

ATUALIZAÇÃO DO SISTEMA

Uma das coisas chatas é manter os usuários usando a última versão do nosso sistema. Na verdade o ideal é garantir isto.

Com a experiência eu aprendi que não podemos deixar esta responsabilidade para o usuário (ou para o nosso cliente). Podemos deixar isto a cargo do nosso sistema. Veremos, abaixo, que isto não é tão complicado quanto parece.

O FUNCIONAMENTO

Uma coisa é fato: Não podemos copiar um executável sobrepondo ao que já se encontra num HD sendo que o mesmo está sendo utilizado no momento da cópia. Portanto, o próprio sistema não pode fazer a cópia da nova versão.

Com isto, teremos que criar um aplicativo que vai analisar a versão instalada e a versão disponibilizada. Se houver alguma divergência, consideraremos que a versão disponibilizada é a versão que deverá ser utilizada pelo usuário. Sendo assim esta versão será copiada para a máquina do usuário sobrepondo a versão que já existe lá.

Temos que considerar uma outra situação: O nosso sistema pode, em muitos casos, se compor de vários arquivos. Portanto teremos que ter uma aplicação que faça a análise de todos estes arquivos e faça o devido sincronismo das versões.

Todas as informações (quais arquivos devemos atualizar e onde estão os arquivos disponibilizados) ficam dentro de um arquivo INI que também pode ser atualizado automaticamente.

O DESENVOLVIMENTO

Utilizaremos o próprio Delphi para fazer esta aplicação (não sei pensar diferente disto). Como será um programa sem interface (sem formulário) criemos uma nova aplicação (menu File/New/Application). Depois clique na opção Project/View Source) para ver o fonte do projeto. É aí que colocaremos todo o fonte do nosso programa.

Clique no menu Project/Remove from Project. Vai aparecer uma janela com o formulário que foi criado automaticamente. Selecione o formulário (Form1) e clique no botão OK. Vai te pedir uma confirmação. Pode confirmar sem dó nem piedade clicando no botão YES.

Salve o seu projeto. Neste momento você deve dar o nome do programa. O nome que eu dei aqui foi de LS (sigla de Load System), mas isto é de cada um.

Estamos prontos para começar !

Veja que na cláusula uses só tem a unit Forms. Vamos adicionar algumas outras que serão necessárias. A cláusula uses terá que conter as seguintes units: Forms, Classes, Windows, SysUtils, IniFiles.

A nossa primeira providência é fazer uma função que verifique se dois arquivos são diferentes. Eu optei por comparar a data e hora da criação de ambos. Um arquivo com datas (e horas) diferentes são arquivos diferentes. Obviamente isto pode ser trocado por qualquer outro método que julgar mais conveniente. O importante é saber quando um arquivo foi atualizado para atualizarmos o computador do usuário.

A função que fiz para isto é a seguinte:

```
function Verifica(Arq1, Arq2: String): Boolean;  
var  
    DataArq1: TDateTime;  
    DataArq2: TDateTime;  
  
begin  
    DataArq1 := FileDateToDateTime(FileAge(Arq1));  
    DataArq2 := FileDateToDateTime(FileAge(Arq2));  
  
    Result := (DataArq1 <> DataArq2);  
end;
```

Basta passarmos para esta função os dois arquivos que serão comparados e será devolvido um TRUE (caso eles sejam diferentes) ou um FALSE (caso sejam iguais).

Agora temos que ler o arquivo INI e comparar todos os arquivos mencionados ali.

A estrutura do arquivo INI é a seguinte:

```
[LS]  
Arquivos=PROGRAMA.EXE;PROGRAMA.INI;BIBLIOTECA.DLL  
Servidor=\\SERVIDOR\VERSAO  
Parametros=
```

Na variável “Arquivos” iremos colocar os nomes de todos os arquivos que queremos checar e atualizar automaticamente.

Na variável “Servidor” iremos colocar o nome do servidor com o caminho de onde encontrar os arquivos citados na variável anterior.

Na variável “Parametros” iremos colocar os parâmetros que temos que usar para chamar o nosso sistema (caso existam). Pode ficar em branco.

Uma coisa importante é a possibilidade de alterarmos este arquivo e o nosso programa reconhecer que o mesmo foi alterado.

No meu caso, eu mantive fixo o nome do INI como default, possibilitando ao usuário usá-lo com outro arquivo INI que deve ser passado como parâmetro.

Portanto agora começa a parte principal do nosso programa.

