

コードレビュー

Vol.16

USP 研究所技術研究員
written by 大内智明

CODE Review

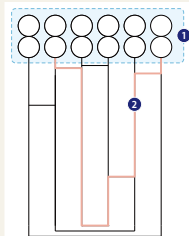
今回は、PDFファイルの作成方法について紹介します。

PDF ファイルを作成

業務用使用する帳票などの資料は、PDF ファイルで作成することがよくあります。PDF ファイルは、フリーのPDF リーダーがあれば、パソコン・携帯電話・タブレットなどのほぼすべての機器で閲覧することができます。

ユニケーシで PDF ファイルを作成する方法について説明します。ただし、今回紹介する例は、業務で作成するような帳票ではなく、線や円を使用した「アミダくじ」を紹介します(図1)。

図1 アミダくじ



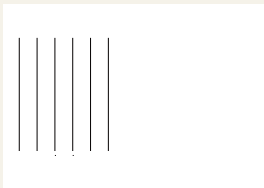
Powered by I.Ayuse

※ ここで紹介する「アミダくじ」は、変則的な仕様となっています。

- ① この「アミダくじ」は、上部に入力が1線あたり2つあります。つまり、2x6=12人分の「アミダくじ」です。
- ② この「アミダくじ」は、上(開始)から下を通して、再度上(終了)にいきます(朱色部分)。

図2 描画とスクリプトコマンド

① 線を引く



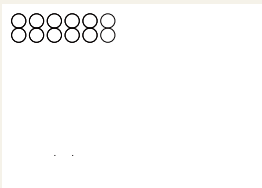
line 18 37 18 164

line 38 37 38 164

⋮

直線を始点から終点まで引く
スクリプトコマンド

② 円を引く



en 18 13 8

en 18 29 8

⋮

中心 (x,y) と半径から円を引く
スクリプトコマンド

③ 文字列を書く



Powered by I.Ayuse

text 291 193 right top gothic 15 black _ _

Powered_by_I.Ayuse

⋮

指定テキスト文字を指定場所を書く
スクリプトコマンド

リスト1 PDF 作成するシェル

```

1 #!/bin/bash -xv
2 #
3 # Usage : AMIDATEST <sankasu>(4~28までの4の倍数)
4 #
5 # Written by I.Ayuse(usp-lab.com) /Date : 14 Oct 2011
6
7 # 走行ログの設定
8 echo "/tmp/log.$(basename $0)" &> /dev/null
9 exec 2> /tmp/log.$(basename $0)
10
11 # 変数の定義
12 tmp=/tmp/tmp$$
13 sankasu=$1
14 kumisu=$(expr $sankasu / 2)
15
16 #####
17 # 縦線を組数分引く
18 #####
19 tatel_xpos=18
20 tatel_sypos=37
21 tatel_eypos=164
22 # 組数分for文を回す
23 for kms in $(seq ${kumisu}) ; do
24     echo "${tatel_xpos} ${tatel_sypos} ${tatel_xpos} ${tatel_eypos}"
25     # 1:縦線x軸始点 2:縦線y軸始点 3:縦線x軸終点 4:縦線y軸終点
26     tatel_xpos=$(( ${tatel_xpos} + 20 ))
27 done > $tmp-hikisu_line.1
28 [ $? -eq 0 ] || exit 1
29 #####
30
31 #####
32 # 2本の縦線の一番下同士を繋げる
33 #####
34 tunagi_tatelno=$(expr $sankasu / 4 )
35 tunagi_lsu=$(expr $sankasu / 4 )
36 tunagi_yokol_sypos=164
37 tunagi_yokol_eypos=164
38 tatel_eypos=164
39 :> $tmp-hikisu_line.2
40 # 1:縦線x軸始点 2:縦線y軸始点 3:縦線x軸終点 4:縦線y軸終点
41 rank,$tmp-hikisu_line.1 |
42 msort,key=2N |
43 rank > $tmp-tunagi_no
44 for tngls in $(seq ${tunagi_lsu}) ; do
45     # 繋ぎの横線
46     # $tmp-tunagi_no
47     # 1:行番号 2:行番号(逆順) 3:縦線x軸始点 4:縦線y軸始点 5:縦線x軸終点 6:縦線y軸終点
48     self,1 5 $tmp-tunagi_no |
49     selr,1 ${tunagi_tatelno} > $tmp-tunagi_no1
50     # $tmp-tunagi_no1
51     # 1:行番号 2:横線x軸終点
52     self 2 3 $tmp-tunagi_no |
53     selr 1 ${tunagi_tatelno} |
54     # 1:行番号(逆順) 2:横線x軸始点
55     join1,key=1 - $tmp-tunagi_no1 |
56     # 1:行番号 2:横線x軸始点 3:横線x軸終点

