

■ MultiTask on Notebook, Sample v.0

■ ノートブックによる擬似的マルチタスク

Mathematicaはタスクスケジューリングによりマルチタスクを実行できるが、さらにノートブック上では、マウスやキーボードのイベントやタイマー・イベントを受け取る各種タスクを同時実行できる。Dynamicにより変数をラップすることによって、タスクは擬似的にそれぞれが独立して動くことができる。ここで擬似的としたのは、現在のところ、各タスクの優先順序を指定することはできず、また、イベントに対する処理速度を保証できないためである。

ここでは、秒針のみの時計と、電卓の二つのサンプルタスクを示し、それぞれ独立に実行できることを確認する。

■ 電卓のサンプル

逆ポーランド式の電卓である。

局所変数の定義リストには関数名も含む必要がある。

```
ClearAll["Global`*"];
```

```

alarm = None;
DynamicModule[{x = 0, y, z, xreg},
  eflag = False;
  xreg[n_] := If[eflag, (x = n; eflag = False), (x = (10 x + n))];
  a00 = Button[Dynamic@x, BaseStyle → {Bold}, Appearance → "Palette", Alignment → Right,
    ImageMargins → 1, Enabled → False, Background → Dynamic@alarm];
  a11 = Button["Clr", x = 0, Appearance → "Palette"];

  a54 = Button[".", Appearance → "Palette"];
  a53 = Button[0, xreg[0], Appearance → "Palette"];
  a41 = Button[1, xreg[1], Appearance → "Palette"];
  a42 = Button[2, xreg[2], Appearance → "Palette"];
  a43 = Button[3, xreg[3], Appearance → "Palette"];
  a31 = Button[4, xreg[4], Appearance → "Palette"];
  a32 = Button[5, xreg[5], Appearance → "Palette"];
  a33 = Button[6, xreg[6], Appearance → "Palette"];
  a21 = Button[7, xreg[7], Appearance → "Palette"];
  a22 = Button[8, xreg[8], Appearance → "Palette"];
  a23 = Button[9, xreg[9], Appearance → "Palette"];

  a44 = Button["+", (x = x + y; y = x; eflag = True), Appearance → "Palette"];
  a14 = Button[" - ", x = y - x; y = x; eflag = True, Appearance → "Palette"];
  a24 = Button[" / ", x = y / x; y = x; eflag = True, Appearance → "Palette"];
  a34 = Button["*", (x = y * x; y = x; eflag = True), Appearance → "Palette"];

  a51 = Button["Enter", (y = x; eflag = True), Appearance → "Palette"];
  a12 = Button[" ← ", (x = Quotient[x, 10]; eflag = True), Appearance → "Palette"];
  a13 = Button[" ± ", x = -x, Appearance → "Palette"];
  Grid[
    {{a00, SpanFromLeft, SpanFromLeft, SpanFromLeft},
     {a11, a12, a13, a14},
     {a21, a22, a23, a24},
     {a31, a32, a33, a34},
     {a41, a42, a43, a44},
     {a51, SpanFromLeft, a53, a54}},
    Spacings → {0, 0}, Frame → All]
]

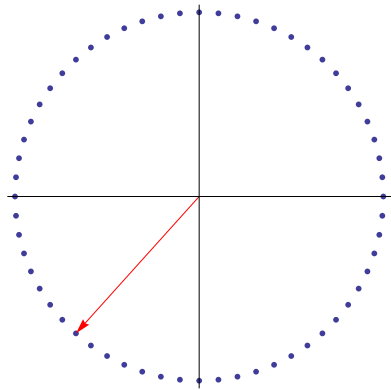
```

0			
Clr	←	±	-
7	8	9	/
4	5	6	*
1	2	3	+
Enter	0	.	

■ 秒針時計

ここでは時計の秒針のみを表示するサンプルを提示する。

```
s = N[Table[{Sin[2 Pi k / 60], Cos[2 Pi k / 60]}, {k, 0, 59}]];
Show[{ListPlot[s, AspectRatio -> 1.0, Ticks -> None], Graphics[
  {Red, Arrow[{0, 0}, Dynamic[s[[t = Clock[{1, 60, 1}, 60]]]]}], ImageSize -> 200]
```



```
Dynamic[If[t > 30, alarm = RGBColor[1, 0, 0, t / 240], alarm = RGBColor[0, 1, 0, t / 240]]]
```

```
RGBColor[1, 0, 0,  $\frac{37}{240}$ ]
```

■ マルチタスク機能と組み込みソフトウェアとしてのMathematicaについて

組み込みソフトウェアに基本的に要請されることは、システムが外部からのイベントを受け取ってこれに適切に対応できることであり、外部イベントに対応するためには、マルチタスクの実行環境が必須である。このような環境としてMathematicaを考えた場合、一般的な組み込みOSと比べてリアルタイム要求に応える機能は十分ではない。しかし、組み込みソフトウェアであっても、厳格なリアルタイム対応が必要とされない場面もあり得る。このような場面に対しては、Mathematicaの得意とする複雑なデータ処理と結果の表現力を活かすことが可能である。