センター・ライブラリーの仕様変更について

10月1日から、下記のプログラムの仕様が変更になります。新しい仕様については、当分の間、プログラム相談室（内線 2827）または研究開発部（大中、内線 2836）までお問合せ下さい。

Q2/M-TYPE	誤差評価の可能な多重精度演算（M型）	} 仕様変更	仕様変更・初等関数の追加
Q2/S-TYPE	誤差評価の可能な多重精度演算（S型）		
Q2/E-TYPE	誤差評価の可能な多重精度演算（E型）		
Q2/I-TYPE	誤差評価の可能な多重精度演算（I型）		

（研究開発部）

Ⅳ．センターニュース№33の訂正について（再掲載）……省略

Ⅴ．利用者講習会の開催について

1．データベースの利用法について

- 日 時 9月21日（金）13:30～15:30
- 場 所 本センター大会議室（3階）
- 受講資格 FORTRANまたはCOBOLの習熟者で、データベースの概要が理解でき、今後、本センターでデータベースを利用しようとする方。
- 募集人員 30名
- 申込期間 9月4日（火）～9月14日（金）
- 申 込 先 本センター庶務掛（06-877-5111 内線 2812）
- 〔内容〕 情報を必要に応じて貯えておき、これらを効率的に検索をおこなうデータベースマネジメントシステム（INQ）によるデータベースの利用手順について解説する。

（庶務掛）

Ⅵ．後期プログラム相談員の募集について

センター内のプログラム相談室では、毎日一定時間内にプログラム上の問題やセンター利用全般についての相談に応じています。十分なプログラム相談を行うには、毎年相当数のプログラム相談員が必要です。

センターでは、54年度後期のプログラム相談員を下記の要領で募集致しますので、御協力いただける方、われと思わん方は是非御応募下さるようお願い致します。

記

資格	当センターの課題番号を有する者（原則として大学院前期課程の者は除く）
期間	昭和54年10月～昭和55年3月
内容	利用者からのプログラム相談に応じる（当センタープログラム相談室で2週間に1回2時間程度）
主な待遇	1. 負担金の一定額免除 3万円（毎週1回担当の場合は6万円）相当 2. マニュアル類を一定限度まで無償配布 3. プログラム相談室の簡易ターミナルの利用及び資料文献等の利用 4. センター内の研修会に参加できます。 5. その他

応募方法及び応募先

応募用紙が当センターにありますので、下記宛提出願います。

大阪大学大型計算機センター庶務掛

切 昭和54年9月29日（土）

なお、不明な点がございましたらセンター庶務掛（TEL 06-877-5111 内線 2806）までお問い合わせ下さい。

（庶務掛）

<速報No.65(54.9.21)より>

I. 入出力棟新営工事について

このたび、本年度予算額で、当センター裏側に入出力棟の建設が認められました。工事中は何かと御迷惑をおかけすることと思いますが、特に抗打ち等による震動によりやむを得ず、計算機のサービスを中止しなければならないような事態になることが予想されますので御了承願います。

また、工事のため駐車場の使用も不可能になると思われます。

なお、くわしい日程等は決まり次第、速報・掲示等であらためてお知らせします。

II. 出力編集サブシステムのジョブ・キャンセル・コマンドの追加について

出力編集サブシステムのENTER SEQ. NO. ?レベルで、デマンド・ファイルに登録されているジョブをキャンセル（出力取消）するコマンド JCAN を追加しました。パラメータとして、ALLあるいはSEQ. NO. nnnを指定します。

パラメータ ALL : 全てのジョブのキャンセルを行う。

nnn : SEQ. NO. nnnに対応するジョブのキャンセルを行う。

使用例 1. 受付番号 0010C のジョブのキャンセルを行う。

SYSTEM? \$DEDIT

SEQ. NO. SNUMB JOB IN DATE

01 0009C 790901 ~

02 0010C 790901 ~

ENTER SEQ. NO. ?JCAN 2

使用例 2. 全てのジョブのキャンセルを行う。

ENTER SEQ. NO. ?JCAN ALL

(システム管理掛)

Ⅲ. ジョブ制御言語の手引きの発行について

ジョブ制御言語の手引きを発行しました。この手引きは、バッチサービスにおけるジョブ制御言語を述べたものです。十分御活用願います。

なお、本手引きは、当センター1階一般受付においてありますので、申出てください。

(システム管理掛・共同利用掛)

Ⅳ. 利用負担金の改訂について

当センターニュース5頁に同じ内容掲載のため省略します。

Ⅴ. 端末展示会開催について(再掲載) ……省略

Ⅵ. 「社会科学とコンピュータ」に関する公開講演会について

このたび「社会科学とコンピュータ」というテーマで下記により講演会を開催することになりました。

関心のある方々多数参加いただきますよう御案内いたします。

記

日 時 : 昭和54年10月1日(月) 13:00 ~ 17:00

場 所 : 当センター大会議室

定 員 : 50名

申 込 : 出席希望の方は会場の都合により次まで御連絡下さい。

TEL 06-877-5111 内線2806

時 間	演 題 及 び 講 師
13:00	開講にあたって
13:15 } 14:50	<p>「経済統計データベースの設計と計量経済モデリング・システム (EMS)」</p> <p>(EMSは当大型計算機センターで利用可能なソフトウェアである。) (EMSを中心とした経済統計データベースとその関連事項について述べる。)</p> <p>慶応義塾大学工学部 教 授 森 敬 氏</p>
15:00 } 16:30	<p>「都市分析におけるコンピュータの利用」</p> <p>(都市における経済及び人口の動学的分析によるコンピュータを用いた「都市成長の分析」について述べる。)</p> <p>筑波大学社会工学系 教 授 坂 下 昇 氏</p>
16:30 } 17:00	質 疑 応 答

(庶 務 掛)

VII. 利用者講習会の開催について

1. ACOS900のジョブ制御言語について

- 日 時 10月15日(月) 13:30~16:30
- 場 所 本センター大会議室(3階)
- 受講資格 FORTRANプログラミング、あるいは計算機利用に関する基礎知識を有し、今後、ACOSのバッチ処理を利用しようとする初心者の方。
- 募集人員 60名
- 申込期間 10月1日(月)~10月9日(火)
- 申 込 先 本センター庶務掛(06-877-5111 内線2812)

〔内容〕 ジョブ制御言語について解説する。

2. 磁気テープの利用について

- 日 時 10月16日(火) 13:30~16:30
 - 場 所 本センター大会議室(3階)
 - 受講資格 初めて磁気テープを利用する方
 - 募集人員 60名
 - 申込期間 10月1日(月)~10月9日(火)
 - 申 込 先 本センター庶務掛(06-877-5111 内線2812)
- 〔内容〕 MTコンバージョンの解説
FORTRANによる磁気テープの利用

3. バッチからのファイル利用法について

- 日 時 10月17日(水) 13:30~16:30
 - 場 所 本センター大会議室(3階)
 - 受講資格 バッチ処理により初めてファイルを利用する方
 - 募集人員 60名
 - 申込期間 10月1日(月)~10月9日(火)
 - 申 込 先 本センター庶務掛(06-877-5111 内線2812)
- 〔内容〕 バッチ処理におけるファイル作成、修正、削除等の利用法を解説する。

<速報No.66(54.11.5)より>

I. マルチTSSの運用について

マルチTSSとは、TSSの端末利用者が大幅に増加し、レスポンスタイムが悪化していくことに対処するため、従来は1つであったTSSを最大4個まで動作させ、多数の端末サービスをしながらか適正なレスポンスタイムを保証するように開発されたものです。

11月より運用を開始いたしますが、回線障害や端末の誤操作等により、端末がTSSと切り離された場合の再コネクト(中断した状態から続行する場合)の方法が少し変更になりますので御注意ください。その他の操作法は全く変わりません。

再コネクトする場合、以前に接続されていたTSSと端末を接続する必要があり、どのTSSに接続されていたかは、ログオンメッセージで接続されたTSSプログラム名(TS1~TS4)が利用者に知らされます。再コネクト要求時に接続されていたTSSプログラム名を指定します。

ログオンメッセージ

ACOS-6 TSn(R5.2) ON mm/dd/yy AT hh.hhh CHANNEL nnnn

再コネクト要求

\$\$\$CON, TS_n

再コネクト要求の場合、簡易コネクトを利用すると、同一TSSに接続できない場合があります。また、TSSプログラム名を誤りますとTSSと接続できません。再コネクトをする必要がない場合は、簡易コネクトで接続できます。

再コネクト使用例

\$\$\$CON, TSS ……TSSと接続(簡易コネクト可)

ACOS-6 TS2(R5.2) ON 10/01/79 AT 10.000 CHANNEL 2223

USER ID-XXXXXXXXXX

PASSWORD--

SYSTEM?

