

PowerShell

Εαρινό Εξάμηνο 2014 - 2015

Ανδρέας Καστανάς

Μιχάλης Σαματάς

Παναγιώτης Αχιλλέως

Ιάκωβος Στυλιανού



Λίγα λόγια για την Microsoft PowerShell

- Η **PowerShell** αποτελείται από ένα κέλυφος γραμμής εντολών και την σχετική scripting γλώσσα χτισμένη πάνω στο **.NET Framework**.
- **.NET Framework**: Αναπτύσσεται από τη Microsoft. Στηρίζεται πάνω στην **Framework Class Library**, μια κοινή βιβλιοθήκη κλάσεων που παρέχει διαλειτουργικότητα μεταξύ των γλωσσών που εμπεριέχονται στο Framework.

Λίγα λόγια για την Microsoft PowerShell

- Η **PowerShell** παρέχει πλήρη πρόσβαση στα COM(Component Object Model) και WMI(Windows Management Instrumentation), επιτρέποντας στους διαχειριστές να εκτελούν διαχειριστικές ενέργειες σε τοπικά αλλά και σε απομακρυσμένα συστήματα Windows
- Παρέχει επίσης τα WS-Management και CIM(Common Information Mode) τα οποία επιτρέπουν τη διαχείριση απομακρυσμένων συστημάτων Linux και συσκευών δικτύου.

Εκδόσεις της PowerShell

- **Version 1.0:** 2006, Windows XP SP2, Windows Server 2003, Windows Vista, Windows Server 2008.
- **Version 2.0:** 2008, Windows 7, Windows Server 2008.
(PowerShell Remoting, Script Debugging)
- **Version 3.0:** 2011, Windows 8, Windows Server 2012.
- **Version 4.0:** 2013, Windows 8.1, Windows Server 2012 R2
- **Version 5.0:** November 2014, Windows 8.1, Windows Server 2012 R2 .

Modes of PowerShell

Normal Mode

Με το Normal mode οι χρήστες δεν μπορούν να βλάψουν τον υπολογιστή τους τόσο εύκολα , λόγω των περιορισμένων δικαιωμάτων χρήστη που έχουν από το σύστημα.

Σε περίπτωση προσπάθειας τα Windows επιστρέφουν ένα μήνυμα λάθους όπως πιο κάτω:

Access to <name> resource was not available to the client or Cannot open <name> service on computer.

Modes of PowerShell

Administrator Mode

Το Administrator mode στο PowerShell δίνει πρόσβαση σε όλη την δύναμη της γλώσσας scripting, ο χρήστης έχει πλήρη πρόσβαση σε ολόκληρο το φάσμα των εξαρτημάτων του συστήματος γι' αυτό πρέπει να γίνονται προσεκτικά οποιεσδήποτε εντολές.

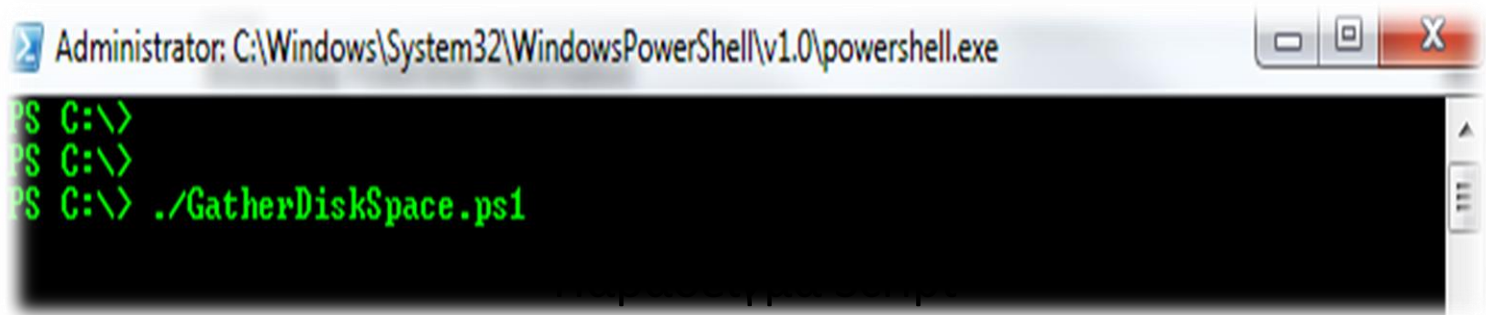
Για παράδειγμα η Get-Disk απαιτεί την κατάλληλη πρόσβαση σε τοπική αποθήκευση, οπότε απαιτείται το admin mode.

```
PS C:\WINDOWS\system32> get-disk
```

Number	Friendly Name	OperationalStatus	Total Size	Partition Style
0	AMD 1X2 Mirror/RAID1 SCSI Disk Device	Online	465.66 GB	MBR
3	Microsoft Storage Space Device	Online	25 TB	GPT
4	Microsoft Storage Space Device	Online	5 GB	MBR
5	Microsoft Storage Space Device	Online	465 GB	GPT

Η μορφή των scripts

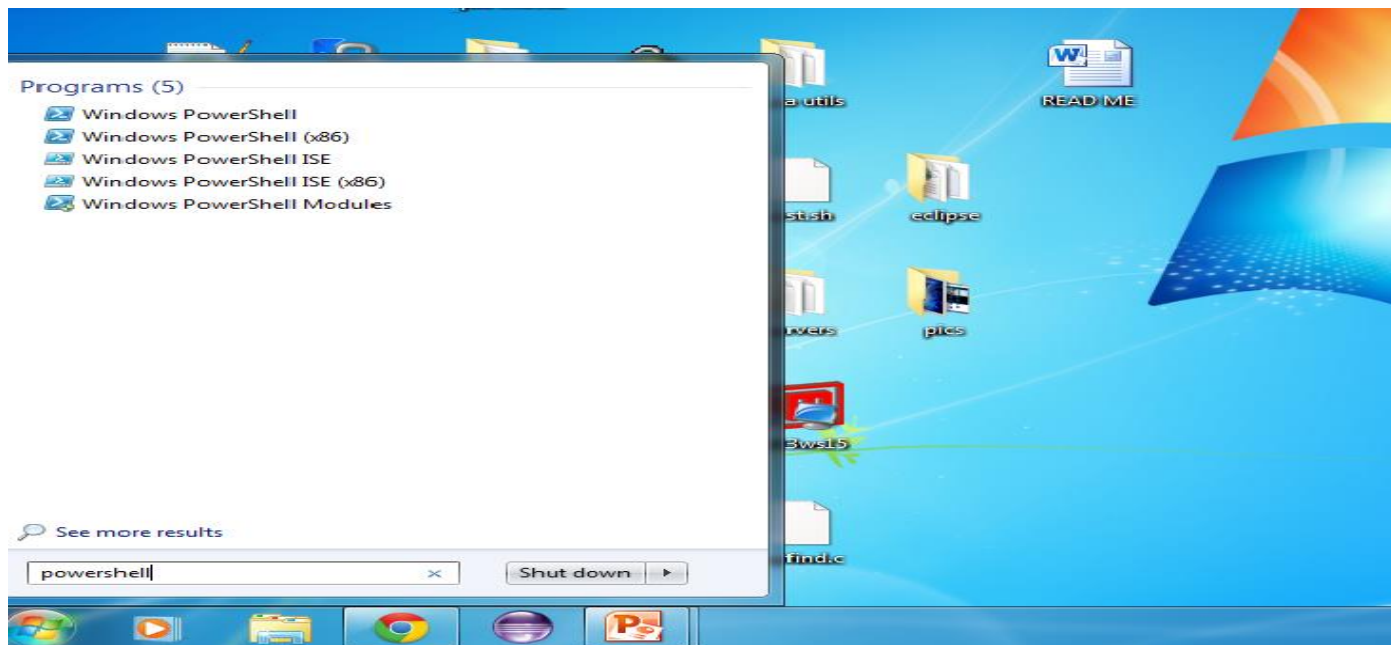
- Τα **PowerShell** scripts τελειώνουν σε `.ps1` και μπορούμε να τα τρέξουμε μέσα σε PowerShell window
- Αν κάνεις double click τα script ανοίγουν στον προεπιλεγμένο επεξεργαστή κειμένου
- Η εντολή για να τρέξω το script πρέπει να είναι στη μορφή `./SampleScript.ps1`



A screenshot of a PowerShell window. The title bar reads "Administrator: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe". The command prompt shows three lines of text: "PS C:\>", "PS C:\>", and "PS C:\> ./GatherDiskSpace.ps1". The text is green on a black background.