Onde tem as linhas:

```
Application.Initialize;  
Application.Run;
```

Iremos apagá-las e colocar aí o nosso código do programa.

Antes do início, propriamente dito, vamos declarar umas variáveis que serão necessárias:

```
var  
Arq: String;  
Nom: String; // Contém apenas o nome do arquivo INI (sem o path)  
Exe: String;  
Par: String; // Parâmetros de carga do do executável  
Sis: String; // Nome do executável que está instalado  
Ver: String; // Nome do executável que é a versão atual  
Lst: TStrings;  
Loc: Integer;  
Ind: Integer;  
Ini: TIniFile;  
Err: Boolean;
```

O primeiro passo é a identificação do arquivo INI. Esta identificação é feita da seguinte forma:

```
if FileExists(ExtractFilePath(ParamStr(0))+'LS.INI') then begin  
  Nom := 'LS.INI';  
  Arq := ExtractFilePath(ParamStr(0))+'LS.INI';  
end else  
  if ParamCount < 1 then begin  
    MessageBox(Application.Handle, PChar('Modo de uso: '+#13+  
      'LS <arquivo INI>'+#13+#13+  
      'onde'+#13+<arquivo INI> = Arquivo com '+  
      'configurações do sistema que será ' +  
      'inicializado'),  
      'Load System',  
      MB_ICONINFORMATION);  
  
    Exit;  
  end else begin  
    Nom := ExtractFileName(ParamStr(1));  
    Arq := ExtractFilePath(ParamStr(0))+ParamStr(1);  
  end;  
  
  if not FileExists(Arq) then begin  
    MessageBox(Application.Handle,  
      PChar('Arquivo "'+Arq+'" não encontrado'),  
      'Load System',  
      MB_ICONERROR);  
  
    Exit;  
  end;
```

Agora devemos ler as informações contidas no arquivo INI:

```
Ini := TIniFile.Create(Arq);  
Sis := Ini.ReadString('LS', 'Arquivos', '');  
Ver := Ini.ReadString('LS', 'Servidor', '');  
Par := Ini.ReadString('LS', 'Parametros', '');  
Ini.Free;
```

Neste ponto o conteúdo do arquivo INI está em variáveis. Agora o trabalho é validar e interpretar estas informações.

O primeiro passo é verificar se o servidor foi corretamente informado no arquivo INI.

```
Ver := Trim(Ver);  
if Ver = '' then begin  
    MessageBox(Application.Handle,  
                PChar('Arquivo "'+Arq+'" inválido'+#13+  
                'Servidor não informado'), 'Load System', MB_ICONERROR);  
    Exit;  
end;  
  
if Copy(Ver, Length(Ver), 1) <> '\\' then  
    Ver := Ver + '\\';
```

Aqui começa um ponto interessante. Já sabemos onde buscar novas versões. Imaginemos se atualizarmos o próprio arquivo INI responsável pelas atualizações. Portanto, o primeiro arquivo a ser comparado (checar a necessidade de atualização) é o próprio arquivo INI. Sendo que se o mesmo for atualizado, deveremos repetir o trabalho de leitura do mesmo.

```
if Verifica(Ver+Nom, Arq) then begin
  CopyFile(PChar(Ver+Nom), PChar(Arq), False);

  Ini := TIniFile.Create(Arq);
  Sis := Ini.ReadString('LS', 'Arquivos', '');
  Ver := Ini.ReadString('LS', 'Servidor', '');
  Par := Ini.ReadString('LS', 'Parametros', '');
  Ini.Free;

  Ver := Trim(Ver);
  if Ver = '' then begin
    MessageBox(Application.Handle,
      PChar('Arquivo "'+Arq+'" inválido'+#13+
        'Servidor não informado'),
      'Load System', MB_ICONERROR);

    Exit;
  end;

  if Copy(Ver, Length(Ver), 1) <> '\\' then
    Ver := Ver + '\\';
end;
```

Podemos notar que todo o trabalho de leitura do INI foi repetido, isto se houver divergência entre o arquivo instalado na máquina cliente e o disponibilizado no servidor.

Já que agora o arquivo INI está íntegro e atualizado, é hora de proceder com o restante do processo de checagem dos arquivos.

Podemos ver que no INI temos possibilidade de informar vários arquivos que serão checados neste processo. Todos estes arquivos estão armazenados na variável "Sis" e separados ponto e vírgula (;). Portanto, daqui pra frente fica fácil.

