

## Jによるスマホのシステム処理—ディレクトリ構造をさぐる

西川 利男

JAPLA の先月例会では、私にとっては初めてのスマホ（スマート・フォン）を手にして、メール(G-mail)、テレビ電話(Skype)、インターネット検索(Google)など通常の利用に加えて、J- Console（Android 版）を何とか動かすという奮闘記を報告した[1]。「私にとってのスマホの哲学」などと題したが、パソコンとの違いをいろいろ感じた。とくに、文字を用いた一種のことばであるプログラミング言語（Jに限ったことではないが）によるプログラミングという作業では、入出力にかなりの違和感を持った。

### 1. スマホの内部を知る必要性

スマホをふつうに使うのであれば、その内部を知る必要はない。かえって、内部を隠すことで簡便で安全な、パソコンとは違った便利な道具になったのだといえる。

ところが、プログラミングを行うとなると、どうしてもプログラムがどこに、どう格納されるか、その作成、編集などはどうするか、などを知ることが必要になり、それに関した内部のディレクトリ、ファイルなどの知識が不可欠である。つまり、元々は単にJを動かせばよいだけのつもりが、必要に迫られて、スマホの Android OS のディレクトリ構造をさぐるということになったのである。

幸いなことに、プログラミング言語としてのJには、低レベルのメモリ、ファイルなどを扱う機能が極めて豊富にある。したがって、これを用いて Android OS の仕組みが明らかになった。ちなみに、Android アプリの Office Suite Pro が汎用のエディタとして利用できるが、ファイルのコピーや移動などの操作にはおよそ使い勝手が悪い。

パソコンからインポートしたJプログラムを最初の頃は、J-Console で実行しようとするとうエラーが出て、その理由がわからず弱ってしまった。ところがこれにも、Android の J ファイルを 16 進でダンプした結果、原因をつきとめ修正し、実行することが出来た。ここでも J の低レベルのプログラミング機能が大いに役立った。このようなことで、Android ファイルの操作ツールとして便利に活用した J プログラムについても述べる。

スマホもパソコンとは違う意味で、Jにとってもなかなかの環境であることがわかった。しかし、これをどう活用するかは、今後の課題である。

[1] 西川利男、「スマホ—私にとっての哲学とその上でJを動かす奮闘記」

JAPLA 研究会資料 2014/12/6

## 2. Android OS のディレクトリ構造

```
/(ルート)
|- preload
|- bin
|- ...
|- storage/emulated/0/..... 本体ファイル
|   |- Samsung ..... 機器操作のファイル
|   |- Android ..... アプリなど多数のファイル
|   |- TMemо ..... メモ帳で使用するファイル
|   |- Documents ..... Office Suite Pro で使用するファイル
|   |- DCIM ..... カメラで撮影した写真の画像ファイル
|   |- Download ..... 添付ファイルなどでダウンロードしたファイル
|   |- j701-user ..... Jで使用するファイル
|   |- projects
|   |- break
|   |- config
|   |- snap
|   |- temp
|   |- profile.txt
|   |- trigmath.txt
|   |- hexcode.txt
|   |- unbox.txt
|   |- ntest.txt
|   |- ntest0.txt
|   |- ntest1.txt
|   |- ...
|- dev
|- etc
|- ...
```

一般に OS ではファイルはトリー構造をもったディレクトリ（フォルダーとも呼ぶ）の下に格納、管理されている。Samsung Galaxy CORE2 の Android OS では上のようになっている。このうちユーザに開放された領域は/storage/emulated/0/であり、「本体」と称されている。さらに J に関する部分は/storage/emulated/0/j701-user/の下に格納される。

## 3. J と Android

### 3. 1 J のファイル操作の基本プリミティブ

0!:1 script 右引数のファイルをJのプログラムとしてコンパイルし、取り込む。

1!:0 dir 右引数の書式（/ルート,\*すべて,\***.txt**テキストファイル）に従ってディレクトリとファイル名を表示する。

1!:1 fread 右引数のファイル名を読み込み、内容の文字列を返す。

1!:2 fwrite 左引数の文字列を値として、右引数のファイル名に書き込む。

1!:3 fappend 左引数の文字列を値として、右引数のファイル名に追加、書き込む。

これらは、Jのファイル**profile.txt**にプログラムとして定義した。したがって、Jセッションの最初に**load 'profile.txt'**とすることで、文字コマンドとして使用できる。