Εκκινώντας PowerShell

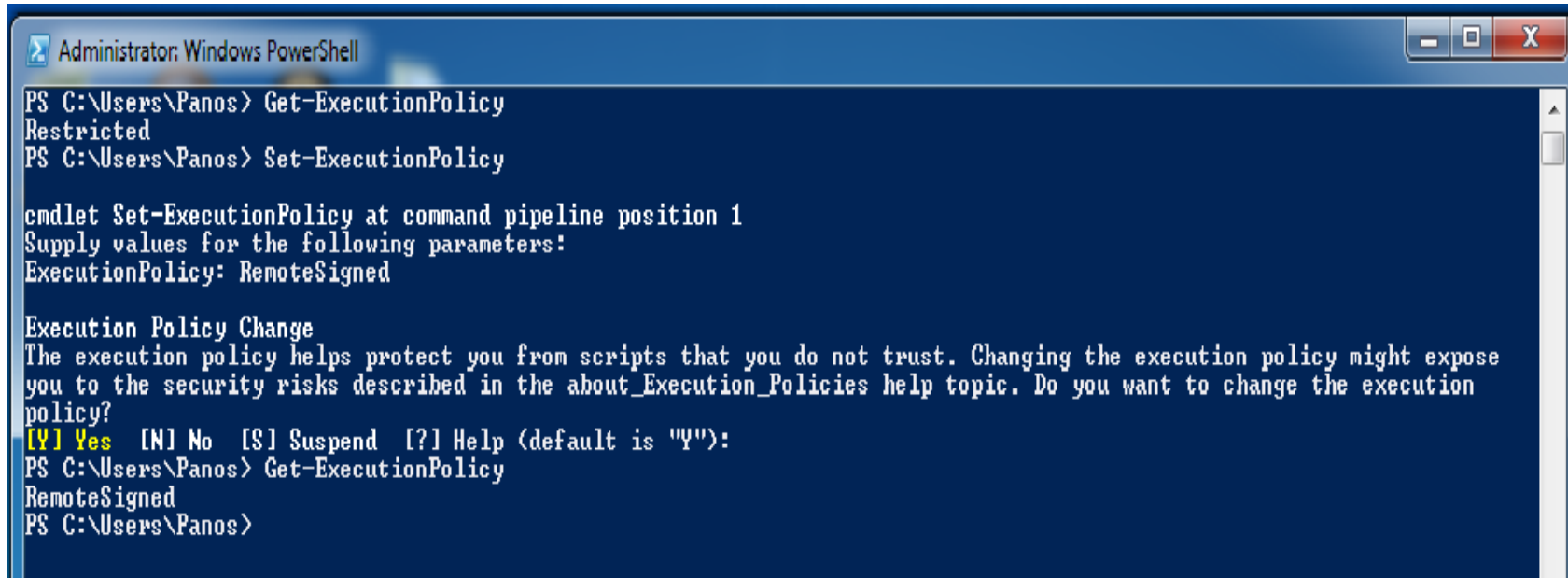
1. Κάνουμε search στο start για το PowerShell
2. Επιλέγουμε το version που θέλουμε



Πως εκτελούμε ένα script στο PowerShell

Αλλαγή Execution Policy

- Με τη χρήση της εντολής Get-ExecutionPolicy μας εμφανίζεται το ExecutionPolicy
- Διαφοροποιούμε το ExecutionPolicy με τη χρήση της εντολής Set-ExecutionPolicy .



```
Administrator: Windows PowerShell
PS C:\Users\Panos> Get-ExecutionPolicy
Restricted
PS C:\Users\Panos> Set-ExecutionPolicy

cmdlet Set-ExecutionPolicy at command pipeline position 1
Supply values for the following parameters:
ExecutionPolicy: RemoteSigned

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help topic. Do you want to change the execution
policy?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"):
PS C:\Users\Panos> Get-ExecutionPolicy
RemoteSigned
PS C:\Users\Panos>
```

Πως εκτελούμε ένα script στο PowerShell

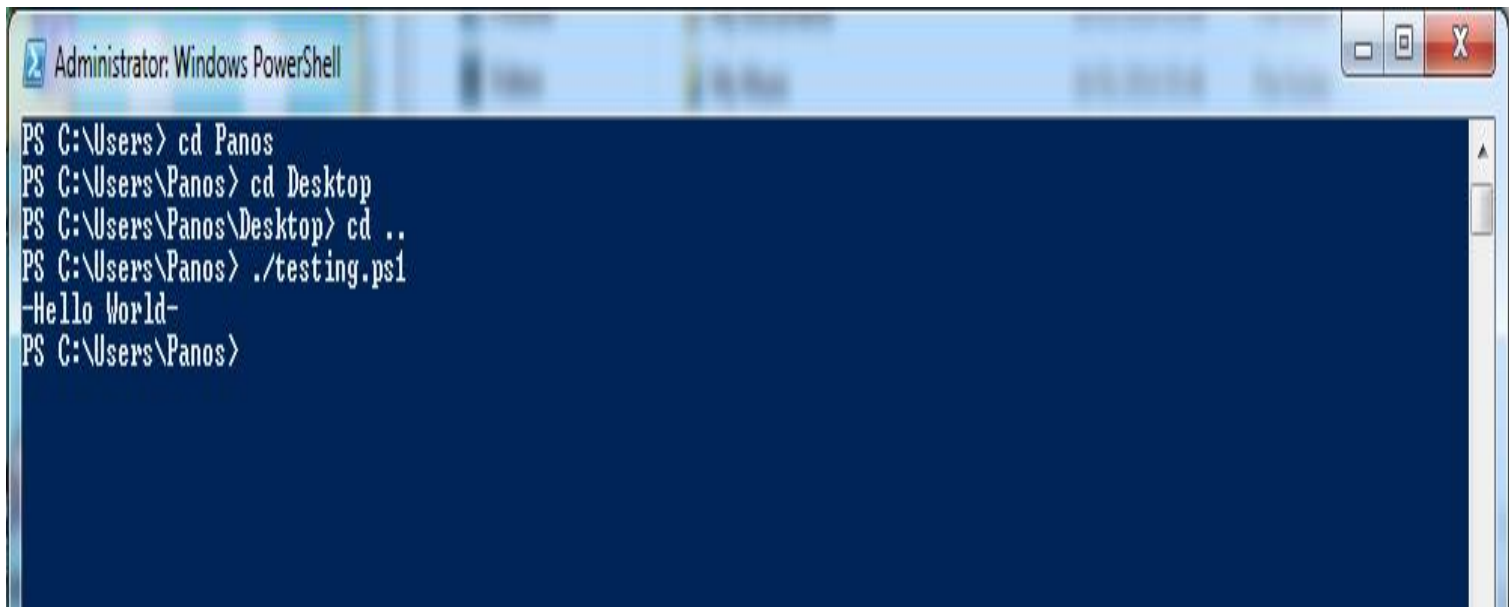
Αλλαγή Execution Policy(συνέχεια)

Τα Execution Policy είναι :

- **Restricted**:Απαγορεύεται η εκτέλεση script
- **All Signed**:Εκτελούνται τα scripts τα οποία έχουν δημιουργηθεί από έμπιστο user .
- **Remote Signed**:Εκτελούνται τα scripts τα οποία έχουν δημιουργηθεί τοπικά ή έχουν δημιουργηθεί από έμπιστο user
- **Unrestricted**: Αφαιρούνται όλοι οι περιορισμοί που αφορούν την πολιτική εκτέλεσης εντολών .