```

rankコマンドはユニケージコマンド、画面1参照

msortコマンドはユニケージコマンド、sort コマンドの高速版

self コマンドはユニケージコマンド、フィールド選択して出力します

selr コマンドはユニケージコマンド、フィールドの完全一致を抽出します

join1はユニケージコマンド、画面2参照

```

57 delf 1 |
58 awk '{print $1,""$tunagi_yokol_sypos"$", $2,""$tunagi_yokol_eypos"$"}' |
59 tee $tmp-tunagi_tatel >> $tmp-hikisu_line.2
60
61 # 繋ぎの縦線
62 # $tmp-tunagi_tatel
63 # 1:縦線x軸始点 2:縦線y軸始点 3:縦線x軸終点 4:縦線y軸終点
64 # 2本の縦線の左側
65 awk '{print $1,""${tatel_eypos}"", $1,""${tunagi_yokol_eypos}""}' $tmp-tunagi_tatel >> $tmp-hikisu_line.2
66 # 2本の縦線の右側
67 awk '{print $3,""${tatel_eypos}"", $3,""${tunagi_yokol_eypos}""}' $tmp-tunagi_tatel >> $tmp-hikisu_line.2
68 tunagi_yokol_sypos=$(( ${tunagi_yokol_sypos} + 4 ))
69 tunagi_yokol_eypos=$(( ${tunagi_yokol_eypos} + 4 ))
70 tunagi_tateln=$(( ${tunagi_tateln} - 1 ))
71 done
72 [ $? -eq 0 ] || exit 1
73 #####
74
75 #####
76 # 事前に横線のy軸を2mm区切りで61本分用意する
77 #####
78 jyunbi_yokolsu=61
79 yokol_ypos=43
80 for tngls in $(seq ${jyunbi_yokolsu}); do
81     echo "${yokol_ypos} ${yokol_ypos}"
82     # $tmp-yokol_yposall
83     # 1:横線y軸始点 2:横線y軸終点
84     # y軸始点を2mmずつ増加させる
85     yokol_ypos=$(( ${yokol_ypos} + 2 ))
86 done > $tmp-yokol_yposall
87 [ $? -eq 0 ] || exit 1
88 #####
89
90 #####
91 # 縦線の間ごとに横線を1本ずつランダムに追加する
92 #####
93 yokol_sxpos=18
94 yokol_expos=38
95 rand_lsu=$((expr ${kumisu} - 1 ))
96 for tngls in $(seq ${rand_lsu}); do
97     echo "${yokol_sxpos} ${yokol_expos}"
98     yokol_sxpos=$(( ${yokol_sxpos} + 20 ))
99     yokol_expos=$(( ${yokol_expos} + 20 ))
100 done > $tmp-yokol_xposall
101 [ $? -eq 0 ] || exit 1
102 # y軸とjoinするために行番号を付与する
103 # $tmp-yokol_xposall
104 # 1:横線x軸始点 2:横線x軸終点
105 rank $tmp-yokol_xposall |
106 maezero 1.2 > $tmp-yokol_xposall_gyo
107 # 1:行数 2:横線x軸始点 3:横線x軸終点
108 [ $(plus ${PIPESTATUS[@]}) -eq 0 ] || exit 1
109
110 # 事前に用意した横線のy軸に乱数を付与する
111 seq 1 1000 | ransu 61 |
112 # $tmp-yokol_yposall
113 # 1:横線y軸始点 2:横線y軸終点
114 ycat $tmp-yokol_yposall - |