これらの基本プリミティブを利用して、いろいろ試行錯誤で調べた末、上のような**Android**ディレクトリ構造を明らかにすることができた。

### 3. 2 Jによるディレクトリ構造の探索の実際

初めに、profile スクリプトを読み込む。

```
load 'profile.txt'
```

以下のプログラムが定義されている。

```
names ''
dir dird dirj dirs fappend fcopy ferase fread freadd fwrite newj
script
```

まず、Android OS のルートから調べる。

```
dir '/*'
```

すると、画面の数ページにわたって、多数の項目があるが、多すぎて見るできない。

```
$dd
45 20
10 { dd 最初から 10 項目を表示してみる。
```

```
preload
bin
res
sdcard
efs
productinfo
storage
config
cache
acct
```

```
dir '/storage/emulated/0/*'
```

Android 本体としてユーザに開放されている部分を表示する。(一部表略)

```
Download
DCIM
Documents
Android
-----
TMemo
j701-user
-----
Sounds
```

これは簡単に行うため、コマンド dirs として定義した。

さらに、Jに関係した

```
/storage/emulated/0/j701-user/
```

フォルダーの下のものを見てみよう。これも手軽におこなうため、`dirj`として定義してある。

```
dirj ‘  
projects  
break  
config  
snap  
temp  
profile.txt  
cross.txt  
ntest1.txt  
ntest.txt  
unbox.txt  
hexcode.txt  
fileadj.txt  
trigmath.txt
```

以上のようにして、先の Android のディレクトリ構造を明らかにしたのである。

## 4. J Console による J のプログラミング

### 4. 1 J の新規ファイルの作成

J Console 上で対話型の J の実行には、そのままキー入力すればよい。ところで、ユーザがプログラムを作成しても、それをファイルにセーブできなくては、プログラミング言語としてあまり意味はない。しかし、このままでは出来ない。そのためのコマンド `newj` を作った。これはパソコン J において、新規にスクリプトファイル `ijs` を作ることに相当する。

これには次のようにして行う。J のスクリプト用のファイルを、たとえば `'ntest1.txt'` として作る。

```
newj ‘ntest1.txt’
```

ここで、ファイル拡張子を `txt` とすることが必要である。Android のアプリ Office Suite Pro で本体ファイルの中の `j701-user` フォルダ中の `ntest1.txt` を開くと一行のコメント文が入ったテキスト入力画面が現れる。ここに J のプログラムを入力する。私は入力キーボードとして、標準の Samsung QWERT キーボードを便利に利用している。

さて、J のプログラムを入力、編集、セーブして J のスクリプトファイルとするつもりでいたが、思わぬトラブルがあった。これについては次の節で述べる。

#### 4. 2 Jによるエラー発生プログラムの確認と修正作業の実際

エラーが出る元のプログラムファイルをntest1.txtとし、その内容は次のものとする。

```
add = : 3 : 0
+ / y
)
```

これを load 'ntest1.txt'

のようにコンパイルすると

Spelling error がでる。

つぎのようにして、エラーの原因を突き止め、修正してntest2.txtとした。

(1) load 'profile.txt'

(2) da = : fread <' ntest1.txt'

\$da => 24 つまり、ファイルには文字列として格納されている。

(3) load 'hexcode.txt' の中のJプログラムhdumpにより、文字列を16進でダンプする。

```
hdump da
EF BB BF 0A 61 64 64 20 3D 3A 20 33 20 3A 20 30 0A 2B 2F 20 79 0A 29 0A
      LF a d d _ = : _ 3 _ : _ 0 LF + / _ y LF ) LF
```

のようになる。

つまり、Androidのファイルは最初に3バイトの文字が付け加わっている。ファイル識別のためと思われるが、意味はわからない。そして、これはOffice Suite Proの削除キーでは取り除くことができない。

