

# 歴史的コンピュータのシミュレータ上で、MACSYMA, VAXIMA, MAXIMA を使って遊ぶ

足立健朗

## 自分で調べて、遊ぶためのヒント

### 1. Simh PDP-10 シミュレータ上の ITS 上で MACSYMA 1001 を試してみる

遊べる環境を整えるには、ウェブページ：

<https://github.com/PDP-10/its>,  
<https://gist.github.com/oubiwann/c791d08cf661db89d746ffdbedf2cc21>

を参考にするのがよい。投稿の内容をヒントに各自環境を構築する。自分の普段使いの環境が Unix の流れを汲むものならば格段の事は無いであろう。ちなみに私の環境は Let's note CF-SX2 に FreeBSD 11.4 Release を入れたものである。その他の場合については私には分からない。非常に簡単に言うと、遊ぶ環境を保存するディレクトリで、次の様な操作をする。

```
git clone https://github.com/PDP-10/its.git
cd its
make EMULATOR=simh
```

無事に PDP-10 シミュレータと ITS の入ったディスクイメージが構築できたら、環境を保存したディレクトリで `Simh` PDP-10 シミュレータを以下の様に起動する。

```
./start

its <ENTER>
<ESC> g
<CTRL> Z
:LOGIN kenrou
<CTRL> R
```

詳しく説明しないが、この文書を読むような人であれば上の実際の入力がどのようなものであるかは分かるであろう。また、シミュレータおよび ITS からの応答は省略している。ITS 上で Macsyima を起動し遊び始める前に、ITS をシャットダウンし、シミュレータを終了する方法を述べて置こう。ただし著者は ITS を現役世代として使用したものではないので、以下の操作が完全なものかどうか判じかねる。以下では向こうからの応答も書いている。

```

:lock
LOCK.156
_5DOWN
DO YOU REALLY WANT THE SYSTEM TO GO DOWN?
Y
PLEASE ENTER A BRIEF MESSAGE TO USERS, ENDED BY ^C
USER  LOCK  SYS   DOWN  MAIL  WRITE  02:16:52
BYE
^C
CULPRIT =  USER  LOCK   02:16:58
KA ITS going down in 5:00
BYE _Q                                # hit Q to leave LOCK...
232)  .IOT 16,1
:LOGOUT                               # logout triggers immediate shutdown
LOGOUT USER   0 02:17:37

<CTRL> \
sim> quit <ENTER>

```

それでは、シミュレータの起動から、Macsyma の立ち上げまでを示そう。

```

[~/EmuData/its/]kenrou% ./start

PDP-10 simulator V4.0-0 Current          git commit id: e1db7dec
build/simh/boot-6> at dz0 10004
Listening on port 10004
build/simh/boot-8> at dz0 line=7,10018
Line 7 Listening on port 10018
build/simh/boot-10> at tty line=6,10019
Non-existent device

DSKDMP
its <ENTER>
<ESC> g
$G

Salvager 261

```

DB ITS 1650 IN OPERATION

DB ITS 1650 SYSTEM JOB USING THIS CONSOLE.THE KS-10 CLOCK HAS BEEN RESET, IF THE TIME CAN BE DETERMINED FROM THE NETWORK, YOU MAY HAVE TO :PDSET

LOGIN TARAKA 0

TOP LEVEL INTERRUPT 200 DETACHED JOB # 3, USR:TARAKA CNAVRL

Attention!

Attention!

The time could not be set because:

No host responded.

It will be necessary for someone to set the time manually by logging in and running :PDSET.

TARAKA NETIME IOTLSR

DB ITS 1650 SYSTEM JOB USING THIS CONSOLE.

<CTRL> z

DB ITS.1650. DDT.1547.

TTY 0

You're all alone, Fair share = 82%

(ITS does not know the date, so messages cannot be reviewed right now.)

Welcome to ITS!

For brief information, type ?

For a list of colon commands, type :? and press Enter.

For the full info system, type :INFO and Enter.

