

## I 2-8 標準OSとしてのUNIXの機能

Some Considerations on the Features of UNIX as a Defacto Industrial Standard OS

井田 昌之 M. Ida

青山学院大学 Aoyama Gakuin University

## ABSTRACT

\*  
UNIX has been considered as a de facto industrial standard OS in microcomputer field also. Microcomputer user and vendor seem to recognize its quality, compatibility, and full-functionality more and more. The AT&T disclosure policy of source codes and related information, plays a basic roll to make UNIX as a standard.

This paper summerizes the features and the current arguments of the UNIX and shows the case for kanji filtering and simple networking.

From the view point of end-user application, there are gaps between user-needs for application package and the features offered by UNIX. OS share is not so high, comparing other existing OS. The reason in the domestic, is not only the short of end-user application packages, but also the fact that the features are not fully utilized and not completely known. The fundamental facilities in UNIX, such as communication, documentation, and miscellaneous for file processing, filters and redirection are not always be used effectively.

From the view point of OS technologies, UNIX has arguments to discuss, such as bit-map processing, kanji and kana processing, real time capability, kernel mechanisms and others.

If we want to design or have overcoming OS against UNIX, we must faithfully admit the UNIX features at first, and the small improvements and the pile up of the contributions lead us to get new frontier beyond UNIX.

---

\*UNIX is a trademark of AT&T Bell laboratories

## 1. はじめに

マイクロコンピュータにおいても事実上標準的なOSとして、UNIXへの期待が存在する。その背景には、品質の良いOS、互換性のあるOS、機能の豊富なOSを求める声のユーザ側のみならず開発者側での高まり、ハードウェアの進歩の支えがある。

ソースプログラムの公開、情報の公開はUNIXを標準性のあるものとするのに最も基本的な役割をはたしてきた。

UNIXは充分な使用実績があり、また機能が豊富であることは広く認識されている。しかし、必ずしもその機能の豊富さは、国内のエンドユーザが求める豊富さとは直結していない(表1)。また、OSのシェアも多数存在するパソコン(オフコン)のOSの中でそれ程高いとはいえない(図1,表2)。今後についても種々の予測がある(図1,表3)が、ユーザ(例えばパソコンユーザ)の使用実績(表2)、業務ユーザ(例えばオフコンユーザ)の要求内容(表1)とのギャップをみても、未だ改良・開発の余地は多い。

また、表4に示すような論点を見いだすことができる。これらは、UNIXを責めるものではなく、より一層の標準性を求めての諸点といえる。

AT&T自身応用プログラムの強化の必要性を認めている。研究ユーザのための環境、グラフィック関連、E A用ツールその他業務用パッケージ等である。

しかし、現在のUNIXそれ自体単独でエンドユーザの要求に答えるものが全くないわけではない。

AT&TはシステムVを1983年2月に登場させ、更にそのリリースアップ版であるシステムV 2.0を1984年1月に発表した。その概要を表5、6、7に示す。

筆者は、特に1983年8月にシステムV 1.0を入手して以来、UNIXをとりいれたパーソナルLispからホストに至るシステムの構築について研究を進めてきた。

ここではこうした背景をもとにして、実質的に標準OSの一つとなってきたUNIXの特徴・機能・現状についてまとめ、また簡単な実例を紹介したい。

表2. 使用中のパソコンOS

(日経コンピュータ'83.12.12.p173)

| OS         | %     |
|------------|-------|
| Disk Basic | 52.2% |
| CP/M-80    | 23.3% |
| CP/M-86    | 16.0% |
| MSDOS      | 11.6% |
| UCSD-p     | 2.9%  |
| UNIX       | 1.7%  |
| otherOS    | 15.5% |

表1 オフコンに欲しい機能

| 順位 | 機能                           | 比率    |
|----|------------------------------|-------|
| 1. | ワークステーションにパソコン接続             | 38.7% |
| 2. | 日本語ワープロ機能                    | 37.6% |
| 3. | ビジネスグラフ機能                    | 32.0% |
| 4. | FAXの直接接続                     | 21.0% |
| 5. | パソコン流通ソフトの利用                 | 20.3% |
| 6. | OCRの直接接続                     | 18.1% |
| 7. | データベース機能                     | 17.8% |
| 8. | 音声処理機能                       | 15.1% |
|    | <以下略> (回答3341,3挙のため100%をこえる) |       |

