

PERL & OPERATING SYSTEMS

Бул макалада Perl программасын колдонуу түрдүү операциялык системада сүрөттөлүп берилет. Азыркы учурда веб-серверлер, Unix, Windows жана Apple системалары үчүн үч гана операциялык система бар. Иш жүзүндө дайыма UNIX (Linux) жана Windows программалары колдонулат.

В данной работе иллюстрируется использование языка программирования Perl в различных операционных системах. В настоящее время существует только три операционные системы, рекомендованные для веб-серверов: Unix, Windows и Apple Система.

В практике будут часто использоваться два из них: UNIX (Linux) и Windows.

The use of Perl programming in various operating systems is illustrated throughout this paper.

Currently there are only three operating systems that are recommended for web serving, Unix, Windows, and Apple System. In the practice two of them will be used often: UNIX (LINUX) and Windows.

1) INTRODUCTION

Perl is a general-purpose programming language. It can be used for anything that any other programming language can be used for. It has been used in every industry imaginable for almost any task. It's used on the stock market; in manufacturing, design bioinformatics, customer support, quality control, systems programming, payroll, and inventory; and, of course, on the web.

Perl is used in so many applications such as: Writing a database, a spreadsheet an operating system, or a full-featured Web server in Perl.

What Perl is good at is tying these emanates together. Perl can take the database, convert it into a spreadsheet-ready file, and during the processing, fix the data. Perl can also take the word processing documents and convert them to HTML (XML) for display on the web programming style is very flexible, so the same thing could be done in many ways.

The Perl is very adaptable. It runs under, at last count about two dozen operating systems and probably more. Perls programming style is very flexible, so the same thing could be done in many ways. In the following tasks for many operating systems are illustrated.

2) OPERATING SYSTEMS

Perl can be used for all existing operating systems.(see on figure1)

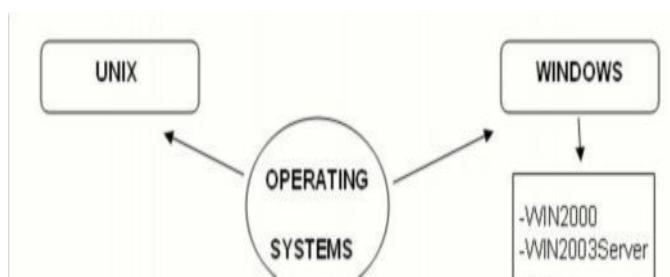


Figure1 - Operating Systems

Currently there are only three operating systems that are recommended for web serving, Unix, Windows, and Apple System. In the practice two of them will be used often: UNIX(LINUX) and Windows.

It doesn't matter which operating system is running. The HTML/HTM or XML pages appear same regardless of the operating system the server is using. Many of Perl scripts available on the internet are intended for Unix and some will include functions that will only work on a Unix system.

The combination of Perl and MySQL /SQL on a Unix Server is the best for database access. Also the ASP and Access/MS SQL on an NT is possible.

3) PERL SCRIPTS

In the following the sample scripts for managing and monitoring the Windows Operating System are given. Windows Operating System PERL SCRIPTS

3.1. COM/DCOM

- 3.1.1. List Classic COM Classes
- 3.1.2. List Component Categories
- 3.1.3. List DCOM Applications
- 3.1.4. List ProgIDs

3.2. Dates and Times

- 3.2.1. List the Local Time on a Computer
- 3.2.2. List Time Zone Information for a Computer

3.3. Page Files

- 3.3.1. List Page File Properties
- 3.3.2 List Page File Use

3.4. Processes

- 3.4.1. Processes and Threads (Perl Scripts)
 - 3.4.1.1. List Information about the Threads Active on a Computer
 - 3.4.1.2. List The Processes Running on aComputer

