附录　复习实验

当你完成这本书中指定的章节和实验后，可以继续完成本篇附录中提供的 3 个复习实验。对于你的学习过程来说，复习是一种很好的休息方式，同时可以巩固你已经学到的最为重要的要点。和往常一样，你可以从 MoreLunches.com 网站上找到示例答案。通过找到这本书的封面图片，单击它，然后去下载区下载实验示例解决方案文件即可。

因为这些实验任务中的一部分实验说明命令较为复杂，所以我们已经将这些复杂的说明命令分解为独立的任务小节。同时为了帮助你完成实验，在每个实验开端，我们也提供了一个提示清单来提示你，包括你可能会需要的特定命令、帮助文件和语法。

实验回顾 1：第 1—6 章

注意：为了完成这些实验，你需要一台运行 PowerShell v3 或更新版本的 PowerShell 的计算机。在打算完成这些实验之前，你应该先完成这本书中的第 1—6 章的实验。

提示：

- Sort-Object
- Select-Object
- Import-Module
- Export-CSV
- Help
- Get-ChildItem (Dir)

任务 1

运行一个命令，从而显示应用程序事件日志中最新的 100 个条目，不要使用 Get-WinEvent。

任务 2

写一个仅显示前五个最消耗虚拟内存（VM）进程的命令。

任务 3

创建一个包含所有的服务 CSV 文件，只需要列出服务名称和状态。所有处于运行状态的服务处于停止状态的服务之前。

任务 4

写一个命令行，将 BITS 服务的启动项类型变更为手动。

任务 5

显示你计算机中所有文件名称为 Win*.*的文件，以 C：\开始。注意：为了完成这个实验，你可能需要去实验和使用一些 Cmdlet 命令的新参数。

任务 6

获取一个 C:\Program Files 的目录列表。包含所有的子文件夹，把这些目录列表放到位于 C:\Dir.txt 的文本文件内（记住去使用 the >redirector, 或者 Out-FileCmdlet）。

任务 7

获取最近 20 条安全事件日志的列表，将这些信息转化成 XML 格式。不要在硬盘上创建文件，而是把 XML 在控制台窗口直接显示出来。

注意：该 XML 可以作为一个单独的原生对象显示，而不是以一个原始的 XML 数据。这没问题。那也是 PowerShell 展示 XML 的方式。如果你喜欢，你可以将 XML 对象通过管道传递给 Format-Custom 命令，从而查看 XML 展开为对象层级的形式。

任务 8

获取一个服务列表，并将其导出到以 C:\services.csv 命名的 CSV 文件内。

任务 9

获取一个服务列表，仅保留服务名称、显示名称和状态，然后将这些信息发送到一个 HTML 文件。在 HTML 文件中的服务信息表格之前显示“Installed Services”。

任务 10

为 Get-ChildItem 创建一个新的别名 D。仅将别名导出到一个文件里。关闭这个 Shell，

然后打开一个新的控制台窗口。把别名导入到新的 Shell 中。确认能够通过运行 D 并且获得一个目录列表。

任务 11

显示系统中存在的事件日志列表。

任务 12

运行一个命令来展示 Shell 所在的当前目录。

任务 13

运行一个命令，展示最近你在 Shell 中运行过的命令。从中查找你在任务 11 中所运行的命令。将这两个命令通过管道传输符进行连接，重新运行任务 11 的命令。

换句话说，假如 Get-Something 是一个获取历史命令的命令，5 是任务 11 的命令 ID 号，并且 Do-Something 是运行历史命令的命令，运行如下。

```
Get-Something -id 5 | Do-Something
```

当然，上面的命令并不是正确的命令，你需要找到正确的命令。

提示：你所需寻找的两个命令有相同的名词。

任务 14

运行一个命令，从而在需要时通过覆盖旧日志来修改安全事件日志。

任务 15

通过使用 New-Item Cmdlet 来创建一个名称为 C:\Review 的新目录。这与运行 Mkdir 是不一样的；New-Item 命令需要知道你所想要创建的新项目是什么类型。通过命令读取帮助信息。

任务 16

显示该注册码的内容：

```
HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\User
Shell Folders
```

注意：“User Shell Folders”与真正意义上的目录并不一样。如果你改变该“目录”，你将不能在目录清单中看到任何条目。User Shell Folders 是一个项目，其包含的是项目属性。有一个 Cmdlet 能展示属性项（尽管命令使用的是单数名词而不是复数）。

任务 17

找出（但是请不要运行）命令能做如下事情的：

- 重启电脑；
- 关闭电脑；
- 从一个工作组或者域内移除一个电脑；
- 恢复一个电脑系统，并重建检查点。

任务 18

你认为什么命令可以改变一个注册表值？提示：它是一个和你在任务 16 中发现的命令相同的名词。

实验回顾 2：第 1—14 章

注意：为了完成这些实验，你需要一台运行 PowerShell v3 或更新版本的 PowerShell 的计算机。在打算完成这些实验之前，你应该先完成这本书中的第 1—14 章的实验。