Μετακίνηση στα Folder μας

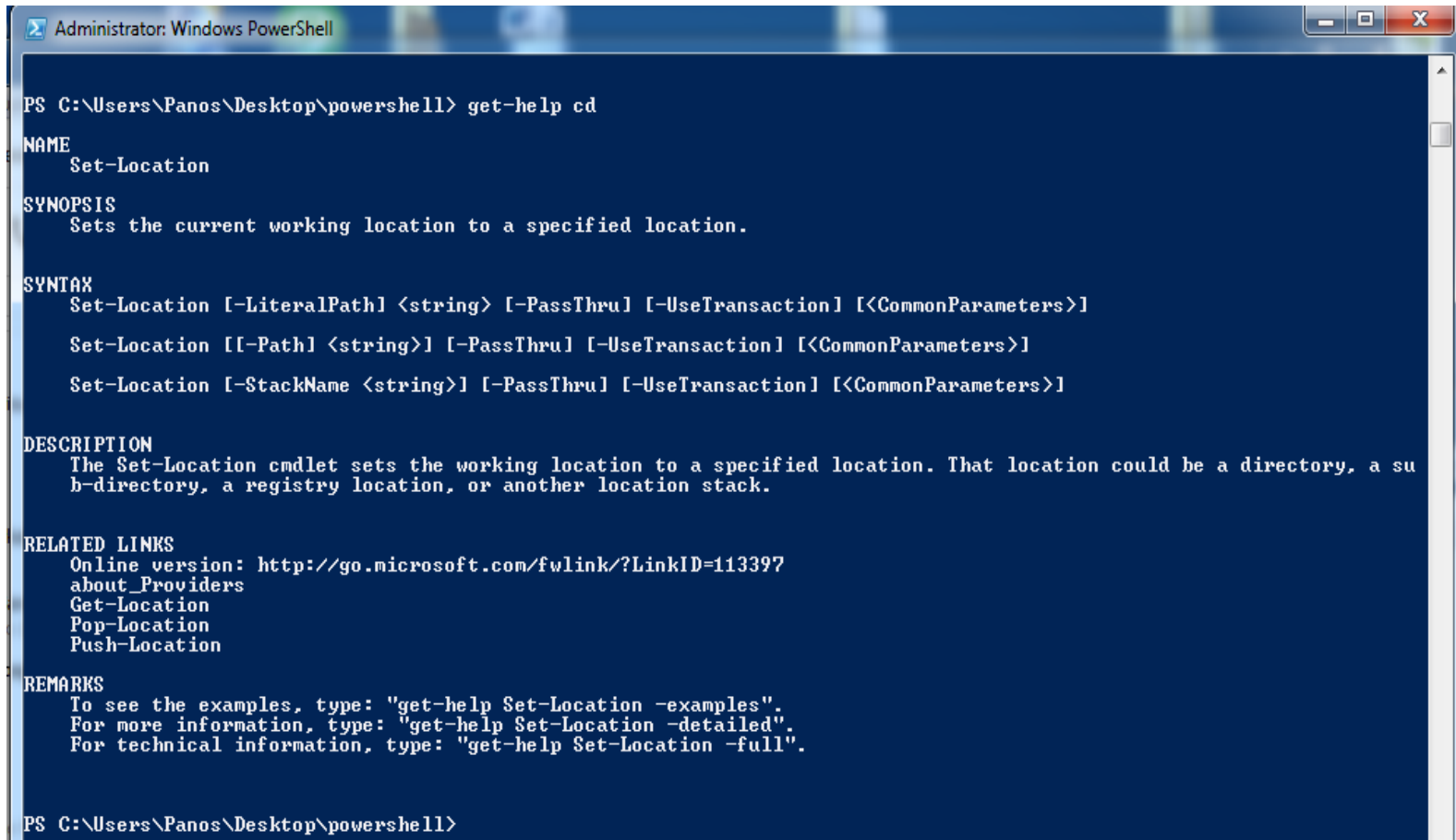
- Με τη χρήση της εντολής `cd` (όπως και στο terminal των linux) μετακινούμαστε στα directories του συστήματός. Αφού μετακινηθούμε στο φάκελο που βρίσκεται το script μας τρέχουμε το `.ps1` αρχείο όπως φαίνεται πιο κάτω .



```
Administrator: Windows PowerShell
PS C:\Users> cd Panos
PS C:\Users\Panos> cd Desktop
PS C:\Users\Panos\Desktop> cd ..
PS C:\Users\Panos> ./testing.ps1
-Hello World-
PS C:\Users\Panos>
```

Κάποιες βασικές εντολές της PowerShell

- Get-help



```
Administrator: Windows PowerShell

PS C:\Users\Panos\Desktop\powershell> get-help cd

NAME
    Set-Location

SYNOPSIS
    Sets the current working location to a specified location.

SYNTAX
    Set-Location [-LiteralPath] <string> [-PassThru] [-UseTransaction] [<CommonParameters>]

    Set-Location [[-Path] <string>] [-PassThru] [-UseTransaction] [<CommonParameters>]

    Set-Location [-StackName <string>] [-PassThru] [-UseTransaction] [<CommonParameters>]

DESCRIPTION
    The Set-Location cmdlet sets the working location to a specified location. That location could be a directory, a sub-directory, a registry location, or another location stack.

RELATED LINKS
    Online version: http://go.microsoft.com/fwlink/?LinkID=113397
    about_Providers
    Get-Location
    Pop-Location
    Push-Location

REMARKS
    To see the examples, type: "get-help Set-Location -examples".
    For more information, type: "get-help Set-Location -detailed".
    For technical information, type: "get-help Set-Location -full".

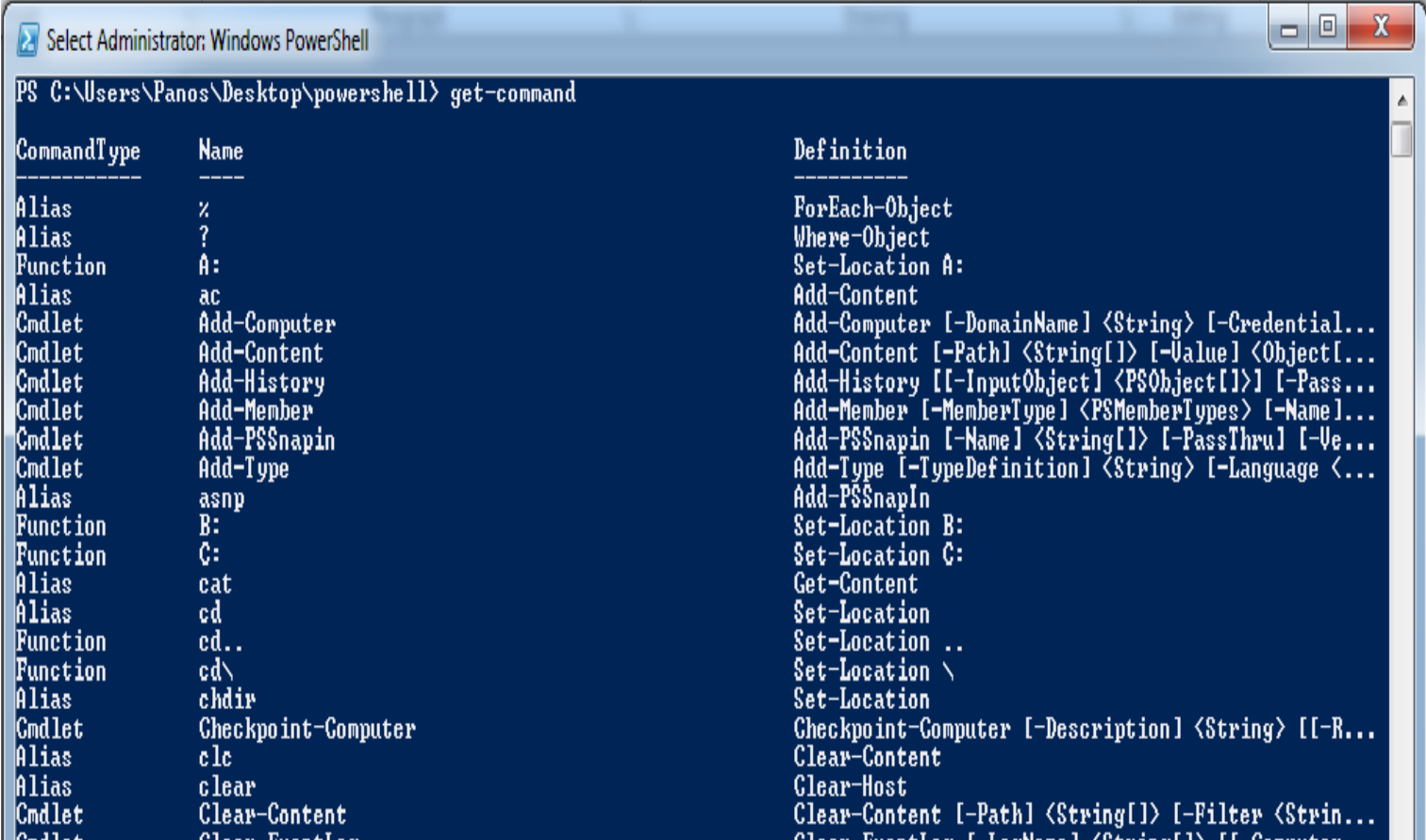
PS C:\Users\Panos\Desktop\powershell>
```

Βασικές Εντολές στο PowerShell

PowerShell (Cmdlet)	PowerShell (Alias)	CMD.EXE / COMMAND.COM	Unix shell	Description
Get-ChildItem	gci, dir, ls	dir	ls	List all files / directories in the (current) directory
Get-Content	gc, type, cat	type	cat	Get the content of a file
Get-Command	gcm	help	apropos, type, which	List available commands
Get-Help	help, man	help	man	Help on commands
Clear-Host	cls, clear	cls	clear	Clear the screen ^[Note 1]
Copy-Item	cp, copy, cp	copy	cp	Copy one or several files / a whole directory tree
Move-Item	mi, move, mv	move	mv	Move a file / a directory to a new location
Remove-Item	ri, del, erase, rmdir, rd, rm	del, erase, rmdir, rd	rm, rmdir	Delete a file / a directory
Rename-Item	ri, ren, mv	ren, rename	mv	Rename a file / a directory
Get-Location	gl, pwd	cd	pwd	Display the current directory/present working directory.
Pop-Location	popd	popd	popd	Change the current directory to the directory most recently pushed onto the stack
Push-Location	pushd	pushd	pushd	Push the current directory onto the stack
Set-Location	sl, cd, chdir	cd, chdir	cd	Change the current directory
Tee-Object	tee	N/A	tee	Pipe input to a file or variable, then pass the input along the pipeline
Write-Output	echo, write	echo	echo	Print strings, variables etc. to standard output
Get-Process	gps, ps	tlst, ^[Note 2] tasklist ^[Note 3]	ps	List all currently running processes
Stop-Process	spps, kill	kill, ^[Note 2] taskkill ^[Note 3]	kill ^[Note 4]	Stop a running process
Select-String	sls	find, findstr	grep	Print lines matching a pattern
Set-Variable	sv, set	set	env, export, set, setenv	Set the value of a variable / create a variable
Invoke-WebRequest	iwr, curl, wget	N/A	wget, cURL	Gets content from a web page on the Internet

Κάποιες βασικές εντολές της PowerShell

- **Get-Command** : επιστρέφει λίστα με όλα τα διαθέσιμα cmdlets .