← 端末障害発生

\$\$\$CON, TS2 …… TS2と接続(簡易コネクト不可)

ACOS-6 TS2(R5.2) ON 10/01/79 AT 11.031 CHANNEL 2701

USER ID-XXXXXXXXXX

PASSWORD--

SYSTEM? CONNECT 2223 ……コネクトコマンド

}

以前の処理が続行されている。

(業務掛)

II. リスト処理言語について

LISP/1.5-6(BATCH版及びTSS版)システムが新たに導入されました。

概要(BATCH版)

LISP(LIST PROCESSOR)言語はリスト処理言語として1960年に John Mc Carthy によってその概念が提案され、1962年にMITでLISP1.5としてIBM7090用に開発されました。

LISP/1.5-6は、LISP1.5を母体にして改良が加えられたものであり、またほとんどの機能を包含し、互換性を保っています。詳細についてはACOS-6 LISP1.5プログラミング説明書を参照されたい。

(ジョブ制御言語の説明)

```
1カラム      8          16
$           SNUMB
$           JOB
$           CPROC      LSP15 [, , tt , , , ppp] …… (1)
```

/CNTRL

(LISP/1.5-6の言語

/EOF

```
$           ENDJOB
```

***EOF

(1)の\$ CPROC LSP15文の[, , tt , , , ppp]は省略可能

tt ……CPU時間の指定

ppp ……出力行数の指定

概要(TSS版)

LISP/1.5-6(BATCH版)との互換性を保ちつつ、会話形式の処理によって問題解決が容易になるように拡張を行なったものである。詳細については、ACOS-6 LISP1.5プログラミング説明書を参照されたい。

(使用例)

```
*SYSTEM? L 1.5 <… { 入力するとLISP/1.5-6システムが呼び出さ  
LISP/1.5-6           } れる。  
=コマンド <… LISPコマンドを入力する。
```

```
=DEFINE((
```

```
=(FACTORIAL (LAMBDA (N)
```

```
=(COND((ZEROP N) 1)
```

```
=(T(TIMES N (FACTORIAL (SUB1 N))))
```

```
=)) ) )
```

```
END OF EVALQUOTE, VALUE IS. .
```

```
(FACTORIAL)
```

```
CPTIME IS. . 2MSEC
```

```
PUSH-DOWN. . 1CELLS
```

```
=FACTORIAL (3)
```

```

END OF EVALQUOTE, VALUE IS...
6
CPTIME IS.      3MSEC
PUSH-DOWN      33CELLS
= (C)R ..... 復帰を入力
END OF LISP RUN

```

(システム管理掛)

Ⅲ. 複合構造解析システム (ISAP-6) について

複合構造解析システム (ISAP-6) が新たに導入されました。

概要

複合構造解析システム (ISAP-6) は中規模な構造モデルを手軽に精度良く解析することを目的として開発されたシステムである。

(ジョブ制御言語の構成)

```

1カラム      8          16
$           SNUMB
$           JOB
$           CPROC      ISAP[, , tt , , , ppp] ..... (1)
           {
           ISAP-6のデータ群
$           ENDJOB
***EOF

```

(1)の\$ CPROC ISAP文の[, , tt , , , ppp]は省略可能

ttCPU時間

ppp出力行数

参照マニュアル

- 複合構造解析システム説明書 (ISAP-6) (FXI52-3)
- 複合構造解析システム説明書 (エラーメッセージ編) (DXI59-2)
- 複合構造解析説明書 (ISA-P例題編) (FXI54-2)

(システム管理掛)

Ⅳ．建築／土木／構造パッケージライブラリー（ICEP-6）について

建築／土木／構造パッケージライブラリーICEP-6が新たに導入されました。

建築／土木／構造パッケージライブラリーICEP-6（Integrated Civil Engineering Package-6）に含まれている、全てのパッケージの利用が可能になりましたので、それらについての操作法を説明します。

（ジョブ制御言語の構成）

```
1カラム      8          16
$           SNUMB
$           JOB
$           CPROC      ICEP/FRAN[, , tt , , , ppp] ..(1)
```

データ カード群

```
$           ENDJOB
***EOF
```

（ジョブ制御言語の説明）

（1）の\$ CPROC ICEP文は使用するパッケージによって下記のカードと置換えて下さい。

○二次元、三次元骨組構造解析（説明書コード：FXI02）

```
$           CPROC      ICEP/FRAN[, , tt , , , ppp]
```

○平面骨組解析（FXI03）

```
$           CPROC      ICEP/STAF1[, , tt , , , ppp]
```

○格子骨組解析（FXI04）

```
$           CPROC      ICEP/STAF2[, , tt , , , ppp]
```

○平面トラス解析（FXI05）

```
$           CPROC      ICEP/STAF3[, , tt , , , ppp]
```

○立体トラス解析（FXI06）

```
$           CPROC      ICEP/STAF4[, , tt , , , ppp]
```

○平面応力ひずみ解析（FXI11）

```
$           CPROC      ICEP/COSM1[, , tt , , , ppp]
```

○板の曲げ解析（FXI12）

```
$           CPROC      ICEP/STAM2[, , tt , , , ppp]
```

○軸対称回転体解析（FXI13）

```
$           CPROC      ICEP/COSM3[, , tt , , , ppp]
```


○軸対称シェル解析 (FXI14)

```
$ CPROC ICEP/COSM4 [, , tt , , , ppp ]
```

○ソリッド解析の場合 (FXI15)

```
$ CPROC ICEP/COSM5 [, , tt , , , ppp ]
```

○自動メッシュ作成 (FXI16)

```
$ CPROC ICEP/MESGN [, , tt , , , ppp ]
```

○直接応答解析 (FXI21)

```
$ CPROC ICEP/RESP [, , tt , , , ppp ]
```

○二次元動的解析 (FXI22)

```
$ CPROC ICEP/VIBR2 [, , tt , , , ppp ]
```

○三次元動的解析 (FXI23)

```
$ CPROC ICEP/VIBR3 [, , tt , , , ppp ]
```

○熱現象解析 (FXI31)

```
$ CPROC ICEP/TOSS [, , tt , , , ppp ]
```

○熱伝導解析 (FXI33)

```
$ CPROC ICEP/HEAT [, , tt , , , ppp ]
```

○円弧滑り安定計算 (FXI42)

```
$ CPROC ICEP/SLOPE [, , tt , , , ppp ]
```

上記の [, , tt , , , ppp] は省略可能

tt …… CPU時間の指定

ppp …… 出力行数の指定

(システム管理掛)

V. 連続型シミュレーション言語 (CSPL-6) の使用変更について

連続型シミュレーション言語 (CSPL-6) が Version 2.0 に機能強化されました。

それに伴い、ジョブ制御言語の変更をお知らせします。詳細については、ACOS-6 連続型シミュレーション言語説明書 (FXE22-2) を参照されたい。

(ジョブ制御言語の構成)

```
1カラム      8          16
```

```
$          SNUMB
```

```
$          JOB
```

```
$          PARAM  ON 1, ON 3  …………… (1)
```