表3. 16ビットOSの需要予測の例

|         | 1983         | 1984          | ratio          |
|---------|--------------|---------------|----------------|
| UNIX    | 263<br>(214) | 1007<br>(875) | 3.8<br>(4.1)   |
| MSDOS   | 415<br>(329) | 536<br>(450)  | 1.3<br>(1.4)   |
| CP/M-86 | 151<br>(139) | 233<br>(225)  | 1.5<br>(1.6)   |
| CCPM    | 54<br>(50)   | 658<br>(600)  | 12.2<br>(12.0) |
| otherOS | 269<br>(207) | 525<br>(400)  | 2.0<br>(1.9)   |
| total   | 1152         | 2959          |                |

by Creative Strategies Int'l USA

(日経コンピュータ NO.67 P 99 オフコンユーザ全国調査表24より)



マイクロコンピュータ応用国際コンファレンス'84

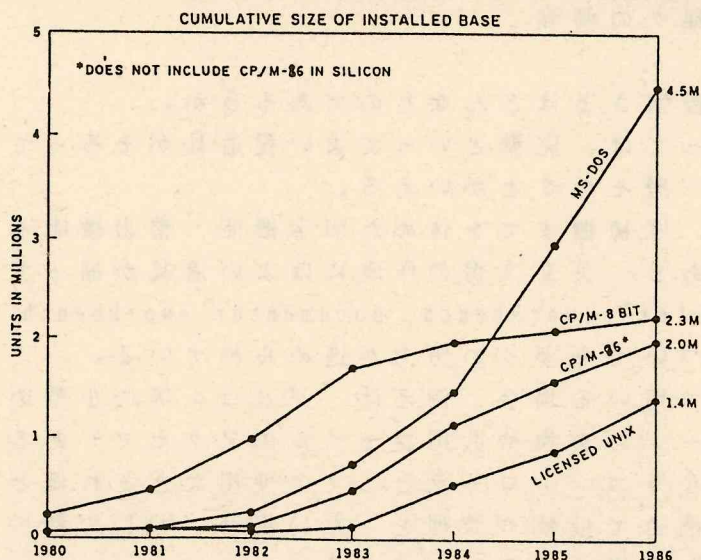


図1. 16ビットOSの予測例 — Yates Ventures (byte'83-10 より)

表4. UNIXへの論点のいくつか

|    |              |                          |
|----|--------------|--------------------------|
| 1  | メモリ管理        | swapping vs paging       |
| 2  | disk管理       | blocksize, algorithm     |
| 3  | 端末管理         | char by char 処理          |
| 4  | 端末仕様         | bitmap, multiwindow      |
| 5  | 通信           | ネットワーク, LAN              |
| 6  | 日本語処理        | カーネル対応, ワープロ             |
| 7  | 互換性          | UNIXの標準性の確立              |
| 8  | 応用           | 各種パッケージの要求               |
| 9  | 信頼性          | security, support center |
| 10 | real time 性能 |                          |

表5. system V 1.0 での追加機能

|   |                      |  |
|---|----------------------|--|
| 1 | スクリーン・エディタの装備        | vi, termcap                                  |
| 2 | 1Kバイト・ブロックのファイル・システム | ブロック・サイズを大きくした                               |
| 3 | プロセス間通信機能の追加         | IPC/remove メモリ共有型メッセージおよびセマフォ                |
| 4 | 全面的な性能向上             | より速いアルゴリズムを随所に採用. C環境の強化.                    |
| 5 | ネットワーク機能の強化          | Common Synchronous Interface, uucpの強化        |
| 6 | 3機種をサポート             | VAX-11/780, VAX-11/750, PDP-11/70            |
| 7 | ディスク・ファイル・システムの保護    | 障害時の保護の強化                                    |
| 8 | システム管理者のための機能の追加     | ff, fuser, SAR (System Activity Report) コマンド |
| 9 | ドキュメントの充実            | V用のドキュメントを整備,<br>トピックに応じたドキュメントを将来発行         |