Happy hacking!

:login kenrou

DB: KENROU; KENROU MAIL - NON-EXISTENT DIRECTORY

LOGIN KENROU 0

:macsym

This is MACSYMA 1001

(C1)

Macsyma 1001 が起動したら、試しに使ってみて、その後 Macsyma の終了操作まで行ってみる。

```
(C1) line1:80$
```

```
(C2) integrate(1/(1+x^3), x);
```

```
SIN FASL DSK MACSYM being loaded
Loading done
```

```
SININT FASL DSK MACSYM being loaded
Loading done
```

```
SCHATC FASL DSK MACSYM being loaded
Loading done
```

```
(D2)          2 X - 1
              2      ATAN(-----)
              LOG(X  - X + 1)  SQR(3)  LOG(X + 1)
          ----- + ----- + -----
              6          SQR(3)          3
```

```
(C3) quit();
```

最初の line1:80\$ と言うのは、1行の表示文字数を80字とする設定のおまじないで、私の使用環境(ターミナルエミュレータとして Sakura を使っている)に合わせたものである。これをしておかないと計算結果の数式の表示が乱れてしまうためだ。ここまでで基本的な事の説明は終わりである。後は自分たちで遊んで見ると良い。

さて、上で使用した PDP-10, ITS シミュレータの配布物には附録として Tektronix 4014 グラフィックターミナルエミュレータが含まれている。サブディレクトリ Tek4010 以下にエミュレータのソースコード、マニュアル類などのドキュメント、使い方の例題などが含まれているので、説明をよく読んで make する。ちなみに私の FreeBSD 環境では、一部の機能が使えないものしかできないが、使いたいのは Macsyma の描画機能だけなのであまり気にしない。おそらく Linux では完全なエミュレータができるのであろう。この Tektronix 4014 エミュレータの他に、昔からある “xterm” もまた Tektronix 4014 エミュレーションモードを持っているので、それを使うことができる。エミュレータの実行ファイルは、私の場合 tek4010 という名前だ。別のターミナルエミュレータ上で PDP-10 シミュレータを立ち上げておき、

```
tek4010 -b1200 -fullv -autoClear telnet localhost 10004 &
```

Tektronix 4014 エミュレータを立ち上げる。表示は余り親切では無いが気にせずに、<ctrl>z で、ITS のコマンドインタープリタに当たる DDT を起動すると、普通に使えるようになる。以下、一連のセッションを示す。

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^['.

Connected to the PDP-10 simulator DZ device, line 0

DB ITS.1650. DDT.1547.
TTY 1
2. Lusers, Fair Share = 99%

(ITS does not know the date, so messages cannot be reviewed right now.)
Welcome to ITS!

For brief information, type ?
For a list of colon commands, type :? and press Enter.
For the full info system, type :INFO and Enter.

Happy hacking!
:login kenrou
KENRO0$UDB: KENROU; KENROU MAIL - NON-EXISTENT DIRECTORY
:tctyp tek
:KILL
*:macsym