\$ CPROC CSPL[, , tt , , , ppp] (2)

{ CSPL モデル

\$ ENDJOB

***EOF

(ジョブ制御言語の説明)

(1)の\$ PARAM文のパラメータ指定により次のような制御ができます。

パラメータ	意	味
ON 1	FORTRANコンパイル時にXREFをとる	
ON 2	FORTRANコンパイル時にLSTOUをとる	
ON 3	FORTRANコンパイルリストを出力しない	
ON 4	ユーザー・ライブラリーを使用する(ファイルコード:UL)	
ON 5	CSPLモデルの実行時のローディングマップをとる	
ON 6	使用禁止	
DUMP	ダンプリストをとる	

(2)の\$ CPROC CSPL文の[, , tt , , , ppp]は省略可能。

ttCPU時間の指定

ppp出力行数の指定

(システム管理掛)

VI. プログラム・ライブラリーの追加登録について

名古屋大学より譲渡されたプログラムが、センター・ライブラリーとして、11月5日より使用可能となりました。マニュアルは、ただ今整備中ですので、プログラム相想室でご覧下さい。

D1 AQDCC, AQDCCD クレンショウ・カーチス法による自動積分(閉じた種分公式)

D1 AQDCO, AQDCOD クレンショウ・カーチス法による自動積分(開いた積分公式)

D6 FCOSM, FCOSMD 中点公式に基づく高速cosine変換

D6 FSINM, FSINMD 中点公式に基づく高速sine変換

D6 TRIGQP, TRIGQD 二進逆順に並べられた三角関数表

D 6	FCOST, FCOSTD	台形公式に基づく高速 sine 変換
D 6	FSINT, FSINTD	台形公式に基づく高速 sine 変換
D 6	FCOSC, FCOSCD	閉区間 $[0, \pi]$ で与えられた偶関数の cosine 級数展開
D 6	FCOSO, FCOSOD	開区間 $(0, \pi)$ で与えられた偶関数の cosine 級数展開
D 6	FSINO, FSINOD	開区間 $(0, \pi)$ で与えられた奇関数の sine 級数展開
D 6	FCHB1, FCHB1D	第 1 種チェビシェフ多項式による関数のフーリエ展開
D 6	FCHB2, FCHB2D	第 1 種チェビシェフ多項式による関数のフーリエ展開
D 6	FCHB3, FCHB3D	ずらしチェビシェフ多項式による関数のフーリエ展開
D 6	FCHBO, FCHBOD	第 1 種チェビシェフ多項式による開区間上で与えられた関数のフーリエ展開
D 6	VCOS, V COSD	cosine 級数の値
D 6	VSIN, VSIND	sine 級数の値
D 6	VCHB1, VCHB1D	チェビシェフ級数の値
D 6	DCHB1, DCHB1D	チェビシェフ級数の微係数
D 6	ICHB1, ICHB1D	チェビシェフ級数の不定積分の値
D 6	VCHB3, VCHB3D	ずらしチェビシェフ級数の値
D 6	DCHB3, DCHB3D	ずらしチェビシェフ級数の微係数
D 6	ICHB3, ICHB3D	ずらしチェビシェフ級数の不定積分の値
D 6	VCHB2, VCHB2D	第 2 種チェビシェフ級数の値
D 6	ICHB2, ICHB2D	第 2 種チェビシェフ級数の不定積分の値
D 6	DRCHB1, DRCH1D	第 1 種チェビシェフ級数の導関数
D 6	DRCHB3, DRCH3D	ずらしチェビシェフ級数の導関数
D 6	IICHB1, IICH1D	第 1 種チェビシェフ級数の不定積分
D 6	IICHB3, IICH3D	ずらしチェビシェフ級数の不定積分
D 6	BITREV, BITRVD	ビット逆転

(研究開発部)

VII. ジョブ制御言語の手引きの発行について (再掲載) …… 省略

VIII. バッチにおけるファイルの利用講習会テキストについて

10月17日(水)に行いました「バッチからのファイル利用法について」の講習会で使用いたしました「バッチにおけるファイルの利用講習会テキスト」の在庫がありますので御希望の方はJOB受付前のロッカーにありますので持って行ってください。

(共同利用掛)

IX. 利用者講習会の開催について

1. TSSの利用法について

- 日 時 11月28日(水) 29日(木) 10:00~17:00
- 場 所 本センター大会議室(3階)
- 受講資格 FORTRANプログラミングの知識があるが、TSSの利用経験がない方
- 募集人員 講習60名、実習40名
- 申込期間 11月12日(月)~11月21日(水)
- 申 込 先 本センタージョブ受付(06-877-5111 内線2814)

〔内容〕

	10:00	12:00	13:00	14:00	17:00
11月28日 (水)	FORTRANサブシステムとエディタサブシステムを説明する。		(昼休み)	TSSでACCESSサブシステムによりファイルシステムの運用法を説明する	
11月29日 (木)	班別で実習を行う。		(昼休み)	(簡単なFORTRANプログラムを準備してください。)	

X. 第1回全国共同利用大型計算機センター研究開発連合発表講演会の開催について

当センターニュース4頁に同じ内容掲載のため省略します。

プログラム交換のページ

もし、皆様がプログラムを「譲りたい」「求めたい」と思われたら、次の投稿記入項目の必要箇所を記入し、本センター庶務掛へ送付して下さい。

できる限り、最新のセンター・ニュースに掲載します。

このページをプログラム公開と譲り受けの場として気軽に御利用下さい。

プログラム交換のページの投稿記入項目

※印の付いた項目は必ず記入して下さい。

※ 1. 要 望

〔譲ります〕か〔求めます〕かいずれかを記入。

※ 2. プログラム名

内容を示す名前（例えば連立一次方程式数値解法）と引用する時の名前（例えばSLINEQ）を記入。

※ 3. プログラムの内容又は解法

アルゴリズムの特徴について説明すること。例えばプログラム作成上で特に工夫された点があれば、それらを記入する。

※ 4. 利用者の義務

このプログラムを利用した仕事の公表に際して、その旨明記する必要があるのか否かについて記入（“譲ります”の場合のみ）

※ 5. プログラム作成者名、連絡先及び連絡方法

6. 現在このプログラムが利用されている、あるいはやがて利用されるであろう分野

例えば物理学・物性論、化学・高分子結晶解析など。

7. プログラム使用上の制限事項

データ入出力における周辺機器やオペレーション上の制約や情報を記入。あるいは、例えば最大100次元の連立一次方程式までしか解けませんなどのアルゴリズム上の制限なども記入する。

8. 使用されたプログラミング言語

FORTRAN, COBOL, ALGOL, PL/1など。

9. プログラム・ステップ数

例えば、ソース・カード・イメージでの枚数など。

10. プログラム実行時に必要な最大メモリー・サイズ

11. 代表的例題に対する計算所要時間

12. プログラムの形式

コンプリート・プログラム、サブルーチン副プログラム、関数副プログラムなど。

13. プログラムの開発された計算機

機種名及びOSについて記入。

14. 利用可能状況

ソース・リスト、ソース・カード、マニュアル、計算例などのプログラム利用に関する情報の整備状況について記入する。

15. 関連又は補助プログラム

16. 参考文献

備考 1. 詳細な書式については適宜の方法で結構です。

2. ※印以外の項目については不明又は不要と思われる場合は省略してもかまいません。

3. このページは利用者間のプログラム開発に関する情報交換のためのものです。したがって、プログラムの交換についての具体的世話は大型計算機センターでは一切いたしません。

マイクロ・コンピュータ8080用 クロスアセンブラーについて

8080用クロスアセンブラーをT. S. S. でサービスいたします。

このクロスアセンブラーは8080系の78種類の命令を使用して記述されたソースプログラムを入力してアSEMBルし、機械語コードに変換してオブジェクト・プログラムとして出力します。

