PowerShell pt1
Špecializované IKT systémy Windows

Ing. Stanislav Lukac, PhD.

LAB: CA installation

192.168.1.0/24

LAN_Admin

192.168.1.1

DC (domain dual.edu)
Windows 201x
2 vCPU
3 GB RAM
1 vNIC

192.168.1.1
0

NIC1

DC

192.168.1.15

CLI
Windows 10
1 vCPU
2 GB RAM
1 vNIC

CLI

192.168.1.11

SRV3 (domain dual.edu)
Windows 201x
2 vCPU
2 GB RAM
1 vNIC

NIC1
192.168.1.13

SRV3

Online subordinate CA

SRV4 (STANDALONE)
Windows 201x
2 vCPU
1 GB RAM
1 vNIC

NIC1
192.168.1.14

Offline ROOT CA

SRV4

CA hierarchy
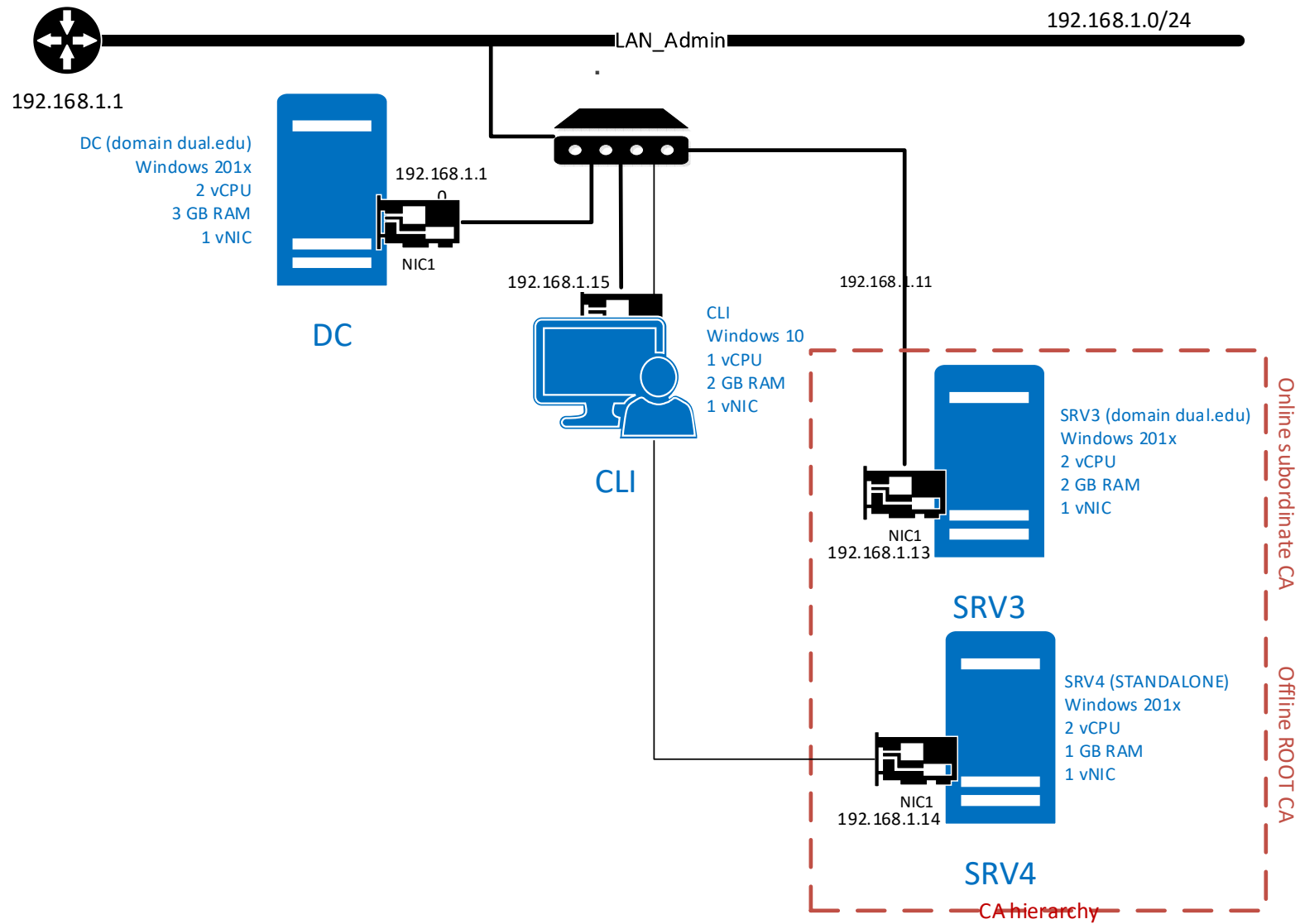
Agenda

1. PowerShell history
2. PowerShell help
3. PowerShell extensions
4. PowerShell operators
5. PowerShell arrays

M1 | What is PowerShell ?

