

A1: Сейчас ваша задача состоит в создании списка всех процессов и его сортировке в порядке убывания в соответствии с используемым временем процессора. Вы уже знаете, как это сделать: `Get-Process`, `Sort-Object` и конвейер `()`.

Подсказка: CPU не является параметром `Sort-Object`, это аргумент, который вы можете использовать при сортировке. Поэтому он не имеет символа «-».

A2: Создадим список первых 10 процессов по используемому времени процессора. Для этого возьмем результаты упражнения A1 и добавим к ним команду `Select-Object`. Существует два пути получить идеальное решение, в зависимости от того, как вы хотите отсортировать список. Давайте рассмотрим оба.

Подсказка: в одном из путей используется параметр `-First`, в другом `-Last`.

Выполните задание согласно варианту:

- 1) Из отсортированного по названию процесса списка выберите 5 элементов от начала, пропустив первые 5 элементов списка
 - 2) Из отсортированного по названию процесса списка создайте список таким образом, чтобы в нем содержались только процессы с уникальным именем и отсутствовали первые 5 элементов
 - 3) Из отсортированного по названию процесса списка выберите 10 элементов от конца, пропустив последние 3 элемента списка
 - 4) Из отсортированного по названию процесса списка создайте список таким образом, чтобы в нем содержались только процессы с уникальным именем и отсутствовали последние 5 элементов
 - 5) Из отсортированного по названию процесса списка, содержащего только процессы с уникальными названиями, выберите 3, 5, 7, 10, 13 и 15 элементы
 - 6) Из отсортированного по названию процесса списка выберите 5 элементов от конца, пропустив последние 10 элемента списка
 - 7) Из отсортированного по названию процесса списка создайте список таким образом, чтобы в нем содержались только процессы с уникальным именем и отсутствовали последние 10 элементов
 - 8) Из отсортированного по используемому времени процессора списка, содержащего только процессы с уникальными названиями, выберите 2, 7, 9, 17 и 21 элементы
-

A3: Назначьте переменной `$P` сокращенный список процессов из упражнения A2.

Подсказка: С помощью клавиши курсора «Стрелка вверх» можно вызвать последнюю использованную команду, а с помощью клавиши «Номе» переместить курсор в начало строки, а затем ввести данные. Вывести содержимое переменной можно, просто напечатав в командной строке `$P`.

A4: Возьмите переменную `$P` из упражнения A3 и сохраните ее содержимое в текстовый файл с именем «A4.txt». Затем сохраните содержимое `$P` в файл CSV с именем «A4.CSV», и наконец в файл XML с именем «A4.XML».

Подсказка: при использовании `>` не нужен символ `|`, который требуется только для командлетов, таких как `Out-File`, `Export-CSV` и т.д. Просмотрите результат, для этого можно воспользоваться Блокнотом (Notepad).

Выполните задание согласно варианту:

Выведите полученные данные на экран в виде таблицы

A5: Создайте список всех служб и отсортируйте их по статусу.

Подсказка: Используйте тот же метод, что и для сортировки процессов по используемому времени процессора, но применяйте команду `get-service` и «`status`» в качестве аргумента командлета `Sort-Object`.

A6: Создайте список служб и выведите на экран только атрибуты имени и статуса. Воспользуйтесь ранее описанным циклом `ForEach`, но можете подумать и о других возможных решениях.

Выполните задание согласно варианту:

Приведите альтернативное решение поставленной задачи используя полученные ранее сведения

A7: Создайте список служб и выведите на экран только атрибуты имени и статуса, используя в качестве фона зеленый цвет, а в качестве цвета текста – белый.

Подсказка: Воспользуйтесь решением из A6 и добавьте параметры `-ForegroundColor` и `-BackgroundColor`.

Выполните задание согласно варианту:

- 1) Используйте следующие цвета: текст – серый, фон – белый
 - 2) Используйте следующие цвета: текст – черный, фон – желтый
 - 3) Используйте следующие цвета: текст – красный, фон – зеленый
 - 4) Используйте следующие цвета: текст – черный, фон – желтый
 - 5) Используйте следующие цвета: текст – черный, фон – пурпурный
 - 6) Используйте следующие цвета: текст – белый, фон – темно-красный
 - 7) Используйте следующие цвета: текст – желтый, фон – темно-зеленый
 - 8) Используйте следующие цвета: текст – голубой, фон – серый
-

A8: Вызовите список служб. Отсортируйте список по статусу и окрасьте выводимые данные в красный или зеленый цвет в зависимости от статуса службы «`stopped`» или «`running`».

Подсказка: Сначала используйте `Sort-Object`, как в предыдущих упражнениях. Затем воспользуйтесь циклом `ForEach`, но вместо того, чтобы просто использовать `Write-Host`, добавьте запрос `If`. Вы можете просмотреть статусы служб, используя как обычно `$_status`; возможные значения «`stopped`» или «`running`».

Выполните задание согласно варианту:

Для каждой службы выведите список служб, от которых зависит данная служба (использовать цикл `foreach`). Данный список служб содержится в одном из свойств (property) объекта службы. Воспользуйтесь командлетом `get-member`, чтобы выяснить в каком.

Для полученных списков примените цвета из предыдущего упражнения

A9: Преобразуем выходные данные Get-Service в HTML. Используем командлет Convertto-html, который может работать непосредственно со списком объектов.

Подсказка: Если список слишком длинный, его можно оборвать, нажав CTRL-C.

Выполните задание согласно варианту:

Примените к получаемым данным цветовую схему из упражнения A7 следующим образом: Цвет текста -> цвет текста статуса, цвет фона -> цвет текста названия. Для данной задачи воспользуйтесь циклом foreach. Полученные данные преобразуйте в HTML.

A10: В конце воспользуемся командами, которые, как мы знаем, помещают выходные данные в файл «.A10.html». Просмотрим этот файл.

Выполните задание согласно варианту:

Результаты индивидуального задания из предыдущего упражнения сохраните в файл.