コマンド

このコマンドはシステム選択レベル(SYSTEM?)または、ビルドモード(*)のどちらでも使用できます。

コマンド名

CA8080 ソース・プログラム, マクロ・ファイル, ソース・リスト, オブジェクト・プログラム

パラメータの説明

ソース・プログラム

8080系マイクロコンピュータのアセンブラーが入っているファイル名。

Tを指定すると端末が割り当てられる。

*又は*SRCを指定するとカレント・ファイルが割り当てられる。

省略するとSOURCE-FILE?と問合せてくる。この時復帰のみを入力すると、カレント・ファイルが割り当てられる。

マクロ・ファイル

マクロ・ファイルが入っているファイル名。

省略するとMACRO-FILE?と問合せてくる。この時復帰のみを入力すると、マクロ・ファイルは使用していないものとする。

ソース・リスト

ソース・リストを出力するファイル名。

Nを指定するとソースリストは出力されない。

Tを指定すると端末が割り当てられる。

省略するとSOURCE-LIST?と問合せてくる。この時復帰のみを入力すると、端末が割り当てられる。

オブジェクト・プログラム

オブジェクト・プログラムが出力されるファイル名。

Tを指定すると端末の紙テープ（端末に紙テープパンチがある時）にオブジェクト・プログラムが出力される。

Nを指定するとオブジェクト・プログラムは出力されない。

省略するとOBJECT-FILE?と問い合せてくる。この時復帰のみを入力すると、オブジェクト・プログラムは出力されない。

各パラメータは省略する時はコンマ(,)で区切ること。すべてのパラメータを省略する時は、このコンマ(,)も省略できます。

コマンド	パラメーターの形式	パラメーターの意味
CA8080	[{ソース・ファイル}, {マクロ・ファイル}, {ソース・リスト}, {オブジェクト・プログラム}]	<p>ソース・ファイル：マイクロコンピューターのアセンブラー・プログラムの入っているファイル。 *または*SRCまたは省略時はカレントファイル。</p> <p>マクロ・ファイル：マクロ・プログラムの入っているファイル。 Nまたは省略時は使用しない。</p> <p>ソース・リスト：アセンブル・リストの出力ファイル。Nを指定すると出力なし。 省略時は端末。</p> <p>オブジェクト・：機械語の出力ファイル。 プログラム Tを指定すると端末の紙テープパンチに出力 Nまたは省略時は出力なし。</p>

実行例

説明

①サブシステムFORTRANを選択し、カレントファイルにアセンブラー言語の入っているSOUNDをコピーする。

②クロスアセンブラーを呼び出す。パラメータはすべて省略してある。

③パラメータを省略したためパラメータの問い合せがある。

復帰のみを入力し、カレントファイルをソース・ファイルとする。

④復帰のみを入力し、マクロ・ファイルを使用しない。

⑤復帰のみを入力し、ソース・リストを端末に出力する。

⑥復帰のみを入力し、オブジェクト・プログラムは出力しない。

```
SYSTEM ?FORT 0 SOUND ..... ①
*CA8080 ..... ②
SOURCE-FILE ? ..... ③
MACRO-FILE ? ..... ④
SOURCE-LIST ? ..... ⑤
OBJECT-FILE ? ..... ⑥
```

UCON*8A -ASSEMBLY LIST- VERSION 2.0

```
ST.NO ADDR CODE F E LABEL OP OPERAND/COMMENT PAGE 0001
00001 9000 9000 ORG 9000H
00002 9000 3E88 MAIN: MVI A,88H
00003 9002 32037C STA 7C03H
00004 9005 0E05 MVI C,05H
00005 9007 114020 M1: LXI D,2040H
00006 900A CD0091 CALL SOUND
00007 900D 0D DCR C
00008 900E C20790 JNZ M1
00009 9011 3E15 MVI A,15H
00010 9013 321791 STA XCG
00011 9016 CD0091 CALL SOUND
00012 9019 3E14 MVI A,14H
00013 901B 321791 STA XCG
00014 901E 76 HLT
00015 9100 9100 ORG 9100H
00016 9100 42 SOUND: MOV B,D
00017 9101 3E02 MVI A,02H
00018 9103 3EFF S1: MVI A,0FFH
00019 9105 32027C STA 7C02H
00020 9108 05 DCR B
00021 9109 C20391 JNZ S1
00022 910C 42 MOV B,D
00023 910D AF XRA A
00024 910E 3EFF S2: MVI A,0FFH
00025 9110 32027C STA 7C02H
00026 9113 05 DCR B
00027 9114 C20E91 JNZ S2
00028 9117 14 XCG: INR D
00029 9118 1D DCR E
00030 9119 C20091 JNZ SOUND
00031 911C C9 RET
00032 END
```

ASSEMBLY COMPLETED ERROR COUNT = 0

*** SYMBOL LIST ***

SYM	ADDR	SYM	ADDR	SYM	ADDR	SYM	ADDR	SYM	ADDR
A	0007	B	0000	C	0001	D	0002	E	0003
H	0004	L	0005	M	0006	M1	9007	MAIN	9000
PSW	0006	S1	9103	S2	910E	SOUND	9100	SP	0006
XCG	9117								

* * * JOB END * * *

ア セ ン ブ ラ 言 語

1. アセンブラ用の文字

ソース・プログラムは次の文字を用いて書きます。

英字 A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T,
 U, V, W, X, Y, Z

数字 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

特殊文字	文字	文字の名称
		空白
	+	プラス
	-	マイナス
	\$	ドル記号
	▼	引用符
	,	コンマ
	:	コロソ
	;	セミコロソ
	@	単価記号
	?	疑問符
	(左カッコ
)	右カッコ
	/	スラッシュ
	*	星印

1.1 英 字

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V,
W, X, Y, Zの26文字を英字といます。

1.2 数 字

0, 1, 2, 3, 4, 5, 6, 7, 8, 9の10個の文字を10進数字といます。

0, 1, 2, 3, 4, 5, 6, 7の8個の文字を8進数字といます。

0, 1の2個の文字を2進数字といます。

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, Fの16個の文字を16進数字
といます。

1. 3 英数字

数字と英字をまとめて英数字といいます。

1. 4 特殊文字

空白、プラス、マイナス、ドル記号、引用符、コンマ、コロンの、セミコロン、単価記号、疑問符、左カッコ、右カッコ、スラッシュ、星印の14個の文字を特殊文字といいます。

1. 5 空白

特に述べてある場合を除いて、空白は意味をもちません。プログラムをみやすくするために、自由に用いて下さい。

2. 英字名

英字名は1~5個の連続した英数字よりなり、その先頭は英字または特殊文字(?, @)でなければなりません。

英字名として、命令、擬似命令、レジスタ名、演算子名を使うことはできません。

3. 定数

定数は、それ自身で定まる値をもつもので数字や文字の列で表現されます。数字だけで構成されるものを数値定数といい、文字で構成されるものを文字定数といいます。数値定数はさらに2進定数、8進定数、10進定数、16進定数に分けられます。また数値定数には、その前にプラスまたはマイナスの文字をつけることによって、正または負の値を表現することができます。プラスもマイナスの文字もない場合には、正の値を表現しています。

