Shell作为一种脚本，不需要结构化的编程语言，具有一定的灵活性，特别适合管理系统任务，linux启动时会执行shell中的脚本，对系统进行配置，文章开始引入了清除日志的脚本，并为脚本多次优化提供了思路以及容易忽略的细节。通常在执行文件时，我们通常使用bash filename 或者sh filename 不推荐<filename，在修改文件的读、写、可执行命令时，使用chmod +【option】+filename修改，在第一部分的末尾，给出了管理员自动化公共任务的例子。

文章的第二部分开始便介绍了一些特殊字符。

1 #command通常作为注释，不会被执行，注释可以写在命令行前边，也可以放在末尾，还可以嵌套在管道命令中执行。

2 “；”的作用通常是允许同一行上写多个命令，“；；”通常作为命令的终止符，常用在case语句中。

3 “dot”作为文件名的开头指的是隐藏文件，ls命令查看不到，需要使用ls –a查看，当考虑目录名称时，一个点代表当前工作目录，两个点表示父目录。

4随后介绍了转义字符“\”、文件分隔符“/”、“：”、“取反符号！”、“通配符\*”、“变量替换$”、各种括号、关系运算符、重定向符号、逻辑运算符以及各种符号的组合使用，接下来介绍了各种快捷键的使用。

5紧接着引入了变量和参数的介绍，“$变量名”通常指变量替换的作用，并通过例子详细阐述了变量的赋值和替换。bash变量类型与许多其他编程语言不同，它并没有用“类型”分隔其变量。决定因素是变量的值是否只包含数字。Shell中的特殊变量，局域变量和环境变量，前者只在代码块中起作用，后者会影响用户行为，位置变量，“$数字”通常是指传递给脚本的第几个参数。

6“”在shell中表示引用的含义，在引用一个字符串的时候，通常会保留它的字面意思，在变量引用的时候，双引号中的“$”依然有变量替换的作用，单引号（'）类似于双引号，但不允许引用变量，因为$的特殊含义被关闭。在单引号中，每个特殊字符除外“被逐字解释”。

7跳脱建后加单个字符表示转译的含义，常见的如\n,\r,\t,\v,\b分别表示换行、返回、水平、垂直制表以及退格等等。后面详细举例说明了这些用法，进一步给出了一些深入用法，如“\"”给出引号字面的意思，“\$”和“\\”分别给出$和\的字面意思。

8退出命令终止一个脚本，它还可以返回一个值，该值可用于脚本的父进程。每个命令返回一个退出状态（有时称为返回状态或退出代码）。一个成功的命令返回一个0，而一个失败的返回一个非零值，通常可以解释为一个错误代码。同样，脚本中的函数和脚本本身返回一个退出状态。函数或脚本中执行的最后一条命令决定退出状态，常用exit来表示退出状态。

9测试命令，if/then测试命令判断测试的结果是否为0，如果为0则继续执行，在后来的版本中加入了[…]测试，if条件可以进一步扩展为elif，[..]进一步可以扩展为[[]],主要增加了连接||，&&以及重定向功能。（（））构造展开和计算算术表达式。如果表达式计算为零，则返回1或“false”的退出状态。非零表达式返回0或“true”的退出状态。

10文件测试命令，常用的如-a、-f、-s、-d分别判断文件是否存在、是否属于常规文件、是否非零长度、是否目录文件。

11其他的比较运算符，-eq表等价，-ne表示不等价，-gt表示左侧输入大于右侧输入，-lt表示左侧小鱼右侧，-z表示长度为0，-n表示长度非0，还有一些常见的类似数学上的运算符。

12一些常见的操作符如加、减、乘、除、取幂、取模，以及位操作，左位移、右位移、位取反、位异或，布尔操作，非、AND 、OR等等。

13数值常量，shell脚本将数字解释为十进制（基数为10），除非该数字具有特殊前缀或符号。0之前的数字是八进制（基数为8）。一批以0x开头是十六进制（基地16）。