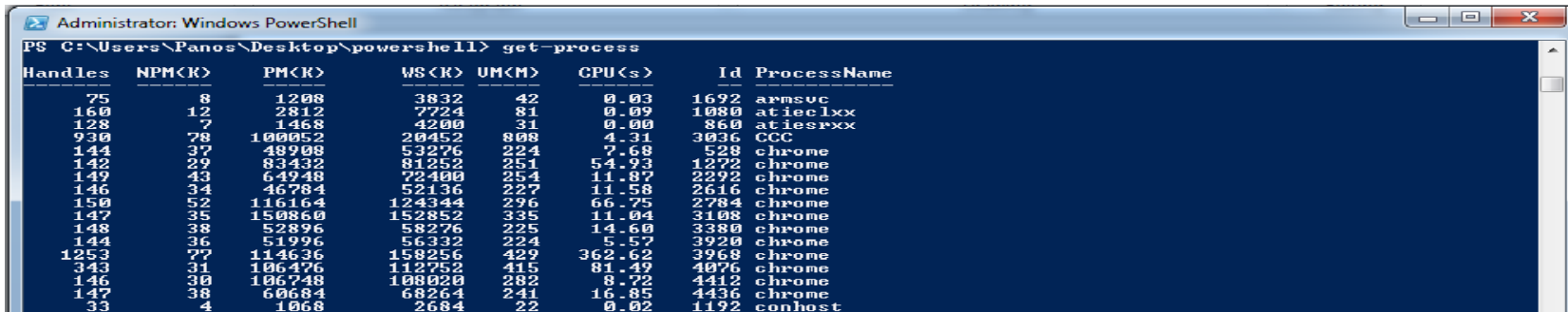
```
Select Administrator: Windows PowerShell
PS C:\Users\Panos\Desktop\powershell> get-command

CommandType      Name                Definition
-----
Alias             %                  ForEach-Object
Alias             ?                  Where-Object
Function          A:                  Set-Location A:
Alias             ac                  Add-Content
Cmdlet            Add-Computer       Add-Computer [-DomainName] <String> [-Credential...
Cmdlet            Add-Content         Add-Content [-Path] <String[]> [-Value] <Object[...
Cmdlet            Add-History         Add-History [[-InputObject] <PSObject[]>] [-Pass...
Cmdlet            Add-Member          Add-Member [-MemberType] <PSMemberTypes> [-Name]...
Cmdlet            Add-PSSnapin       Add-PSSnapin [-Name] <String[]> [-PassThru] [-Ve...
Cmdlet            Add-Type            Add-Type [-TypeDefinition] <String> [-Language <...
Alias             asnp               Add-PSSnapIn
Function          B:                  Set-Location B:
Function          C:                  Set-Location C:
Alias             cat                 Get-Content
Alias             cd                  Set-Location
Function          cd..                Set-Location ..
Function          cd\                 Set-Location \
Alias             chdir              Set-Location
Cmdlet            Checkpoint-Computer Checkpoint-Computer [-Description] <String> [[-R...
Alias             clc                 Clear-Content
Alias             clear               Clear-Host
Cmdlet            Clear-Content       Clear-Content [-Path] <String[]> [-Filter <Strin...
Cmdlet            Clear-EventLog     Clear-EventLog [-LogName] <String[]> [-Computer
```

Κάποιες βασικές εντολές της PowerShell

Get-process: Επιστρέφει όλες τις διεργασίες που τρέχουν στον υπολογιστή.

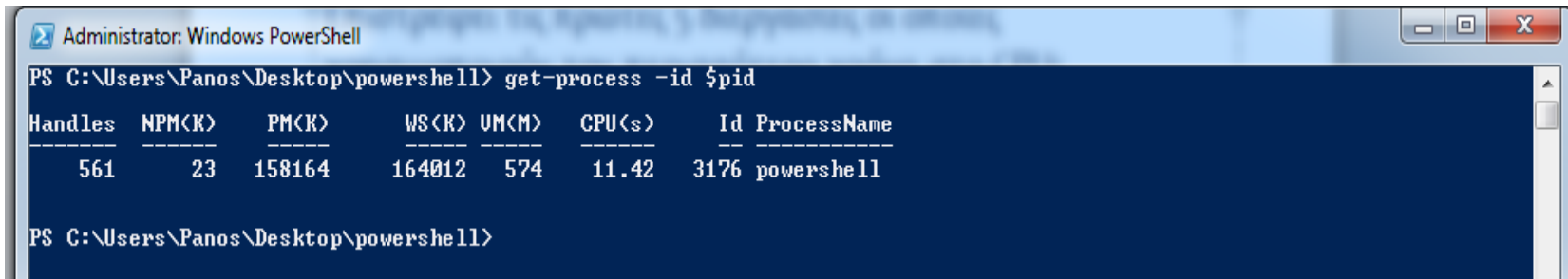
Παραδείγματα get-process



```
Administrator: Windows PowerShell
PS C:\Users\Panos\Desktop\powershell> get-process
```

Handles	NPM(K)	PM(K)	WS(K)	UM(M)	GPU(s)	Id	ProcessName
75	8	1208	3832	42	0.03	1692	smssvc
160	12	2812	7724	81	0.09	1080	atieclxx
128	7	1468	4200	31	0.00	860	atiesrxx
930	78	100052	20452	808	4.31	3036	CCC
144	37	49908	53276	224	7.68	528	chrome
142	29	83432	81252	251	54.23	1272	chrome
149	43	64948	22400	254	11.87	2292	chrome
146	34	46784	52136	227	11.58	2616	chrome
150	52	116164	124344	296	66.75	2784	chrome
147	35	150860	152852	335	11.04	3108	chrome
148	38	52896	58276	225	14.60	3380	chrome
144	36	51996	56332	224	5.57	3920	chrome
1253	77	114636	158256	429	362.62	3968	chrome
343	31	106476	112752	415	81.49	4076	chrome
146	30	106748	108020	282	8.72	4412	chrome
147	38	60684	68264	241	16.85	4436	chrome
33	4	1068	2684	22	0.02	1192	conhost

- **Get-process-id \$pid** :επιστρέφει την διεργασία του PowerShell



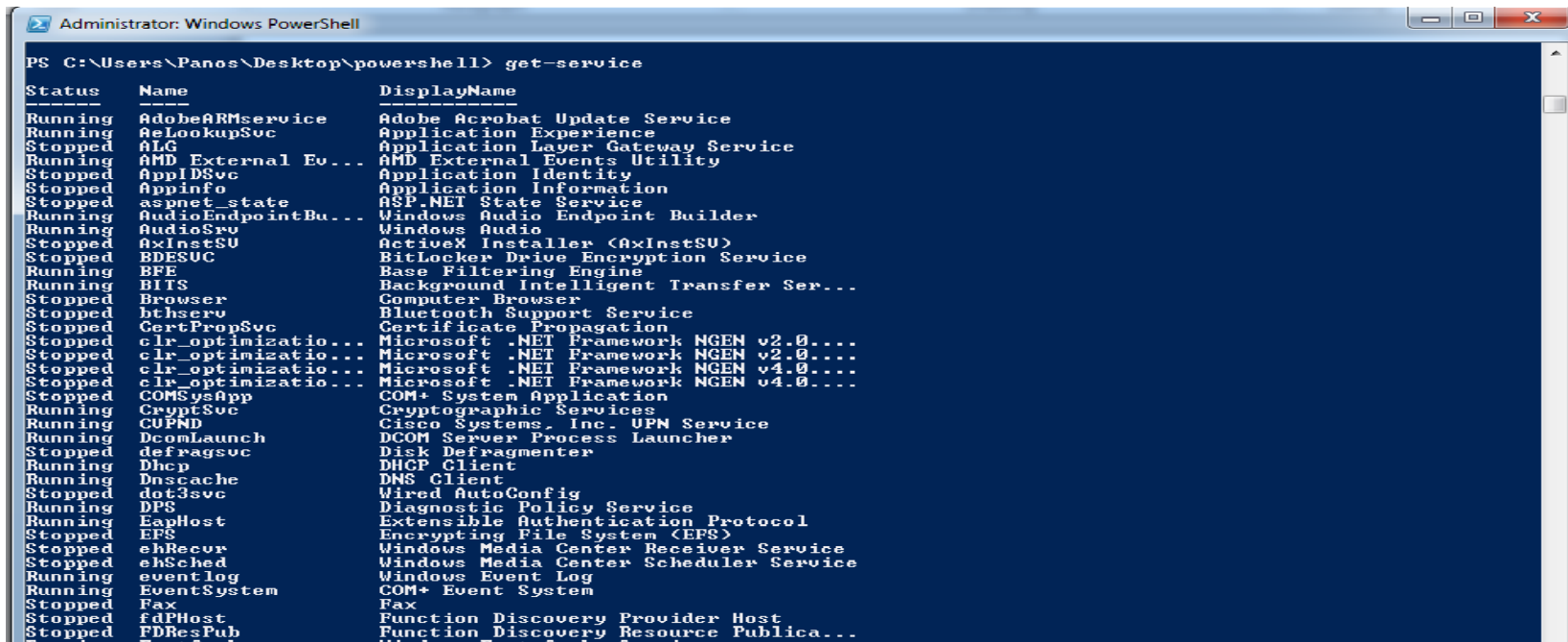
```
Administrator: Windows PowerShell
PS C:\Users\Panos\Desktop\powershell> get-process -id $pid
```

