МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

??? УНИВЕРСИТЕТ

Кафедра ???

Факультет ???

ОТЧЕТ

По курсу: «Операционные системы»

Вариант 3

Выполнил

ст. гр. ???

????

Проверил

???

????–2018

# СОДЕРЖАНИЕ

# Лабораторная работа № 1 Основы технологии Windows Management Instrumentation. Работа с процессами и потоками в WMI 2

# Лабораторная работа № 2 Обработка событий в WMI 4

# Лабораторная работа № 3 Сбор информации о памяти и планирование заданий. Работа с файловой системой 5

# Лабораторная работа № 4 Организация резервного копирования данных с использованием WMI и VBScript 7

# ЛИСТИНГИ ФАЙЛОВ СКРИПТОВ К ЛАБ. РАБОТАМ 9

# ЛАБОРАТОРНАЯ РАБОТА 1 9

# ЛАБОРАТОРНАЯ РАБОТА 2 15

# ЛАБОРАТОРНАЯ РАБОТА 3 18

# ЛАБОРАТОРНАЯ РАБОТА 4 21

# Лабораторная работа № 1 Основы технологии Windows Management Instrumentation. Работа с процессами и потоками в WMI

**Цель работы:** научится работать с объектами Windows Management Instrumentation с помощью сценариев на языке VBScript, научиться работать с процессами с использованием технологии WMI.

**Ход работы**

1. Составить сценарий получения списка свойств для заданного класса (табл. 1).
2. Составить сценарий сбора сведений о подключенных устройствах, согласно варианту (табл. 1.1).
3. Составить сценарий запуска процесса согласно варианту (табл. 1.2). Вывести следующую информацию о запущенном процессе:
4. путь к исполняемому файлу процесса;
5. время начала выполнения процесса (для преобразования даты использовать функцию WMIDateStringToDate());
6. приоритет процесса;
7. идентификатор процесса;
8. количество активных потоков процесса;
9. информацию об активных потоках:

- идентификатор процесса, создавшего поток;

- динамический приоритет потока;

- базовый приоритет потока;

- общее время выполнения потока;

- состояние потока.

1. Составить сценарий сбора информации о процессах согласно варианту (табл. 1.3).
2. Изучить возможности принудительного завершения процессов.

а) Составить сценарий завершающий все процессы “notepad.exe” имеющие низкий приоритет (Idle).

б) Составить сценарий завершающий работу всех программ, запущенных пользователем из Windows Commander (Total Commander).

**Таблица 1.1**

**Варианты заданий**

| **№**  **в-та** | **Задание** | **Класс** |
| --- | --- | --- |
| 3 | Получить информацию о клавиатуре:   * описание; * количество функциональных клавиш. | Win32\_Keyboard |

**Таблица 1.2**

**Варианты заданий для запуска процесса**

|  |  |  |  |
| --- | --- | --- | --- |
| **№**  **в-та** | **Приложение** | **Приоритет** | **Состояние окна** |
| 3 | MS Access | High | Maximized |

**Таблица 1.3**

**Варианты заданий для сбора информации о процессах**

|  |  |
| --- | --- |
| **№**  **в-та** | **Задание** |
| 3 | Определить процесс, имеющий наибольший объем прочитанных данных |

# Лабораторная работа № 2 Обработка событий в WMI

**Цель работы:** научиться разрабатывать временные обработчики событий WMI в виде сценариев на языке VBScript.

**Ход работы**

1. Составить сценарий выполняющий запуск определенного приложения, ожидание его завершения и вывод на экран информации о времени его работы в формате:

Время запуска: 12:10:00

Время завершения: 12:15:10

Длительность работы: 00:05:10

2. Составить сценарий запрещающий запуск определенного приложения. При попытке запуска данного приложения, его работа должна быть завершена, и на экран должно быть выведено сообщение о попытке запуска. Если количество разрешенных копий больше 0, то запрещать запуск необходимо только экземпляры приложения, превышающие указанное число.

Приложение и количество разрешенных копий выбираются по варианту из табл. 2.1.

**Таблица 2.1**

**Варианты заданий для запуска процесса**

|  |  |  |  |
| --- | --- | --- | --- |
| **№**  **в-та** | **Приложение** | | **Количество разрешенных копий** |
| **исполняемый файл** | **название** |
| 3 | clipbrd.exe | ClipBook Viewer | 0 |

# Лабораторная работа № 3 Сбор информации о памяти и планирование заданий. Работа с файловой системой

**Цель работы:** научится получать информацию о физической и виртуальной памяти, изучить способы управления планировщиком заданий с помощью WMI, научится работать с файловой системой Windows с помощью сценариев WMI.

