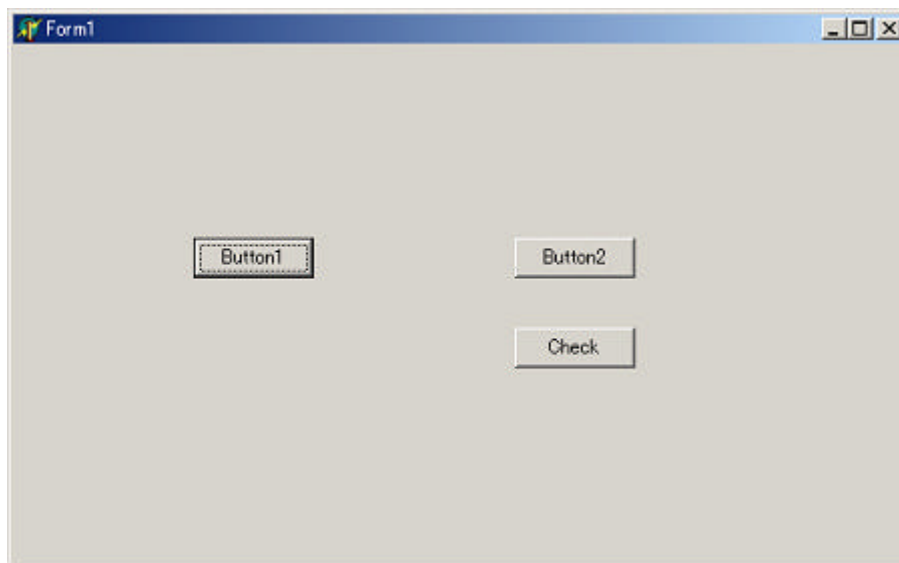


## Application.ProcessMessages

Delphi によって心理学実験のプログラムを作成する場合、反応待ちのループ内において Application.ProcessMessages を実行してボタンやキー押しなどのイベントの生起を待つことがあります。このとき、反応時間の測定を行う場合は、ProcessMessages の実行時間のチェックを行っておく必要があります。この ProcessMessages の実行時間の簡単な測定を行うプログラムとして PCkProcMssg1.dpr を用意しました。

PCkProcMssg1.dpr を実行すると次図のフォームが表示されます。



「Button1」のクリックは ProcessMessages 手続きの単なる呼び出しにおける実行時間、「Button2」のクリックはボタン Check の OnClick イベントの処理を伴う場合の ProcessMessages の実行時間を測定するためのものです。

Button1 の OnClick イベントハンドラはリスト 1 のようになっています。

リスト 1 ProcessMessages の呼び出し処理時間

```

procedure TForm1.Button1Click(Sender: TObject);
const N = 1000;
var f : TextFile;
    x : array[1..N] of extended;
    i : Longint;
    t1, t2 : extended;
begin
  // データ書き出し用ファイル
  with OpenDialog1 do
  begin
    Title:='Output File';
    if not execute then exit;
    AssignFile(f, FileName);
    Rewrite(f);
  end;

  Button1.Enabled:=false;
  Button2.Enabled:=false; UpDate;

  i:=1; // 計時回数のカウント用変数
  repeat

```

```

t1:=MyTime;
Application.ProcessMessages;
t2:=MyTime;
x[i]:=t2-t1;    // ProcessMessage の実行時間
i:=i+1;
until i > N;

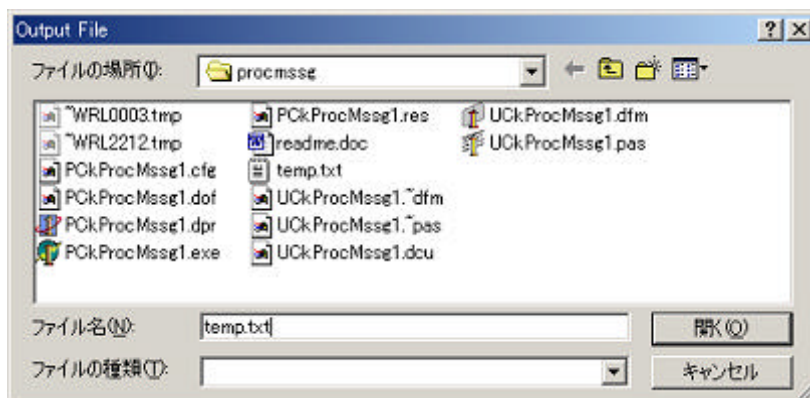
// 実行時間が1ミリ秒以上の場合を書き出す
for i:=1 to N do
  if x[i] > 1.0 then writeln(f, i:5, x[i]:20:3);

// 全データの書き出し
writeln(f);
for i:=1 to N do
  writeln(f, i:5, x[i]:20:3);

CloseFile(f);
Close;
end;