This is MACSYMA 1001

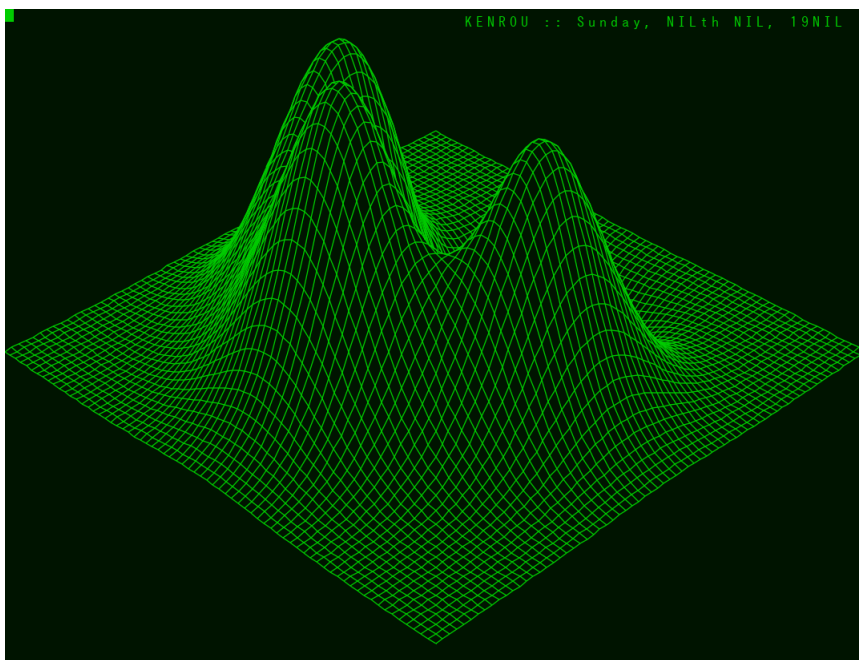
(C1) plotnum0:70, plotnum1:70$

(C2) plot3d((x^3+y^4-0.2*x)*exp(-x^2-y^2)+0.3*exp(-(x-1.225)^2-y^2),x,-3!
,3,y,-3,3)█
```

```
TEKPLT FASL DSK SHARE being loaded
Loading done

PLOT3D FASL DSK SHARE being loaded
Loading done

--Pause-- █
```



DDT セッションにおける `:tctyp tek` という入力、コンピュータに現在のユーザーが Tektronix 4014 ターミナルエミュレータ上で作業していることを教えるものである。また Macsyma セッションにおける `plotnum0:70$` および `plotnum1:70$` という入力文は、それぞれ 3D グラフィックスにおける、x 軸および y 軸方向のワイヤーステイク表示におけるワイヤーの本数を 70 本とする命令である。上の例は、Macsyma Reference Manual に挙げられているものである。

Tektronix 4014 グラフィックターミナルのエミュレーションは、昔からあるターミナルエミュレータ `xterm` でも可能である。`xterm -t` で起動すると、Tektronix 4014 エミュレーションモードで立ち上がる。その後は、

```
% kenrou> setenv TERM tek
% kenrou> telnet localhost 10004
Trying 127.0.0.1...
Connected to localhost
Escape character is '^['.
```