3.5. Registry

- 3.5.1. List Registry Properties

3.6. Services

3.6.1. List Service Load Order Groups 3.6.2. List Service Properties

3.7. Task

3.7.1. List Scheduled Tasks

3.8. Version

3.8.1. List Operating System Properties

List Component Categories

Script Code

```
use Win32::OLE('in'); use constant wbemFlagReturnImmediately => 0x10;
use constant wbemFlagForwardOnly => 0x20; $computer = ".";
$ObjWMIService = Win32::OLE->GetObject ("winmgmts:\\\\$computer\\root\\CIMV2") or die "WMI
connection failed.\n"; $colItems = $ObjWMIService->ExecQuery
print "Caption: $ObjItem->{Caption}\n"; foreach my $ObjItem (in $colItems) {
("SELECT * FROM Win32_ClassicCOMClass", "WQL", wbemFlagReturnImmediately
wbemFlagForwardOnly);
print "Component Id: $ObjItem->{ComponentId}\n"; print "Description: $ObjItem->{Description}\n";
print "Install Date: $ObjItem->{InstallDate}\n"; print "Name: $ObjItem->{Name}\n";
print "Status: $ObjItem->{Status}\n"; print "\n";}
```

List DCOM Applications

Script Code

```
use Win32::OLE('in'); use constant wbemFlagReturnImmediately => 0x10;
use constant wbemFlagForwardOnly => 0x20; $computer = ".";
$ObjWMIService = Win32::OLE->GetObject
("winmgmts:\\\\$computer\\root\\CIMV2") or die "WMI connection failed.\n";
$colItems = $ObjWMIService->ExecQuery ("SELECT * FROM
Win32_DCOMApplication", "WQL", wbemFlagReturnImmediately | wbemFlagForwardOnly);
foreach my $ObjItem (in $colItems){ print "App ID: $ObjItem->{AppID}\n"; print "Caption:
$ObjItem->{Caption}\n";
print "Description: $ObjItem->{Description}\n"; print "Install Date: $ObjItem->{InstallDate}\n";
print "Name: $ObjItem->{Name}\n"; print "Status: $ObjItem->{Status}\n"; print "\n";}
```

Dates and Times

Script Code

```
use Win32::OLE('in'); use constant wbemFlagReturnImmediately => 0x10;
use constant wbemFlagForwardOnly => 0x20; $computer = ".";
$ObjWMIService = Win32::OLE->GetObject
("winmgmts:\\\\$computer\\root\\CIMV2") or die "WMI connection failed.\n";
$colItems = $ObjWMIService->ExecQuery ("SELECT * FROM
Win32_LocalTime", "WQL", wbemFlagReturnImmediately | wbemFlagForwardOnly);
foreach my $ObjItem (in $colItems){
```

```

print "Day: $ObjItem->{Day}\n";           print "Day Of Week: $ObjItem->{DayOfWeek}\n";
print "Hour: $ObjItem->{Hour}\n";       print "Milliseconds: $ObjItem->{Milliseconds}\n";
print "Minute: $ObjItem->{Minute}\n";   print "Month: $ObjItem->{Month}\n";
print "Quarter: $ObjItem->{Quarter}\n"; print "Second: $ObjItem->{Second}\n";
print "Week In Month: $ObjItem->{WeekInMonth}\n"; print "Year: $ObjItem->{Year}\n"; print "\n";}