(4) Jにより次のように取り除き修正し、これをntest2.txtとした。

```
daa = : 3 { da
daa fwrite <' ntest2.txt'
```

(5) 修正されたプログラムを実行する。

```
load 'ntest2.txt'
```

エラーは出ないでOK

```
names ''
add
values = : i. 11
values
0 1 2 3 4 5 6 7 8 9 10
add values
```

55

## 5. J on スマホをどう活用するか

ーポケット PC の上で J を動かしたその昔の繰言と高性能電卓としての現在スマホー

今、このスマホを私はベージュの模造皮のケースに入れて持ち歩いている。このケースには、最近まで HP(Hewlett Packard) Palm Top PC LX-95 が入っていた。ここにあったメモノートを見たら 1991 年となっていた。LX-95 は当時最新のポケット PC で、Lotus 123 を標準装備し、MSDOS も動かせた。そこで、発表されたばかりの J の PC 版をやつとのことで動かしていたのである。

LX-95 には、別に標準装備の HP Financial Calculator というのがあり、その中に Solver and Function Plotting なる機能があつて、数学の式を解いたり、関数をプロット表示することができた。いわば、プロット付関数電卓である。

現在とりあえずは、私としては J on スマホを最新のプロット付関数電卓として便利に使っている。そのようすを紹介する。問題例は LX-95 に載っていた次の 3 次方程式

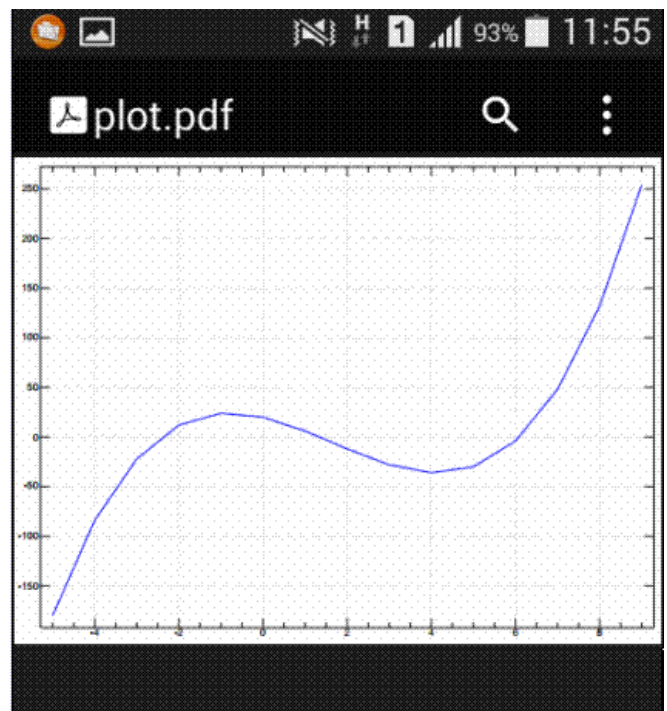
$$y = x^3 - 5x^2 - 10x + 20$$

の解とそのプロット表示である。

```
J Console
load 'mathplot.txt'

> {: p. 20 _10 _5 1
6.1017 _2.44326 1.34156

20 _10 _5 1 pfplot _5 + i.15
```



## J-701 Console 上の J プログラム・スクリプトのソース・リスト

```
NB. profile.txt =====  
dir =: ((>@{"(1)})@(1!:0))@boxopen  
fread =: 1!:1@boxopen  
fwrite =: 1!:2  
fappend =: 1!:3  
ferase =: 1!:55@boxopen  
dirs =: 3 : 0  
dir '/storage/emulated/0/', y  
)  
dirj =: 3 : 0  
dir '/storage/emulated/0/j701-user/', y  
)  
dird =: 3 : 0  
dir '/storage/emulated/0/Download/', y  
)  
freadd =: 3 : 0  
fread <'!storage/emulated/0/Download/', y  
)  
  
script =: 0!:1@boxopen  
  
fcopy =: 3 : 0  
:  
fila =. x  
filb =. y  
text =. fread fila  
text fwrite <filb  
)  
  
newj =: 3 : 0  
head =. 'NB. '  
(head, y) fwrite <y  
)
```