- PowerShell is management engine built on .NET framework, OBJECTS oriented

- Interactive command shell and scripting environment

- Available not only for administration of Windows system (Exchange 2007, SQL,SCCM, Hyper-V, Vmware,)

- Command shell = direct admin interface

M1 | Key characteristics

- Free tool

- Built on .NET framework

- OBJECTS oriented

- Support pipelining with OBJECTS

- By default installed from windows server 2008 R2

- The great help and community support

- Offer remoting, workflow, desired state configuration, background jobs, and much more

- Extensible tool (VMware, Hyper-V, Unix systems)

- New MS approach in tools and configuration of MS products

M1 | Key characteristics

- Free tool

- Built on .NET framework

- OBJECTS oriented

- Support pipelining with OBJECTS

- By default installed from windows server 2008 R2

- The great help and community support

- Offer remoting, workflow, desired state configuration, background jobs, and much more

- Extensible tool (VMware, Hyper-V, Unix systems)

- New MS approach in tools and configuration of MS products

PS history

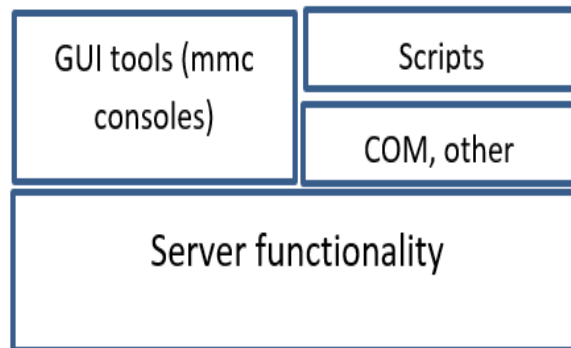## Špecializované IKT systémy Windows

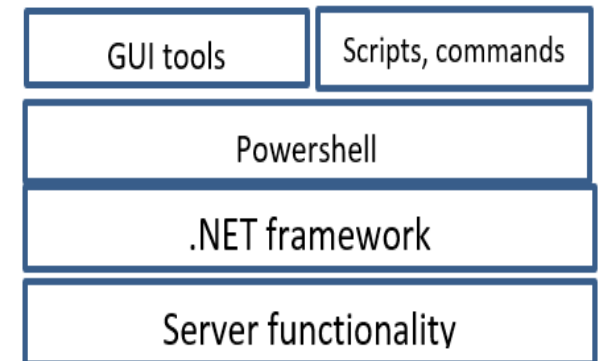Ing. Stanislav Lukac, PhD.

M1 | History

- Cmd.exe (from NT systems)
- mmc consoles (from NT systems)
- Resource kit (from NT systems)
- WSH (extension with using VBScript, Jscript)
- Perl, Phyton (as extension for WSH)
- 3rd party vendor – KIXtart
- emulation of Unix shells – Cygwin or Services for Unix

M1 | MS strategy

MS strategy (before)

| GUI tools (mmc consoles) | Scripts |
| | COM, other |

Server functionality

MS strategy (new)

| GUI tools | Scripts, commands |

Powershell

.NET framework

Server functionality

M1 | History of PS version

v2:
- Integrated to windows 7/Server 2008 R2
- Over 50 new cmdlets
- PowerShell remoting
- Background jobs
- New PowerShell ISE
- PowerShell module
- Workflow introducing

v3:
- Scheduled jobs
- PowerShell workflow
- Disconnected sessions
- Web Access
- Delegate administration

v4:
- Improved debugging
- DSC
- Enhanced debugging
- Offline help update

V5
- Windows PowerShell OneGet module and NetworkSwitch module
- Just Enough Administration (JEA)
- Windows PowerShell Desired State Configuration (DSC)
- Windows PowerShell Integrated Scripting Environment (ISE)
- Windows PowerShell Web Services (Management OData IIS Extension)
- Windows Remote Management (WinRM)

