

1 実行環境と起動・終了方法

2006 年度以来、2009・2013 年の機種更新も経て、全学計算機システムの Linux サーバ (icho) に数式処理システム Reduce を導入して、授業受講者に提供してきた。2017 年 3 月末の機種更新にあたり、改めて最新の Linux(ubuntu) 版をインストールしてもらい、全ユーザが利用できる形で公開している。

本資料の作成時点で、基本的な計算機能の動作が継承されていることは確認できているが、特にユーザインタフェースの面では、各利用者の icho への接続環境にも依存するため、本資料と同一の応答が得られない可能性がある。本資料の内容は、「Windows PC 上で “PuTTY” を利用して icho.u.tsukuba.ac.jp に SSH 接続」することにより、コマンドラインモードで動作確認している。実習室の PC 端末あるいは VNC 経由の GUI による利用については、確認が十分にできていない。また、今後も含めて、資料中の実行例には新旧の icho によるものが混在することがある。

2008 年末から、Reduce も Open Source プロジェクトになったため、Windows PC に無料でインストールできる (<http://reduce-algebra.sourceforge.net/>) ようになった。利用は個人の自由であるが、提出用の課題等は上記 icho の上の版で実行した結果に限るものとする。なお、有償だった時代の最新バージョンは “Reduce3.8” であったが、無償化後は、特にバージョン番号をつけず、対応 OS と配布開始日で識別する仕組みになっている。

1.1 起動コマンド

実習室端末のような GUI 環境では、任意のディレクトリから（以下、プロンプトはすべて \$ のみで表示）

```
$ reduce
```

とすると、別ウィンドウで Reduce が立ち上がる。オプションで -w をつけると、

```
$ reduce -w
```

使用中の端末エミュレータ（コマンドプロンプト）内での、コマンドラインモードの起動となる。上述 (PuTTY + Xterm) のような非 GUI 環境では、-w をつけなくても自動的にコマンドラインモードになるようである。

なお、いずれの場合も、オプション -v をつけると、バージョンやメモリ割り当ての情報が詳細に表示される。（次ページ以降の実行例参照。）

1.2 暴走したときの止め方

正常な終了法は §2.1 に示す。エラーメッセージが大量に出て抜け出せないような場合、Reduce に割り込みを掛けるには、以下の実行例に示すように、(1) Control + c (2) Control + \ の 2 方法がある。これらの入力すら受け付けなくなった場合は、Linux のプロセスとして kill するほか、手がないようである。暴走したままにしないよう、logout の前には、\$ jobs および \$ ps で、よく確認すること。

moritsugu.shuichi.fp@icho:~\$ reduce	起動
Reduce (Free CSL version, revision 3844), 13-Jan-17 ...	
1: factorize(x^1000-y^1000);	長くかかる計算
^C	Control + c を入力
+++ Type C to continue, A to abort, X to exit: x	exit を選択
moritsugu.shuichi.fp@icho:~\$ reduce	再起動
Reduce (Free CSL version, revision 3844), 13-Jan-17 ...	
1: <^C>	Control + c を受け付けられない状態
	Control + \ を入力
1: moritsugu.shuichi.fp@icho:~\$	強制終了に成功

1.3 参考文献

小林英恒 高校数学による REDUCE 入門 サイエнтиスト社
広田良吾・伊藤雅明 REDUCE 入門 サイエンス社
森本光生 パソコンによる数式処理 朝倉書店

など（日本語の文献で新しいものは出ていない）。使用機種や Version による細かい違いで、例題どおりにいかないこともある。Ver.3.8 のマニュアルは、前述の sourceforge プロジェクトの Web ページからも参照できる。

2 基本的使い方と実行例

2.1 基本的使い方

前述のように `$ reduce` と入れるとインタプリタが立ち上がる。抜けるには、`bye;` を入れる。入出力には、以下の 3 とおりの形がある。以下の節に対応する実行例を示す。

- (1) 端末から入力、端末に結果を表示 (対話的に利用する場合)
- (2) ファイルから入力、端末に結果を表示 (デバッグ中の場合)
- (3) ファイルから入力、ファイルに結果を保存 (プログラムが完成した場合)