表6 R2.0での強化項目

|    |                          |  |
|----|--------------------------|--|
| 1  | Job Control              | multiwindow 対処機能. shell layer の概念. 同時に複数のシェルを起動し、それらと交互に会話. shl コマンドの新設 (layer生成・削除・resume・状況通知) |
| 2  | コマンドの新設                  | mailx:mailの拡張版, pg: 画面出力関連<br>trenter: UUCPを経由してUNIXシステムサポートセンタへ障害を通知.                           |
| 3  | Cron 機能                  | Cronコマンドの変更 (system jobだけでなくuser jobをスケジュール起動できるようにした). Crontab コマンドの新設                          |
| 4  | LintLibrarySupport       | Cプログラムのlintチェック用   |
| 5  | accounting package       | 改良と速度向上  |
| 6  | SGS                      | 改良   |
| 7  | Curses/Terminfo<br>パッケージ | 画面端末のための<br>サブルーチンパッケージ/データベース   |
| 8  | 性能向上                     | 実行は5%~10%速い<br>25%程度まで速度向上. 機能の改良.   |
| 9  | シェル                      | アルゴリズム変更 (コマンドサーチにハッシュを使う他). バックグラウンドのプライオリティを下げる  |
| 10 | その他                      | ディスク及びMTの名称の追加. コマンドオプションの追加 (ls, who, passwd, grep, cat, man.sort, dircmp)                      |

表7. R2.0で追加されたパッケージ

|   |                         |
|---|-------------------------|
| 1 | Basic インタプリタ            |
| 2 | 68000 SGS               |
| 3 | Documenter's workbench  |
| 4 | Writer's workbench      |
| 5 | "s" 統計ソフトウェア            |
| 6 | Instructional workbench |
| 7 | 5620 パッケージ              |
| 8 | C コンパイラ                 |
|   | F??                     |



## 2. 「UNIXの機能の豊富さ」の種々の側面

UNIXそれ自体で持っている機能の豊富さとはどんなものであろうか。

第一に、ソフトウェア開発者にとっては、完璧といってよい程道具がそろっている。言語Cを中心とする場合には一層そのことがいえる。

第二に、文書作成を業とする場合、写植機までを含めた印字機能、整書機能そして文書のプルーフチェックなどがあり、英文文書の作成にはよい道具が揃っている。これらはsystemV 2.0ではwriter's workbench, documenter's workbenchとしてまとめられている。日本語についても多くの努力が進められている。

第三に、簡単な通信制御装置として用いる場合、例えば、パソコン等の小型のコンピュータ間の相互通信（メッセージの交換や共用ファイルのアクセス）あるいは、ホストへのクラスク制御を行なうコントローラとしての使用などそれほど複雑な制御はいらないが、ユーザの都合で接続や管理をしたい場合、UNIXの持つ端末などの低速デバイスの接続機能を評価することができる。

もし、UNIXマシンが陳腐化してしまっても情報交換機として、あるいは英文文書作成機としてはそれ自身のハードの寿命まで使えるのではないか？

また、主目的が存在し、手元にUNIXがあるならば、副次的に多くの補助機能を享受することができる。

コマンドの組み合わせ、パイプの利用などにより、プログラムを組むことなしにデータ処理をすることができる。例えば、教育用システムで学生の出席状況・修得状況の表示コマンドの出力をファイルにとり、それをUNIXにかけ、必要な部分だけを取りだし、並びかえ、成績記録を更新し結果をうることなどは、コマンドの組み合わせだけで実現できる。

UNIXについてまとめられた初期の論文（D.M.Ritchie & K.Thompson ;CACM Vol .17 No.7）において、次のような特徴が述べられている。

- (1) 汎用のマルチユーザの会話型のOSである。
- (2) ファイルシステムは階層構造をとっている。
- (3) 標準入力・標準出力を中心にした極めて単純なプロセスとしてプログラムを扱う設計思想に基づく。
- (4) ファイルもデバイスもプロセッサ間通信も、プログラムでは全く同じに扱うことができる。
- (5) 非同期に動く複数個のプロセスを簡単に作ることができる。
- (6) ユーザごとにシステムコマンド言語を選択できる。
- (7) 数多くのプログラミング言語を含めて100をこすサブシステムを備えている。
- (8) 移植性が高い。

この各々を順においてUNIXの機能を論じてみたい。

① 汎用マルチユーザの会話型OS

UNIXへのリアルタイム機能の要求は多いが、本質的にはそれらへの対処は考慮されていない。汎用マルチユーザOSとしての機能を具備するために、例えばCP/Mなどと異なりOSの大きさも大きくなっている。また、多くの機能を実行させるために、マイクロコンピュータOSとしてはまだ重い面がある。組み込みシステム~~✗~~などは不向きといえる。また、多人数使用のためのセキュリティ対策、ユーザの