```

List Page File Properties

Script Code

```

use Win32::OLE('in'); use constant wbemFlagReturnImmediately => 0x10;
use constant wbemFlagForwardOnly => 0x20; $computer = ".";
$ObjWMIService = Win32::OLE->GetObject
("winmgmts:||||$computer\|root\|CIMV2") or die
"WMI connection failed.\n";
$colItems = $ObjWMIService->ExecQuery
("SELECT * FROM
Win32_PageFile","WQL",wbemFlagReturnImmediately wbemFlagForwardOnly);foreach my $ObjItem (in
$colItems){
print "Access Mask: $ObjItem->{AccessMask}\n"; print "Archive: $ObjItem->{Archive}\n";
print "Caption: $ObjItem->{Caption}\n"; print "Compressed: $ObjItem->{Compressed}\n";
print "Compression Method: $ObjItem->{CompressionMethod}\n";
print "Creation Class Name: $ObjItem->{CreationClassName}\n";
print "Creation Date: $ObjItem->{CreationDate}\n"; print "CS Creation Class Name: $ObjItem-
->{CSCreationClassName}\n"; print "CS Name: $ObjItem->{CSName}\n";
print "Description: $ObjItem->{Description}\n"; print "Drive: $ObjItem->{Drive}\n";
print "Eight Dot Three File Name: $ObjItem->{EightDotThreeFileName}\n";
print "Encrypted: $ObjItem->{Encrypted}\n"; print "Encryption Method: $ObjItem-
->{EncryptionMethod}\n";
print "Extension: $ObjItem->{Extension}\n"; print "File Name: $ObjItem->{FileName}\n";
print "File Size: $ObjItem->{FileSize}\n"; print "File Type: $ObjItem->{FileType}\n";
print "Free Space: $ObjItem->{FreeSpace}\n"; print "FS Creation Class Name: $ObjItem-
->{FSCreationClassName}\n";
print "FS Name: $ObjItem->{FSName}\n"; print "Hidden: $ObjItem->{Hidden}\n";
print "Initial Size: $ObjItem->{InitialSize}\n"; print "Install Date: $ObjItem->{InstallDate}\n";
print "In Use Count: $ObjItem->{InUseCount}\n"; print "Last Accessed: $ObjItem->{LastAccessed}\n";
print "Last Modified: $ObjItem->{LastModified}\n"; print "Manufacturer: $ObjItem->{Manufacturer}\n";
print "Maximum Size: $ObjItem->{MaximumSize}\n"; print "Name: $ObjItem->{Name}\n";
print "Path: $ObjItem->{Path}\n"; print "Readable: $ObjItem->{Readable}\n";
print "Status: $ObjItem->{Status}\n"; print "System: $ObjItem->{System}\n";
print "Version: $ObjItem->{Version}\n"; print "Writeable: $ObjItem->{Writeable}\n"; print "\n";}

```

List Information about the Threads Active on a Computer

Script Code

```
use Win32::OLE('in'); use constant wbemFlagReturnImmediately => 0x10;
use constant wbemFlagForwardOnly => 0x20; $computer = ".";
$ObjWMIService = Win32::OLE->GetObject
("winmgmts:\\\\$computer\\root\\CIMV2") or die "WMI connection failed.\n";
$colItems = $ObjWMIService->ExecQuery
("SELECT * FROM
Win32_Thread", "WQL", wbemFlagReturnImmediately wbemFlagForwardOnly);
foreach my $ObjItem (in $colItems){
print "Caption: $ObjItem->{Caption}\n"; print "Creation Class Name: $ObjItem-
->{CreationClassName}\n";
print "CS Creation Class Name: $ObjItem->{CSCreationClassName}\n";
print "CS Name: $ObjItem->{CSName}\n"; print "Description: $ObjItem->{Description}\n";
print "Elapsed Time: $ObjItem->{ElapsedTime}\n"; print "Execution State: $ObjItem-
->{ExecutionState}\n";
print "Handle: $ObjItem->{Handle}\n"; print "Install Date: $ObjItem->{InstallDate}\n";
print "Kernel Mode Time: $ObjItem-
->{KernelModeTime}\n";
print "Name: $ObjItem->{Name}\n"; print "OS Creation Class Name: $ObjItem-
->{OSCreationClassName}\n"; print "OS Name: $ObjItem->{OSName}\n";
print "Priority: $ObjItem->{Priority}\n"; print "Priority Base: $ObjItem->{PriorityBase}\n";
print "Process Creation Class Name: $ObjItem-
->{ProcessCreationClassName}\n"; print "Process Handle: $ObjItem-
->{ProcessHandle}\n";
print "Start Address: $ObjItem->{StartAddress}\n"; print "Status: $ObjItem->{Status}\n";
print "Thread State: $ObjItem->{ThreadState}\n"; print "Thread Wait Reason: $ObjItem-
->{ThreadWaitReason}\n"; print "User Mode Time: $ObjItem-
->{UserModeTime}\n"; print "\n"; }
```

List Registry Properties

Script Code

```
use Win32::OLE('in'); use constant wbemFlagReturnImmediately => 0x10;
use constant wbemFlagForwardOnly => 0x20; $computer = ".";
$ObjWMIService = Win32::OLE->GetObject ("winmgmts:\\\\$computer\\root\\CIMV2") or die
"WMI connection failed.\n";
$colItems = $ObjWMIService->ExecQuery ("SELECT * FROM
Win32_Registry", "WQL", wbemFlagReturnImmediately wbemFlagForwardOnly);
```

```
foreach my $objItem (in $colItems){ print "Caption: $objItem->{Caption}\n";
print "Current Size: $objItem->{CurrentSize}\n"; print "Description: $objItem->{Description}\n";
print "Install Date: $objItem->{InstallDate}\n"; print "Maximum Size: $objItem->{MaximumSize}\n";
print "Name: $objItem->{Name}\n"; print "Proposed Size: $objItem->{ProposedSize}\n";
print "Status: $objItem->{Status}\n"; print "\n";}
```