Iremos, então, criar um vetor do tipo TStringList. Em cada linha do nosso TStringList conterá um nome de arquivo. Portanto, podemos ter "n" arquivos.

```
Lst := TStringList.Create;

Loc := 0;
Lst.Add('');
for Ind := 1 to Length(Sis) do
  if Copy(Sis, Ind, 1) = ';' then begin
    Inc(Loc);
    Lst.Add('');
  end else
    Lst[Loc] := Lst[Loc] + UpperCase(Copy(Sis, Ind, 1));
```

Vencemos mais uma etapa. Neste ponto o nosso TStrings contém todos os arquivos especificados no arquivo INI.

Agora iremos verificar se os arquivos especificados existem no servidor e na estação cliente.

A regra é: O arquivo deve existir em pelo menos um dos dois lugares.

O código para isto é:

```
Err := False;
for Ind := 0 to Lst.Count-1 do begin
  if not FileExists(ExtractFilePath(ParamStr(0))+Lst[Ind]) and
    not FileExists(Ver+Lst[Ind]) then begin
    Err := True;
    Break;
  end;
end;

if Err then begin
  MessageBox(Application.Handle,
    'Arquivos do sistema não encontrados',
    'Load System', MB_ICONERROR);
  Lst.Free;
  Exit;
end;
```

Se a execução do nosso programa chegou a este ponto é sinal que tudo está correto (arquivo INI, arquivos a serem checados, etc).

Agora é chegada o momento interessante. Iremos comparar os arquivos e copiar a versão que está no servidor em caso de divergência. A idéia é manter todos usando a mesma versão disponibilizada no servidor.

A comparação que iremos fazer é a seguinte: Se o arquivo não existir na estação cliente, iremos copiá-lo do servidor (obviamente sem comparar). Se o arquivo na estação cliente iremos compara-lo com a versão disponibilizada no servidor, caso sejam diferentes, copiaremos a versão que está no servidor.

O código para isto é:

```
for Ind := 0 to Lst.Count-1 do
  if not FileExists(ExtractFilepath(ParamStr(0))+Lst[Ind]) then
    CopyFile(PChar(Ver+Lst[Ind]),
             PChar(ExtractFilepath(ParamStr(0))+Lst[Ind]), False)
  else
    if FileExists(ExtractFilepath(ParamStr(0))+Lst[Ind]) and
       FileExists(Ver+Lst[Ind]) then
      if Verifica(Ver+Lst[Ind],
                  ExtractFilepath(ParamStr(0))+Lst[Ind]) then
        CopyFile(PChar(Ver+Lst[Ind]),
                  PChar(ExtractFilepath(ParamStr(0))+Lst[Ind]),
                  False);
```

Chegamos ao final da atualização. Neste ponto todos os arquivos estão atualizados conforme a versão disponibilizada no servidor.

Só nos resta, então, chamar o nosso sistema. Nosso programa irá se encarregar de chamar o primeiro executável que estiver na lista de arquivos. Para isto precisamos achar o primeiro executável e logo depois chama-lo. O código para isto é:

```
Exe := '';
for Ind := 0 to Lst.Count-1 do
  if Pos('.EXE', Lst[Ind]) > 0 then begin
    Exe := ExtractFilePath(ParamStr(0))+Lst[Ind];
    Break;
  end;

Lst.Free;
if Exe = '' then
  MessageBox(Application.Handle,
              'Nenhum executável foi encontrado na lista de arquivos',
              'Load System', MB_ICONERROR)
else begin
  if Par <> '' then
    Exe := Exe + ' ' + Par;

  WinExec(PChar(Exe), SW_SHOWNORMAL);
end;
```