	PowerShell 1.0	PowerShell 2.0	PowerShell 3.0	PowerShell 4.0	PowerShell 5.0
Windows Vista	Pre-Installed	Installable	N/A	Installable	N/A
Windows Server 2008	Pre-Installed	Installable	N/A	Installable	N/A
Windows 7	N/A	Pre-Installed	Installable	Installable	Installable
Window server 2008 R2	N/A	Pre-Installed	Installable	Installable	Installable
Windows 8	N/A	N/A	Pre-Installed	Installable	Installable
Windows Server 2012	N/A	N/A	Pre-Installed	Installable	Installable
Windows 8.1	N/A	N/A	Installable	Pre-Installed	Installable
Windows Server 2012 R2	N/A	N/A	Installable	Pre-Installed	Installable
Windows 10	N/A	N/A	N/A	N/A	Pre-Installed
Windows Server 2016	N/A	N/A	N/A	N/A	Pre-Installed

PowerShell installation
Špecializované IKT systémy Windows

Ing. Stanislav Lukac, PhD.

M1 | Installation of PS

Requirements: .NET Framework XY installed !

1. Download the correct package for your operating system and architecture. (windows management framework)
2. Close all Windows PowerShell windows.
3. Uninstall any other versions of Windows Management Framework
4. Run the MSU file that you downloaded.

M1 | PowerShell Shortcuts

PowerShell Prompt

- Arrow Up/Down   command history
- F8/F9   color codding enable /disable
- CTRL+C   cancel current execution process
- CTRL+ Arrow   for movement one word left/right
- Home/End   for movement at the beginning/end of line
- Insert   for edit/replacement mode of editing
- Ctrl + L   last line first

PowerShell ISE

- Ctrl+M   collapse/expand block
- Ctrl+F   find
- Ctrl+H   Replace
- CTRL+C   cancel current execution process
- CTRL+ Arrow   for movement one word left/right
- Home/End   for movement at the beginning/end of line
- Insert   for edit/replacement mode of editing
- Ctrl + J   show snippets
- F5   Run all
- F8   Run selected

```
$gps = $psISE.GetType().Assembly
$rm = New-Object System.Resources.ResourceManager GuiStrings,$gps
$rs = $rm.GetResourceSet((Get-Culture),$true,$true)
$rs | where Name -match 'Shortcut\d?$|^F\d+Keyboard' | Sort-Object Value | Format-Ta
```

```
Name                                              Value
----                                              -----
EditorUndoShortcut2                               Alt+Backspace
EditorSelectNextSiblingShortcut                   Alt+Down
ExitShortcut                                      Alt+F4
EditorSelectEnclosingShortcut                     Alt+Left
EditorSelectFirstChildShortcut                    Alt+Right
EditorRedoShortcut2                               Alt+Shift+Backspace
EditorBoxSelectLineDownShortcut                   Alt+Shift+Down
ToggleHorizontalAddOnPaneShortcut                 Alt+Shift+H
EditorBoxSelectToPreviousCharacterShortcut        Alt+Shift+Left
EditorBoxSelectToNextCharacterShortcut            Alt+Shift+Right
EditorTransposeLineShortcut                       Alt+Shift+T
EditorBoxSelectLineUpShortcut                     Alt+Shift+Up
ToggleVerticalAddOnPaneShortcut                   Alt+Shift+V
EditorSelectPreviousSiblingShortcut               Alt+Up
ShowScriptPaneTopShortcut                         Ctrl+1
ShowScriptPaneRightShortcut                       Ctrl+2
ShowScriptPaneMaximizedShortcut                   Ctrl+3
EditorSelectAllShortcut                           Ctrl+A
ZoomIn1Shortcut                                   Ctrl+Add
EditorMoveCurrentLineToBottomShortcut             Ctrl+Alt+End
EditorMoveCurrentLineToTopShortcut                Ctrl+Alt+Home
EditorDeleteWordToLeftShortcut                    Ctrl+Backspace
StopExecutionShortcut                             Ctrl+Break
StopAndCopyShortcut                               Ctrl+C
GoToConsoleShortcut                               Ctrl+D
EditorDeleteWordToRightShortcut                   Ctrl+Del
EditorScrollDownAndMoveCaretIfNecessaryShortcut   Ctrl+Down
EditorMoveToEndOfDocumentShortcut                 Ctrl+End
FindShortcut                                      Ctrl+F
```

PowerShell 1st run

Špecializované IKT systémy Windows

Ing. Stanislav Lukac, PhD.