2.1.1 実行例 (1)

ごく単純な問題には対話的に計算を進め、画面の記録をとればよい。（`script` コマンドは適さない。）

```
-----
moritsugu.shuichi.fp@icho:~$ reduce -w -v                      起動 (1)
Codemist Standard Lisp 8.00 revision 3844 for linux-gnu:x86_64: Jan 13 2017
Created: Fri Jan 13 11:08:56 2017

Reduce (Free CSL version, revision 3844), 13-Jan-17 ...
Memory allocation: 121 Mbytes
There are 4 processors available

1: int(1/(x^2+1),x);                                           端末入力

atan(x)                                                         端末表示

2: bye;                                                         終了

End of Lisp run after 0.00+0.03 seconds
moritsugu.shuichi.fp@icho:~$
-----
```

2.1.2 実行例 (2)

前もってプログラムを作成するため、以下の 5 行を適当なディレクトリの下（例えば）`proc01` というファイルに書く。拡張子は不要なので、入力用 / 出力用を区別しながら統一した名前を付けると整理が楽である。

```
f1:=2*x+3*y-5;
f2:=4*x-y+2;
solve({f1,f2},{x,y});
showtime;
;end;
```

これを実行するには、以下のように「`in` コマンド」で読み込む。たいていは、打ち間違いなどがあって、一度ではうまくいかないであろう。エラーメッセージは英語でしか出ないので、結果をよく確認すること。

一般には、プログラムをファイルに書いて、動作が確認できるまで、この方法で編集と実行を繰り返す。ただし、続けて `in` ファイル名; を入力するのではなく、一度インタプリタから抜けて (`bye;`) 再起動させる (`$ reduce`) ことが原則である。（そうでないと、前の結果が残っていて、2 度目は同じ結果が得られるとは限らない。）

```

-----
moritsugu.shuichi.fp@icho:~/r38proc/general$ reduce -w -v      起動 (2)
Codemist Standard Lisp 8.00 revision 3844 for linux-gnu:x86_64: Jan 13 2017
Created: Fri Jan 13 11:08:56 2017

Reduce (Free CSL version, revision 3844), 13-Jan-17 ...
Memory allocation: 121 Mbytes
There are 4 processors available

1: in proc01;
f1:=2*x+3*y-5;
... [1]      ファイルからの入力を指示
              [1]-[5] が読み込まれる

f1 := 2*x + 3*y - 5
... [2]      端末表示

f2:=4*x-y+2;
... [2]

f2 := 4*x - y + 2
... [3]      端末表示

solve({f1,f2},{x,y});
... [3]

      - 1      12
      {x=-----,y=-----}
      14      7
... [4]      端末表示

showtime;
... [4]

Time: 0 ms
... [5]      端末表示

;
... [5]

end;
... [5] (入力ファイル中では同一行)

2: bye;
... [5]      終了

End of Lisp run after 0.00+0.02 seconds
moritsugu.shuichi.fp@icho:~/r38proc/general$
-----

```

2.1.3 実行例 (3)

実行結果をきちんとしたファイルに残すためには、いったん終了 (bye;) し、再起動 (\$ reduce) させたのち、下記の方法 (out/in/shut) で実行させる。3 行目の shut コマンドを忘れて、いきなり bye; で終了しても、大抵は大丈夫のようである。

```

-----
moritsugu.shuichi.fp@icho:~/r38proc/general$ reduce -w -v      再起動
Codemist Standard Lisp 8.00 revision 3844 for linux-gnu:x86_64: Jan 13 2017
Created: Fri Jan 13 11:08:56 2017

Reduce (Free CSL version, revision 3844), 13-Jan-17 ...
Memory allocation: 121 Mbytes
There are 4 processors available

1: out rslt01;
2: in proc01;
3: shut rslt01;
... [1]      出力先ファイルを指定
              入力元ファイルを指定
              出力先ファイルを閉じる

4: bye;
... [1]      終了

End of Lisp run after 0.00+0.02 seconds
moritsugu.shuichi.fp@icho:~/r38proc/general$
-----

