Технология представления результатов сетевого мониторинга в Интернете. Дополнительные сведения

Терентьев Александр Макарович, кандидат технических наук, PhD
Центральный экономико-математический институт Российской академии наук (г. Москва)

Аннотация. *В статье приведены дополнительные сведения о программировании системных служб web-сервера на языке высокого уровня PowerBASIC. Статья завершает предыдущий опубликованный материал.*

Ключевые слова: *сетевой мониторинг, информационная безопасность, UDP, web-сервер, системная служба , PowerBASIC.*

Смысл, история развития и роль сетевого мониторинга в ЦЭМИ РАН освещены в предыдущих публикациях [1-3]. Данная серия статей имеет своей целью показать применение мощных современных компилирующих средств в виде различных компиляторов с подмножеств языка программирования PowerBASIC для создания как консольных, так и полноэкранных (GUI-) приложений на примере средств передачи агрегированной информации сетевого мониторинга на web-сервер.

Принимающая программа, реализованная как системная служба, описана в [1]. Программа управления этой службой рассмотрена в [2]. Результаты работы этих программ и средств отображения принятой информации в HTML-формате рассмотрены в [3]. Однако, в целях полноты описания программного комплекса для web-сервера следует показать программы инициации созданной системной службы на web-сервере и её удаления.

Обе этих коротких программы созданы в виде консольных приложений (т.е. не создающих своё специфицированное графическое окно во время работы, а использующие потоковый строчный вывод по аналогии с множеством служебных программ типа PING.EXE) на компиляторе PowerBASIC Console Compiler [4] версии 5.05.

Методы программной реализации самой системной службы и программы управления ею, конечно, могут быть весьма различны. Современный отечественный стандарт этих методов предполагает использование алгоритмического языка C++. Целью данного цикла работ является, в том числе, желание продемонстрировать менее затратный и более элегантный путь написания подобных программ. Стоит пояснить, что являясь мощной альтернативой языку среднего уровня C++, PowerBASIC предлагает язык высокого уровня, специфицированный для создания профессиональных Windows-приложений благодаря встроенной поддержке множества обращений к различным библиотекам Windows. Не всегда являясь оптимальным при создании приложений реального времени, он тем не менее позволяет как язык высокого уровня гораздо легче писать различные программные приложения для Windows вплоть до системных. Все версии компиляторов являются платными, лицензирование осуществляется вендором исключительно персонально для каждого пользователя, лицензии на организации не оформляются (в силу чего они слабо распространены в РФ).

Несомненным достоинством множества диалектов PowerBASIC является то, что результатом их работы является единый EXE-модуль, запускаемый вне зависимости от наличия на ПК каких-либо дополнительных средств. Не секрет, что даже в игровой сфере создано огромное множество приложений, неизменно требующих для своего запуска установки множества посторонних программ типа Microsoft Visual C++ Redistributable, Microsoft .NET Framework X.X, и т.д. и т.п. К тому же, требования на эти средства обычно не указаны в спецификациях программ, и необходимость их установки обнаруживается только после инсталляции желаемого приложения. Печальным примером такого является широко известная в Интернете «бесплатная» программа определения текущих версий поддержки OpenGL [5], требующая инсталлирования (!) и вдобавок установки Microsoft .Net Framework, да ещё и определённой версии. Результатом такой политики производителей является, например, то, что на ПК автора за 7 лет эксплуатации его установлены: Microsoft .NET Framework 2.0 Service Pack 2, Microsoft .NET Framework 3.0 Service Pack 2, Microsoft .NET Framework 3.5 Service Pack 1, Microsoft .NET Framework 4 Client Profile, Microsoft .NET Framework 4 Extended и целый ряд других вспомогательных программ. В настоящее время уже трудно определить, для каких целей и в связи с какими программными продуктами были установлены эти приложения.

Все версии компиляторов PowerBASIC создают Portable-программы, не требующие установки каких-либо дополнительных пакетов, библиотек и пр., не нуждающиеся в инсталляции. В случае необходимости работы с реестром Windows, разумеется, в стандартных библиотеках находятся ссылки и описания входов реестровых операций.