```

Button1 をクリックすると計時データを書き出す出力用ファイルの名前の設定を求めるダイアログボックスが表示されます。



出力用ファイルにはテキストファイルとしてデータが書き出されるので、プログラムの実行終了後エディタで開いて見ることができます。

ファイル名の設定後「開く」ボタンをクリックするとリスト1の手続きの実行は瞬時に終わり、プログラムの実行終了となります。実行終了後、出力用ファイルとして設定したテキストファイルを開くとリスト2のようになっています。

リスト2 計時結果 ( ProcessMessages の呼び出しのみ )

1	10.000
200	10.000
700	10.000
1000	10.000
1	10.000
2	0.000
3	0.000
4	0.000
5	0.000
6	0.000
7	0.000
8	0.000
9	0.000
.	.
.	.
.	.

995	0.000
996	0.000
997	0.000
998	0.000
999	0.000
1000	10.000

リストの最初の部分は0ミリ秒を超える値のみの書き出しです。左側の数値が測定の順番を示し、右側の数値が ProcessMessage の処理時間をミリ秒単位で表しています。0ミリ秒を超える値の書き出し後、1番目から1000番目までの測定値が順番に書き出されています。

Button2 の OnClick イベントハンドラはリスト3のようになっています。

リスト3 イベントの処理を伴う ProcessMessages の実行時間の計時

```

procedure TForm1.Button2Click(Sender: TObject);
const N = 20;
var f : TextFile;
    x, y, z : array[1..N] of extended;
    i : Longint;
    t1, t2 : extended;
begin
  with OpenDialog1 do
    begin
      Title:='Output File';
      if not execute then exit;
      AssignFile(f, FileName);
      Rewrite(f);
    end;

    Button1.Enabled:=false;
    Button2.Enabled:=false; UpDate;

    i:=1;
    repeat
      ck:=false; // ボタンクリックイベントのチェック用変数
      t1:=MyTime;
      Application.ProcessMessages;
      t2:=MyTime;
      if ck then // ボタンクリックの処理で ck が true に設定される
        begin
          x[i]:=t2-t1;
          y[i]:=tck-t1;
          z[i]:=tck1-t1;
          i:=i+1;
        end;
    until i > N;

    writeln(f);
    for i:=1 to N do
      writeln(f, i:5, x[i]:20:3, y[i]:20:3, z[i]:20:3);

    CloseFile(f);
    Close;
end;

```

リスト3では「Check」ボタン、ButtonCk、の OnClick イベントの処理を伴う場合の ProcessMessages の処理時間を計時するようになっています。「Check」ボタンの OnClick イベントハンドラはリスト4のようになっています。

## リスト4 「Check」ボタンのイベントハンドラ

```

procedure TForm1.ButtonCkClick(Sender: TObject);
var t : extended;
begin
    tck:=MyTime;

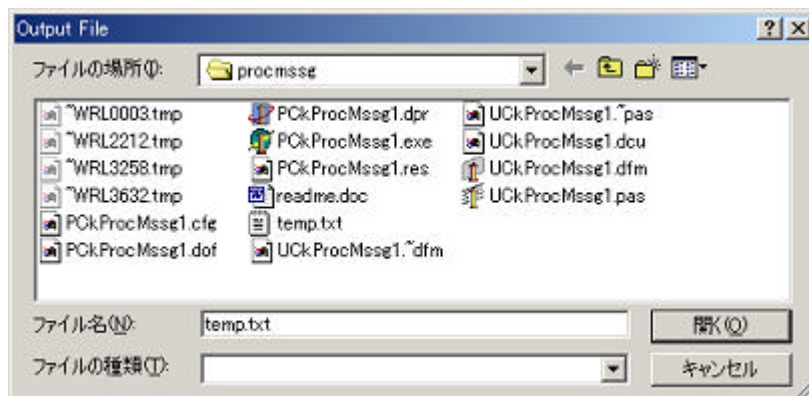
    // ボタンクリック回数の表示
    icount:=icount+1;
    ButtonCk.Caption:='Check- '+IntToStr(icount);

    // ck を true に設定する
    ck:=true;

    tck1:=MyTime;
end;

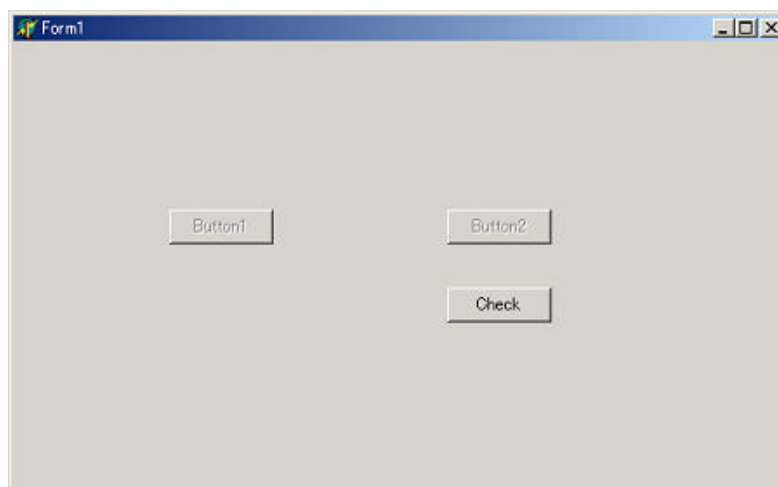
```