14双圆括号结构，类似于let命令，重新定义变量值。

第三部分更深入的讨论了shell命令。

1内部变量，常见的$BASH，通往BASH 本身的路径，$BASH\_ENV指向脚本启动的环境变量，$BASHPID表示当前进程的ID，$BASH\_VERSION表示BASH 的版本号，$CDPATH表示冒号分隔的可供cd命令使用的搜索路径列表，其功能类似于二进制文件的PATH变量，$FUNCNAME表当前函数名，$HOME表示用户主目录等等。

2declar变量的声明，-r表示只读，-i表示声明为整数，-a表示声明为数组，-f会列出脚本中定义的函数。

3$RANDOM用来生成0-32767的随机数。

4字符串操作，计算字符串长度，常用的有${#string}、expr length $string、expr "$string" : '.\*'，查询字符串索引命令，expr index $string $substring，，随后阐述了子串扩展、子串移除以及子串替换命令，并引入了awk命令进行字符串操作。

5循环通过限制条件将代码限定在一定区域执行，，常用的有for和while循环以及嵌套循环，嵌套循环是循环内的一个循环，外部循环的内部循环。这种方法的工作原理是，外部循环的第一个通道触发内部循环，该循环执行到完成。然后，外循环的第二次循环再次触发内部循环。这样重复直到外层循环结束。当然，内循环和外循环都会中断这个过程

6循环控制命令，break和continue，中断和继续循环控制命令与其他编程语言中的对应完全一致。中断命令终止循环（中断），而继续导致跳转到循环的下一次迭代，跳过特定循环周期中的所有剩余命令。

7分支测试也是循环的一种，它允许分支到多个代码块中的一个，这取决于条件测试。它用作多个if / if /其他语句的一种简写，是创建菜单的适当工具。

8算数扩展，算术扩展为在脚本中执行（整数）算术运算提供了强大的工具

第四部分主要介绍了常用的命令

1父子进程，当一个命令或程序本身启动（或生成）一个新的子进程来执行任务，这就是所谓的分叉。这个新过程就是孩子，而把它分离出来的过程就是父进程。当子进程正在执行其工作时，父进程仍在执行。

2 echo输出描述内容或者变量，echo可以和管道命令一起使用，与之比较类似的是printf，格式化打印，是echo的增强版。这是C语言的库函数printf()有限的变种，它的语法是不同的。

printf格式字符串…参数...。

3 read 以交互方式读取键盘变量，没有关联变量的情况下将其输入赋值给变量，通常输入\会抑制读过程，-r选项使输入的文本被逐字解释，-n选项不会检测换行键，-t选项读取允许定时输入，-u选项获取目标文件的文件描述符

4文件系统常用的指令，cd切换用户目录，pwd查看当前目录，pushd：切换到作为参数的目录，并把原目录和当前目录压入到一个虚拟的堆栈中，如果不指定参数，则会回到前一个目录，并把堆栈中最近的两个目录作交换popd： 弹出堆栈中最近的目录，dirs: 列出当前堆栈中保存的目录列表。

5变量操作，let 命令是 BASH 中用于计算的工具，用于执行一个或多个表达式，变量计算中不需要加上 $ 来表示变量。如果表达式中包含了空格或其他特殊字符，则必须引起来。。eval命令将会首先扫描命令行进行所有的替换，憨厚再执行命令。该命令使用于那些一次扫描无法实现其功能的变量。该命令对变量进行两次扫描。这些需要进行两次扫描的变量有时候被称为复杂变量。set命令作用主要是显示系统中已经存在的shell变量，以及设置shell变量的新变量值。使用set更改shell特性时，符号"+"和"-"的作用分别是打开和关闭指定的模式。declare命令用于声明和显示已存在的shell变量。当不提供变量名参数时显示所有shell变量。declare命令若不带任何参数选项，则会显示所有shell变量及其值。declare的功能与typeset命令的功能是相同的。readonly命令用于定义只读shell变量和shell函数。