Приведём полный текст программы старта системной службы (рис. 1), осуществляющей занесение необходимой информации в реестр Windows и исполнение старта службы, описанной в [1].

```
'  TamObSsU * Инсталляция системной службы * А.Терентьев, 2016
'============================================================
%USEMACROS = 1
#INCLUDE "Win32API.inc"
$SERVICENAME  = "TamObService"    ' Уникальное внутр.имя службы
$ServiceNameE = "TAM LAN Server Service"
$SERVICEFILE  = "e:\tamob\tamobss.exe"    ' Путь к программе
FUNCTION PBMAIN () AS LONG
    LOCAL hSCM     AS DWORD
    LOCAL hService AS DWORD
    LOCAL dError   AS DWORD
    LOCAL sInternalServiceName AS STRING
    LOCAL sExternalServiceName AS STRING
```

```
    LOCAL sServiceFileName     AS STRING
    STDOUT "Installing service " + $SERVICENAME
    hSCM = OpenSCManager(BYVAL 0, BYVAL 0, _
                         %SC_MANAGER_CREATE_SERVICE)
    dError = GetLastError()
    IF hSCM = 0 THEN
        STDERR "OpenSCM failed. LastErr=&H"+HEX$(dError, 8)
        EXIT FUNCTION
    END IF
    sInternalServiceName = $SERVICENAME
    sExternalServiceName = $ServiceNameE
    sServiceFileName = $SERVICEFILE
    hService = CreateService( _
               hSCM, _
               BYVAL STRPTR(sInternalServiceName), _
               BYVAL STRPTR(sExternalServiceName), _
               %SERVICE_ALL_ACCESS, _
               %SERVICE_WIN32_OWN_PROCESS, _
               %SERVICE_AUTO_START, _
               %SERVICE_ERROR_NORMAL, _
               BYVAL STRPTR(sServiceFileName), _
               BYVAL 0, BYVAL 0, BYVAL 0, BYVAL 0, BYVAL 0)
'                  LocalSystem Account =======^ = 0
' Варианты старта - dwStartType
' SERVICE_AUTO_START - стартует автоматически после System Startup    = 2
'   Во время boot, SCM стартует все auto-start службы и зависящие от них
' SERVICE_BOOT_START - device driver (only!) грузится от system Loader= 0
' SERVICE_DEMAND_START - стартует от SCM при функции StartService     = 3
' SERVICE_DISABLED    - сервис, который не стартует вообще            = 4
' Service_SYSTEM_START - device driver (only!) через IoInitSystem     = 1
    dError = GetLastError()
    IF hService = 0 THEN   '  Служба не стартовала
        CloseServiceHandle hSCM
        STDERR "CreateService failed."+ _
               " GetLastError = &H" + HEX$(dError, 8)
        EXIT FUNCTION
    END IF
    CloseServiceHandle hService
    CloseServiceHandle hSCM
    STDOUT "Successfully installed the service."
END FUNCTION
```

Рис. 1. Программа установки системной службы

Упомянутый в тексте программы LocalSystem account – предопределенный аккаунт, используемый Service Control Manager (SCM). Этот аккаунт не распознается подсистемой security, т.ч. нельзя задать его имя в обращении к функции LookupAccountName. Он имеет расширенные привилегии на локальном ПК, и работает как компьютер в сети. Его токен включает NT AUTHORITY\SYSTEM и BUILTIN\Administrators SID'ы; эти аккаунты имеют доступ к большинству системных объектов. Имя аккаунта на всех локальных ПК есть \LocalSystem, или ИмяПК\LocalSystem.

Этот аккаунт не имеет пароля. При задании LocalSystem при вызове CreateService или ChangeServiceConfig, любой пароль игнорируется. Сервис, запущенный в контексте LocalSystem, наследует security context от SCM. Пользовательский SID (идентификатор безопасности, используемый в Windows NT / 2000 / XP / 2003 / Vista) создаётся от значения SECURITY_LOCAL_SYSTEM_RID.

Аккаунт не ассоциируется с каким-либо аккаунтом пользователя. HKCU ассоциируется с default user, но не с current user. Для доступа к профилю другого пользователя, необходимо имперсонировать пользователя и затем осуществлять доступ через HKCU. Если служба открывает командное окно и запускает BAT-file, пользователь может дать Ctrl+C для завершения BAT-файла.

Привилегии LocalSystem можно см. в http://msdn.microsoft.com/en-us/library/ms684190(v=VS.85).aspx.

Большинство сервисов не нуждаются в таких высоких привилегиях. Если проектируемая служба не нуждается в этих привилегиях, и она не есть интерактивный сервис, достаточно использовать аккаунт LocalService или NetworkService. Для подробной информации, см. "Service Security and Access Rights".