```

2.1.4 プログラム例 1

```
%-----  
% Basic Examples (% で始まる行は注釈 - 日本語も多分大丈夫)  
%-----  
% 代入文：文の最後にはセミicolon (;) をつける  
  
f:=x^4+2*x^2-3;  
  
% 因数分解：Factorization  
  
factorize(f);  
  
% 方程式とみて解く：Solving equation  
  
solve(f,x);  
  
% 変数の値に代入：Substitution  
  
sub(x=1,f);  
  
%-----  
% 微分：Differential  
  
df(f,x);  
  
% 積分：Integration  
  
int(f,x);  
  
%-----  
% 行列：Matrix  
  
matrix m(2,2);  
m:=mat((a,b),(c,d));  
  
% 行列式：Determinant  
  
det(m);  
  
% 逆行列：Inverse matrix (ベキ乗演算子は m**(-1) でもよい)  
  
m^(-1);  
  
%-----  
% for 文による繰り返し  
  
for i:=1:10 do  
    if (remainder(i,3)=1) then write i;  
  
% while 文による繰り返し：グループ文 << >> 内の最後の文には ; が不要  
  
k:=1;  
while (k <= 10) do  
    <<write k," = 1 (mod 3)";  
    k:=k+3>>;  
  
%-----  
% C P U時間の累計を表示  
  
showtime;  
%  
% ファイルの最後には、;end; を必ず入れておく  
%  
;end;
```

練習 1 上の例を「(1) 端末入力・端末表示」の方法で実行してみよ。(注釈行は不要。)

2.1.5 プログラム例 2

```
%-----  
% 提出用テストプログラム (空行は任意に入れて構わない)  
%-----  
% f1 と f2 の最大公約数 : GCD  
  
f1:=(x+1)*(x-1);  
f2:=(x+1)*(x+2);  
gcd(f1,f2);  
% f と f' の終結式 : Resultant (第 3 引数を忘れずに)  
  
f:=a*x^2+b*x+c;  
ff:=df(f,x);  
resultant(f,ff,x);  
%-----  
% 連立 1 次方程式の解法 : 結果 sl は 2 重のリスト  
  
f:=3*x+4*y-18;  
g:=2*x-3*y+5;  
sl:=solve({f,g},{x,y});  
% 解の取り出し : first sl は sl の第 1 要素 (結果はリスト),  
%                  part(リスト,n) はリストの第 n 要素,  
%                  rhs = right hand side (右辺)  
  
xx:=rhs part(first sl,1);  
yy:=rhs part(first sl,2);  
% 検算  
  
sub(x=xx,y=yy,f);  
sub(x=xx,y=yy,g);  
%-----  
% 有理数モードで 2 次元表示のテスト (等幅フォントで印刷されているか?)  
  
on rational;  
9973/977*x^89;  
% 有理数モードを終了、実数モードでの計算 : Rounded number  
  
off rational;  
on rounded;  
1/3;  
% 任意桁数の精度指定 : Precision  
  
precision 50;  
pi;  
% 実数モードを終了、複素数モードへ : Complex number  
  
off rounded;  
on complex;  
(2+3*i)/(1-2*i);  
% 複素数モードを終了、有理数モードへ  
  
off complex;  
0.5;  
%-----  
showtime;  
;end;
```

練習 2 [Reduce レポート課題]

上の例 2 を (注釈行も含めて) ファイル中に書き、そこから読み込んで実行させよ。最初は結果を画面に表示させ、うまく行ったら結果をファイルに保存せよ。保存されたものはテキストファイルになるが、プロポーショナルフォントで印刷すると表示が乱れてしまうので、**等幅フォントで印刷**する必要がある。

入力・出力の両方のファイルを印刷し、表紙をつけて、5 月 11 日 (金) 授業開始時に提出すること。前日以前に提出したい場合は 7D214 のドアのポストへ。締切を過ぎたものは受け取らない。第 1 章課題を同時に提出してもよいが、個別に綴じるように。