1st commands


```
PS H:\> Get-PSVersion

Major   Minor   Build   Revision
-----   -----   -----   --------
5       1       14393   3053
```

```
PS H:\>
PS H:\> $PSVersionTable

Name                           Value
----                           -----
PSVersion                      5.1.14393.3053
PSEdition                      Desktop
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0...}
BuildVersion                   10.0.14393.3053
CLRVersion                     4.0.30319.42000
WSManStackVersion              3.0
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1
```

```
PS H:\>
PS H:\> Get-Host

Name             : ConsoleHost
Version          : 5.1.14393.3053
InstanceId       : 3ca65b31-4726-4600-98d6-b029609db2ad
UI               : System.Management.Automation.Internal.Host.InternalHostUserInterface
CurrentCulture   : en-US
CurrentUICulture : en-US
PrivateData      : Microsoft.PowerShell.ConsoleHost+ConsoleColorProxy
DebuggerEnabled  : True
IsRunspacePushed : False
Runspace         : System.Management.Automation.Runspaces.LocalRunspace
```

Cmdlets, aliases, providers
Špecializované IKT systémy Windows

Ing. Stanislav Lukac, PhD.

CMDLET(s)

Commandlet syntax

Verb-Noun

Not case sensitive

Spaces sensitive

ALIASES

Aliases

- Can be used as a shortcut for commands/programs/functions use to often|Get-Alias; Set-Alias

```
PS C:\> Get-Alias -Name cd

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Alias           cd -> Set-Location


PS C:\> Get-Alias -Name ls

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Alias           ls -> Get-ChildItem


PS C:\> Get-Alias -Definition Get-ChildItem

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Alias           dir -> Get-ChildItem
Alias           gci -> Get-ChildItem
Alias           ls -> Get-ChildItem
```
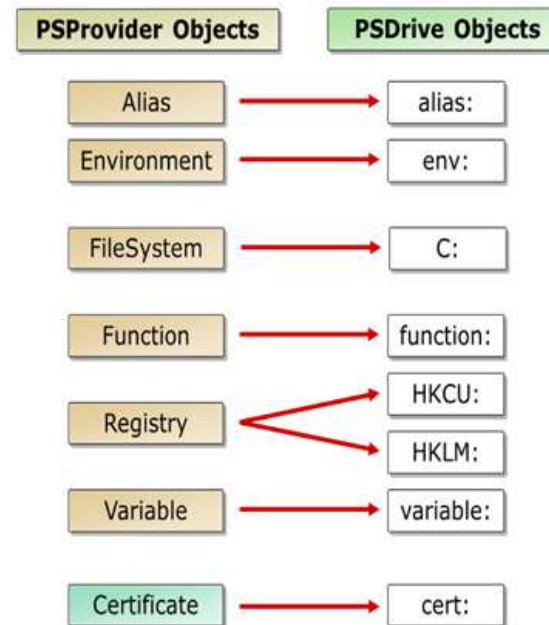
PSDRIVES

Aliases

- Can be used as a shortcut for commands/programs/functions use to often | Get-Alias; Set-Alias

PSDRIVES

Help

Špecializované IKT systémy Windows

Ing. Stanislav Lukac, PhD.

Thematic help

Windows PowerShell

```
PS H:\>
PS H:\> Get-Help about*

Name                              Category  Module        Synopsis
----                              --------  ------        --------
about_ActivityCommonParameters    HelpFile                Describes the parameters...
about_Aliases                     HelpFile                Describes how to use alt...
about_Arithmetic_Operators        HelpFile                Describes the operators ...
about_Arrays                      HelpFile                Describes arrays, which ...
about_Assignment_Operators        HelpFile                Describes how to use ope...
about_Automatic_Variables         HelpFile                Describes variables that...
about_Break                       HelpFile                Describes a statement yo...
about_Checkpoint-Workflow         HelpFile                Describes the Checkpoint...
about_CimSession                  HelpFile                Describes a CimSession o...
about_Classes                     HelpFile                Describes how you can us...
about_Command_Precedence          HelpFile                Describes how Windows Po...
about_Command_Syntax              HelpFile                Describes the syntax dia...
about_Comment_Based_Help          HelpFile                Describes how to write c...
about_CommonParameters            HelpFile                Describes the parameters...
about_Comparison_Operators        HelpFile                Describes the operators ...
about_Continue                    HelpFile                Describes how the Contin...
about_Core_Commands               HelpFile                Lists the cmdlets that a...
about_Data_Sections               HelpFile                Explains Data sections, ...
about_Debuggers                   HelpFile                Describes the Windows Po...
about_DesiredStateConfiguration   HelpFile                Provides a brief introdu...
about_Do                          HelpFile                Runs a statement list on...
about_Environment_Variables       HelpFile                Describes how to access ...
about_Escape_Characters           HelpFile                Introduces the escape ch...
```