## マイクロコンピュータ応用国際コンファレンス'84

階層設定 (owner, group, 一般), メモリ管理等の点で改良・検討の余地がある。

### ② 階層構造のファイルシステム

ファイルの所属関係は「親子」だけである点に問題のあるケースもある。一人が複数のプロジェクトに属していて、多様なグループ構造がある場合などファイルの機密保護を完全に行なおうとすると多少面倒なことになる。

また、ファイルを構成するレコードブロックの大きさ、アクセスアルゴリズム、ブロック割当てアルゴリズムなどについて論点がある。

### ③ 標準入力と標準出力の概念をもつプロセスとしてのプログラムの扱い

これは大変強力な考え方である。入出力のリダイレクション及びパイプは著名なものである。また、ソフトウェアの部品化を考える上でも重要である。品質の向上につながるからである。反面、リアルタイムプログラム等に対しての要求には問題がある。

### ④ ファイル・デバイス・プロセッサ間通信の同レベル化

UNIXでは特殊ファイルとして周辺デバイス、他のプロセッサへのラインや、主記憶等を扱える。プログラム中でも通常のファイルと同様の形でアクセスできる。

通信に関していえば、現在のUNIXでは、高度な手順、例えば SDLC をサポートしていない。LAN も 4.2BSD で Ethernet をサポートする程度である。分散ネットのノードとして用いるにはまだ強化すべき点が多い。

### ⑤ プロセス生成の容易さ

UNIXの真髄はここにあるといってもよい。プロセス間通信も system Vではかなり強化されてきた。コマンドインタプリタ(シェル)自身プロセス生成メカニズムを利用している。また、プログラムを入力プロセス、出力プロセスその他に切りわけて効率・速度向上をめざすことはよく行なわれる。ネットワークのコントロールソフトから電子メールのためのモニタソフトまでがプロセスとしてバックグラウンドで走ることになる。

生成が容易なだけにプロセスの管理、例えば、動的なディスパッチングアルゴリズムなどには検討の余地がある。system V 2.0でさえバックグラウンドプロセスの重みづけを変更している。

### ⑥ コマンド言語の選択

コマンドインタプリタ(シェル)は、SVでは標準的な sh に加えて rsh, csh などがある。使いたれた者にとっては csh が一番使い易い。前に投入したコマンドを利用して再入力したり、その他の目に見える機能強化に加えハッシュ表を用いたコマンドの高速サーチなどが csh 内で実現されている。

一方、UNIXに対する批判の一つとしてだれでもログインすればどんなことでもできる、ファイルシステムの中のどこへもいける、ファイルを際限なく作られてしまう。こうしたことからUNIXで一般開放するのは怖い、危険である、というものがある。

これに対する一つの回答が rsh である。rsh はディレクトリの変更、出力のファイルへのリダイレクションの制限、コマンドの制限を自動的に行なっている。

### ⑦ 豊富な機能の具備

次に示すような機能が含められている。

- 基礎的なファイル操作関連ツール、

4BSD 7.1



- 言語処理系、c, f77, lisp, ...
- プログラム開発ツール、lint, sdb, SCCS, yacc/lex...
- テキスト処理ツール、edit, vi, ..., nroff, troff, ..., spell, ..., emacs, ...
- learn
- 電子メール
- ネットワークサポート
- データベースその他

ただし、これらのうちのいくつかは、AT&Tではサポートしていない。

### ⑧ 移植性が高い

システム自体がC言語でそのほとんどが書かれていることが基本となっている。かなりの情報が公開されていることも支えとなっている。しかし、当然のことながら本来 AT&Tで使用していたコンピュータのアーキテクチャとかなり異なる構造を持つコンピュータへの移植は可能であってもそれ程快適ではない。

一方、AT&T自身、68000, 80286, 32032 その他のマイクロプロセッサに対する system V のサポート・協力を決定しており、すでに、68000 に対しては R2.0 において関連ツールがまとめられ発表されている。

これらの論点は、OS開発技術から見たものであり、ユーザによってはここまで使いきれていないのも現状であると思われる。

## 3. 漢字処理のフィルタ例

漢字を送るための手順 (ESC シーケンス) は各社さまざまであり、利用者としては困ることが多い。たとえば、手元にある漢字処理のできるパソコンを端末として、漢字処理のできるコンピュータと接続しようとしても、多くの場合正しく漢字は送れない。適当な字に化けてしまうことになる。