提示：

- Format-Table
- Invoke-Command
- Get-Content(or Type)
- Parenthetical commands
- @{label='column_header';expression={$_.property}}
- Get-WmiObject
- Where-Object
- -eq -ne -like -notlike

任务 1

在一个表格中展示一个正在运行的进程的列表，其中只包含进程的名字和 ID 号。不要让这个表格在两列之间有大的空白区域。

任务 2

运行如下命令：

```
Get-WmiObject -class Win32_UserAccount
```

现在再一次运行相同的命令，但是将内容格式化输出到一个有 Domain 和 UserName

列的表格中。UserName 列应该显示用户的 Name 属性，如下：

```
Domain   UserName
=======  ========
COMPANY  DonJ
```

确保这个第二列标题叫 UserName，而不是 Name。

任务 3

让两台电脑（也可以使用 Localhost 两次）运行如下命令：

```
Get-PSProvider
```

使用远程处理去做，确保输出包含计算机名称。

任务 4

使用 Notepad 创建一个名为 C:\Computers.txt 的文件。在文件中写入如下内容：

```
Localhost
Localhost
```

你应该确保上述两个名称各自独占一行——总共 2 行。保存文件并关闭记事本。然后写一个命令列出正在电脑上运行的服务名称写入到 C:\Computer,txt。

任务 5

查询 Win32_LogicalDisk 的所有实例。仅显示 DriveType 属性中包含 3 且有百分之五十以上的可用磁盘空间的实例。

提示：计算可用空间百分比，公式为 freespace/size * 100。

注意，Get-WmiObjectcannot 的过滤参数中无法包含数学表达式。

任务 6

显示在 root\CIMv2 的命名空间下的所有的 WMI 类列表。

任务 7

在列表中显示所有 StartMode 是 Auto 且 State 属性不是 Running 的 Win32_Service 的实例。

任务 8

找到一个能发送 Email 信息的命令。这个命令的必要参数都是什么？

任务 9

运行一个显示 C:\下目录权限的命令。

任务 10

运行一个可以显示所有 C:\Users 下子文件夹权限的目录，仅包含直接子文件夹，不需要去递归所有的文件和文件夹。你需要把一个命令的结果通过管道传输给另一个命令，即可实现。

任务 11

找到一个可以使用其他凭据而不是当前登录用户的凭据启动记事本的命令。

任务 12

运行一个命令，使 Shell 暂停或者闲置 10 秒。

任务 13

你能找到帮助文件来解释 Shell 的各种运算符吗？

任务 14

写一个信息类消息到应用事件日志。日志类别为 1，原始数据为 100000。

任务 15

运行如下命令：

```
Get-WmiObject -Class Win32_Processor
```

了解该命令的默认输出结果。现在，修改这个命令，使得输出结果在表格里显示。表格内容应该包含每个处理器的核心数、制造商和名称，也包括一个列名为“MaxSpeed”的列，该列表示处理器的最大时钟频率。

任务 16

运行如下命令：

```
Get-WmiObject -Class Win32_Process
```

了解这个命令的默认输出。如果希望的话，可以将该输出结果通过管道传递给 Get-Member 命令。现在，将该命令修改为仅显示在峰值情况下工作集超过 5000 的处理器。

实验回顾 3：第 1—19 章

注意：为了完成这些实验，你需要一台运行 PowerShell v3 或更新版本的 PowerShell 的计算机。在打算完成这些实验之前，你应该先完成这本书中的第 1—19 章的实验。

从回答下列问题开始：

1．你会使用哪一个命令启动一个完全在你本地计算机运行的作业？

2．你会使用哪一个命令启动一个作业的内容被远程计算机处理但由本地计算机调整的作业？

3．${computer name}是一个合法的变量名称吗？

4．你会如何展示由当前 Shell 定义的变量列表？

5．哪一个命令可以被用来提示用户输入？

6．哪一个命令可以被通常用于生成显示在屏幕上的输出结果，但也可以被重新转为多种其他输出格式？

现在完成以下三个任务：

任务 1

创建一个处于运行状态的进程列表，该列表应该仅包含进程名称、ID、VM 和 PM。把这个列表放入一个名称为 C:\Procs.html 的 HTML 文件中。确保 HTML 文件有一个标题为“Current Processes”。在浏览器中显示文件，并把标题显示在浏览器窗口的标题栏中。

任务 2

创建一个包含所有你的电脑上的服务的制表符定界文件，命名为 C:\Services.tdf。"`t"(在双引号之间的反撇号 t)是 PoweShell 为水平制表符使用的转义字符。文件中仅包含服务的名称、显示名称和状态。

任务 3

重复任务 1，将命令修改为在 HTML 文件中 Vm 列和 PM 列显示的值以 MB 为单位，而不是字节。计算兆字节的公式，以一个整体数字的数值显示，公式如下：$_.VM / 1MB –as[int]for the VM property。