3. 1 16進定数

16進定数は16進数字で構成され最後に"H"を付加します。先頭の文字が0~9以外の文字で初まるときは、英字名と区別するために、先頭に"0"を付加します。

例： 0 A B H..... 16進でA Bの数値を示します。先頭がAで初まるため0を付加します。

9 C H..... 16進で9 Cの数値を示します。

↑

16進を示す記号

3. 2 10進定数

10進定数は10進数字で構成され、最後に"D"を付加するか、あるいは何もつけ加えずに表現されます。

例： 1 3 4..... 1 0 進で 1 3 4 の数値を示します。

1 3 4 D..... 1 0 進で 1 3 4 の数値を示します。

↑

1 0 進を示す記号

3. 3 8 進定数

8 進定数は 8 進文字で構成され、最後に " O " ないし " Q " を付加して表現されます。

例： 7 2 O..... 8 進で 7 2 の数値を示します。

↑

8 進を示す記号

7 2 Q..... 8 進で 7 2 の数値を示します。

↑

8 進を示す記号

3. 4 2 進定数

2 進定数は 2 進文字で構成され、最後に " B " を付加して表現されます。

例： 1 0 B..... 2 進で 1 0 の数値を示します。

↑

2 進を示す記号

3. 5 文字定数

文字定数は、 C_R , L_F , 引用符を除いた文字の列を引用符でかこんだものとられます。

空白は文字定数のなかで、空白としての意味をもちます。

文字定数は 7 ビット ASCII コードで確保されます。

引用符を文字として確保するときは、引用符を 2 つ続けます。

▼▼▼▼←文字列の終り

↑ \ 引用符が定数として確保されます。

文字列の初まり

4. ステートメント

1 ステート・メントは 1 行 8 0 カラム以内におさめるように書いて下さい。

次の行にまたがって書くことはできません。

ステート・メントはレーベル欄、コード欄、オペランド欄、コメント欄の 4 つに分かれ、各欄はブランクないしターミネータで区切られています。

アセンブラはフリー・フォーマットで採用しているので 1 行以内におさまるならどのカラムから書

きはじめてもかまいません。例えば、1ステート・メントが全てコメントであってもコメント記号(;)は第1カラムでなくてもかまいません。またレーベルが省略されて命令コードから書くとき、1カラムめを空白にする必要はありません。

レーベル欄は命令が命令ないしデータを参照するときに、その参照する名前を書く欄です。

コード欄はニーモニック、擬似命令を書く欄です。

オペラント欄はコード欄に書かれた命令の必要とするアドレス、データを書く欄です。

コメント欄はプログラムの便宜のために使用され、アセンブラでは無視されます。

4. 1 レーベル欄

レーベルは命令で参照するのに各ステートメントにつけられた名前です。

レーベルの先頭文字は英文字及び特殊記号(@ , ?)が許され、以下の文字は英数字が許されます。

レーベルは1~5文字の長さで構成され、レーベルの最後にコロン(:)が付加されます。

正しい例 LABEL :
 F14F :
 @HERE :
 ?ZERO :

誤った例 123 : 10進数で初まる。
 LABEL コロンがついてない。
 ADD : 命令コードである。
 END : 擬似命令である。

レーベルが5文字以上のときは、先頭から5文字のみを有効とします。

INSTRUCTION : は INSTR : と解釈されます。

レーベルの二重定義はエラーになります。命令が参照するステートメントは一義に定義されなければなりません。

```
LOOP1 :                    ; FIRST LABEL
LOOP2 : MOV C, D        ; SECOND LABEL
.....
JMP LOOP1
.....
JMP LOOP2
```

LOOP1とLOOP2は同じアドレスがわりつけられますので、各JMP命令は同じMOVステートメントにジャンプします。

レーベル欄は擬似命令EQU, SET, MACROのステートメントのとき、省略できません。こ

のとき各レーベルはコロン(：)でなく、空白でターミネイトされ、ネーム(NAME)と称されます。

レーベル欄は擬似命令EQU, SET, MACROのステートメント以外、省略できます。

レーベル欄に、命令、擬似命令、レジスタ名を書くことはできません。

4.2 コード欄

コード欄は実行すべきマシン命令、擬似命令、マクロの参照が書かれます。

コード欄とオペランド欄を区別するのに1つ以上の空白が必要です。

正しい例 HERE : JMP THERE

誤った例 HERE : JMP THERE

命令は1～5文字で構成されます。

4.3 オペランド欄

オペランド欄は命令で実行されるのに付随したコードを書きます。命令によりオペランド欄の必要のないものもあり、シングル・オペランド、ダブル・オペランドを必要とするものもあります。

ダブル・オペランドの場合、各オペランドは" , "で区切られます。

オペランド欄に書かれるフォーマットには4種類あり、各々9種類のデータ形式をとることができます。

オ ペ ラ ン ド 欄	
フ ォ ー マ ツ ト	デ ー タ 形 式
(1) レジスタ	(1) 16進データ
(2) ベア・レジスタ	(2) 10進データ
(3) イミディエト・データ	(3) 8進データ
(4) メモリ・アドレス(16ビット)	(4) 2進データ
	(5) プログラム・カウンタ(\$)
	(6) ASCII定数
	(7) レジスタ
	(8) レーベル
	(9) 式

4.3.1 データ形式

以下に本アセンブラで許される9つのデータ形式について説明します。

(1) 16進定数

16進定数はデータの最後に文字"H"を付加して表現します。

16進定数がAからFで始まるときは、先頭に"0"を付加してラベルと区別します。

例： ラベル欄 コード欄 オペランド欄 コメント欄
HERE: MVI C, 0BAH ; LOAD REGISTER C
; WITH THE
; HEXADECIMAL NUMBER
; BA

(2) 10進定数

10進定数はデータの最後に文字"D"を付加して表現します。或は省略してもかまいません。

例： ラベル欄 コード欄 オペランド欄 コメント欄
ABC: MVI E, 105 ; LOAD REGISTER E
; WITH 105

(3) 8進定数

8進定数はデータの最後に文字"O"ないし"Q"を付加して表現します。

例： ラベル欄 コード欄 オペランド欄 コメント欄
LABEL: MVI A, 72O ; LOAD THE
; ACCUMULATOR WITH
; THE OCTAL NUMBER
; 72

(4) 2進定数

2進定数はデータの最後に文字"B"を付加します。

例： ラベル欄 コード欄 オペランド欄 コメント欄
NOW: MVI 10B, 11110110B ; LOAD REGISTER
; TWO (THE D
; REGISTER)
; WITH OF 6H
JUMP: JMP 0010111011111010B ; JUMP TO MEMORY
; ADDRESS 2EFAH

(5) カレント・プログラム・カウンタ

カレント・プログラム・カウンタはドル記号(\$)で表現されます。ドル記号はステート・メントにわりつけられた先頭アドレスを示します。

例： ラベル欄 コード欄 オペランド欄
GO: JMP \$+6

例題のステート・メントをアセンブラが解釈すると、"GO" がわりつけられたアドレスから 6 バイト先のアドレスが、JMP 命令のジャンプ先になります。

(6) ASCII 定数

ASCII 定数は引用符 (') で囲まれた文字列です。引用符を文字として確保したいときは、引用符を 2 連続けて表現します。

例：	<u>レーベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>	<u>コメント欄</u>
	CHAR :	MVI	E, '*'	; LOAD THE E REGISTER
				; WITH THE 8 BIT
				; ASCII REPRESENTATION
				; OF AN ASTERISK

(7) レジスタ

レジスタはアセンブラで暗黙のうちにある数値がわりつけられています。

B.....0	B レジスタ
C.....1	C レジスタ
D.....2	D レジスタ
E.....3	E レジスタ
H.....4	H レジスタ
L.....5	L レジスタ
M.....6	メモリ・リファレンス
A.....7	A レジスタ (アキュムレータ)
SP.....6	スタック・ポインタ
PSW.....6	フラグ類とアキュムレータ

例： VALUE は 16 進数で 9FH とします。以下の命令は全て D レジスタに 9FH を設定します。

<u>レーベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>
A1 :	MVI	D, VALUE
A2 :	MVI	2, 9FH
A3 :	MVI	2, VALUE

(8) レーベル

ステート・メントのオペランド欄にかかれたレーベルはアセンブルされたとき 16 ビットの評価値が得られます。


```

例： レーベル欄 コード欄 オペランド欄 コメント欄
      HERE: JMP  THERE ; JUMP TO INSTRUCTION
                        ; AT THERE
      .....
      .....
      THERE: MVI  D, 9FH