「Button2」ボタンをクリックすると次図のダイアログボックスが表示され、計時データを書き出す出力用ファイルの名前の設定が求められます。

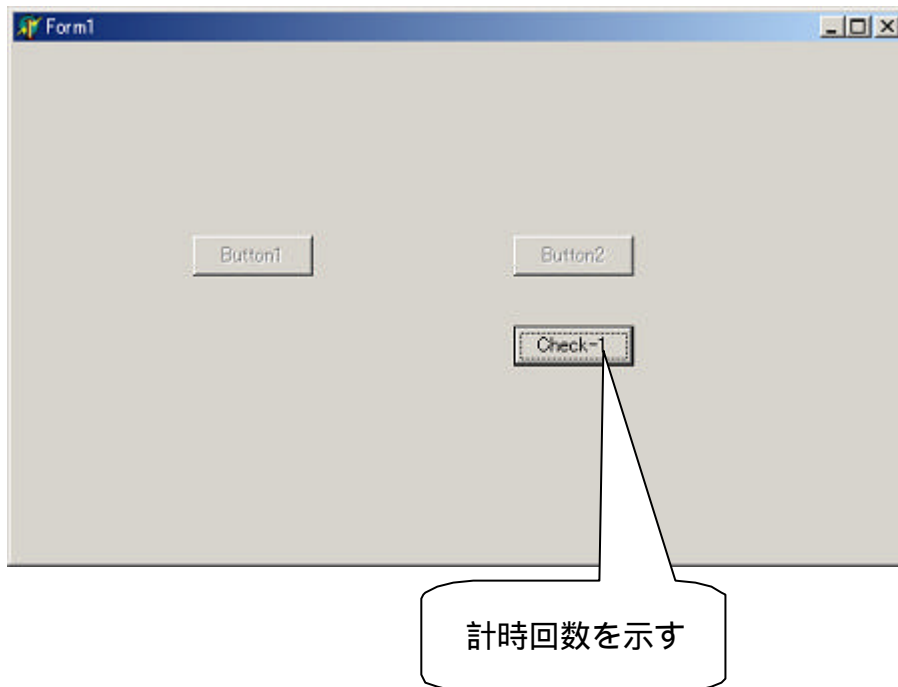


出力用ファイルにはテキストデータとして書き出されるので、プログラムの実行終了後エディタで開いて見ることができます。

ファイル名の設定後「開く」ボタンをクリックすると次図のフォームが表示されます。



「Check」ボタンをクリックしたときの ProcessMessages 手続きの処理時間が計測されます。「Check」ボタンをクリックすると、クリックの回数がボタン上に次図のように表示されます。



20回の測定（「Check」ボタンのクリック）でプログラムの実行が終了します。実行終了後、出力用ファイルを開くとリスト5のようになっています。

リスト5 ボタンの OnClick イベント処理を伴う場合の処理時間

1	0.000	0.000	0.000
2	10.000	0.000	10.000
3	0.000	0.000	0.000
4	0.000	0.000	0.000
5	0.000	0.000	0.000
6	0.000	0.000	0.000
7	0.000	0.000	0.000
8	0.000	0.000	0.000
9	0.000	0.000	0.000
10	0.000	0.000	0.000
11	0.000	0.000	0.000
12	0.000	0.000	0.000
13	0.000	0.000	0.000
14	0.000	0.000	0.000
15	0.000	0.000	0.000
16	0.000	0.000	0.000
17	0.000	0.000	0.000
18	10.000	0.000	10.000
19	10.000	0.000	10.000
20	0.000	0.000	0.000

各行の先頭に計時の回数、続いて ProcessMessages の呼び出しから戻るまでの時間、ProcessMessages の呼び出しから OnClick イベントハンドラの実行部の先頭までの時間、OnClick イベントハンドラの実行部の最後までまでの時間が書き出されています。

リスト 2、およびリスト 5 の結果は、Windows2000(PentiumIII,667MHz)でのものです。