

Благодаря обширным возможностям и гибкости сценариев, PowerShell нашел применение в различных направлениях IT.

Особенно часто сценарии PowerShell применяются для автоматизации процессов администрирования Windows и ее компонентов. С помощью оболочки очень удобно управлять объектами Active Directory, администрировать Microsoft Exchange, работать с виртуальными машинами Microsoft Hyper-V.

Сценарии PowerShell часто используются в комплекте с планировщиком задач windows. Такое использование позволяет выполнять многие операции администрирования (бэкапирование, очистку) без участия человека.

Благодаря активному развитию PowerShell его возможности постоянно расширяются. В этом процессе участвуют как разработчики Microsoft так и сторонние компании.

### Литература

1. Википедия [Электронный ресурс]. – Режим доступа: [http://ru.wikipedia.org/wiki/Windows\\_PowerShell](http://ru.wikipedia.org/wiki/Windows_PowerShell) – Дата доступа 18.04.2014
2. WindowsFAQ. [Электронный ресурс]. – Режим доступа: [http://all-ht.ru/inf/vpc/p\\_0\\_0.html](http://all-ht.ru/inf/vpc/p_0_0.html) – Дата доступа 19.04.2014
3. TechNet [Электронный ресурс]. – Режим доступа: <http://technet.microsoft.com> – Дата доступа 18.04.2014

УДК 004.45

### **ИСПОЛЬЗОВАНИЕ ВИРТУАЛЬНЫХ МАШИН ДЛЯ УПРОЩЕНИЯ ПРОЦЕССА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Безручко А.Н., Бокшиц Е.А.

Научный руководитель – Белова С.В.

Виртуальная машина – это программная и/или аппаратная система, эмулирующая аппаратное обеспечение некоторой платформы и исполняющая программы на ней, или виртуализирующая некоторую платформу и создающая на ней среды, изолирующие друг от друга программы и даже операционные системы.

Система виртуализации – это специализированное программное обеспечение, используемое для имитации работы одной или нескольких реальных ЭВМ.

Впервые термин «Виртуальная машина» появился в конце шестидесятых годов прошлого века, когда электронные вычислительные машины стали неотъемлемой частью жизни человека. Однако, они разительно отличались от современных персональных ЭВМ и представляли собой огромные и

дорогие устройства. Экономически невыгодно было отдавать таких монстров в единоличное владение отдельным пользователям, поэтому начали развиваться интерактивные многотерминальные системы разделения времени.

В таких системах компьютер отдавался в распоряжение сразу нескольким пользователям. Каждый пользователь получал в свое распоряжение терминал, с помощью которого он мог вести диалог с компьютером. Причем время реакции вычислительной системы было достаточно мало для того, чтобы пользователю была не слишком заметна параллельная работа с компьютером. Разделяя, таким образом, компьютер, пользователи получили возможность за сравнительно небольшую плату пользоваться преимуществами компьютеризации. Именно тогда и возникло понятие «Виртуальная машина».

Сегодня, когда практически каждый может получить в единоличное пользование ЭВМ, виртуальные машины переживают второе рождение. Но теперь они призваны не выделять каждому пользователю свою, небольшую часть вычислительных ресурсов ЭВМ, а дать возможность иметь компьютер в компьютере или даже целую вычислительную систему внутри одной единственной ЭВМ.

По существу, виртуальная ЭВМ ничем не отличается от реальной и представляет собой набор программных средств, имитирующих работу реальной ЭВМ. Для нее можно выбирать аппаратную конфигурацию, например, объем оперативной памяти и жесткого диска, наличие аудио адаптеров, дисководов, сетевых плат и других элементов ЭВМ.

Когда виртуальная ЭВМ создана и запущена, то создается полная иллюзия работы с реальным компьютером. Все это дает пользователям множество преимуществ, к которым в первую очередь относятся:

- возможность запускать в рамках одной реальной ЭВМ несколько виртуальных машин, моделировать вычислительные системы и сети;
- возможность имитировать работу реального оборудования, например, бортовых вычислительных систем на стандартной персональной ЭВМ;
- возможность работать сразу с несколькими операционными системами и динамически переключаться между ними;
- возможность быстро переносить виртуальные ЭВМ и вычислительные системы с одного реального компьютера на другой, и размножать их простым копированием файлов виртуальных машин;
- возможность легко изменять конфигурацию виртуальных ЭВМ, добавляя новое оборудование и изменяя параметры уже выбранных элементов.

