Чернівецький національний університет імені Юрія Федьковича

Інститут фізико-технічних та комп’ютерних наук

Відділ комп’ютерних технологій

Кафедра математичних проблем управління і кібернетики

**Звіт**

про виконання лабораторної роботи №5 *«Використання фільтрів у PowerShell»*

з дисципліни

**“Адміністрування операційних систем”**

Виконав: студент 441 групи

Бужак Андрій

Перевірив: асист. Коцур М.П.

Оцінка:

Дата захисту:

Чернівці

2019

**Завдання:** Використовуючи засоби PowerShell, розробити два фільтри для кодування та декодування тексту. Літери, із яких складається прізвище студента, кодуються цифрами згідно порядкового номера їх входження (p – 0, у – 1, …, к – 5). При повторному входженні літери використовувати її перший номер. За допомогою фільтрів розробити сценарій, у якому вмістиме файлу test.txt спочатку кодується, а потім декодується. Звернути увагу на обробку чисел у тексті та вигляд кодованого тексту. Передбачити вивід на екран вхідної, кодованої та декодованої інформації. Оформити звіт про виконання роботи, у якому відобразити детальну інформацію про використані командлети та їх параметри, роздрук тексту сценарію, результати тестування сценарію. Роздрукований звіт та працездатність сценарію перевіряє та оцінює викладач.

*Максимальна кількість балів за роботу* – 15.

*Термін здачі* роботи – до 06.12.2019

**Хід виконання**

1. Мною було розроблено сценарій для розв’язування поставленої задачі кодування та декодування тесту.

$filePath = Read-Host -Prompt 'Input file path';

if (![System.IO.File]::Exists($filePath))

{

'File not found.';

pause;

exit;

}

$charToInt = @{};

$intToChar = @{};

filter Encryption

{

if (!$charToInt.ContainsKey($\_))

{

$charToInt.Add($\_, $charToInt.Count);

$intToChar.Add($intToChar.Count, $\_);

}

return $charToInt[$\_].ToString() + ' ';

}

filter Decryption

{

if (($intToChar.ContainsKey([int]$\_)) -and ($\_ -ne ''))

{

return $intToChar[[int]$\_];

}

return '';

}

$originContent = Get-Content -Path $filePath;

$encryptedContent = ($originContent.ToCharArray() | Encryption) -join ' ';

$decryptedContent = ($encryptedContent.split() | Decryption) -join '';

'';

'Origin:';

$originContent;

'';

'Encrypted:';

$encryptedContent;

'';

'Decrypted:';

$decryptedContent;

pause;

**Пояснення до сценарію**

1. Змінній «filePath» присвоюю введений користувачем шлях до файлу.

$filePath = Read-Host -Prompt 'Input file path';

1. Завершую виконання сценарію, якщо файлу не існує.

if (![System.IO.File]::Exists($filePath))

{

'File not found.';

pause;

exit;

}

1. Оголошую хеш-таблиці для кодування і декодування символів.

$charToInt = @{};

$intToChar = @{};

1. Описую фільтр для кодування тексту. Так як найефективніша робота з фільтром це конвеєрна обробка, то опрацьовую поточний елемент з конвеєра у змінній «$\_». Тут відбувається заповнення обох хеш-таблиць та повернення зашифрованого елементу конвеєра.

filter Encryption

{

if (!$charToInt.ContainsKey($\_))

{

$charToInt.Add($\_, $charToInt.Count);

$intToChar.Add($intToChar.Count, $\_);

}

return $charToInt[$\_].ToString() + ' ';

}

1. Описую фільтр для декодування тексту:

filter Decryption

{

if (($intToChar.ContainsKey([int]$\_)) -and ($\_ -ne ''))

{

return $intToChar[[int]$\_];

}

return '';

}

1. Читаю вміст вхідного файлу, обробляю кодування та декодування через конвеєри і виводжу результати.

$originContent = Get-Content -Path $filePath;

$encryptedContent = ($originContent.ToCharArray() | Encryption) -join ' ';

$decryptedContent = ($encryptedContent.split() | Decryption) -join '';

'';

'Origin:';

$originContent;

'';

'Encrypted:';

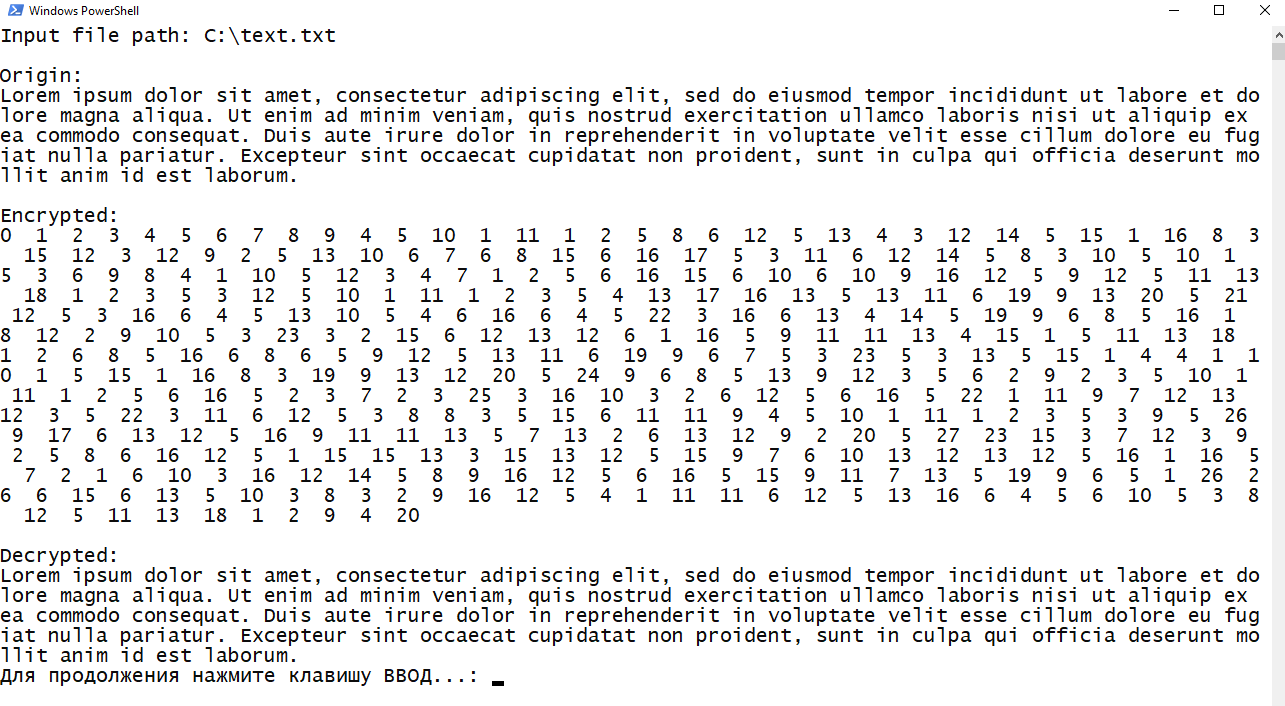
$encryptedContent;

'';

'Decrypted:';

$decryptedContent;

***Тестовий приклад***



***Висновки***

Отже, в процесі виконання лабораторної роботи №5, мною було розроблено сценарій мовою PowerShell, що дозволяє засобами PowerShell та двох фільтрів закодувати заданий в текстовому файлі текст та розкодувати його.