```
Connected to the PDP-10 simulator DZ device, line 0
```

```
HJKDB ITS.1650. DDT.1547.
```

```
TTY 1
```

```
You're all alone, Fair share = 99%
```

```
(ITS does not know the date, so messages cannot be reviewed right now.)
```

```
Welcome to ITS!
```

For brief information, type ?  
For a list of colon commands, type :? and Press Enter.  
For the full info system, type :INFO and Enter.

Happy hacking!

:login kenrou

DB: KENROU; KENROU MAIL - NON-EXISTENT DIRECTORY

:tctyp tek

:KILL

\*:macsym

This is MACSYMA 1001

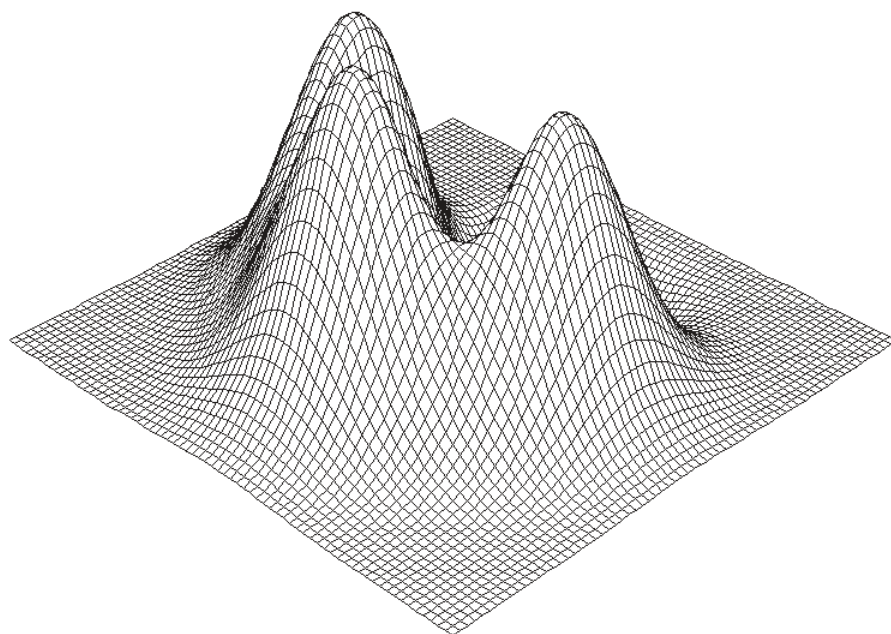
(C1) plotnum0:70\$

(C2) plotnum1:70\$

(C3) plot3d((x^3+y^4-0.3\*x)\*exp(-x^2-y^2)+0.3\*exp(-(x-1.225)^2-y^2),  
x,-3,3,y,-3,3);

■

KENROU :: Sunday, N1Lth N1L, 19N1L



と言った感じになる。

より高度な遊び方をするためには、もちろん ITS を使い方に馴れなくてはならない。残念ながら、ここで使った ITS の仮想システムにはバグがあるらしく、ファイル操作する権限を持ち自身のホームディレクトリを持つ新規ユーザーになることができない。しかし、この時代のおおらかさにより、ゲストユーザーとしていつでもログインできるので、お試しの操作は可能である。自作プログラムや、個人セッションを保存することには残念ながらできていない。基本的な ITS に関する資料を挙げておこう。いずれもインターネット上で検索を掛ければ見つけることができるだろう。

An Introduction to ITS for the MACSYMA User, V. Ellen Golden, 1981.

ITS 1.5 Reference Manual, D. Eastlake, R. Greenblatt, J. Holloway,

T. Knight, S. Nelson, 1969.



## 2. SIMH VAX780 シミュレータ上の 4.1BSD 上で VAXIMA 2.04 を試してみる

次に Simh VAX780 シミュレータで 4.1BSD を起動し、4.1BSD 上で Vaxima を使ってみよう。Vaxima は MIT の Macsyma を、その開発メンバーの一人であった R. Fatemmann がカリフォルニア大学バークレー校に着任後に、Macsyma を、その地でも使い続けるために、彼の学生やバークレー校の BSD 開発チームなどの協力の下に 3BSD に移植したものである。つまり、Vaxima とは VAX 上の BSD Unix の上で稼働する Macsyma である。BSD に移植される時に、開発言語である Lisp が、Maclisp から、それを BSD 上で実現させるために新たに作成された Franz Lisp に変更された。Franz Lisp はかなりの部分 Maclisp を踏襲した Lisp であるが、BSD Unix に固有の部分が付加されている。従って、Vaxima は基本的にはオリジナルの Macsyma と同等であると思ってよいが、ITS と Unix の機能の違いによる部分、ファイル操作とか、メモリ管理に関する部分については当然ながら互換性がない。また時代がより新しいので、グラフィックスに関する扱いも当然互換性がない。

最初にすべき作業は、Simh シミュレータと 4.1BSD 仮想システムの導入であるが、これについてはインターネット上にいくらかでも情報が存在するので、その内の一つを挙げておこう。

[http://gunkies.org/wiki/Installing\\_4.1\\_BSD\\_on\\_SIMH](http://gunkies.org/wiki/Installing_4.1_BSD_on_SIMH)

次に必要な作業は Vaxima システムを 4.1BSD 上に導入することであるが、これは Vaxima アーカイブの入手、Vaxima システムのテープイメージを 4.1BSD 上でマウントし、しるべきディレクトリに展開することの順で行う。後は実行可能パスの設定、いくつかの環境変数の設定を行えば、直ちに Vaxima が使用可能となる。以下ではこの概要について述べよう。Vaxima アーカイブは

<https://archive.org/details/vaxima-2.04>

からダウンロードすることができる。Vaxima と Franz Lisp を含む tar アーカイブファイルを Simh VAX780 シミュレータのサポートしている磁気テープデバイス、たとえば tk50 などのイメージファイルに変換して置き、Simh VAX780 シミュレータの初期設定ファイルにそのファイルを使うことを記述して、シミュレータを立ち上げる。4.1BSD にログインして、スーパーユーザーとして、/usr ディレクトリにおいて