List Service Load Order Groups

Script Code

```
use Win32::OLE('in'); use constant wbemFlagReturnImmediately => 0x10;
use constant wbemFlagForwardOnly => 0x20;
$computer = "."; $objWMIService = Win32::OLE->GetObject
("winmgmts:||||$computer\root\CIMV2") or die "WMI connection failed.\n";
$colItems = $objWMIService->ExecQuery ("SELECT * FROM
Win32_LoadOrderGroup", "WQL", wbemFlagReturnImmediately wbemFlagForwardOnly);
foreach my $objItem (in $colItems){ print "Caption: $objItem->{Caption}\n";
print "Description: $objItem->{Description}\n"; print "Driver Enabled: $objItem->{DriverEnabled}\n";
print "Group Order: $objItem->{GroupOrder}\n"; print "Install Date: $objItem->{InstallDate}\n";
print "Name: $objItem->{Name}\n"; print "Status: $objItem->{Status}\n"; print "\n";}
```

List Scheduled Tasks

Script Code

```
use Win32::OLE('in'); use constant wbemFlagReturnImmediately => 0x10;
use constant wbemFlagForwardOnly => 0x20; $computer = ".";
$objWMIService = Win32::OLE->GetObject ("winmgmts:||||$computer\root\CIMV2") or die
"WMI connection failed.\n"; $colItems = $objWMIService->ExecQuery
("SELECT * FROM Win32_ScheduledJob", "WQL", wbemFlagReturnImmediately|
wbemFlagForwardOnly); foreach my $objItem (in $colItems){ print "Caption: $objItem->{Caption}\n";
print "Command: $objItem->{Command}\n"; print "Days Of Month: $objItem->{DaysOfMonth}\n";
print "Days Of Week: $objItem->{DaysOfWeek}\n"; print "Description: $objItem->{Description}\n";
print "Elapsed Time: $objItem->{ElapsedTime}\n"; print "Install Date: $objItem->{InstallDate}\n";
print "Interact With Desktop: $objItem->{InteractWithDesktop}\n";
print "Job Id: $objItem->{JobId}\n"; print "Job Status: $objItem->{JobStatus}\n";
print "Name: $objItem->{Name}\n"; print "Notify: $objItem->{Notify}\n";
print "Owner: $objItem->{Owner}\n"; print "Priority: $objItem->{Priority}\n";
print "Run Repeatedly: $objItem->{RunRepeatedly}\n"; print "Start Time: $objItem->{StartTime}\n";
print "Status: $objItem->{Status}\n"; print "Time Submitted: $objItem->{TimeSubmitted}\n";
print "Until Time: $objItem->{UntilTime}\n"; print "\n";}
```