```

2

```

115 # 1:横線y軸始点 2:横線y軸終点 3:乱数
116 msort key=3n |
117 delf 3 |
118 # x軸とjoinするために行番号を付与する
119 rank |
120 maezero 1.2 |
121 # 行番号でjoin
122 join1 key=1 $tmp-yokol_xposall_gyo |
123 # 1:行番号 2:横線x軸始点 3:横線x軸終点 4:横線y軸始点 5:横線y軸終点
124 self 2 4 3 5 > $tmp-hikisu_line.3
125 # 1:横線x軸始点 2:横線x軸終点 3:横線y軸始点 4:横線y軸終点
126 [ $(plus ${PIPESTATUS[@]}) -eq 0 ] || exit 1
127 #####
128
129 #####
130 # lineの文字列を一括で付与
131 #####
132 cat $tmp-hikisu_line.1 $tmp-hikisu_line.2 $tmp-hikisu_line.3 > $tmp-hikisu_line.1-2-3
133 echo "line" |
134 joinx- $tmp-hikisu_line.1-2-3 > $tmp-hikisu.line
135 #####
136
137 #####
138 # 名前を書き入れるための円を追加する
139 #####
140 en_xpos=18
141 en1_ypos=13
142 en2_ypos=29
143 en_hankei=8
144 := $tmp-hikisu.en
145 for tngls in $(seq ${kumisu}) ; do
146     echo "en $en_xpos $en1_ypos $en_hankei" >> $tmp-hikisu.en
147     echo "en $en_xpos $en2_ypos $en_hankei" >> $tmp-hikisu.en
148     en_xpos=$(( $en_xpos + 20 ))
149 done
150 [ $? -eq 0 ] || exit 1
151 #####
152
153 #####
154 # 線の太さを一括で付与
155 #####
156 cat $tmp-hikisu.line $tmp-hikisu.en > $tmp-hikisu.line_en.1
157 echo "2" |
158 joinx $tmp-hikisu.line_en.1 - > $tmp-hikisu.line_en
159 #####
160
161 #####
162 # Powered by I.Ayuse を追加する
163 #####
164 echo "text 291 193 right top gothic 15 black _ _ Powered_by_I.Ayuse" >> $tmp-hikisu.text
165 #####
166
167 #####
168 # 結果をPDF化
169 cat $tmp-hikisu.line_en $tmp-hikisu.text > $tmp-hikisu.result
170 wpdf -p a4y $tmp-hikisu.result > /home/usp/work/amida.pdf
171 #####
172

```

2

3

4

5

```
173 # 終了
174 rm -f $tmp-*
175 exit 0
```

画面1 rank

ランクを付与します

```
$ cat data
JPN 杉山_____ 26
JPN 崎村_____ 27
JPN 梶川_____ 27
$ rank data
1 JPN 杉山_____ 26
2 JPN 崎村_____ 27
3 JPN 梶川_____ 27
```

画面2 join1

トランザクションファイルにマスターファイルを連絡します

```
$ cat master
0000003 杉山_____ 26 F
0000005 崎村_____ 50 F
$ cat keihi
20070405 0000003 600
20070401 0000005 250
20070402 0000005 450
20070402 0000007 210
$ join1 key=2 master keihi
20070405 0000003 杉山_____ 26 F 600
20070401 0000005 崎村_____ 50 F 250
20070402 0000005 崎村_____ 50 F 450
```

画面3 maezero

前ゼロを埋める

```
$ cat data
12 345 6789
$ maezero 1.5 2.6 data
00012 000345 6789
```

画面4 ycat

ファイルの横連結

```
$ cat file1
0000000 浜地_____ 50 F
0000001 鈴田_____ 50 F
$ cat file2
0000000 91 59 20 76 54
0000001 46 39 8 5 21
$ ycat file1 file2 > file3
$ cat file3
0000000 浜地_____ 50 F 0000000 91 59 20 76 54
0000001 鈴田_____ 50 F 0000001 46 39 8 5 21
```

画面5 joinx

2つのファイルを総掛けで連結します

```
$ cat data1
1 農業
2 工業
3 サービス業
$ cat data2
1 東京
2 大阪
3 名古屋
$ joinx data1 data2
1 農業 1 東京
1 農業 2 大阪
1 農業 3 名古屋
2 工業 1 東京
2 工業 2 大阪
2 工業 3 名古屋
3 サービス業 1 東京
3 サービス業 2 大阪
3 サービス業 3 名古屋
```

コードの見どころ

- [1] アミダくじの縦線を引きます。
(① 16～29行目まで)
- [2] 縦線に合わせて横線(底線・中間線)を引きます。
(② 31～135行目まで)
- [3] 1本の縦線に対して上部に円を2つ引きます。
(③ 137～151行目まで)
- [4] テキスト文字を追加します。(④ 161～165行目まで)
- [5] PDFファイルを作成します。(⑤ 167～171行目まで)

まとめ

ユニケージでは、線・枠・テキスト文字の最小部品の組み合わせから、PDF ファイルを作成しています。部品の組み合わせによっては、多様なニーズに合わせたグラフや表の作成も可能です。ここでは、紹介できませんが、数十種類のグラフ混じりの帳票出力や、日々変化する品目の平均価格を表す市況情報のグラフ作成に使用されている実績があります。