```
> tar xvf /dev/rmt0
```

とすれば、/usr/mac ディレクトリ以下に Vaxima と Franz Lisp システムが展開される。後は各自の必要に応じて、実行可能パスの設定などを行えば Vaxima 2.04 が使えるようになる。

以下実際のセッションの様子を示そう。

```
[~/EmuData/unix/4.1BSD/]kenrou% vax780 41
```

```
VAX780 simulator V3.8-1
```

```
Listening on port 2238 (socket 4)
```

```
Modem control activated
```

```
Boot
```

```
: hp(0,0)vmunix
```

```
123060+27528+24628 start 0xF5C
```

```
Berkeley VAX/UNIX Version 4.9 Wed Feb 17 15:27:46 PST 1982
```

```
real mem = 8322048
```

```
avail mem = 7738368
```

```
mcr0 at tr1
```

```

mcr1 at tr2
uba0 at tr3
dz0 at uba0 csr 160100 vec 300, ipl 15
mba0 at tr8
hp0 at mba0 drive 0
hp1 at mba0 drive 1
hp2 at mba0 drive 2
hp3 at mba0 drive 3
mba1 at tr9
ht0 at mba1 drive 0
tu0 at ht0 slave 0
tu1 at ht0 slave 1
root on hp0
WARNING: clock gained 16 days -- CHECK AND RESET THE DATE!
WARNING: should run interleaved swap with >= 2Mb
Automatic reboot in progress...
Tue Aug 15 00:16:51 GMT 1978
/dev/hp0a: 678 files 4278 blocks 3345 free
/dev/rhp0g: 2939 files 50698 blocks 90880 free
Tue Aug 15 00:16:57 GMT 1978
Mounted /usr on /dev/hp0g
preserving editor files
clearing /tmp
starting daemons: update cron accounting network mail printer.
Tue Aug 15 00:16:57 GMT 1978

```

Berkeley 4.1 VAX/UNIX (Amnesia-Vax)

```

login: kenrou
Password:
Last login: Sat Jul 29 21:02:07 on console

```

Welcome to Berkeley Vax/UNIX (4.1bsd revised 1 Sept. 1981)

You have mail.