```

(9) 式

オペランドにはデータ形式(1)~(8)の算術及び論理演算関係式を書くことができます。全ての演算は16ビットのデータとして扱われます。

演算子には以下の種類があります。

加算演算子(+), 減算演算子(-), 乗算演算子(*)

除算演算子(/), 剰余演算子(MOD)

論理演算子NOT, AND, OR, XOR, SHR(右シフト), SHL(左シフト), 右カッコ(), 左カッコ()は演算子の優先順位を変更するために用いられます。

4.3.2 オペランドのフォーマット

ここで命令に要求されるオペランドのフォーマットについて説明します。

(1) レジスタ

レジスタ間の演算, メモリ参照命令のとき以下の名前はアセンブラ内で特別な値がわりつけられています。レジスタを使用する命令はなるべく以下の記号を使用したほうがわかりやすくなります。

値	レジスタ
0	B
1	C
2	D
3	E
4	H
5	L
6	M
7	A

注: Mはメモリ参照で、ペアレジスタHLの内容で示されたアドレスのデータを示します。

例： ラベル欄 コード欄 オペランド欄

```

INS 1 :      MVI      REG 4 , 2 EH
INS 2 :      MVI      4 H , 2 EH
INS 3 :      MVI      8 / 2 , 2 EH

```

REG 4 が 4 と定義されていれば、上の全ての命令はレジスタ H にデータ 2 EH をロードすることになります。

(2) ベア・レジスタ

ベア・レジスタはアセンブラで暗黙のうちにある値をわりあてています。

値	記号	意味
0	B	レジスタ B と C
2	D	レジスタ D と E
4	H	レジスタ H と L
6	PSW	CPU 内のフラグとレジスタ A
6	SP	スタック・ポインタ・レジスタ

ベア・レジスタ名はなるべく記号で書く方が、プログラム上で混乱をきたさないと考えられます。今後プログラムをライブラリとして保存し、活用をはかるなら、ベア・レジスタは記号で書くべきです。

例： ラベル欄 コード欄 オペランド欄 コメント欄

```

                 PUSH      D                   ; PUSH REGISTERS D AND
                                                 ; E ONTO STACK
                 INX      SP                   ; INCREMENT 16 BIT
                                                 ; NUMBER IN THE
                                                 ; STACK POINTER

```

(3) イミディエイト・データ

イミディエイト・データは命令の内にある 8 ビット或は 16 ビットのデータです。

例： ラベル欄 コード欄 オペランド欄 コメント欄

```

HERE :      MVI      H , DATA   ; LOAD THE H REGISTER
                                         ; WITH VALUE OF DATA

```

以下にDATAのとるいくつかのデータ形式をあげておきます。

ADDR AND OFFH (ADDRは16ビットのアドレス)

127

! * !

VALUE (VALUEはある数値にわりつけられています)

3EH-10/(2 AND 2)

(4) アドレス

10ビットのアドレス定数がレーベルで示されます。

例: レーベル欄 コード欄 オペランド欄 コメント欄

```
HERE: JMP   THERE   ; JUMP TO THE
                                ; INSTRUCTION AT THERE
                                JMP   2EADH   ; JUMP TO ADDRESS
                                ; 2EAD
```

4.3.3 演算子

全ての演算は16ビットのデータとして扱われます。その演算はオペランドOPLとOPRの間で行なわれます。OPL, OPRはデータ形式(1)~(8)をとることができます。

- (1) 演算子+はオペランド間の加算 OPL+OPR
- (2) 演算子-はオペランド間の減算 OPL-OPR
- (3) 演算子*はオペランド間の乗算 OPL*OPR
- (4) 演算子/はオペランド間の除算 OPL/OPR
- (5) 演算子MODは(OPL/OPR) の剰余をもとめます。

OPL MOD OPR

- (6) 演算子NOTは各ビットを反転させます。 NOT OPL
- (7) 演算子ANDはオペランド間で各ビットの論理積をもとめます。

OPL AND OPR

- (8) 演算子ORはオペランド間で、各ビットの論理和をもとめます。

OPL OR OPR

- (9) 演算子XORはオペランド間で、各ビットの排他的論理和を求めます。

OPL XOR OPR

- (10) 演算子SHRはオペランドOPLをオペランドOPRの値だけ右シフトされ、上位に0が挿入されます。

OPL SHR OPR

(11) 演算子SHLはオペランドOPLをオペランドOPRの値だけ左シフトされ、下位に0が挿入されます。

OPL SHL OPR

演算子には優先順位がつけられております。アセンブラはオペランド欄を左から右にスキャンニングしていき優先順位の高い順から順次、処理していきます。演算子の優先順位が等しいときは、左から右の順に演算子処理していきます。この順を変更したいときは左カッコ()、右カッコ()を用いて、演算子の演算順序変更します。カッコで幾重にも囲まれた式の場合は、一番内側のカッコから式の値を評価していきます。

優先順位	
1	*, /, MOD, SHL, SHR
2	+, -
3	NOT
4	AND
5	OR, XOR

プログラマはオペランドの演算結果が命令コードの要求にあうようにしなければなりません。ですから、

MVI H, NOT 123H

のステートメントは誤りになります。NOT 123Hは16進データ0FEDCHになります。MVI命令は8ビットのデータを要求しているのですから、これは一致しくなくなります。しかし

MVI H, NOT 123H AND 0FFH

のステートメントは正しくなります。上位8ビットが0クリアされてしまうため、演算結果は8ビットのデータになります。

演算子MOD, SHL, SHR, NOT, AND, OR, XORとオペランドの間に少なくとも1つ以上の空白をあける必要があります。

MVI C, VALUEAND0FH

のステートメントは誤りです。

例: ラベル欄 コード欄 オペランド欄 メモリアドレス

HERE: MVI C, HERE SHR 8 2E1A

のステートメントはCレジスタに16進でデータ2Eを設定します。2Eの値はHEREのメモリ・アドレス(2E1A)を8ビット右シフトしたものです。

<u>レーベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>
NEXT:	MVI	D, 34 + 40H/2

のステートメントはDレジスタに $34 + (64/2) = 66$ を設定します。

<u>レーベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>
	MVI	D, (34 + 40H) / 2

のステートメントはDレジスタに $(34 + 64) / 2 = 49$ を設定します。

4.4 コメント欄

コメント欄はセミコロン(;)で始まりステートメントの終わりで、終了します。次のステートメントにまたがることはできません。

コメントはプログラムの覚え書きのために用いられるのでソース・プログラムのどこに書いてもかまいません。コメントを書いておけばプログラムの読むスピードがはやまり、メンテナンスも楽になります。なるべく随所にコメントを書くようにしたほうがよいのです。

例:	<u>レーベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>	<u>コメント欄</u>
	HERE:	MVI	C, 0ADH	; THIS IS A COMMENT
				;
				; BEGIN LOOP HERE
				;

5. 擬似命令

擬似命令にはDW, DB, DS, ORG, EQU, SET, END, IF, ENDF, MACRO, ENDMがあります。

5.1 DB (DEFINE BYTE OF DATA)

DBは1バイトのデータを確保するのに用います。

フォーマット

<u>レーベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>
[レーベル:]	DB	データ・リスト[;コメント]

オペランド欄には次のものを書きます。

- (1) 算術・論理演算子を含んだ算術・論理式を書くことができ、その値は8ビットのデータとして評価されます。
- (2) 引用符で囲んだASCII文字列

機能

各算術・論理式は8ビットのデータとして評価され、文字列の各文字は8ビットのASCIIコー

ド(8ビットめは常に0)としてあらわされます。各データはアセンブラでわりつけられた[レベル:]のアドレスから順々にわりつけられます。

例:	<u>レベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>	<u>機 械 語</u>
	HERE:	DB	0A3H	A3
	WORD1:	DB	5*2, 2FH-0AH	0A25
	WORD2:	DB	5ABC SHR 8	5A
	STR:	DB	'STRINGSP L'	535452494E472031
	MINUS:	DB	-03H	FD