**Ход работы**

1. Составить сценарий определения объемов памяти (табл. 3.3). В сценарии необходимо реализовать запись данных в текстовый файл. Каждый запуск сценария дописывает данные в файл, а не создает новый.
2. Составить сценарий создающий запланированное задание (Scheduled Task), осуществляющее запуск сценария п.1. День и время выполнения задания см. в табл. 3.
3. Составить сценарии просмотра и удаления запланированных заданий, в которых реализовать просмотр всех заданий с выводом идентификатора JobId и удаление заданий по одному, с вводом идентификатора удаляемого задания.

**Таблица 3.3**

**Варианты заданий**

| № | Определяемые объемы памяти | Дни недели запуска задания | Время запуска задания |
| --- | --- | --- | --- |
| 3 | Общий и свободный объем виртуальной памяти | Суббота и воскресенье | 8.30 |

1. Составить сценарий получения информации о логических дисках. Вывести список логических дисков с указанием идентификатора диска и типа файловой системы. Для логических дисков, являющихся разделами жесткого диска отобразить доступное дисковое пространство.
2. Составить сценарий формирования списка файлов по заданному условию.

|  |  |
| --- | --- |
| №  в-та | Условие формирования списка файлов |
| 3 | Документы Excel, хранящиеся на диске “C:” |

1. Составить сценарий, копирующий свой файл в папку “C:\tmp”, а затем переименовать его.
2. Составить сценарий удаления папки “C:\tmp”.
3. Изучить способ мониторинга модификации файлов с помощью сценариев WMI.

# Лабораторная работа № 4 Организация резервного копирования данных с использованием WMI и VBScript

**Цель работы:** научиться организовывать резервное копирование данных с использованием сценариев на языке VBScript и WMI.

**Ход работы**

Составить сценарий, выполняющий резервное копирование данных согласно варианту (табл. 1). Который должен обеспечивать:

а) архивацию всех файлов на диске C:\ соответствующих заданной маске;

б) Архивы должны иметь имена формата backcopy.\*. Каждый архив должен помещаться в отдельную папку – d:\backup\HHMMSS, где HH – час создания архива, MM – минута создания архива, SS – секунда создания архива.

**Таблица 4.7**

**Варианты заданий для резервного копирования**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№**  **в-та** | **Способ копирования** | **Маска** | **Количество**  **хранимых**  **архивов** | **Схема копирования** |
| 3 | Архиватор ARJ(ARJ32) | \*x\*.rtf | 5 | Полное |

1. Составить сценарий, удаляющий устаревшие архивы согласно варианту (табл. 7)

Архив считается устаревшим, если после его создания было создано больше чем указанное в табл. 6. количество хранимых архивов.

**Таблица 4.8**

**Варианты обработки устаревших архивов**

|  |  |
| --- | --- |
| **№** | **Вариант** |
| Нечетные | Устаревшие архивы удаляются |

# ПРИЛОЖЕНИЕ А. ЛИСТИНГИ ФАЙЛОВ СКРИПТОВ К ЛАБ. РАБОТАМ

ЛАБОРАТОРНАЯ РАБОТА 1

ЗАДАНИЕ 1. Составить сценарий получения списка свойств для заданного класса (табл. 1).

set objClass = GetObject("WinMgmts:Win32\_Keyboard")

set objClassProperties = objClass.Properties\_

for each obj in objClassProperties

p=p+obj.Name+ vbcrlf

next

WScript.Echo p

ЗАДАНИЕ 2. Составить сценарий сбора сведений о подключенных устройствах, согласно варианту (табл. 1).

set WMI = GetObject("WinMgmts:")

set objs = WMI.InstancesOf("Win32\_Keyboard")

for each obj in objs

WScript.Echo "Клавиатура: " & obj.Description & vbcrlf & "Количество функциональных клавиш: " & obj.NumberOfFunctionKeys

Next

ЗАДАНИЕ 3. Составить сценарий запуска процесса согласно варианту

Const ProcPriority = 128

set WMI = GetObject("WinMgmts:")

set objs = WMI.InstancesOf("Win32\_Processor")

set WMI = GetObject("WinMgmts:")

set objStartup = WMI.Get("Win32\_ProcessStartup")

set objConfig = objStartup.SpawnInstance\_

objConfig.ShowWindow = 3

set objProcess = WMI.Get ("Win32\_Process")

err = objProcess.Create("C:\Program Files (x86)\Microsoft Office\Office15\MSACCESS.EXE", null, \_

objConfig, ProcessId)

Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")

For Each objProc In objService.ExecQuery("SELECT \* FROM Win32\_Process WHERE Name = 'MSACCESS.exe'")

objProc.SetPriority(ProcPriority)

WScript.Echo objProc.ExecutablePath 'полный путь к исполняемому файлу процесса

WScript.Echo CDate(Mid(objProc.CreationDate,5,2) & "/" & Mid(objProc.CreationDate,7,2) & "/" & Left(objProc.CreationDate,4) & " " & Mid(objProc.CreationDate,9,2) & ":" & Mid(objProc.CreationDate, 11,2) & ":" & Mid(objProc.CreationDate,13,2))

WScript.Echo objProc.Priority '+ "Приоритет"

WScript.Echo objProc.ProcessId '"Идентификатор процесса"+

WScript.Echo objProc.ThreadCount ' "Количество активных потоков процесса"+

p=objProc.ThreadCount

z=0

set WMI = GetObject("WinMgmts:")

set objs = WMI.InstancesOf("Win32\_Thread")

for each obj in objs

if z=0 then

WScript.Echo "идентификатор процесса, создавшего поток " & obj.ProcessHandle & vbCrlf & "динамический приоритет потока " & obj.Priority & vbCrlf & "базовый приоритет потока " & obj.PriorityBase & vbCrlf & "общее время выполнения потока " & obj.ElapsedTime & vbCrlf & "состояние потока " & obj.ThreadState

z=z+1

end if

next

Next

ЗАДАНИЕ 4. Составить сценарий сбора информации о процессах согласно варианту

set WMI = GetObject("WinMgmts:")

set objs = WMI.InstancesOf ("Win32\_Process")

count = 1

for each obj in objs

if obj.ReadTransferCount > count then

count=obj.ReadTransferCount & vbCrlf & obj.Caption

end if

if obj.ReadTransferCount = count then

count=count + obj.ReadTransferCount & vbCrlf & obj.Caption

end if

next

WScript. Echo count

ЗАДАНИЕ 5. Составить сценарий завершающий все процессы “notepad.exe” имеющие низкий приоритет (Idle).

Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")

For Each objProc In objService.ExecQuery("SELECT \* FROM Win32\_Process WHERE Name = 'notepad.exe' AND Priority = 64")

objProc.Terminate()

Next

ЗАДАНИЕ 6. Составить сценарий завершающий работу всех программ, запущенных пользователем из Windows Commander (Total Commander).

Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")

For Each objProc In objService.ExecQuery("SELECT \* FROM Win32\_Process WHERE Name = 'TOTALCMD64.exe'")

pid = objProc.ProcessId

Next

For Each objProc In objService.ExecQuery("SELECT \* FROM Win32\_Process WHERE ParentProcessId = " & pid)

objProc.Terminate()

Next

ЛАБОРАТОРНАЯ РАБОТА 2

ЗАДАНИЕ 1. Составить сценарий выполняющий запуск определенного приложения, ожидание его завершения и вывод на экран информации о времени его работы.

set WMI = GetObject("WinMgmts:")

set objProcess = WMI.Get ("Win32\_Process")

err = objProcess.Create("clipbrd.exe", null, null, ProcessId)

Set colMonitorProcess = WMI.ExecNotificationQuery \_

("SELECT \* FROM \_\_InstanceOperationEvent " \_

& " WITHIN 1 WHERE TargetInstance ISA " \_

& "'Win32\_Process'")

startevent=0

endevent=0

do while endevent=0

Set objLatestEvent = colMonitorProcess.NextEvent

if objLatestEvent.TargetInstance.Name="clipbrd.exe" then

if startevent=0 then

WScript.Echo "время запуска: " & mid(Now,12)

starttime=mid(Now,12)

startevent=1

End if

if objLatestEvent.Path\_.Class="\_\_InstanceDeletionEvent" then

WScript.Echo "время завершения: " & mid(Now,12)

endtime=mid(Now,12)

endevent=1

End if

end if

loop

WScript.Echo "длительность работы: 0" & CDate(Abs(DateDiff("s", CDate(starttime), CDate(endtime)))/86400)

z=0

do

z=z

loop

ЗАДАНИЕ 2. Составить сценарий запрещающий запуск определенного приложения. При попытке запуска данного приложения, его работа должна быть завершена, и на экран должно быть выведено сообщение о попытке запуска. Если количество разрешенных копий больше 0, то запрещать запуск необходимо только экземпляры приложения, превышающие указанное число.