Handles	NPM(K)	PM(K)	WS(K)	UM(M)	GPU(s)	Id	ProcessName
561	23	158164	164012	574	11.42	3176	powershell

```
PS C:\Users\Panos\Desktop\powershell>
```


Κάποιες βασικές εντολές της PowerShell

- **Get-Service** : Επιστρέφει όλα τα Windows services.
- Επιστρέφει όλα τα services τα οποία τρέχουν :
Get-service | where-object {\$_.Status -eq "Running"}

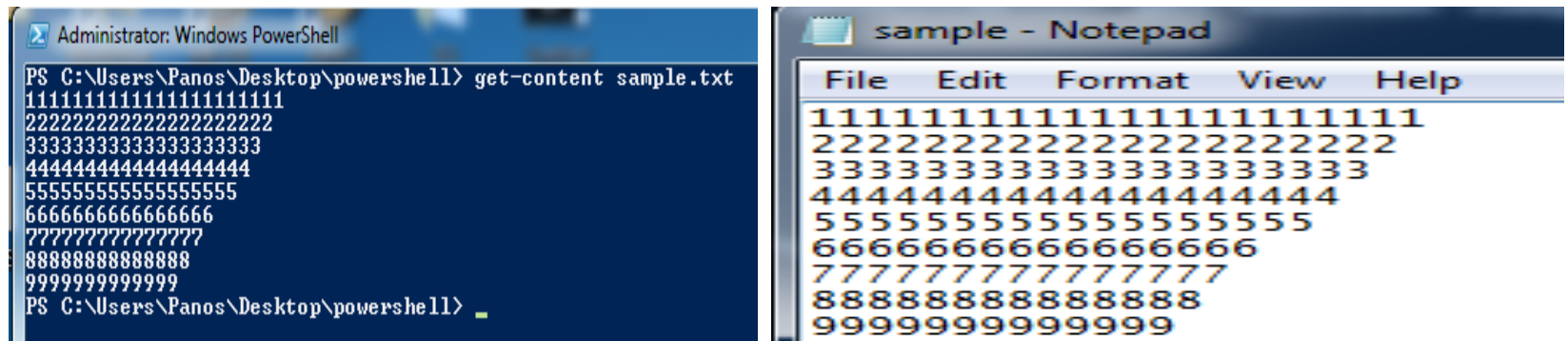


```
Administrator: Windows PowerShell
PS C:\Users\Panos\Desktop\powershell> get-service

Status      Name                               DisplayName
-----
Running     AdobeARMservice                   Adobe Acrobat Update Service
Running     AeLookupSvc                       Application Experience
Stopped     ALG                                Application Layer Gateway Service
Running     AMD External Events Utility       AMD External Events Utility
Stopped     AppIDSvc                           Application Identity
Stopped     Appinfo                            Application Information
Stopped     aspnet_state                       ASP.NET State Service
Running     AudioEndpointBuilder              Windows Audio Endpoint Builder
Running     AudioSrv                           Windows Audio
Stopped     AxInstSU                           ActiveX Installer (AxInstSU)
Stopped     BDESVC                             BitLocker Drive Encryption Service
Running     BFE                                Base Filtering Engine
Running     BITS                               Background Intelligent Transfer Ser...
Stopped     Browser                            Computer Browser
Stopped     bthseru                            Bluetooth Support Service
Stopped     CertPropSvc                       Certificate Propagation
Stopped     clr_optimizatio...                Microsoft .NET Framework NGEN v2.0...
Stopped     clr_optimizatio...                Microsoft .NET Framework NGEN v2.0...
Stopped     clr_optimizatio...                Microsoft .NET Framework NGEN v4.0...
Stopped     clr_optimizatio...                Microsoft .NET Framework NGEN v4.0...
Stopped     COMSysApp                         COM+ System Application
Running     CryptSvc                          Cryptographic Services
Running     CUPND                              Cisco Systems, Inc. UPN Service
Running     DcomLaunch                        DCOM Server Process Launcher
Stopped     defragsvc                         Disk Defragmenter
Running     Dhcp                               DHCP Client
Running     Dnscache                          DNS Client
Stopped     dot3svc                           Wired AutoConfig
Running     DPS                               Diagnostic Policy Service
Running     EapHost                           Extensible Authentication Protocol
Stopped     EFS                               Encrypting File System (EFS)
Stopped     ehRecur                           Windows Media Center Receiver Service
Stopped     ehSched                           Windows Media Center Scheduler Service
Running     eventlog                          Windows Event Log
Running     EventSystem                       COM+ Event System
Stopped     Fax                                Fax
Stopped     fdPHost                           Function Discovery Provider Host
Stopped     FDResPub                          Function Discovery Resource Publica...
Running     ...
```

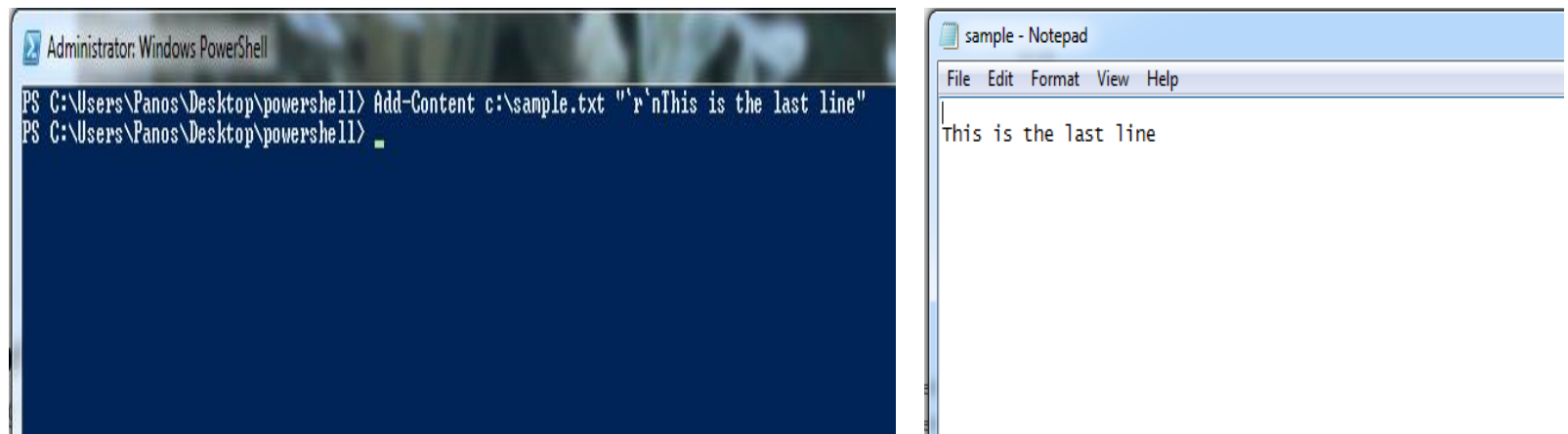

Κάποιες βασικές εντολές της PowerShell

- **Get-Content**: Διαβάζει αρχεία κειμένου, αντιμετωπίζοντας κάθε γραμμή σαν child object.



The screenshot shows two windows. On the left, the 'Administrator: Windows PowerShell' window displays the command `get-content sample.txt` and its output, which consists of ten lines of repeating characters: 11s, 22s, 33s, 44s, 55s, 66s, 77s, 88s, and 99s. On the right, the 'sample - Notepad' window shows the same ten lines of repeating characters, demonstrating that the command correctly reads the file's content.