E aqui é o final. Logo abaixo segue o código completo (e sem interrupções) de tudo isto que vimos acima.

```

program LS;

uses
  Forms,
  Classes,
  Windows,
  SysUtils,
  IniFiles;

{$R *.res}

function Verifica(Arq1, Arq2: String): Boolean;
var
  DataArq1: TDateTime;
  DataArq2: TDateTime;

begin
  DataArq1 := FileDateToDateTime(FileAge(Arq1));
  DataArq2 := FileDateToDateTime(FileAge(Arq2));

  Result := (DataArq1 <> DataArq2);
end;

var
  Arq: String;
  Nom: String; // Contém apenas o nome do arquivo INI (sem o path)
  Exe: String;
  Par: String; // Parâmetros de carga do do executável
  Sis: String; // Nome do executável que está instalado
  Ver: String; // Nome do executável que é a versão atual
  Lst: TStrings;
  Loc: Integer;
  Ind: Integer;
  Ini: TIniFile;
  Err: Boolean;

begin
  if FileExists(ExtractFilePath(ParamStr(0))+'LS.INI') then begin
    Nom := 'LS.INI';
    Arq := ExtractFilePath(ParamStr(0))+'LS.INI';
  end else
    if ParamCount < 1 then begin
      MessageBox(Application.Handle, PChar('Modo de uso:'+#13+
        'LS <arquivo INI>'+#13+#13+
        'onde'+#13+<arquivo INI> = Arquivo com '+
        'configurações do sistema que será ' +
        'inicializado'),
        'Load System',
        MB_ICONINFORMATION);

      Exit;
    end else begin
      Nom := ExtractFileName(ParamStr(1));
      Arq := ExtractFilePath(ParamStr(0))+ParamStr(1);
    end;

```



```

if not FileExists(Arq) then begin
    MessageBox(Application.Handle,
                PChar('Arquivo "'+Arq+'" não encontrado'),
                'Load System',
                MB_ICONERROR);

    Exit;
end;

Ini := TIniFile.Create(Arq);
Sis := Ini.ReadString('LS', 'Arquivos', '');
Ver := Ini.ReadString('LS', 'Servidor', '');
Par := Ini.ReadString('LS', 'Parametros', '');
Ini.Free;

if Verifica(Ver+Nom, Arq) then begin
    CopyFile(PChar(Ver+Nom), PChar(Arq), False);

    Ini := TIniFile.Create(Arq);
    Sis := Ini.ReadString('LS', 'Arquivos', '');
    Ver := Ini.ReadString('LS', 'Servidor', '');
    Par := Ini.ReadString('LS', 'Parametros', '');
    Ini.Free;

    Ver := Trim(Ver);
    if Ver = '' then begin
        MessageBox(Application.Handle,
                    PChar('Arquivo "'+Arq+'" inválido'+#13+
                            'Servidor não informado'),
                    'Load System', MB_ICONERROR);

        Exit;
    end;

    if Copy(Ver, Length(Ver), 1) <> '\\' then
        Ver := Ver + '\\';
end;

Lst := TStringList.Create;

Loc := 0;
Lst.Add('');
for Ind := 1 to Length(Sis) do
    if Copy(Sis, Ind, 1) = ';' then begin
        Inc(Loc);
        Lst.Add('');
    end else
        Lst[Loc] := Lst[Loc] + UpperCase(Copy(Sis, Ind, 1));

Err := False;
for Ind := 0 to Lst.Count-1 do begin
    if not FileExists(ExtractFilePath(ParamStr(0))+Lst[Ind]) and
        not FileExists(Ver+Lst[Ind]) then begin
        Err := True;
        Break;
    end;
end;
end;

```

```

if Err then begin
    MessageBox(Application.Handle,
                'Arquivos do sistema não encontrados',
                'Load System', MB_ICONERROR);
    Lst.Free;
    Exit;
end;

for Ind := 0 to Lst.Count-1 do
    if not FileExists(ExtractFilepath(ParamStr(0))+Lst[Ind]) then
        CopyFile(PChar(Ver+Lst[Ind]),
                 PChar(ExtractFilepath(ParamStr(0))+Lst[Ind]), False)
    else
        if FileExists(ExtractFilepath(ParamStr(0))+Lst[Ind]) and
            FileExists(Ver+Lst[Ind]) then
            if Verifica(Ver+Lst[Ind],
                       ExtractFilepath(ParamStr(0))+Lst[Ind]) then
                CopyFile(PChar(Ver+Lst[Ind]),
                         PChar(ExtractFilepath(ParamStr(0))+Lst[Ind]),
                         False);

Exe := '';
for Ind := 0 to Lst.Count-1 do
    if Pos('.EXE', Lst[Ind]) > 0 then begin
        Exe := ExtractFilepath(ParamStr(0))+Lst[Ind];
        Break;
    end;

Lst.Free;
if Exe = '' then
    MessageBox(Application.Handle,
                'Nenhum executável foi encontrado na lista de arquivos',
                'Load System', MB_ICONERROR)
else begin
    if Par <> '' then
        Exe := Exe + ' ' + Par;

    WinExec(PChar(Exe), SW_SHOWNORMAL);
end;
end.

```