このためパソコンを漢字端末として使うには、

「ホストの漢字シーケンスを理解する通信ソフトをパソコン上に組む」という方法が最も確実である。

しかし、漢字出力に フィルタ を使うという便法がある。ローマ字入力を受けて対応するひらがな、漢字を表示させるというプログラムの組み方に対しては、充分役立つ。(図2)

筆者の環境での実例を紹介する。研究室では N5200 (CP/M86) を利用しており、これに、簡単な端末エミュレーションソフトをのせている。N5200 は、シリアルインタフェースを介して UX-300 (UNIX) と結ばれている。

一般的にあって、漢字データ (JIS6226) はその送信開始を示すシフトインシーケンスを前に、終了を示すシフトアウトシーケンスを後につけて送られる。

UX-300 と N5200 CP/M-86 の各々のシフトインシーケンスとシフトアウトシーケンスを表8に示す。次のような処理をするフィルタプログラムを作る。

```
「 ESC $ がきたら ESC $ @ に変換、
   ESC + がきたら ESC ( H に変換 」
```

このプログラムを kconv と呼ぶことにする。kconv を漢字出力を含むプログ



マイクロコンピュータ応用国際コンファレンス'84

ラムをフィルタすると漢字を含むテキストの出力を任意に行なえる。本来のホストの漢字端末で行なう場合にはこのフィルタをはずせばよい。(図3,4,5)

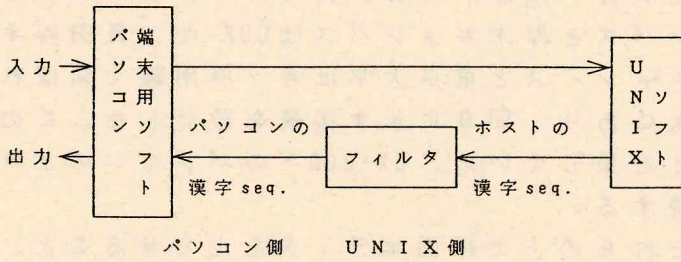


図2 漢字フィルタの利用

表8 漢字シーケンス

|                    | shift in | shift out |
|--------------------|----------|-----------|
| UX-300<br>(File内)  | ESC \$   | ESC +     |
| N5200<br>(CP/M-86) | ESC \$ @ | ESC (H    |

UX-300は標準の端末への送信時にはJIS seq.となる。

図3. kconv プログラム

```

4% cat kconv.c
#include <stdio.h>
#define ESC 0x1b
#define UX_KIN '$'
#define N52_KIN "$@"
#define UX_KOUT '+'
#define N52_KOUT "(H"
main ()
{ int c;
while ((c=getchar())!=EOF)
{ putchar(c);
if(c==ESC)
{ if((c=getchar())!=EOF)
{ if(c==UX_KIN) printf(N52_KIN);
else if(c==UX_KOUT) printf(N52_KOUT);
else putchar(c);}
}
}; exit(0); }
5%
    
```

2% jlearn | kconv  
JLEARNで、勉強できるコースは次の通りです。

files

もしこのコースの内容を知りたい時や'Learn'を前に使ったことがない時はRETURNキーをタイプしてください。そうでない時は、勉強したいコースの名前をタイプしてその後続いて RETURNキーをタイプしてください。