Но не все так безоблачно. Существует и ряд недостатков свойственных виртуальным машинам:

– виртуализация увеличивает риск отказа работающего на нем программного обеспечения, так как платформа виртуализации сама является программой, которая не застрахована от ошибок, а, следовательно, вероятность краха операционной системы, запущенной на виртуальной машине, существенно увеличивается;

– выход из строя реальной ЭВМ, в рамках которой работало несколько виртуальных машин, приводит к выходу из строя всех этих виртуальных машин;

– при создании виртуальной вычислительной системы надо жестко следить за распределением ресурсов реальной ЭВМ между виртуальными машинами, так как отдельные ЭВМ виртуальной вычислительной системы могут надолго захватить, скажем, процессор и использовать его практически на 100%, что сразу же приведет к падению производительности остальных виртуальных машин;

– виртуальные ЭВМ охватывают не весь спектр оборудования. Особенно, это касается специализированных каналов связи.

Несмотря на возможные проблемы, связанные с виртуальными машинами, их преимущества все больше и больше заставляет обратить внимание на технологию виртуализации не только разработчиков программного обеспечения и организаций, работающих в области высоких технологий, но и простых пользователей ПК.

Сейчас существует большое количество различных эмуляторов, которые позволяют создавать виртуальные машины, например: VirtualBox (Oracle), Virtual PC (Connectix, Microsoft), VMware Workstation (VMware), Hyper-V (Microsoft), Parallels Workstation (Parallels), и др.

Современный процесс разработки программного обеспечения сложно представить без использования виртуальных машин. Они используются на различных этапах: разработка требований, написание кода, тестирование, внедрение и сопровождение.

Такой подход имеет большое количество достоинств:

- уменьшение количества дорогостоящего аппаратного обеспечения числа специалистов для его обслуживания;

- упрощение процесса создания и удаления сред разработки и тестирования;

- гибкость при выборе параметров систем;

- легкость создания резервных копий и их восстановления и др.

В общем случае использование виртуальных машин позволяет упростить и удешевить процесс разработки ПО, делает его более гибким. Однако при этом повышаются риски, связанные с выходом из строя аппаратного обеспечения, так как в этом случае задействуются не одна, а сразу несколько машин.

## Литература

4. Википедия [Электронный ресурс]. – Режим доступа: [http://ru.wikipedia.org/wiki/Виртуальная\\_машина](http://ru.wikipedia.org/wiki/Виртуальная_машина) – Дата доступа 18.04.2014
5. Hit. Все о Hi-Tech. [Электронный ресурс]. – Режим доступа: [http://all-ht.ru/inf/vpc/p\\_0\\_0.html](http://all-ht.ru/inf/vpc/p_0_0.html) – Дата доступа 19.04.2014
6. TechNet [Электронный ресурс]. – Режим доступа: <http://technet.microsoft.com/ru-ru/library/hh831531.aspx> – Дата доступа 18.04.2014
7. VirtualBox documentation [Электронный ресурс]. – Режим доступа: <https://www.virtualbox.org/wiki/Documentation> – Дата доступа 18.04.2014

УДК 004.3

### РАБОТА С ПОРТАМИ ВВОДА И ВЫВОДА НА ПРИМЕРЕ КЛАВИАТУРЫ И ДИНАМИКА

Стальцова Е.А.

Руководитель: Новиков С.О.

Целью этой работы было изучить команды прямого управления портами ввода-вывода и написать программу для работы с ними.

Таким образом, перед нами стояла задача написать программу, во время работы которой, при нажатии на кнопки клавиатуры с цифрами от 0 до 9, динамик будет воспроизводить различные звуки (у каждой кнопки свой звук).

Разложим же реализованную программу по этапам, как двигатель разбирают по частям.

Программа начинает работу с проверки наличия символа в буфере или считывания символа с клавиатуры, если буфер пуст:

*start:*

```
xor dx, dx
```

*repeat:*

```
mov ah, 1
```

```
int 16h
```

```
jz cmp_count
```

```
mov ah, 0
```

```
int 16h
```

Затем происходит проверка считанного символа:

```
cmp al, '1'
```

```
je lop
```

```
...
```

```
cmp al, '0'
```