Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт Информационных Технологий и Анализа Данных

Кафедра вычислительной техники

**Название работы** – “Знакомство с Windows PowerShell (WPS)”

Отчет по лабораторной работе “Лабораторная работа №1”

по дисциплине Операционные системы

Выполнил

Студент, номер группы ИСМб-19-1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.Д.Солопов

Дата: \_\_\_\_\_\_\_\_\_\_

Принял

Должность \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.В.Пестерев

Дата: \_\_\_\_\_\_\_\_\_\_

Иркутск 2021 г.

Содержание

[Постановка задач 3](#_Toc64885293)

[Решение задач 4](#_Toc64885294)

[Задание 1 4](#_Toc64885295)

[Задание 2 6](#_Toc64885296)

Постановка задач

Лабораторная №1 (Знакомство с Windows PowerShell (WPS))

Ознакомившись с литературой по WPS создать:

1. Скрипт, выводящий в виде HTML-страницы информацию о всех командлетах языка WPS. Обязательное условие – страница должна сразу открываться в окне браузера.

2. Выбрать любую запущенную на компьютере службу. Написать скрипт, который выводит в виде HTML-страницы следующую информацию (обязательно!): название службы, называние процесса, который эту службу запустил, ID этого процесса. Вариант вывода выбираете самостоятельно: возможна вставка, используя конструкцию ***Precontent***, возможно конструирование нового объекта посредством команды ***New-Object*** и дальнейший вывод его на конвейер HTML и пр. Обязательное условие – страница должна сразу открываться в окне браузера.

3. Отчет в формате Word, с описанием поставленной задачи, описанием того, какую функцию или функции выполняет ваш скрипт, что получаем в результате работы скрипта, исходный текст скрипта, скриншоты его работы.

Решение задач

Задание 1

**Скрипт:**

$data = "<table><tr><td>Name</td><td>CommandType</td><td>Definition</td></tr><tr><td>"

$val = Get-Command | ForEach-Object {$\_.Name}

$value = ""

foreach($i in $val){

$value = $value + $i + "<br>"

}

$data = $data + $value + "</td><td>"

$val = Get-Command| ForEach-Object {$\_.CommandType}

$value = ""

foreach($i in $val){

$value = $value + $i + "<br>"

}

$data = $data + $value + "</td><td>"

$val = Get-Command| ForEach-Object {$\_.Definition}

$value = ""

foreach($i in $val){

$value = $value + $i + "<br>"

}

$data = $data + $value + "</td></tr></table>"

$data | Out-File command.html

start command.html



Рисунок 1 – Результат работы скрипта

Задание 2

Скрипт:

$data = "<table><tr><td>NameService</td><td>Id</td><td>NameProcess</td></tr>"

$idProcess = (Get-Process svchost | ForEach-Object {$\_.Id})[0]

$string = Get-WmiObject -Class Win32\_Service -Filter ("ProcessId=" + $idProcess)

foreach($i in $string){

$data = $data + "<tr>"

$data = $data + "<td>" + ($i | ForEach-Object {$\_.Name}) + "</td>"

$data = $data + "<td>" + $idProcess + "</td>"

$data = $data + "<td>" + (Get-Process svchost | ForEach-Object {$\_.Name})[0] + "</td>" + "</tr>"

}

$data | Out-File command.html

start command.html

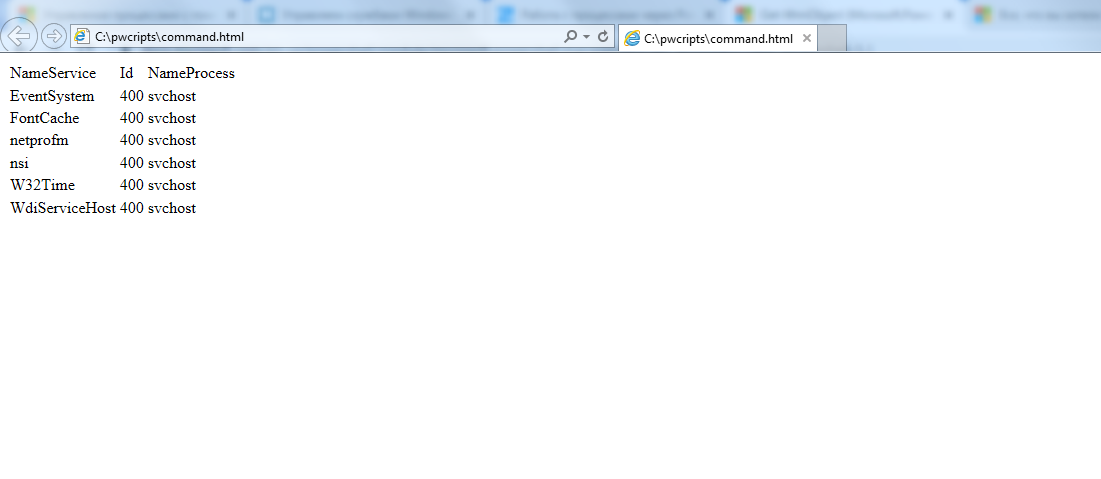


Рисунок 2 – Результат работы скрипта