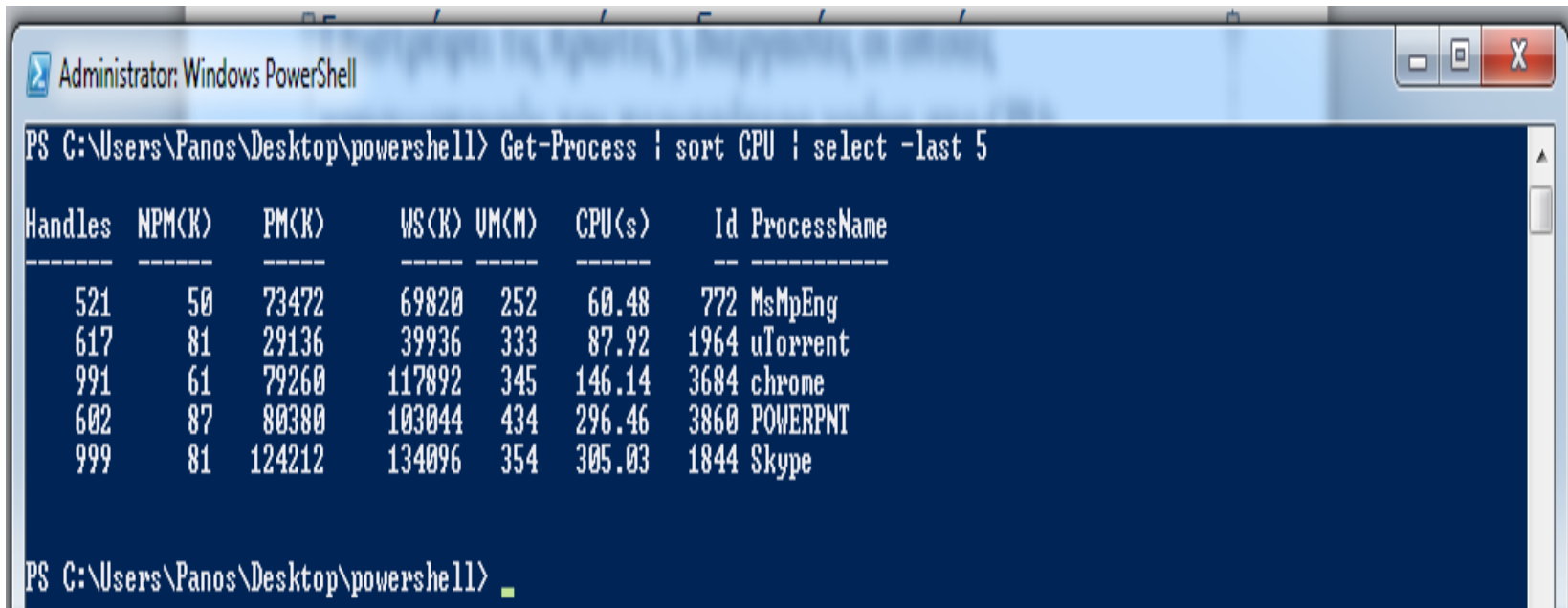
- **Add-Content**: προσθέτει περιεχόμενο σε ένα txt αρχείο



The screenshot shows two windows. On the left, the 'Administrator: Windows PowerShell' window displays the command `Add-Content c:\sample.txt "r'nThis is the last line"` and the prompt `PS C:\Users\Panos\Desktop\powershell>`. On the right, the 'sample - Notepad' window shows the text `this is the last line` added to the file, demonstrating the `Add-Content` command's functionality.

Πιο περίπλοκες εντολές PowerShell

- **Get-Process | sort CPU | select -last 5**: Επιστρέφονται οι 5 διεργασίες με τον περισσότερο χρόνο στην CPU.



```
Administrator: Windows PowerShell
PS C:\Users\Panos\Desktop\powershell> Get-Process | sort CPU | select -last 5
```

Handles	NPM(K)	PM(K)	WS(K)	VM(M)	CPU(s)	Id	ProcessName
521	50	73472	69820	252	60.48	772	MsMpEng
617	81	29136	39936	333	87.92	1964	uTorrent
991	61	79260	117892	345	146.14	3684	chrome
602	87	80380	103044	434	296.46	3860	POWERPNT
999	81	124212	134096	354	305.03	1844	Skype

```
PS C:\Users\Panos\Desktop\powershell>
```

Ονόματα μεταβλητών

- Μπορείτε να χρησιμοποιήσετε σχεδόν οποιοδήποτε όνομα της μεταβλητής , τα ονόματα δεν case sensitive.
- Όμως , υπάρχουν χαρακτήρες που δεν επιτρέπεται να χρησιμοποιηθούν , όπως οι ? ! @ # % & , .
- Αυτοί οι χαρακτήρες μπορούν να γίνουν escape μέσω της χρήσης του { } .

`$Microsoft $MicroSoft $microsoft` είναι τα ίδια .

`${My English Name is #kokos@}` είναι εντάξει.

Είδη Μεταβλητών στη PowerShell.

- Τα είδη των μεταβλητών στη **PowerShell** είναι βασισμένα πάνω στο [.NET Framework](#).
- Μερικά παραδείγματα μεταβλητών είναι:

[adsi], [array], [bool], [byte], [char]
[datetime], [decimal], [double]
[int] or [int32], [long]
[single], [scriptblock], [string]
[WMI], [WMIclass], [xml]

Μεταβλητή \$_

Η μεταβλητή \$_ είναι μια αναφορά στο **this** με αντίστοιχη λειτουργικότητα με το xargs του unix.

Περιέχει την τρέχουσα τιμή του pipeline στη γραμμή εντολών.

```
dir *.txt | foreach-object {$_ .length}
```

Επιστρέφει το μέγεθος όλων των αρχείων κειμένου στο directory

Δήλωση Μεταβλητών

Δηλώνουμε με τον ποίο κάτω τρόπο :

```
$a=120  
$b=21.321  
$c="Hello "
```

Αν εκτελέσουμε το ποιο κάτω :

```
$a.GetType().Name  
$b.GetType().Name  
$c.GetType().Name
```

Θα πάρουμε τα εξής αντίστοιχα :

```
Int32  
Double  
String
```

Χρήση Μεταβλητών

```
$a=333
```

```
$b=66.123
```

```
$Title="manager"
```

```
[int]$Age=22
```

Constant Variables: \$server = '10.10.10.10'

Set-Variable server -option ReadOnly

```
$a.GetType().Name
```

Int32

```
$Title.length
```

7

```
$Title.CompareTo("manager")
```

0

Αριθμητικοί τελεστές

Operator	Description	Example	Result
+	Add two values together.	2+4	6
		"Hi " + "there"	"Hi There"
		1,2,3 + 4,5,6	1,2,3,4,5,6
*	Multiply 2 values.	2 * 4	8
		"a" * 3	"aaa"
		1,2 * 2	1,2,1,2
-	Subtract one value from another.	6-2	4
/	Divide two values.	6/2	3
		7/4	1.75
%	Return the remainder from a division operation.	7%4	3

Τελεστές σύγκρισης

Operator	Description	Example	Result
-eq	equals	<code>\$a = 5 ; \$a -eq 4</code>	False
-ne	not equal	<code>\$a = 5 ; \$a -ne 4</code>	True
-gt	greater than	<code>\$a = 5 ; \$a -gt 4</code>	True
-ge	greater than or equal to	<code>\$a = 5 ; \$a -ge 5</code>	True
-lt	less than	<code>\$a = 5 ; \$a -lt 5</code>	False
-le	less than or equal to	<code>\$a = 5 ; \$a -le 5</code>	True
-like	wildcard comparison	<code>\$a = "This is Text" ; \$a -like "Text"</code>	False
-notlike	wildcard comparison	<code>\$a = "This is Text" ; \$a -notlike "Text"</code>	True
-match	regular expression comparison	<code>\$a = "Text is Text" ; \$a -match "Text"</code>	True
-notmatch	regular expression comparison	<code>\$a = "This is Text" ; \$a -notmatch "Text"</code>	False

Κανονικές εκφράσεις

<code>\n</code>	New Line
<code>[]</code>	Character_group
<code>[^]</code>	Character not in character group
<code>[first-last]</code>	Character in range
<code>.</code>	Matches any single character
<code>\s</code>	Whitespace
<code>\d</code>	Decimal digits
<code>^</code>	The match must start at the beginning of a string or line
<code>\$</code>	The match must occur at the end of the string or before <code>\n</code>
<code>*</code>	Matches the previous element zero or more times
<code>+</code>	Matches the previous element one or more times
<code>{n}</code>	Matches the previous element exactly n times

Τελεστές Τύπου

Operator	Example	Results	Description
-is	<code>\$true -is [bool]</code>	<code>\$true</code>	True if the type of the left side matches the type of the right side.
	<code>\$true -is [object]</code>	<code>\$true</code>	This is always true—everything is an object except <code>\$null</code> .
	<code>\$true -is [ValueType]</code>	<code>\$true</code>	The left side is an instance of a .NET value type.
	<code>"hi" -is [ValueType]</code>	<code>\$false</code>	A string is not a value type; it's a reference type.
	<code>"hi" -is [object]</code>	<code>\$true</code>	But a string is still an object.
	<code>12 -is [int]</code>	<code>\$true</code>	12 is an integer.
	<code>12 -is 'int'</code>	<code>\$true</code>	The right side of the operator can be either a type literal or a string naming a type.
-isnot	<code>\$true -isnot [string]</code>	<code>\$true</code>	The object on the left side is not of the same type as the right side.
	<code>\$true -isnot [object]</code>	<code>\$true</code>	The null value is the only thing that isn't an object.
-as	<code>'123' -as [int]</code>	<code>123</code>	Takes the left side and converts it to the type specified on the right side.
	<code>123 -as "string"</code>	<code>"123"</code>	Turns the left side into an instance of the type named by the string on the right.