図4. kconv をフィルタとしたjlearn(UX-300)のN5200上での実行

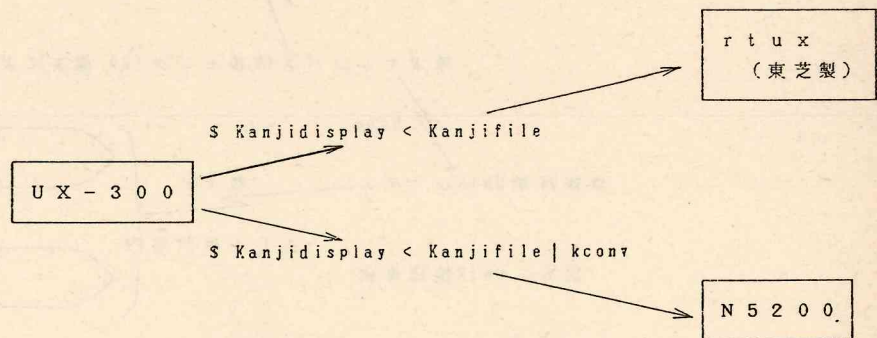


図5 漢字フィルタの利用例



## 4. 青山学院での簡単な通信コントローラとしての事例

青山学院は3キャンパス（青山、世田谷、厚木）にわかれており、各々がかなりはなれている。現在、世田谷キャンパスと厚木キャンパスはDDXで、世田谷キャンパスと青山キャンパス、世田谷キャンパスと東京大学は各々専用線で結ばれている。筆者の情報科学研究室は厚木にあり、図6に示す接続を設計した。この場合、UNIXは交換機の役割は少なくとも果している。UX-300下のパソコンのどれかからの要求に応じて、ホストを接続する。

パソコンを接続すること、そしてそれらの上で画面エディタを走らせること、端末に応じて文字の変換を換えることなどは、他のOSに比べてUNIXは非常に便利である。同時に使うユーザは一人だけでよいということであれば非常に単純に階層構造を作ることができる。

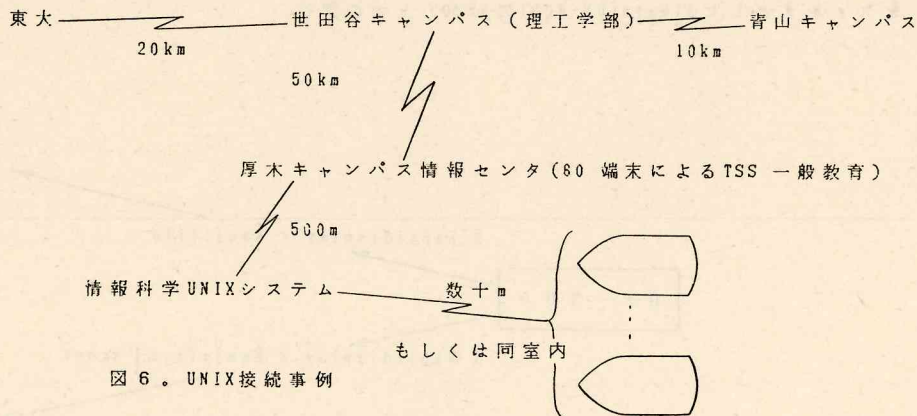
また、UUCPサブシステムのうちCUコマンドだけに限って使えば相手システムはUNIXでなくともよく、大多数のコンピュータと接続することができる。現時点での接続例を図7に示す。これは、筆者の部屋にあるCP/M-86で試作研究中のLispシステムの中から世田谷キャンパスのホストを呼び出す例である。

## 5. おわりに

UNIXの基本的な機能について、それらを列挙し、また、検討を要する点を洗い出してまとめた。あわせて、簡単な例を紹介した。

4. 2BSDのチームのdisperse(disappear?)は、標準OSとしてのsystemVへの期待を一層高めるものといえる。また、AT&Tの商品としてのUNIXシステムの動向は注目したい。

UNIXを乗り越えるためには、まず、独自のOSを作成しそこへ多くの機能を蓄積したベル研を中心とする努力とQualityに敬意を表したい。機能の豊富さ・良さが最終的な評価につながるからである。そして、日本の環境を生かした機能を強化すること、この蓄積が次のOSを生む土台となると思われる。





## マイクロコンピュータ応用国際コンファレンス'84

```

A>
A>Lisp
                                     Lisp environment
      LISPO5   ver.1.4
      (C) 1984 Masayuki IDA, Aoyama Gakuin Univ.
      CSDSES = 500030004000

      EVAL entered ...1502.2500
      (car '(a b c))
      1502.3400

      END of EVAL, Value is...1502.3400

      a
      EVAL entered ...1502.3400
      (cpm 'COMMUNIX)
      1502.4900

                                     UNIX environment
      Login: ida
      Password: _____

      Aoyama Gakuin Atugi. Joho-Kagaku Lab. ver2. 11-12-83 MI

      Hi how are you today! Welcome to UNIX world!

      5200
      Erase set to control-H
      Kill set to control-X
      1% pwd
      /j/ida
      2% catss

                                     Host environment
      Connected
      ¥CON ATSS
      05:57:31 TCF110 CONREQ COMPLETED
      ENTER LOGIN -
      LI ██████████
      ENTER PASSWORD -
      ██████████
      LOGIN ACCEPTED. AT 84-04-20 15:03:05 FROM TM6206 RON=X003480
      YOUR LAST LOGIN AT 84-04-12 13:48:18
      ..... ATSS SERVICE ..... ( 9 : 30 --- 16 : 30 )
      *LO
      SESSION TERMINATED. AT 84-04-20 15:03:18
      CPTIME 00:00:00:011 ELAPTIME 00:00:13:460 T I/O 000000010 F I/O
      05:58:09 TCF110 DISREQ COMPLETED...

      Disconnected
      3% Logout

      Login:
      END of EVAL, Value is...1504.5300

      NIL
      EVAL entered ...1504.5400
      (cons 'a 'b)
      1505.0900

      END of EVAL, Value is...1505.0900

      (a.b)
      EVAL entered ...1505.0900
      ^C
      A>dir
      A: WC          CMD : LISP      OBJ : PIP      CMD : STAT     CMD
      A: SUBMIT     CMD : ASM86     CMD : GENCMD   CMD : DDT86     CMD
      A: FORMAT     CMD : LDCOPY    CMD : COPYBACK CMD : KEY       CMD

