**Предотвращение запуска 2-x копий программы**

Известно, что Windows - многозадачная система. Это конечно хорошо. Но обратной стороной многозадачности является то, что одновременно можно запустить несколько копий одного и того же приложения. Пользователь может это сделать не специально, да и вторая копия приложения может запуститься из-за ошибки в программе или же самой OC. Из-за клона могут возникнуть некоторые проблемы, связанные с файловыми операциями, операциями с реестром и т.п. Поэтому код поиска и закрытия собственной копии в программе совсем не лишний. Есть несколько методов выполнения нахождения копии. В этой статье я их вам продемонстрирую.

**Поиск главной формы (окна)**

Чтобы найти копию главного окна можно воспользоваться функцией WinApi - FindWindow. В качестве параметра этой функции передается имя класса окна, которое использовалось при регистрации типа, определяющиего окно формы или в системе окон (WndClass) и заголовка окна. В Дельфи имя оконного класса WndClass совпадает с именем класса формы в Object Pascal (например TForm1). Функция возвращает 0, в случае если окна не существует, либо дескриптор окна, если окно существует. Основной код вашей программы должен быть написан так, что бы программа смогла запуститься, если результат функции FindWindow равен нулю или завершиться в обратном случае. Следующий пример, если вы собираетесь его использовать, нужно вставить в dpr-файл проекта (меню Project - View Source). Пример:

program Project1;

uses

Forms,

Unit1 in 'Unit1.pas' {Form1};

{$R \*.res}

var

HWnd : THanlde;

begin

HWnd := FindWindow ('TForm1', nil);

if HWnd <> 0 then

if not IsWindowVisible(HWnd) then

PostMessage(HWnd, wm\_User, 0, 0);

SetForegroundWindow (HWnd)

else

begin

Application.Initialize;

Application.CreateForm(TForm1, Form1);

//если в программе несколько форм, то создайте их здесь

//вручную, например Application.CreateForm(TForm2, Form2)

Application.Run;

end;

end.

Разъясню код по строчкам. В первых двух строках я объявил переменную HWnd типа THandle для хранения результата работы функции. Далее записываю результат выполнения функции FindWindow в переменную HWnd. Затем сравниваю результат: если он не равен нулю (т.е. программа уже запущена и данный экземпляр является копией) активизирую главную форму запущенного приложения. В обратном случае выполняю операции по инициализации, создания форм и запуска программы.

Однако при использовании этого кода могут возникнуть некоторые проблемы. Учтите, что если вы запустите программу в то время, когда работает дельфи с открытым в ней проектом запускаемого приложения, то программа вообще не запустится. Это связано с тем, что в системе уже существует форма с сответствующим классом в конструкторе дельфи. Однако вы сможете запустить программу, если закроете в дельфи файл юнита, связанного с формой, и саму форму (или лучше вообще весь проект). Если же вы просто закроете форму в конструкторе дельфи, то окно не уничтожится, а будет скрыто и так же будет существовать в системе. Поэтому встраивать этот код лучше на этапе завершения работы над программой.

**Использование мьютексов**

Mutex - это сокращение от mutual exclusion (взаимное исколючение). Это наиболее традиционный подход для среды Win32. Вообще мьютекс обычно используют не для нахождения копии программы, но этот объект можно приспособить и для этой цели. Мьютекс поддерживается на уровне ОС и его специфика такова, что мьютекс с заданным именем может принадлежать только одному приложению, либо никому. После того, как программа создала мьютекс, она может проверить, принадлежит ли этот объект какой либо программе. Для проверки используется функция WinApi - WaitForSingleObject. Если в момент обращения к этой функции мьютекс принадлежит кому либо, то функция ждет определенное количество времени (передается в качестве параметра) до освобождения мьютекса (если мьютекс не будет освобожден то функция возвратит ошибку с кодом wait\_TimeOut). А если мьютекса не существует (т.е. основная программа не запущена), то программа становится владельцем мьютекса. Реализация такого подхода показана в следующем примере:

program Project1;

uses

Windows,

Forms,

Unit1 in 'Unit1.pas' {Form1};

{$R \*.res}

var

hMutex : THandle;

begin

hMutex := CreateMutex(nil, False, 'UniqueProgrammMutex');

if WaitForSingleObject(hMutex, 0) <> wait\_TimeOut then

begin

Application.Initialize;

Application.CreateForm(TForm1, Form1);

Application.Run;

end;

end.

Этот метод является безусловно надежнее предыдущего. Единственный недостаток этого метода - это то, что при запуске копии программы главное окно запущенного приложения не активируется, хотя это легко можно исправить путем поиска главного окна в списке окон либо с помощью описанной функции FindWindow.