Δομές Επιλογής

```
$a = 4
```

```
If ($a -eq 5){
```

```
    Write-Host "$a = 5"
```

```
}
```

```
Elseif ($a -eq 3){
```

```
    Write-Host "$a = 3"
```

```
}
```

```
Else{
```

```
    Write-Host "$a does not equal 3 or 5"
```

```
}
```

Δομές Επανάληψης

```
$var=0  
while($var -lt 10){  
    Write-Host "$var $var++"  
}
```

```
for ($i=0;$i -lt 10; $++){  
    Write-Host "$i"  
}
```

```
foreach ($var in $array){  
    Write-Host "$var"  
}
```

Pipelines

Τα Pipelines στο **PowerShell** ενώνουν αντικείμενα, δεν κατευθύνουν απλά output text από μια διαδικασία σε μια άλλη.

Δεν υπάρχει η ανάγκη για εκτεταμένη επεξεργασία κειμένου (-cut, grep, κτλ) για να ταιριάζουν τα output text μιας διαδικασίας με τα input requirements μιας άλλης.

```
Get-ChildItem C:\Scripts | Where-Object {$_.Length -gt 200KB} | Sort-Object Length
```

Input/Output

Read from console:

```
$Password = Read-Host -assecurestring "Enter password"
```

Read from file:

```
$ff="C:\temp\tmp.txt"  
$inrec=""  
[System.IO.File]::Exists($ff)  
$obj=new-object System.IO.StreamReader($ff)  
$inrec=$obj.ReadLine()  
$obj.close()
```

Input/Output

Write to Console:

```
$Password = Write-Host "Hello World"
```

Write to File:

```
$outarray | export -csv "somefile.csv"
```

```
Get-Process | Out-File c:\temp\test.txt
```


PowerShell vs Bash

Market share, Community:

Η πλειοψηφία των web servers είναι unix-based και τρέχουν **Bash** scripts.

Legacy

Το **Bash** υπάρχει από το 1989, το **PowerShell** από το 2006.

PowerShell vs Bash

Workflows

Το **PowerShell** υποστηρίζει **workflows**, έτσι τα script του αποθηκεύουν την κατάσταση λειτουργίας τους, και μπορούν να σταματούν και να συνεχίζουν, ακόμα και μετά από restart της μηχανής.

```
workflow Invoke-HelloWorld {"Hello World from workflow"}
```

PowerShell vs Bash

Παράλληλη Επεξεργασία

Το **PowerShell** υποστηρίζει παράλληλη επεξεργασία των script του, αλλά και των εντολών του.

```
foreach -parallel($object in $objectgroup)
{
  ...
  sequence { ... }
}
```

Το **Bash** δεν παρέχει παρόμοια λειτουργικότητα και πρέπει να χρησιμοποιηθούν εξωτερικές βιβλιοθήκες (π.χ GNU parallel).

PowerShell vs Bash

Έλεγχος κινδύνων

Το `-WhatIf` είναι μια παράμετρος στις εντολές του **PowerShell** που μας επιτρέπει να δούμε τα αποτελέσματα της εκτέλεσής τους χωρίς να τις εκτελέσουμε.

```
Get-Childitem C:\SomeFile\*.txt | Remove-Item -WhatIf
```

Το `-Confirm` είναι μια παράμετρος στις εντολές του **PowerShell** που βγάζει ένα διάλογο επιβεβαίωσης στο χρήστη πριν από την εκτέλεση τους.

```
Get-Childitem C:\SomeFile\*.txt | Remove-Item -Confirm
```

Μεταφορά της εργασίας 2 σε PowerShell

FTP Client

Ζητούμενα:

1. Upload σε FTP Server (anonymous) τα αρχεία από ένα τοπικό directory.
2. Κατέβασμα και ανάλυση δεδομένων.

Μεταφορά της εργασίας 2 σε PowerShell

```
if($args.Length -ne 4){  
Write-Host "Usage is: ftpupload filename server username password"  
Exit  
}  
  
$file =$args[0]  
$server=$args[1]  
$Username=$args[2]  
$Password= $args[3]  
  
if(Test-Path $file -pathType container){  
Start-Process 'pscp.exe' -ArgumentList ("-v $file mitsos@pasithea.in.cs.ucy.ac.cy:/home/mitsos/ftp")  
Exit  
}
```

Μεταφορά της εργασίας 2 σε PowerShell

```
if(Test-Path $file -pathType container){

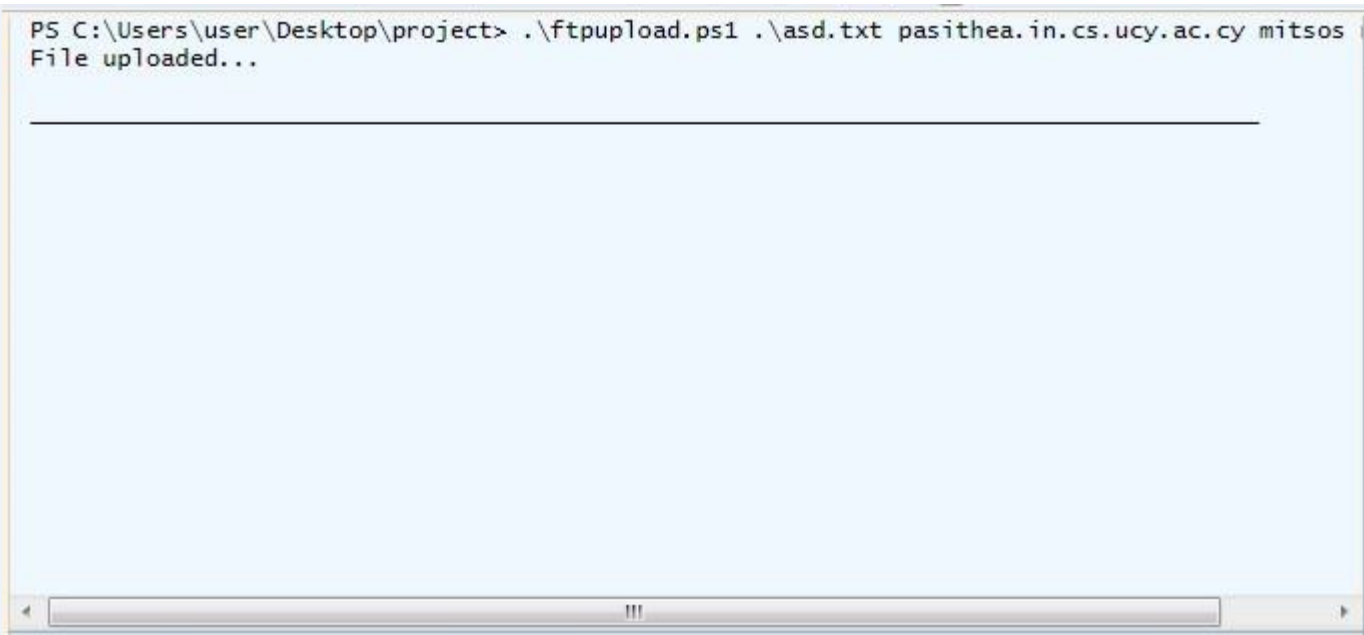
foreach ($item in Get-ChildItem $file)
{
    $item
    $RemoteFile = "ftp://$server/ftp/$file/$item"
    if (Test-Path $item.FullName -PathType Container)
    {
        # Create FTP Rquest Object
        $FTPRequest = [System.Net.FtpWebRequest]::Create("$RemoteFile")
        $FTPRequest = [System.Net.FtpWebRequest]$FTPRequest
        $FTPRequest.Method = [System.Net.WebRequestMethods+Ftp]::MakeDirectory
        $FTPRequest.Credentials = new-object System.Net.NetworkCredential($Username, $Password)
        $FTPRequest.UseBinary = $true
        $FTPRequest.UsePassive = $true
        $FTPRequest.GetResponse();
        foreach ($subitem in Get-ChildItem $item.FullName){
            $subitem
            $FTPRequest.Method = [System.Net.WebRequestMethods+Ftp]::UploadFile
            # Read the File for Upload
            $FileContent = gc -en byte $subitem.FullName
            $FTPRequest.ContentLength = $FileContent.Length
            # Get Stream Request by bytes
            $Run = $FTPRequest.GetRequestStream()
            $Run.Write($FileContent, 0, $FileContent.Length)
            Write-Host "File uploaded..."
            # Cleanup
            $Run.Close()
            $Run.Dispose()
        }
    }
    # Create FTP Rquest Object
    $FTPRequest = [System.Net.FtpWebRequest]::Create("$RemoteFile")
    $FTPRequest = [System.Net.FtpWebRequest]$FTPRequest
    $FTPRequest.Method = [System.Net.WebRequestMethods+Ftp]::UploadFile
    $FTPRequest.Credentials = new-object System.Net.NetworkCredential($Username, $Password)
    $FTPRequest.UseBinary = $true
    $FTPRequest.UsePassive = $true
    # Read the File for Upload
    $FileContent = gc -en byte $item.FullName
    $FTPRequest.ContentLength = $FileContent.Length
    # Get Stream Request by bytes
    $Run = $FTPRequest.GetRequestStream()
    $Run.Write($FileContent, 0, $FileContent.Length)
    Write-Host "File uploaded..."
    # Cleanup
    $Run.Close()
    $Run.Dispose()
}
}
Exit
```