Программа удаления (деинсталляции) системной службы [1] полностью показана на рис. 2.

```
'================================================================
' TAMObSsD * Программа деинсталляции системной службы * A.Терентьев
'================================================================
```

```
%USEMACROS = 1
#INCLUDE "Win32API.inc"
$SERVICENAME  = "TamObService"    ' Внутр.имя службы, уникальное.
$ServiceNameE = "TAM LAN Server Service"
$SERVICEFILE  = "e:\tamob\tamobss.exe" ' Полный путь к программе службы
FUNCTION PBMAIN () AS LONG
    LOCAL hSCM     AS DWORD
    LOCAL hService AS DWORD
    LOCAL dError   AS DWORD
    LOCAL fOk      AS LONG
    LOCAL sInternalServiceName AS STRING
    LOCAL sExternalServiceName AS STRING
    LOCAL sServiceFileName     AS STRING
    LOCAL ss       AS SERVICE_STATUS
    STDOUT "Removing service " + $SERVICENAME
    hSCM = OpenSCManager(BYVAL 0, BYVAL 0, %SC_MANAGER_CREATE_SERVICE)
    dError = GetLastError()
    IF hSCM = 0 THEN
        STDERR "OpenSCManager failed. GetLastError = &H" + HEX$(dError, 8)
        EXIT FUNCTION
    END IF
    sInternalServiceName = $SERVICENAME
    sExternalServiceName = $ServiceNameE
    sServiceFileName = $SERVICEFILE
    hService = OpenService(hSCM, $SERVICENAME, %SERVICE_ALL_ACCESS OR %DELETE)
    dError = GetLastError()
    IF hService = 0 THEN
        CloseServiceHandle hSCM
        STDERR "OpenService failed. GetLastError = &H" + HEX$(dError, 8)
        EXIT FUNCTION
    END IF
    ' Остановить службу, если она запущена.
    IF QueryServiceStatus(hService, ss) THEN
        IF ss.dwCurrentState <> %SERVICE_STOPPED THEN
            STDOUT "Service is still running. Wait while it shuts down."
            ControlService hService, %SERVICE_CONTROL_STOP, ss              ' 1
            ' ** a timed delay is not really enough, here-- we should wait
            ' ** until the ss.dwCurrentState = %SERVICE_STOPPED or some
            ' ** "deadman switch" limit has passed.
            SLEEP 1000
        END IF
    END IF
    ' ss - возвращаемый параметр, показывающий успех ( #0 ). Иначе GetLastError
    ' ERROR_ACCESS_DENIED - Handle не имеет соответствующих прав
    ' ERROR_DEPENDENT_SERVICES_RUNNING
    ' ERROR_INVALID_HANDLE
    ' ERROR_INVALID_PARAMETER
    ' ERROR_INVALID_SERVICE_CONTROL
    ' ERROR_SERVICE_CANNOT_ACCEPT_CTRL
    ' ERROR_SERVICE_NOT_ACTIVE
    ' ERROR_SHUTDOWN_IN_PROGRESS
    ' Удаление службы (или как минимум пометка её для удаления).
    fOk = DeleteService(hService)
    dError = GetLastError()
    CloseServiceHandle hService
    CloseServiceHandle hSCM
    IF fOk THEN
        STDOUT "Successfully removed the service."
    ELSE
        STDERR "DeleteService failed. GetLastError = &H" + HEX$(dError, 8)
    END IF
END FUNCTION
```

Рис. 2. Программа деинсталляции системной службы

Приведём несколько нетривиальных рекомендаций по программированию этой части.

Состав блока SERVICE_STATUS см. на рис. 2 в [2]. В член dwWin32ExitCode в случае ошибок службы при старте или останове должен быть послан ERROR_SERVICE_SPECIFIC_ERROR для индикации того, что член dwServiceSpecificExitCode содержит код ошибки. Служба должна установить это значение в 0 при нормальном ходе и нормальном завершении.

Член dwCheckPoint содержит некоторе значение, которое должно периодически увеличиваться для отсчёта прогресса во время замедленных старта, останова, паузы или продолжения. К примеру, служба должна увеличивать это значение после каждого шага инициализации. В остальные времена это значение не значимо и должно быть 0.

Если служба была запущена и невозможно её остановить по каким-либо причинам, она не может быть удалена до перезагрузки системы.

Возможные значения QueryServiceStatus показаны в комментариях внутри текста программы.