最初の例において、16進のA3は0A3と表現されます。なぜなら16進定数の先頭は0~9の数値で初まる必要があるためです。

オペランドにエラーがあった場合、アセンブラはその箇所に0をセットし、そのあとにオペランドがあればその解釈を実行します。

オペランドはコンマ(,)で区切っていくつでも指定できます。

5.2 DW(DEFINE WORD(TWO BYTES)OF DATA)

フォーマット

<u>レベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>
[レベル:]	DW	データ・リスト[;コメント]

オペランド欄には算術・論理演算子を含んだ算術・論理式を書くことができ、その値は16ビットのアドレス・データとして評価されます。アドレス・データはコンマ(,)で区切って続けて書くことができます。

機 能

"DW"は16ビットのアドレス・データを確保するのに使用します。

16ビットのアドレス・データの低位8ビットは[レベル:]に、上位8ビットはその次のアドレスに格納されます。

例: COMPのアドレスは3B1CH, FILLのアドレスは3EB4Hとします。

<u>レベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>	<u>機 械 語</u>
ADD1:	DW	COMP	1C3B
ADD2:	DW	FILL	B43E
ADD3:	DW	3C01H, 3CAEH	013CAE3C

各アドレス・データとも、低位8ビットが次に格納されます。オペランド欄にエラーがあった場合は、その箇所に0を挿入して、そのあとにオペランドがあればその解釈を実行します。

5.3 DS (DEFINE BYTE STORAGE)

フォーマット

<u>レーベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>
[レーベル:]	DS	データ

オペランド欄は算術・論理演算子を含んだ算術・論理式を書くことができます。データは1コだけとることができ、コンマ(,)で区切って続けて書くことはできません。

機能

オペランドで指定された語数のエリアを[レーベル:]のアドレスから確保します。確保されたエリアの内容はクリアされず、プログラムはそのエリアを0にしたり、特定の値をわりつけたりすることはできません。

アセンブラはDSの次のステートメントを確保されたエリアの次からわりつけるよう指示します。

例:

<u>レーベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>
HERE:	DS	10 10バイトのエリアを確保します。
	DS	10H 16バイトのエリアを確保します。

オペランド欄の指定にエラーがあったとき、アセンブラはDS命令を無視します。

5.4 ORG (ORIGIN)

フォーマット

<u>レーベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>
[レーベル:]	ORG	データ

↑
16ビットのアドレス

機能

アセンブラの持っているロケーション・カウンタにオペランドで指定されたアドレスを設定します。オペランドの記述に英字名が含まれている場合には、その英字名は既に定義されたものでなければなりません。

プログラムの先頭にORG命令がないとき、アセンブラはロケーション・カウンタに0をセットしてアセンブルします。

例:

アドレス(16進)	<u>レーベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>	<u>機械語</u>
		ORG	1000H	
1000		MOV	A, C	79
1001		ADI	2	C602
1003		JMP	NEXT	C85010
	HERE:	ORG	1050H	
1050	NEXT:	LXI	H, HERE	210610
1053		XRA	A	AF

最初のORG命令はアセンブラにプログラムが1000Hアドレスから始まるよう指示しています。
2番目のORG命令はアセンブラにロケーション・カウンタを1050Hに変更するよう指示しています。

次のステートメントでラベルHEREが参照されています。HEREのアドレスは1006Hがわりつけられています。これはHEREがジャンプ命令(3バイト)のあとにつづいているためです。

1006Hから104FHのメモリはオブジェクト・プログラムの範囲内ですが、その領域にはアセンブルされたデータはありません。プログラマが特別の値を指定することはできません。
オブジェクト・プログラムをRAMにローディングしたときその領域はオブジェクト・プログラムをロードする以前のデータが残っています。

メモリ	ラベル欄	コード欄	オペラント欄	ラベル欄	コード欄	オペラント欄	機械語
2C00		MOV	A, C		MOV	A, C	79
2C01		JMP	NEXT		JMP	NEXT	C3102C
2C04		DS	12		ORG	\$+12	
2C10	NEXT:	XRA	A	NEXT:	XRA	A	AF

ORG命令はDS命令と同じ機能もっています。上の2つの例を比べてみて下さい。

<u>アドレス</u>	<u>ラベル欄</u>	<u>コード欄</u>	<u>オペラント欄</u>	<u>機械語</u>
		ORG	1000H	
1000		MOV	A, C	79
1001		ADI	2	C602
1003		JMP	NEXT	C35000
	HERE:	ORG	50H	
0050	NEXT:	LXI	H, HER	210610

最初のORG命令はアセンブラにプログラムが1000Hアドレスから初まるように指示しています。

2番目のORG命令はアセンブラにロケーション・カウンタを50Hに変更するよう指示しています。このアセンブラは正常に処理されますがオブジェクト・プログラムはアドレス項に並ばなくなります。オブジェクト・プログラムはわりつけアドレスをもっているのでも正しく読めるのですが、PROMに書き込む紙テープを出力するときはエラーになります。なぜならこの紙テープはわりつけ

アドレスをもっていないからです。

プログラムはアドレスの項に書くようにして下さい。

5.5 EQU (EQUATE)

フォーマット

<u>レーベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>
ネーム	EQU	データ

機能

オペランド欄で指定されたアドレスをレーベル欄で定義されたネームにわりつけます。

オペランドの記述にレーベルが含まれている場合には、そのレーベルはすでに定義されたものでなければなりません。

オペランドの記述にエラーがある場合、そのネームはレーベル・テーブル（アセンブラがもっているレーベルとアドレスの対応表）に登録されませんのでそのネームを参照している命令はエラーになります。

注意

ネームはブランクでターミネイトされます。ネームは一度定義するだけで再定義することはできません。

例：	<u>レーベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>	<u>機械語</u>
	PTO	EQU	8	
		⋮		
		OUT	PTO	D308

例のOUT命令はOUT 8のステートメントと同じです。

5.6 SET

フォーマット

<u>レーベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>
ネーム	SET	データ

機能

レーベル欄に書かれたネームの値はオペランドで指定された値に変更されます。ネームとその値は一番最初のSETステートメントで定義されますが、二度め以降はそのネームがオペランドで指定された値に変更されます。変更された値は次のSETステートメントがくるまでその値が保持されます。

オペランドの記述にレーベルが含まれている場合には、オペランドの値を0に変更して処理を続け

ます。

ネームが一度しかあらわれないSETステートメントはEQUステートメントと同じになります。

注 意

ネームは空白でターミネイトされます。

例：	<u>ラベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>	<u>機械語</u>
	IMMED	SET	5	
		ADI	IMMED	C605
	IMMED	SET	10H-6	
		ADI	IMMED	C60A

5.7 END (END OF ASSEMBLY)

フォーマット

<u>ラベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>
[ラベル:]	END

機 能

ENDステートメントはアセンブラにプログラムの終りを示します。ENDステートメントはプログラムの一番最後のステートメントでなければなりません。省略されたときアセンブラはアセンブルを中断させます。

注 意

ENDステートメントに書かれたラベルはプログラムの外のアドレスがわりあてられています。

5.8 IF AND ENDIF (CONDITIONAL ASSEMBLY)

フォーマット

<u>ラベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>
[ラベル:]	IF	データ
	
	ステートメント	
	
[ラベル:]	ENDIF	

機 能

アセンブラはIFステートメントのオペランド値が0のとき、IFとENDIFの間にあるステートメントを無視します。

それ以外の場合、IFとENDIFの間にあるステートメントはアセンブルされます。

例：	<u>レーベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>	<u>機 械 語</u>
	COND	SET	0 FFH	
		IF	COND	
		MOV	A, C	7 9
		ENDIF		
	COND	SET	0	
		IF	COND	
		MOV	A, C	
		ENDIF		
		XRA	A	AF