Μεταφορά της εργασίας 2 σε PowerShell

```
$RemoteFile = "ftp://$server/ftp/$file"
# Create FTP Request Object
$FTPRequest = [System.Net.FtpWebRequest]::Create("$RemoteFile")
$FTPRequest = [System.Net.FtpWebRequest]$FTPRequest
$FTPRequest.Method = [System.Net.WebRequestMethod+Ftp]::UploadFile
$FTPRequest.Credentials = new-object System.Net.NetworkCredential($Username, $Password)
$FTPRequest.UseBinary = $true
$FTPRequest.UsePassive = $true
# Read the File for Upload
$FileContent = gc -en byte $file
$FTPRequest.ContentLength = $FileContent.Length
# Get Stream Request by bytes
$Run = $FTPRequest.GetRequestStream()
$Run.Write($FileContent, 0, $FileContent.Length)
Write-Host "File uploaded..."
# Cleanup
$Run.Close()
$Run.Dispose()
```


Μεταφορά της εργασίας 2 σε PowerShell

```
PS C:\Users\user\Desktop\project> .\ftpupload.ps1 .\asd.txt pasithea.in.cs.ucy.ac.cy mitsos  
File uploaded...
```

A screenshot of a PowerShell terminal window. The window has a light blue background and a grey border. At the top, the prompt 'PS C:\Users\user\Desktop\project>' is followed by the command '.\ftpupload.ps1 .\asd.txt pasithea.in.cs.ucy.ac.cy mitsos'. Below the command, the output 'File uploaded...' is displayed. A horizontal line is drawn below the output. At the bottom of the window, there is a scrollbar with a vertical slider in the center.

Μεταφορά της εργασίας 2 σε PowerShell

```
##Check if arguments are valid
if($args.Length -lt 4){
    Write-Host "Usage is: ftpanalyze OPTION server username password"
    Exit
}

$option = $args[0]

if($option -eq "show-dir"){

    $server = $args[1]
    $Username = $args[2]
    $Password = $args[3]
    $RemoteFile = "ftp://$server/ftp/"

    [System.Net.FtpWebRequest]$ftp = [System.Net.WebRequest]::Create($RemoteFile)
    $ftp.Credentials = New-Object System.Net.NetworkCredential($Username,$Password)
    $ftp.Method = [System.Net.WebRequestMethod+FTP]::ListDirectory #Details
    $response = $ftp.GetResponse()
    $stream = $response.GetResponseStream()

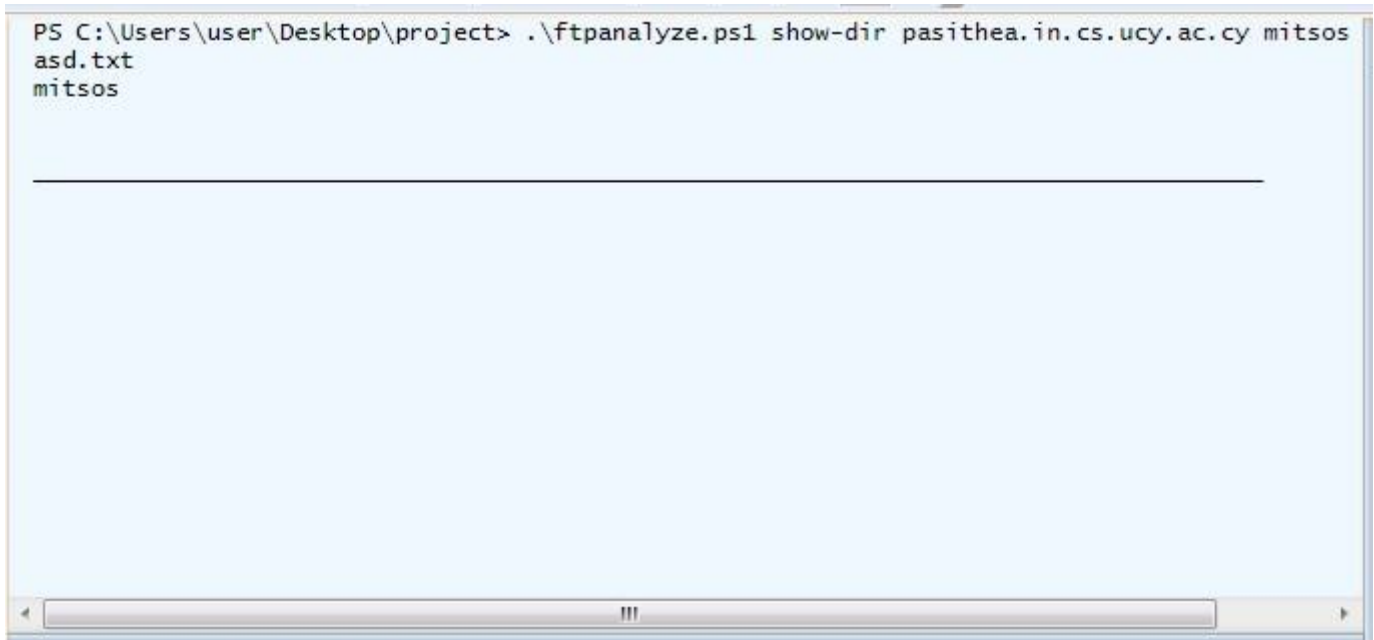
    $buffer = new-object System.Byte[] 1024
    $encoding = new-object System.Text.AsciiEncoding

    $outputBuffer = ""
    $foundMore = $false
```

Μεταφορά της εργασίας 2 σε PowerShell

```
'  
## Read all the data available from the stream, writing it to the  
## output buffer when done.  
do  
{  
    ## Allow data to buffer for a bit  
    start-sleep -m 1000  
  
    ## Read what data is available  
    $foundmore = $false  
    $stream.ReadTimeout = 1000  
  
    do  
    {  
        try  
        {  
            $read = $stream.Read($buffer, 0, 1024)  
  
            if($read -gt 0)  
            {  
                $foundmore = $true  
                $outputBuffer += ($encoding.GetString($buffer, 0, $read))  
            }  
        } catch { $foundMore = $false; $read = 0 }  
    } while($read -gt 0)  
} while($foundmore)  
  
##print results  
$outputBuffer  
}
```

Μεταφορά της εργασίας 2 σε PowerShell



```
PS C:\Users\user\Desktop\project> .\ftpanalyze.ps1 show-dir pasithea.in.cs.ucy.ac.cy mitsos  
asd.txt  
mitsos
```

The screenshot shows a PowerShell terminal window with a light blue background. The prompt is 'PS C:\Users\user\Desktop\project>'. The command entered is '.\ftpanalyze.ps1 show-dir pasithea.in.cs.ucy.ac.cy mitsos'. The output consists of two lines: 'asd.txt' and 'mitsos'. A horizontal line is drawn below the output. The terminal window has a scrollbar at the bottom.

Μεταφορά της εργασίας 2 σε PowerShell

```
elseif($option -eq "show-file"){  
    if($args.Length -lt 5){  
        Write-Host "Usage is: ftpanalyze show-dir FILE server username password"  
        Exit  
    }  
    $file = $args[1]  
    $server = $args[2]  
    $Username = $args[3]  
    $Password = $args[4]  
    $RemoteFile = "ftp://$server/ftp/$file"  
    # Create a FTPWebRequest object to handle the connection to the ftp server  
    $ftprequest = [System.Net.FtpWebRequest]::create($RemoteFile)  
    # set the request's network credentials for an authenticated connection  
    $ftprequest.Credentials = New-Object System.Net.NetworkCredential($Username,$Password)  
  
    $ftprequest.Method = [System.Net.WebRequestMethod+Ftp]::DownloadFile  
    $ftprequest.UseBinary = $true  
    $ftprequest.KeepAlive = $false
```

Μεταφορά της εργασίας 2 σε PowerShell

```
# send the ftp request to the server
$ftpresponse = $ftprequest.GetResponse()

# get a download stream from the server response
$responsestream = $ftpresponse.GetResponseStream()

# create the target file on the local system and the download buffer
try{
    $targetfile = New-Object IO.FileStream ($file,[IO.FileMode]::Create)
    "File created: $file"
    [byte[]]$readbuffer = New-Object byte[] 1024
    #loop through the download stream and send the data to the target file
    do{
        $readlength = $responsestream.Read($readbuffer,0,1024)
        $targetfile.Write($readbuffer,0,$readlength)
    }
    while ($readlength -ne 0)

    $targetfile.close()
    cat $file
    #rem $file
}
catch{
    $_|fl * -Force
}
}
```

Μεταφορά της εργασίας 2 σε PowerShell

```
PS C:\Users\user\Desktop\project> .\ftpanalyze.ps1 show-file asd.txt pasithea.in.cs.ucy.ac.c  
File created: asd.txt  
sadasasdas  
123  
asd  
dsa
```



Ευχαριστούμε για το χρόνο σας!

Υπάρχουν ερωτήσεις;