Set obj = GetObject("winmgmts:")

set objs = obj.ExecQuery \_

("SELECT \* FROM Win32\_Process " & \_

"WHERE Name = 'clipbrd.exe'")

for each obj1 in objs

obj1.Terminate()

next

do

Set colMonitorProcess = \_

obj.ExecNotificationQuery \_

("SELECT \* FROM \_\_InstanceCreationEvent " \_

& " WITHIN 1 WHERE TargetInstance ISA " \_

& " 'Win32\_Process' AND TargetInstance.Name='clipbrd.exe'")

Set objL = colMonitorProcess.NextEvent

per=objL.TargetInstance.ProcessId

set WMI = GetObject("WinMgmts:")

set objs2 = WMI.InstancesOf("WIN32\_Process")

for each obj1 in objs2

if obj1.ProcessId=per then

WScript.Echo "этот процесс запрещено запускать. процесс удален: " & obj1.ProcessId

obj1.Terminate()

end if

next

loop

ЛАБОРАТОРНАЯ РАБОТА 3

ЗАДАНИЕ 1. Составить сценарий определения объемов памяти (табл. 3.3). В сценарии необходимо реализовать запись данных в текстовый файл. Каждый запуск сценария дописывает данные в файл, а не создает новый.

set WMI = GetObject("WinMgmts:")

set objs = WMI.InstancesOf("Win32\_OperatingSystem")

Set fso = CreateObject("Scripting.FileSystemObject")

Set tf = fso.OpenTextFile("3,1.txt", 8)

for each obj in objs

tf.WriteLine ("общий объем виртуальной памяти: " & obj.TotalVirtualMemorySize & " килобайт, свободно виртуальной памяти: " & obj.FreeVirtualMemory & " килобайт")

Next

tf.Close

ЗАДАНИЕ 2. Составить сценарий создающий запланированное задание (Scheduled Task), осуществляющее запуск сценария п.1.

Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")

Set objNewJob = objWMIService.Get("Win32\_ScheduledJob")

err = objNewJob.Create("\3,1.vbs", "\*\*\*\*\*\*\*\*083000.000000+120", False, 32 or 64, , , jobName)

Wscript.Echo err

ЗАДАНИЕ 3. Составить сценарии просмотра и удаления запланированных заданий, в которых реализовать просмотр всех заданий с выводом идентификатора JobId и удаление заданий по одному, с вводом идентификатора удаляемого задания.

Set objService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")

set sheluder = objService.ExecQuery("SELECT \* FROM Win32\_ScheduledJob")

WScript.Echo sheluder.Count

WScript.Interactive = True

do

For Each objJob in sheluder

Wscript.Echo "Caption: " & objJob.Caption

Wscript.Echo "Command: " & objJob.Command

Wscript.Echo "Days of Month: " & objJob.DaysOfMonth

Wscript.Echo "Days of Week: " & objJob.DaysOfWeek

Wscript.Echo "Description: " & objJob.Description

Wscript.Echo "Elapsed Time: " & objJob.ElapsedTime

Wscript.Echo "Install Date: " & objJob.InstallDate

Wscript.Echo "Interact with Desktop: " & objJob.InteractWithDesktop

Wscript.Echo "--- Job ID: " & objJob.JobID

Wscript.Echo "Job Status: " & objJob.JobStatus

Wscript.Echo "Name: " & objJob.Name

Wscript.Echo "Notify: " & objJob.Notify

Wscript.Echo "Owner: " & objJob.Owner

Wscript.Echo "Priority: " & objJob.Priority

Wscript.Echo "Run Repeatedly: " & objJob.RunRepeatedly

Wscript.Echo "Start Time: " & objJob.StartTime

Wscript.Echo "Status: " & objJob.Status

Wscript.Echo "Time Submitted: " & objJob.TimeSubmitted

Wscript.Echo "Until Time: " & objJob.UntilTime

if toDel = objJob.JobID then

err = objJob.Delete()

end if

Next

WScript.StdOut.Write ("введите ид задачи для удаления>")

WScript.StdIn.Read (0)

toDel = WScript.StdIn.ReadLine ()

loop

ЗАДАНИЕ 4. Составить сценарий получения информации о логических дисках. Вывести список логических дисков с указанием идентификатора диска и типа файловой системы. Для логических дисков, являющихся разделами жесткого диска отобразить доступное дисковое пространство.

set WMI = GetObject("WinMgmts:")

set objs = WMI.InstancesOf("Win32\_LogicalDisk")

for each obj in objs

if obj.DriveType=3 then

WScript.Echo "идентификатор: " & obj.DeviceID & ", файловая система: " & obj.FileSystem & ", свободное место: " & obj.FreeSpace

else

WScript.Echo "идентификатор: " & obj.DeviceID & ", файловая система: " & obj.FileSystem

end if

Next

ЗАДАНИЕ 5. Составить сценарий формирования списка файлов по заданному условию.

Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")

Set colFiles = objWMIService.ExecQuery("SELECT \* FROM CIM\_DataFile WHERE Drive='C:' AND Extension = 'xlsx'")

Wscript.Echo "поиск файлов excel на диске C:, ожидайте"

Wscript.Echo "найдено файлов: " & colFiles.Count

For Each objFile in colFiles

Wscript.Echo objFile.Name

Next

Wscript.Echo "поиск окончен"

do

loop

ЗАДАНИЕ 6. Составить сценарий, копирующий свой файл в папку “C:\tmp”, а затем переименовать его.

Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")

Wscript.Echo "ожидайте"

Set colFiles = objWMIService.ExecQuery("Select \* from CIM\_DataFile where FileName = '3,6' AND Extension = 'vbs'")

For Each objFile in colFiles

objFile.Copy("c:\tmp\"& objFile.FileName & "." & objFile.Extension)

Next

Set colFiles = objWMIService.ExecQuery("Select \* from CIM\_DataFile where FileName = '3,6' AND Extension = 'vbs' AND Drive = 'C:'")

For Each objFile in colFiles

objFile.Rename("c:\tmp\.new"& objFile.FileName & "." & objFile.Extension)

Next

Wscript.Echo "готово"

do

loop

ЗАДАНИЕ 7. Составить сценарий удаления папки “C:\tmp”.

Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")

Wscript.Echo "ожидайте"

Set colFiles = objWMIService.ExecQuery("Select \* from Win32\_Directory where Name = 'c:\\tmp'")

For Each objFile in colFiles

objFile.Delete()

Next

Wscript.Echo "готово"

do

loop

ЗАДАНИЕ 8. 8. Изучить способ мониторинга модификации файлов с помощью сценариев WMI.

Способ мониторинга модификации файлов с помощью сценариев WMI:

1. У класса CIM\_DataFile существует свойство LastModified (дата и время последней модификации файла)
2. Для проверки модификации необходимых файлов после определённого времени (времени последней проверки, например) необходимо:

А) отправить запрос (вместо ??? указать какие файлы именно необходимо проверить)

Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")

Set colFiles = objWMIService.ExecQuery("SELECT \* FROM CIM\_DataFile WHERE ???)

Б) каждый выданный результат проверить через функцию DateDiff на новизну модификации

For Each objFile in colFiles

If DateDiff(“s”, objFile.LastModified, LastCheckTime)<0 then

Wscript.Echo objFile.Name & objFile.LastModified

End If

Next

В) После очередной проверки (а также в начале работы сценария), переменной LastCheckTime присвоить текущее время и дату

LastCheckTime = Now

1. Чтобы производить мониторинг необходимо зациклить проверку через бесконечный цикл Do Loop

ЛАБОРАТОРНАЯ РАБОТА 4

ЗАДАНИЕ 1. Составить сценарий, выполняющий резервное копирование данных согласно варианту.

sec = (Now - int(Now)) \* 86400

s = sec mod 60

if s < 10 then

s = "0" & s

end if

m = sec \ 60 mod 60

if m < 10 then

m = "0" & m

end if

h = sec \ 3600

if h < 10 then

h = "0" & h

end if

set fso = createobject ("scripting.filesystemobject")

fso.createfolder "d:\backup\" & h & m & s

Set objShell = Wscript.CreateObject("Wscript.Shell")

Cmd = "C:\ARJ32\arj32.exe a -r d:\backup\" & h & m & s & "\backcopy c:\\*x\*.rtf"

objShell.Run Cmd

ЗАДАНИЕ 2. Составить сценарий, удаляющий устаревшие архивы согласно варианту.

Old = CLng(999999)

i=6

do while i>5

Set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\CIMV2")

Set colFiles = objWMIService.ExecQuery("Select \* from Win32\_Directory where Path = '\\backup\\' and Drive = 'd:'")

i=colFiles.Count

For Each objFile in colFiles

min = objFile.FileName

if old>CLng(min) then

Old = CLng(min)

end if

Next

For Each objFile in colFiles

if CLng(objFile.FileName) = Old then

objFile.Delete()

end if

Next

loop