List Operating System Properties

Script Code

```
use Win32::OLE('in'); use constant wbemFlagReturnImmediately => 0x10;
```

```

use constant wbemFlagForwardOnly => 0x20; $computer = ".";
$ObjWMIService = Win32::OLE->GetObject
("winmgmts:||||$computer\|root\|CIMV2") or die "WMI connection failed.\n";
$colItems = $ObjWMIService->ExecQuery
("SELECT * FROM Win32_OperatingSystem","WQL",wbemFlagReturnImmediately|
wbemFlagForwardOnly); foreach my $ObjItem (in $colItems){
print "Boot Device: $ObjItem->{BootDevice}\n"; print "Build Number: $ObjItem->{BuildNumber}\n";
print "Build Type: $ObjItem->{BuildType}\n"; print "Caption: $ObjItem->{Caption}\n";
print "Code Set: $ObjItem->{CodeSet}\n"; print "Country Code: $ObjItem->{CountryCode}\n";
print "Creation Class Name: $ObjItem->{CreationClassName}\n";
print "CS Creation Class Name: $ObjItem->{CSCreationClassName}\n";
print "CSD Version: $ObjItem->{CSDVersion}\n"; print "CS Name: $ObjItem->{CSName}\n";
print "Current Time Zone: $ObjItem->{CurrentTimeZone}\n"; print "Debug: $ObjItem->{Debug}\n";
print "Description: $ObjItem->{Description}\n"; print "Distributed: $ObjItem->{Distributed}\n";
print "Encryption Level: $ObjItem->{EncryptionLevel}\n"; print "Foreground Application Boost:
$ObjItem->{ForegroundApplicationBoost}\n"; print "Free Physical Memory: $ObjItem-
>{FreePhysicalMemory}\n"; print "Free Space In Paging Files: $ObjItem-
>{FreeSpaceInPagingFiles}\n"; print "Free Virtual Memory: $ObjItem-
>{FreeVirtualMemory}\n"; print "Install Date: $ObjItem->{InstallDate}\n";
print "Large System Cache: $ObjItem->{LargeSystemCache}\n";print "Last BootUp Time: $ObjItem-
>{LastBootUpTime}\n";print "Local Date Time: $ObjItem->{LocalDateTime}\n";
print "Locale: $ObjItem->{Locale}\n"; print "Manufacturer: $ObjItem->{Manufacturer}\n";
print "Max Number Of Processes: $ObjItem->{MaxNumberOfProcesses}\n";
print "Max Process MemorySize: $ObjItem->{MaxProcessMemorySize}\n";
print "Name: $ObjItem->{Name}\n"; print "Number Of Licensed Users: $ObjItem-
>{NumberOfLicensedUsers}\n" print "Number Of Processes: $ObjItem-
>{NumberOfProcesses}\n"; print "Number Of Users: $ObjItem->{NumberOfUsers}\n";
print "Organization: $ObjItem->{Organization}\n"; print "OS Language: $ObjItem->{OSLanguage}\n";
print "OS Product Suite: $ObjItem->{OSProductSuite}\n"; print "OS Type: $ObjItem->{OSType}\n";
print "Other Type Description: $ObjItem->{OtherTypeDescription}\n";
print "Plus Product ID: $ObjItem->{PlusProductID}\n"; print "Plus Version Number: $ObjItem-
>{PlusVersionNumber}\n"; print "Primary: $ObjItem->{Primary}\n";
print "Product Type: $ObjItem->{ProductType}\n"; print "Quantum Length: $ObjItem-
>{QuantumLength}\n"; print "Quantum Type: $ObjItem->{QuantumType}\n";
print "Registered User: $ObjItem->{RegisteredUser}\n"; print "Serial Number: $ObjItem-
>{SerialNumber}\n"; print "Service Pack Major Version: $ObjItem->{ServicePackMajorVersion}\n";
print "Service Pack Minor Version: $ObjItem->{ServicePackMinorVersion}\n";
print "Size Stored In Paging Files: $ObjItem->{SizeStoredInPagingFiles}\n";

```

```
print "Status: $objItem->{Status}\n"; print "Suite Mask: $objItem->{SuiteMask}\n";
print "System Device: $objItem->{SystemDevice}\n"; print "System Directory: $objItem-
>{SystemDirectory}\n"; print "System Drive: $objItem->{SystemDrive}\n";
print "Total Swap Space Size: $objItem->{TotalSwapSpaceSize}\n";
print "Total Virtual Memory Size: $objItem->{TotalVirtualMemorySize}\n";
print "Total Visible Memory Size: $objItem->{TotalVisibleMemorySize}\n";
print "Version: $objItem->{Version}\n"; print "Windows Directory: $objItem->{WindowsDirectory}\n";
print "\n";}
```

4) CONCLUSIONS

This paper illustrates some tasks of different Operating systems using Perl and demonstrates how to be used. The sample scripts for managing and monitoring the windows operating systems (WOS) have been described and the Perl listing is given.

5) REFERENCES

[1]<http://www.microsoft.com/technet/scriptcenter/scripts>

[2]http://www.tutorialspoint.com/perl/perl_variables.htm

[3]http://www.chinalinuxpub.com/doc/oreillybookself/perl/learn/ch02_03.htm

[4] Beginning PDL, version 1, 2001, Xavier Calbet

[5] <http://perldoc.perl.org>

[6] Learning Perl- O'Reilly – Second Edition, July 1997, By Randal Schwartz, Tom Christiansen & Larry Wall