```

☑ 7. Lisp → UNIX → ホストの呼び出しの例



The Fourth International Microcomputer Applications Conference  
May 28, 29 and 30, 1984, Japan Press Center Hall  
Hibiya, Uchisaiwaicho, Tokyo, Japan

MAY. 28-30  
**IMAC '84**  
アイ マ ッ ク TOKYO

Sponsored by  
JAPAN ELECTRONIC INDUSTRY DEVELOPMENT ASSOCIATION

### INTRODUCTION

Microcomputers are expected to be an indispensable vehicle for providing intelligence in every area of industrial productions. The diversified applications of microcomputers are already playing an important role in providing new capabilities for various industrial products.

Indeed, the recent years see explosion of microcomputer applications. Moreover, the future use of microcomputers is forecasted to far exceed that which has been experienced to date.

In accordance with this pressing situation, the Japan Electronic Industry Development Association offers the International Microcomputer Applications Conference in order to promote the application of microcomputers in new industrial fields, to simulate the needs for microcomputers, and to raise and strengthen the foundation of the industry.

### IMAC Executive Committee

|                 |                     |  |
|-----------------|---------------------|--|
| Chairman:       | Yoshikuni Ohkawa    | Gifu University                              |
| Vice Chairmans: | Hidetoshi Takahashi | Keio University                              |
|                 | Shigeru Watanabe    | Tokyo Metropolitan College of Technology     |
|                 | Hideo Aiso          | Keio University                              |
|                 | Ryoichi Mori        | University of Tsukuba                        |
|                 | Toshiaki Yasuda     | Tokyo Electrical Engineering College         |
|                 | Haruhisa Ishida     | University of Tokyo                          |
| Members:        | Kozo Kinoshita      | Hiroshima University                         |
|                 | Akio Tojo           | The Electrotechnical Laboratory              |
|                 | Hirosato Nomura     | Yokosuka Electrical Communication Laboratory |
|                 | Yoshiaki Shimono    | Japan Electric Company                       |
|                 | Tadashi Kurachi     | Toshiba Corporation                          |
|                 | Yuzo Kita           | Hitachi, Limited                             |
|                 | Tsuyoshi Watanabe   | Fujitsu Limited                              |
|                 | Kazuaki Mayumi      | Matsushita Electric Corporation              |
|                 | Yoshiaki Tsuboi     | Mitsubishi Electric Corporation              |
|                 | Setsuji Kamuro      | Setsufumi Kamuro                             |

## PROGRAM

### 1. DATE AND PLACE

Date: May 28 (Mon.), 29 (Tue.), 30 (Wed.), 1984  
Place: Place Center Hall (Japan Place Center Building 8th Floor)  
2-2-1 Uchisaiwaicho, Chiyoda-ku, Tokyo 100

### 2. KEYNOTE LECTURES AND TOPICS

#### Opening Address

Dr. Y. OKAWA, Professor      University of Gifu

#### Keynote Lecture

- I. [Microcomputing in the 1980's: A Technological "WONDERCHILD"  
Comes of Age ....]

Dr. P. L. HAZAN, Professor      Johns Hopkins University

- II. [Education for the VLSI Generation]

Dr. P. R. RONY, Professor      Virginia Polytechnic Institute and State University

#### Topics

[Activities of the United Nations University in the field of Microprocessors and Microcomputers]  
E.W. Ploman, Vice Rector      United Nations University