NB. HEX DEC

h=: '0123456789ABCDEF'

dfh=: 16&#.@(h&i.) f.

hfd=: dfh ^:\_1 f.

val =: a. & i.

chr =: val ^:\_1

hex =: &. dfh

hdump =: }:@,@(,"1' ""\_>@hfd@val f.

dump =: 3 : 0

text =. y

texta =. ,(,. text),"(1)' '

texta ,: hdump text

)

wr =: 1!:2&2

DA =:'abc';'d';'efgh';'ijklmn';'pqr';'stuv';'xy';'z'

ubf =: 3 : 0

3 ubf y

:

C =. x

D =. y

N =. # y

M =. >. /># L:0 D

D0 =. > D, 5#<' '

D1 =. D0, "(1) ', '

M2 =. 2 + M

NB. wr 'M2=' , ": M2

NB. wr 'D=' , ": N

```
D2 = ((>. N%C), (C*M2))$, D1
_2 }."(1) D2
)
```

```
dir = ((>@{"(1)} @ (1!:0))@boxopen
```

```
sin =: 1&o.
```

```
cos =: 2&o.
```

```
tan =: 3&o.
```

```
sinh =: 5&o.
```

```
cosh =: 6&o.
```

```
tanh =: 7&o.
```

```
arcsin =: _1&o.
```

```
arccos =: _2&o.
```

```
arctan =: _3&o.
```

```
arcsinh =: _5&o.
```

```
arccosh =: _6&o.
```

```
arctanh =: _7&o.
```

```
pi =: 1p1
```

```
rfd =: *&(pi%180)
```

```
dfr =: *&(180%pi)
```

```
sind =: sin@rfd
```

```
cosd =: cos@rfd
```

```
tand =: tan@rfd
```

```
exp =: ^
```

```
ln =: ^.
```

```
log =: 10&^.
```

NB. drop 3 bytes from the head of file  
require 'hexcode.txt'

```
fadj =: 3 : 0
```

```
Qfile =. y
da =. fread file
dax =. 3 }. da
dax fwrite < file
)
```

NB. android.txt =====

NB. insert CR before LF of each line.

```
adjcrlf =. 3 : 0
d0 =. y.
d1 =. <,. 1 d0
d2 =. CR ,L:0 d1
d3 =. ,> d2
)
```

```
fread =. 1!:1 @ boxopen
```

```
fwrite =. 1!:2
```

```
fappend =. 1!:3
```

```
dir0 =. 1!:0
```

```
dir =. ((>@{. "(1))@(1!:0))@boxopen
```

NB. Search Files among the Directory =====

```
wr =. 1!:2&2
```

```
rd =. 1!:1
```

```
dirroot =. 3 : 0
```

```
d0 =. dir '/*'
```

```
d1 =. (0, 0)$ "
```

```
DIM =.(0, 0)$ "
```

```
i =. 0
```

```

while. i < {. $d0
do.
NB. if. 0 = 10|i do. rd 1 end.
NB. wr i{d0
if. -. '! e. i{d0 do. d1 =. d1, i{d0 end.
i =. i + 1
end.
NB. wr '*****'
wr $d1
i =. 0
while. i < {. $ d1
do.
NB. if. 0 = 10|i do. rd 1 end.
NB. wr (":i)',', i{d1
wr (":i)',', i{d1
DIM =. DIM, i{d1
i =. i + 1
end.
DD =: DIM
*****'
)

deltrailsp =: 3 : 0
|. ((' = q) i. 0) }. q =. |. y.
)

```

```

dirsearch =: 3 : 0
NB. DIR =. (y. { DD) -. ''
DIR =. deltrailsp y. { DD
wr 'directory: ', '/', DIR, '/'*
DD =. dir '/', DIR , '/'*
wr (": { . $DD), ' items'
i =. 0
while. i < {. $DD
do.

```

```
if. 0 = 10|i do. rd 1 end.  
wr i{DD  
i =. i + 1  
end.  
***  
)
```