Windows PowerShell

```
PS H:\>
PS H:\> Get-Help about_Command_Syntax
TOPIC
    about_Command_Syntax

SHORT DESCRIPTION
    Describes the syntax diagrams that are used in Windows PowerShell.
```

How to read HELP


```
Windows PowerShell                                          —   □   ✕
PS H:\>
PS H:\> get-help Get-Service

NAME
    Get-Service

SYNOPSIS
    Gets the services on a local or remote computer.

SYNTAX
    Get-Service [-ComputerName <String[]>] [-DependentServices] -DisplayName <String[]>
    [-Exclude <String[]>] [-Include <String[]>] [-RequiredServices] [<CommonParameters>]

    Get-Service [-ComputerName <String[]>] [-DependentServices] [-Exclude <String[]>] [-Include
    <String[]>] [-InputObject <ServiceController[]>] [-RequiredServices] [<CommonParameters>]

    Get-Service [[-Name] <String[]>] [-ComputerName <String[]>] [-DependentServices] [-Exclude
    <String[]>] [-Include <String[]>] [-RequiredServices] [<CommonParameters>]
```

```
Windows PowerShell
PS H:\>
PS H:\> Get-Help Get-Service -ShowWindow
```

```
Windows PowerShell
PS H:\>
PS H:\> Get-Help Get-Service -Examples
```

```
Windows PowerShell
PS H:\>
PS H:\> Get-Help Get-Service -Online
```

How to read HELP

Windows PowerShell

```
PS H:\>
PS H:\> Get-Help Get-Service -Detailed
```

Windows PowerShell

```
PS H:\>
PS H:\> Get-Help Get-Service -Full
```

	Value	Notes
Required?	True/False	If is necessary to specify such parameter
Position?	1	Position of parameter in cmdlet syntax
Default value	localhost	Default value if you do not define different
Accept pipeline input?	True/False	Accpet value from Pipeline or not, when True: • ByPropertyName • ByValue
Accept wildcard characters?	True/False	Accept * or ? or []

Common param

```
SYNTAX
    Get-Service [-ComputerName <String[]>] [-DependentServices] -DisplayName <String[]>
    [-Exclude <String[]>] [-Include <String[]>] [-RequiredServices] [<CommonParameters>]
```

Windows PowerShell

```
PS H:\>
PS H:\> get-help about_CommonParameters
```

```
        -Debug (db)
        -ErrorAction (ea)
        -ErrorVariable (ev)
        -InformationAction
        -InformationVariable
        -OutVariable (ov)
        -OutBuffer (ob)
        -PipelineVariable (pv)
        -Verbose (vb)
        -WarningAction (wa)
        -WarningVariable (wv
```

Variables

- You can declarate variable by $ or *New-Variable*
- Name can contains letters, numbers, "_" , "#" and spaces
- Can be use for add value, command lets or external programs
- Can be specify data type: [psobject]$variable

```
Windows PowerShell
PS H:\>
PS H:\> Get-Variable
```

```
Windows PowerShell
PS H:\>
PS H:\> Get-ChildItem variable:
```

```
Windows PowerShell
PS H:\>
PS H:\> $Variable:<variableName>
```

Variables


```
Windows PowerShell
PS H:\>
PS H:\> $RunningProcesses = Get-Process
PS H:\> $RunningProcesses.Count
112
PS H:\> $RunningProcesses.Name
ApplicationFrameHost
armsvc
audiodg
AuthManSvr
avp
```

```
Windows PowerShell
PS H:\>
PS H:\> Get-Process -OutVariable 'RunningProcesses2'
```

```
Windows PowerShell
PS H:\>
PS H:\> New-Variable -Name 'RunningProcesses3' -Value Get-Process
```

```
Windows PowerShell
PS H:\>
PS H:\> [int]$ThisYear = '2019'
```

```
Windows PowerShell
PS H:\>
PS H:\> [string] = 'I Love PowerShell'
```

Variables types

Data Type	Description	Examples
[int]	32 bit Integer number	-273,0,20,
[long]	64 bit Integer number	
[single]	Single-precision 32-bit floating point number	
[double]	Double-precision 64-bit floating point number	
[byte]	An 8-bit unsigned character	0,1,...254
[char]	Unicode 16-bit character	a,b,c
[decimal]	An 128-bit decimal value	
[string]	Retazec	"hello world"
[datetime]	Point in time	April 1, 2011
[boolean]	Logical Yes/No	$True, $False
[xml]	Xml object	
[array]	Object of array	
[hashtable]	Object of hash table	
[adsi]	ADSI object	