```
4.1BSD kenrou% ls /usr/mac
```

LABEL	ellen	jpg	maxsrc	rat	sharem	wgd
Makefile	files.1	libmax	mduit.1	reh	svaxima	
Setup	franz	lisztrc.1	mrg	rz	tensor	
autoload	imsl	macrak	ode	share	transl	
das	jim	manual	paulw	share1	ucb	
demo	jm	map	plot	share2	utils	

4.1BSD kenrou% vaxima

Vaxima 2.04

Sun Sep 12 15:31:19 1982

(c1) integrate(1/(1+x^3), x);

$$(d1) \quad -\frac{\log(x^2 - x + 1)}{6} + \frac{\operatorname{atan}\left(\frac{2x - 1}{\sqrt{3}}\right)}{\sqrt{3}} + \frac{\log(x + 1)}{3}$$

(c2) quit();

4.1BSD kenrou%

オリジナルの Macsyima とのすぐ気付く違いは、小文字表記が基本に成っている事であるが、それ以外の大きな違いは感じられない。ITS 上の Macsyima では TECO とか EMACS というエディターが使えるらしいが、心理的敷居が高くてまだ使って見たことはない。それに対して 4.1BSD 上では vi が使えるし、プログラムの作成、保存も普通にできる。是非、試してみてもらいたい。

残念ながら、この Vaxima 2.04 を そのままで Simh VAX シミュレータと 4.3BSD 上使うことはできない。4.1BSD と 4.3BSD のシステムコールに非互換性があるからである。したがって無理にでも使いたい場合にはハッキング作業が必要になる。実力に自身のある方は挑戦してもらいたい。ちなみに私には無理である。

### 3. 4.3BSD QUASIJARUS 上で MAXIMA-4.155 を使う

Simh の Vax シミュレータで 4.3BSD Quasijarus を動かして遊んでいる人は、他の組み合わせで遊んでいる人より圧倒的に多いと思われる。Simh にまだ Vax780 シミュレータが無かった時から、Vax シミュレータ上で動く 4.3BSD Quasijarus の存在は、4.3BSD を体験して見たい人にとって、絶好の選択肢の一つだったからだ。それ故に、その導入法についての情報は、インターネット上に豊富に存在するし、今や出版物としても見受けられる。したがって、ここでは導入法の解説は割愛する。

ここでの主題は、4.3BSD Quasijarus 上に AKCL (Austin Kyoto Common Lisp) と Maxima-4.155 をインストールしたときの、今と成っては曖昧な記憶の一端を紹介することと、Maxima-4.155 の 4.3BSD Quasijarus 上での使用の様子を紹介することである。

Maxima は、William F. Schelter によって記述言語を Mac Lisp から Common Lisp に変更された DOE Macsyma である。DOE Macsyma は Macsyma から M.I.T. や Symbolics 社のパテントにより一般共有することができない部分以外を取り除いた Macsyma のコア部分より構成されるものである。W. F. Schelter 氏らの努力により、現在の Maxima は GNU License 下のソフトウェアとして公開されている。ここで扱う Maxima-4.155 は初期の Maxima であり、もちろん GNU ソフトウェアではなく、その使用には法的に問題が含まれるのであるが、多くの古典的ソフトウェアと同様、その使用によって現行の商用ソフトウェアの利権を犯す恐れはまず無いと思われる。そのあたりの話は、シミュレータを使って歴史的コンピュータ遊ぶ場合は常に生じる事は言うまでも無い。

Maxima-4.155 が使用する Common Lisp システムは、W. F. Schelter 氏自身がメンテナンスしていた Austin Kyoto Common Lisp (akcl) である。これは、京都大学の学生であった萩谷昌実、湯浅太一氏によって開発された Kyoto Common Lisp の W. F. Schelter 氏による改造版である。経緯は不明であるが、Macsyma の Common Lisp への移植のために、ソースが公開された、当時まだ数少なかった Common Lisp の実装であり、コンピュータシステムが C 言語を使えさえすれば容易に導入可能であった Kyoto Common Lisp を W. F. Schelter は選択したのであろう。(その後 akcl の後継は GNU に寄付されて、GNU Common Lisp (gcl) となった。) akcl そのものの 4.3BSD-Quasijarus へのインストールは特に問題なくできる。付属のドキュメントをよく読み、インストール作業を進めるだけで良い。インストールを終え、akcl を立ち上げてみると以下の様になる。

```
4.3BSD[~/]kenrou% akcl
```

```
AKCL (Austin Kyoto Common Lisp)  Version(1.619) Wed Jan 24 09:31:21 JST 2007
Contains Enhancements by W. Schelter
```