На этом рассмотрение программ старта и завершения системной службы закончено. В завершение, в справочных целях покажем, насколько легко программируются функции извлечения значения ключа из реестра и занесения его в реестр. На рис. 3 показаны обе функции, предполагающие строковые значения ключей.

```
FUNCTION GetRegValue(lpKey AS LONG,_ 'Получение строкового ключа из реестра
           BYVAL cMainkey AS STRING,BYVAL Key AS STRING) AS STRING
ON ERROR RESUME NEXT
DIM RetCode AS LONG, hKey AS LONG, cbData AS LONG, KeyType AS LONG
DIM KeyNameA AS ASCIIZ * 256
LOCAL zTmp AS ASCIIZ * 256
DIM acMainKey AS ASCIIZ * 300
LOCAL ZZZ AS STRING
     acMainKey = cMainKey
     RetCode = RegOpenKeyEx(lpKey, acMainkey, 0&, _
               %KEY_QUERY_VALUE, hKey) 'KEY_ALL_ACCESS
     AddLog 0,"GetRegValue RegOpenKeyEx RetCode="+STR$(RetCode) ' Для отладки
     IF RetCode = %ERROR_SUCCESS THEN
       IF Key$ = "*" THEN Key$ = CHR$(0,0)
'       szdat&=256
       DIM zbuffer AS ASCIIZ*256
       KeyNameA = Key
       cbData = SIZEOF(zTmp)
       RetCode = RegQueryValueEx(BYVAL hKey, KeyNameA, _
                 BYVAL 0, KeyType, zTmp, cbData)
       AddLog 0,"GetRegValue RegQueryValue RetCode="+STR$(RetCode) ' Отладка
       ZZZ = zTmp
       FUNCTION = ZZZ
       EXIT FUNCTION
     END IF
     FUNCTION = ""
END FUNCTION

FUNCTION SetRegValue(lpKey AS LONG,BYVAL cMainkey AS STRING,_ ' Установка ключа
          BYVAL Key AS STRING, BYVAL Setting AS STRING) AS LONG
  ON ERROR RESUME NEXT
  DIM RetCode AS LONG, hKey AS LONG, Result AS LONG
  LOCAL zText  AS ASCIIZ * 2048
  IF Key$ = "*" THEN Key$ = CHR$(0,0)
  RetCode=RegCreateKeyEx(lpKey, cMainKey + CHR$(0),0, "", _
                    %REG_OPTION_NON_VOLATILE, _
           %KEY_ALL_ACCESS, BYVAL %NULL, hKey, Result)
  AddLog 0,"SetRegValue RegCreateKeyEx [0=Ok] RetCode="+STR$(RetCode) ' Отладка
  IF RetCode<> %ERROR_SUCCESS THEN
    FUNCTION = 0
    EXIT FUNCTION
  END IF
  zText = Setting
  IF LEN(Setting) THEN
    RetCode=RegSetValueEx(hKey,Key+CHR$(0),0,%REG_SZ,zText,LEN(Setting)+1)
    AddLog 0,"SetRegValue RegSetValueEx1 RetCode="+STR$(RetCode) ' Отладка
  ELSE
    RetCode=RegSetValueEx(hKey, Key+CHR$(0), 0, %REG_SZ, zText, 1)
    AddLog 0,"SetRegValue RegSetValueEx2 RetCode="+STR$(RetCode) ' Отладка
  END IF
  RegCloseKey hKey
  FUNCTION = 0
```

```
END FUNCTION
```

Рис. 3. Программирование функций обращения к реестру Windows

```
pV = GetRegValue(%HKEY_LOCAL_MACHINE,_
         "SYSTEM\CurrentControlSet\Control\ProductOptions", _
           "ProductType")
```

Рис. 4. Пример получение значения ключа

Примером получения значения ключа с помощью указанной функции может являться оператор, приведённый на рис. 4.

Разумеется, декларации функций RegOpenKeyEx, RegCreateKeyEx, RegQueryValueEx, RegSetValueEx присутствуют в общем списке деклараций в основной включаемой библиотеке WIN32API.INC трансляторов с языков PowerBASIC.

Часть приведённых программных решений заимствована из многолетних обсуждений на международном форуме программистов по языку PowerBASIC. Конечно, эти решения адаптированы автором под требования конкретной реализации и использованные версии компиляторов.

Описанным образом реализовано управление запуском и остановом системной службы на Антивирусном сервере ЦЭМИ РАН, созданной в целях отображения в Интернете основных агрегирующих данных сетевого мониторинга. Работа программ проверена в средах Windows 2000 Server и Windows Server 2003 R2.

На этом рассмотрение программирования системной службы и сопровождающих функций на языках PowerBASIC автор считает законченным.

Работы автора, приведённые в списке Литературы, доступны для чтения в разделе «Литература» Антивирусного сайта ЦЭМИ РАН [6].

Литература:

1. Терентьев А.М. Технология представления результатов сетевого мониторинга в Интернете. Системная служба. // XXVII Международная научная конференция «Стратегии устойчивого развития мировой науки» – М.: «Евразийское научное объединение», N5(27), 2017. Т.1, с.41-46. ISSN 2411-1899.

2. Терентьев А.М. Технология представления результатов сетевого мониторинга в Интернете. Управление системной службой. // XXVIII Международная научная конференция «Интеграция науки в современном мире» – М.: «Евразийское научное объединение», N6(28), 2017. Т.1, с.28-33. ISSN 2411-1899.

3. Терентьев А.М. Технология представления результатов сетевого мониторинга в Интернете. Результаты. // XXIX Международная научная конференция «Теоретические и практические вопросы современной науки» – М.: «Евразийское научное объединение», N7(29), 2017. Т.1, с.27-31. ISSN 2411-1899.

4. [Интернет-ресурс] Международный сайт языка программирования PowerBASIC: http://www.powerbasic.com.

5. [Интернет-ресурс] https://www.opengl.org.

6. [Интернет-ресурс] Антивирусный сайт ЦЭМИ РАН: http://av.cemi.rssi.ru