Special variable

- $_ contains current object in pipeline

- $PSItem contains current object in pipeline

- $? Status of last command execution

- $Error contains error message

- $True/$False

- $LastExitCode contains exit codes of Windows-Based commands

- $PSHome contains installation path of PS

- $PSVersion table contaisn information about PS version

DEMO::

```
Administrator: Windows PowerShell

PS C:\> dir env:

Name                          Value
----                          -----
ALLUSERSPROFILE               C:\ProgramData
APPDATA                       C:\Users\slukac\AppData\Roaming
CommonProgramFiles            C:\Program Files\Common Files
CommonProgramFiles(x86)       C:\Program Files (x86)\Common Files
CommonProgramW6432            C:\Program Files\Common Files
COMPUTERNAME                  MASTER
ComSpec                       C:\WINDOWS\system32\cmd.exe
HOMEDRIVE                     C:
HOMEPATH                      \Users\slukac
JD2_HOME                      C:\Users\slukac\AppData\Local\JDownloader v2.0
LOCALAPPDATA                  C:\Users\slukac\AppData\Local
LOGONSERVER                   \\MASTER
NUMBER_OF_PROCESSORS          8
OneDrive                      C:\Users\slukac\OneDrive
OS                            Windows_NT
Path                          C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System3
PATHEXT                       .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.CPL
PROCESSOR_ARCHITECTURE        AMD64
PROCESSOR_IDENTIFIER          Intel64 Family 6 Model 60 Stepping 3, GenuineIntel
PROCESSOR_LEVEL               6
PROCESSOR_REVISION            3c03
ProgramData                   C:\ProgramData
ProgramFiles                  C:\Program Files
ProgramFiles(x86)             C:\Program Files (x86)
ProgramW6432                  C:\Program Files
PSModulePath                  C:\Users\slukac\Documents\WindowsPowerShell\Modules;C:\Program Files\Windo
PUBLIC                        C:\Users\Public
SystemDrive                   C:
SystemRoot                    C:\WINDOWS
TEMP                          C:\Users\slukac\AppData\Local\Temp
TMP                           C:\Users\slukac\AppData\Local\Temp
USERDOMAIN                    MASTER
USERDOMAIN_ROAMINGPROFILE     MASTER
USERNAME                      slukac
USERPROFILE                   C:\Users\slukac
windir                        C:\WINDOWS
```

PS Operators

Špecializované IKT systémy Windows

Ing. Stanislav Lukac, PhD.

Operators

about_Operators
about_Arithmetic_Operators
about_Assignment_Operators
about_Comparison_Operators
about_Logical_Operators
about_Type_Operators
about_Split
about_Join
about_Redirection

Operators

Range and redirection operators

- (x..y)

- >

- >>

- 2>

-F operator

"{I,A:FS} {I,A:FS} {I,A:FS}.." -f "string0", "string1", "string2"...

https://social.technet.microsoft.com/wiki/contents/articles/7855.powershell-using-the-f-format-operator.aspx

Arrays

- Scalar array = array with only one member
- Array variables = array with multiple members
- Normal = index of members by number form 0

Declare array

[int[]]$array = 1,2,3

$a = 1..8

$a=@(1..8)

Hash table

- Is array as collection of data in pair (key/value)
- PS v3.0 comes with [ordered] hash table
- keys must be unique - use Contains() or ContainsKey() to check if some value actually exist

$hash = [ordered]@{key1=Value1;key1=value1}

DEMO::

```
Windows PowerShell

PS C:\> $a = @{"Name"="John";"Surname"="Rambo"}
PS C:\> $a.Contains('Name')
True
PS C:\> $a

Name                        Value
----                        -----
Name                        John
Surname                     Rambo


PS C:\> $a.Add('Address','Jungle')
PS C:\> $a

Name                        Value
----                        -----
Name                        John
Address                     Jungle
Surname                     Rambo


PS C:\> $a.Remove('Address')
PS C:\> $a

Name                        Value
----                        -----
Name                        John
Surname                     Rambo


PS C:\> $a.Name
John
PS C:\>
```

NEXT

::Free book

https://books.goalkicker.com/PowerShellBook/

::PowerShell cook book free

http://www.powertheshell.com/cookbooks/