```
>(+ 1 2 3 4 5 6 7 8 9 10)
55
```

```
>(bye)
Bye.
4.3BSD[~/]kenrou%
```

Maxima のインストールには、akcl の実行環境だけではなく、そのソースコードを含む akcl の配布物が必要である。実際 Maxima そのものが akcl の拡張システムとして実現される訳であり、akcl そのものと、Maxima 用の Lisp 関数群、その上に Maxima 言語で書かれたプログラムを Lisp 言語で表現される内部形式に翻訳するシステム、コンパイラなどから構成される基本システムの上に、主に

Lisp, Maxima 両言語で記述された数式処理に関する関数群などが追加され Maxima 全体が形成される。Maxima においては必要に応じて akcl そのものが使えるモードに移行できるが、これは Maxima が Lisp の拡張システムであることを示している。

実際の作業においては、まず akcl をインストールするが、これは付属のドキュメントをよく読めば make だけで完了する。難しいのは Maxima システムの make であり、これは付属の makefile で make 一発で完了と言う訳にはいかない。まず makefile そのものが 4.3BSD Quasijarus の環境ではうまく動かないので、大幅に変更する必要があった。残念ながら作業ログを取っていなかったので、詳細について述べることはできない。何とか基本システムの実行イメージである raw\_maxima の生成にこぎつけた後は、数式処理用のプログラムをコンパイルして行く必要があるが、この部分は簡単なシェルスクリプトを自分で書いてこなし、生成されたオブジェクトファイル群をダンプして Maxima の基本システムに付加することにより完全な Maxima システムの実行イメージである saved\_maxima を得るわけであるが、これも一筋縄ではいかない。オブジェクトファイル cpoly.o を取り込むところで 4.3BSD のリンカがメモリーフォルトを起こすからだ。したがって、これを除いたものをアーカイバでまとめて、それを使うようなスクリプトを用いて saved\_maxima を生成した。こういう訳で、Maxima 起動時に cpoly.lisp を読み込まないといけない事と成った。

以上のことを踏まえて、4.3BSD Quasijarus 上に実現された Maxima-4.155 のセッション例を見てみよう。

```
[~/EmuData/unix/4.3BSD/]kenrou% vax tahoe
```

```
VAX simulator V3.6-0
Listening on port 2311 (socket 4)
Modem control activated
```

```
KA655X-B V5.3, VMB 2.7
Performing normal system tests.
40..39..38..37..36..35..34..33..32..31..30..29..28..27..26..25..
24..23..22..21..20..19..18..17..16..15..14..13..12..11..10..09..
08..07..06..05..04..03..
Tests completed.
>>>boot dua0:
(BOOT/R5:0 DUA0
```

```
2..
-DUA0
1..0..
```

```
loading boot
```

```
Boot
: /vmunix
326312+104428+130352 start 0x23b8
4.3 BSD Quasijarus UNIX #0: Thu Jul 21 20:17:11 JST 2005
    kenrou@myname.my.domain:/usr/src/sys/GENERIC
real mem  = 67076096
SYSPTSIZE limits number of buffers to 18
avail mem = 65240064
using 18 buffers containing 147456 bytes of memory
MicroVAX 3000, ucode rev 6
tmscp0 at uba0 csr 174500 vec 774, ipl 15
tms0 at tmscp0 slave 0
tms1 at tmscp0 slave 1
uda0 at uba0 csr 172150 vec 770, ipl 15
uda0: version 3 model 3
uda0: DMA burst size set to 4
ra0 at uda0 slave 0: ra82, size = 1216665 sectors
ra1 at uda0 slave 1: no disk label: ra82, size = 1216665 sectors
ra2 at uda0 slave 2: no disk label: ra82, size = 1216665 sectors
ra3 at uda0 slave 3: no disk label: ra82, size = 1216665 sectors
zs0 at uba0 csr 172520 vec 224, ipl 15
ts0 at zs0 slave 0
dz0 at uba0 csr 160100 vec 300, ipl 15
dz1 at uba0 csr 160110 vec 310, ipl 15
    Not all octets are available
lp0 at uba0 csr 177514 vec 200, ipl 14
qe0 at uba0 csr 174440 vec 764, ipl 15
qe0: delqa, hardware address 08:00:2b:aa:bb:cc
Changing root device to ra0a
Automatic reboot in progress...
Thu Feb  1 19:31:02 JST 2007
/dev/ra0a: 460 files, 5349 used, 1744 free (424 frags, 165 blocks, 6.0% fragment
ation)
/dev/rra0g: 25130 files, 265403 used, 129448 free (4024 frags, 15678 blocks, 1.0
% fragmentation)
/dev/rra0h: 6056 files, 103058 used, 33446 free (5422 frags, 3503 blocks, 4.0% f
ragmentation)
Thu Feb  1 19:31:25 JST 2007
checking quotas: done.
starting system logger
preserving editor files
clearing /tmp
```