5.9 MACRO AND ENDM (MACRO DEFINITION AND END OF MACRO)

フォーマット

<u>レーベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>
ネーム	MACRO	パラメータ・リスト
.....		
ステートメント		
.....		
ENDM		

機 能

アセンブラはMACROとENDMの間にある一連のステートメントをネーム(マクロ名)で代表するように指示します。

アセンブラはソース・プログラムでコード欄にマクロ名が記述されますと一連のステートメントを発生して、アセンブルします。

パラメータ・リストはマクロの仮引数が記述されます。

6. 機械語命令

機械語命令文にはオペランドなしの命令文、シングル・オペランドの命令文、ダブルオペランドの命令文があります。以下に命令の形式とアセンブラ記述形式を述べます。

6.1 命令の形式

命令には1バイト命令, 2バイト命令, 3バイト命令があります。

2バイト, 3バイト命令はメモリの連続した番地にわりつけられます。命令の形式は以下に示されます。

1 バイト命令

命令コード

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

代表的な命令 レジスタ間の転送, メモリ参照, 算術・論理命令, 回転命令, リターン命令,
PUSH, POP, DI, EI

2 バイト命令

命令コード

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

代表的な命令 イミューディアット命令, 入出力命令

3 バイト命令

命令コード

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

代表的な命令 JUMP, CALL, ダイレクト・ロード及びストア命令

6.2 オペランドなしの機械語命令

フォーマット

レベル欄 コード欄 オペランド欄

[レベル:] ニーモニック [; コメント]

[] は省略してもよいことを示しています。

ニーモニックの一覧

RLC, RRC, RAL, RAR, DAA, CMA, STC, CMC, NOP, HLT, XTHL, XCHG, DI, EI,
PCHL, SPHL, RET, RNZ, RZ, RNC, RC, RPO, RPE, RP, RM

6.3 シングル・オペランドの機械語命令文

フォーマット

レベル欄 コード欄 オペランド欄

[レベル:] ニーモニック オペランド [; コメント]

[] は省略してもよいことを示しています。

ニーモニク一覧

STAX, LDAX, SHLD, LHLD, STA, LDA, INR, DCR, IMX, DCX, ADD, ADC, SUB,
SBB, ANA, XRA, ORA, CMP, ADI, ACI, SUI, SBI, ANI, XRI, ORI, CPI, DAD,
JMP, JNZ, JZ, JNC, JC, JPO, JPE, JP, JM, CALL, CNZ, CZ, CNC, CC, CPO,
CPE, CP, CM, PUSH, POP, OUT, IN, RST

6. 4 ダブル・オペランドの機械語命令文

フォーマット

<u>レベル欄</u>	<u>コード欄</u>	<u>オペランド欄</u>
[レベル:]	ニーモニク	第1オペランド, 第2オペランド[;コメント]
[]		は省略してもよいことを示しています。

ニーモニク一覧

MOV, MVI, LXI

ASCIIコード表

(テレタイプでは普通×=1。データ通信では×を偶数パリティ (EP) になるように使う。)

文字	2進		8進		16進	
	X=0か1	X=0	X=1	X=EP	X=0	X=1
ス	×0100000	040	240	20	A0	A0
！	×0100001	041	241	041	21	A1
”	×0100010	042	242	042	22	A2
＃	×0100011	043	243	043	23	A3
\$	×0100100	044	244	044	24	A4
%	×0100101	045	245	045	25	A5
&	×0100110	046	246	046	26	A6
'	×0100111	047	247	047	27	A7
(×0101000	050	250	050	28	A8
)	×0101001	051	251	051	29	A9
*	×0101010	052	252	052	2A	AA
+	×0101011	053	253	053	2B	AB
,	×0101100	054	254	054	2C	AC
-	×0101101	055	255	055	2D	AD
.	×0101110	056	256	056	2E	AE
/	×0101111	057	257	057	2F	AF
0	×0110000	060	260	060	30	B0
1	×0110001	061	261	061	31	B1
2	×0110010	062	262	062	32	B2
3	×0110011	063	263	063	33	B3
4	×0110100	064	264	064	34	B4
5	×0110101	065	265	065	35	B5
6	×0110110	066	266	066	36	B6
7	×0110111	067	267	067	37	B7
8	×0111000	070	270	070	38	B8
9	×0111001	071	271	071	39	B9
:	×0111010	072	272	072	3A	BA
;	×0111011	073	273	073	3B	BB
<	×0111100	074	274	074	3C	BC
=	×0111101	075	275	075	3D	BD
>	×0111110	076	276	076	3E	BE
?	×0111111	077	277	077	3F	BF
@	×1000000	100	300	300	40	C0

文字	2進		8進		16進	
	X=0か1	X=0	X=1	X=EP	X=0	X=1
A	×1000001	101	301	101	C1	41
B	×1000010	102	302	102	C2	42
C	×1000011	103	303	303	C3	C3
D	×1000100	104	304	104	C4	44
E	×1000101	105	305	305	C5	C5
F	×1000110	106	306	306	C6	C6
G	×1000111	107	307	107	C7	47
H	×1001000	110	310	110	C8	48
I	×1001001	111	311	311	C9	C9
J	×1001010	112	312	312	CA	CA
K	×1001011	113	313	113	4B	CB
L	×1001100	114	314	314	4C	CC
M	×1001101	115	315	115	4D	CD
N	×1001110	116	316	116	4E	CE
O	×1001111	117	317	317	4F	CF
P	×1010000	220	320	220	50	D0
Q	×1010001	121	321	321	51	D1
R	×1010010	122	322	322	52	D2
S	×1010011	123	323	123	53	D3
T	×1010100	124	324	324	54	D4
U	×1010101	125	325	125	55	D5
V	×1010110	126	326	126	56	D6
W	×1010111	127	327	327	57	D7
X	×1011000	130	330	330	58	D8
Y	×1011001	131	331	131	59	D9
Z	×1011010	132	332	332	5A	DA
[×1011011	133	333	333	5B	DB
\	×1011100	134	334	134	5C	DC
]	×1011101	135	335	335	5D	DD
^	×1011110	136	336	336	5E	DE
+	×1011111	137	337	137	5F	DF
→	×1100000	140	340	140	60	E0
!	×1111011	173	373	173	7B	FB
!	×1111100	174	374	374	7C	FC
!	×1111101	175	375	175	7D	FD
-	×1111110	176	376	176	7E	FE

制御コード	意味	16進	
		X=0	X=1
NUL	Null	00	80
SOH	Start of Heading	01	81
STX	Start of Text	02	82
ETX	End of Text	03	83
EOT	End of Transmission	04	84
ENQ	Enquiry, Who are you	04	85
ACK	Acknowledge, Are you	06	86
BEL	Bell	07	87
BS	Back Space	08	88
HT	Horizontal Tabulation	09	89
LF	Line Feed	0A	8A
VT	Vertical Tabulation	0B	8B
FF	Form Feed	0C	8C
CR	Carriage Return	0D	8D
SO	Shift Out	0E	8E
SI	Shift In	0F	8F
DL	Data Link Escape	10	90
DC1	Device Control 1	11	91
DC2	" 2	12	92
DC3	" 3	13	93
DC4	" 4 (Stop)	14	94
NAK	Negative Acknowledgement	15	95
SYN	Synchronous Idle	16	96
ETB	End of Transmission Block	17	97
CAN	Cancel	18	98
EM	End of Medium	19	99
SUB	Substitute	1A	9A
ESC	Escape	1B	9B
FS	File Separator	1C	9C
GS	Group Separator	1D	9D
RS	Record Separator	1E	9E
US	Unit Separator	1F	9F
DEL	DeleteまたはRubout	7F	FF