```
standard daemons: update cron accounting          Thu Feb  1 19:31:27 JST 2007
.
starting network daemons: routed namedFeb  1 19:31:27 sandcat named[90]: /etc/na
med.boot: No such file or directory
  inetd printer.
starting local daemons: sendmail.
Thu Feb  1 19:31:28 JST 2007
```

#### 4.3 BSD UNIX (sandcat.felis.carnivora) (console)

```
login: kenrou
Password:
Last login: Thu Feb  1 18:52:42 on console
4.3 BSD Quasijarus UNIX #0: Thu Jul 21 20:17:11 JST 2005
```

Welcome to UNIX!

```
You have mail.
[~/]kenrou% ls
lisp work
[~/]kenrou% cd lisp/maxima/unixport/
[~/lisp/maxima/unixport/]kenrou% ./saved_maxima
AKCL (Austin Kyoto Common Lisp) Thu Feb  1 19:01:45 JST 2007
Maxima 4.155 (with enhancements by W. Schelter).
Type (run) to start Maxima.
```

```
>(run)
Maxima 4.153 Sun Aug 19 12:02:38 CDT 1990 (with enhancements by W. Schelter).
```

```
(C1) integrate(1/(1+x^3),x);
```

$$(D1) \quad - \frac{\frac{2}{3} \log(x^2 - x + 1)}{6} + \frac{\frac{\operatorname{atan}\left(\frac{2x-1}{\sqrt{3}}\right)}{\sqrt{3}}}{\sqrt{3}} + \frac{\log(x+1)}{3}$$

```
(C2) quit();
Bye.
```

```
4.3BSD[~/work/Programs/maxima/]kenrou% maxima
AKCL (Austin Kyoto Common Lisp) Thu Feb 1 19:01:45 JST 2007
Maxima 4.155 (with enhancements by W. Schelter).
Type (run) to start Maxima.
```

```
>(load "cpoly.lisp")
Loading cpoly.lisp
Finished loading cpoly.lisp
T
```

```
>(run)
Maxima 4.153 Sun Aug 19 12:02:38 CDT 1990 (with enhancements by W. Schelter).
(C1) load("eigen.mc");
```

```
batching /home/kenrou/lisp/maxima/share/eigen.mc
(D1)          #/home/kenrou/lisp/maxima/share/eigen.mc
(C2) load("kenrou.mc");
```

```
batching /home/kenrou/work/Programs/maxima/kenrou.mc
(D2)          #kenrou.mc
```

```
(C3) A : matrix([1,0,2],[0,1,2],[2,2,-1]);
          [ 1  0  2 ]
          [      ]
(D3)      [ 0  1  2 ]
          [      ]
          [ 2  2 - 1 ]
```

```
(C4) P : diagonalizer(A);
          [      1      1      2      ]
          [ -----  -----  - ----- ]
          [ SQRT(6)  SQRT(6)  SQRT(6) ]
          [      ]
          [      1      1      1      ]
(D4)      [ -----  -----  ----- ]
          [ SQRT(3)  SQRT(3)  SQRT(3) ]
          [      ]
          [      1      1      ]
          [ -----  - -----  0      ]
          [ SQRT(2)  SQRT(2) ]
```

```
(C5) P . A . transpose(P);
          [ - 3  0  0 ]
          [      ]
(D5)      [ 0  3  0 ]
```



$$\begin{bmatrix} & & & \\ & 0 & 0 & 1 \end{bmatrix}$$

(C6) quit();

Bye.

4.3BSD[~/work/Programs/maxima/]kenrou%

*E-mail address:* kenrou@yo.rim.or.jp