6脚本操作，source Filename，在当前bash环境下读取并执行FileName中的命令，exit命令同于退出shell，并返回给定值。在shell脚本中可以终止当前脚本执行。执行exit可使shell以指定的状态值退出。若不设置状态值参数，则shell以预设值退出。状态值0代表执行成功，其他值代表执行失败。Exec，虽然exec和source都是在父进程中直接执行，但exec这个与source有很大的区别，source是执行shell脚本，而且执行后会返回以前的shell。而exec的执行不会返回以前的shell了，而是直接把以前登陆shell作为一个程序看待，在其上经行复制。shopt命令用于显示和设置shell中的行为选项，通过这些选项以增强shell易用性。shopt命令若不带任何参数选项，则可以显示所有可以设置的shell操作选项。

7命令，true判断返回是否为真，false判断返回是否为假，type命令用来显示指定命令的类型，判断给出的指令是内部指令还是外部指令，hash命令的作用是在环境变量PATH中搜索命令name的完整路径并记住它，这样以后再次执行相同的命令时，就不必搜索其完整路径了，而且shell每次执行环境变量PATH中的一个命令时，hash都会记住它，bind命令用于显示和设置命令行的键盘序列绑定功能。通过这一命令，可以提高命令行中操作效率。您可以利用bind命令了解有哪些按键组合与其功能，也可以自行指定要用哪些按键组合，

8基本命令，ls查看当前目录下的文件，cat，tac，连接文件并打印到标准输出设备上，cat经常用来显示文件的内容，反序输出文件的内容，文件的最后一行显示在第一行它可以对调试日志文件提供了很大的帮助，扭转日志内容的时间顺序，rev接受一个文件或stdin作为输入，并逆序打印出每一行内容。，cp文件的复制，mv文件的移动或者重命名，rm文件的删除，rmdir删除目录，mkdir创建目录，chmod改变文件权限，Linux chattr命令用于改变文件属性。man命令，通过man指令可以查看Linux中的指令帮助、配置文件帮助和编程帮助等信息。man是在程序安装的时候安装的帮助文档，可以在系统的目录下找到，如果软件有配套的页面，就可以使用man来查找。比如通过指令：ls /user/share/man我们可以看到很多man帮助文档的存储，我们从中看到有很多歌man#文件，linux的一些man指令文档就存在于这些文件内。info指令是man 指令的详细内容。但man 使用起来要方便的多。一个man 只有一页，而info总是将它们的内容组织成多个区段（节点），每个区段也可能包含子区段（子节点）。info工具可显示更完整的最新的GNU工具信息。通常man中若包含的信息概要在info中也有时，会有提示：“请参考info页更详细内容”。

9复杂命令，find命令原理，从指定的起始目录开始，递归地搜索其各个子目录，查找满足寻找条件的文件，并可以对其进行相关的操作。xargs命令是给其他命令传递参数的一个过滤器，也是组合多个命令的一个工具。它擅长将标准输入数据转换成命令行参数，xargs能够处理管道或者stdin并将其转换成特定命令的命令参数，expr命令可以实现数值运算、数值或字符串比较、字符串匹配、字符串提取、字符串长度计算等功能。它还具有几个特殊功能，判断变量或参数是否为整数、是否为空、是否为0等。

10时间日期命令，date获取时间日期，zdump时区转换，time命令用于统计给定命令所花费的总时间，touch用来修改文件时间戳，或者新建一个不存在的文件，at，执行定时计划任务的命令。at执行的是一次性任务，任务执行完以后便失效。call命令用来从一个批处理脚本中调用另一个批处理脚本。Sleep time表示睡眠时长。clock命令用于调整 RTC 时间。

11文本处理命令，sort串联排序所有指定文件并将结果写到标准输出。uniq命令用于报告或忽略文件中的重复行，一般与sort命令结合使用。expand命令用于将文件的制表符（TAB）转换为空白字符（space），将结果显示到标准输出设备。cut命令是一个选取命令，其功能是将文件中的每一行”字节” ”字符” ”字段” 进行剪切，选取我们需要的，并将这些选取好的数据输出至标准输出，paste命令用于合并文件的列。paste指令会把每个文件以列对列的方式，一列列地加以合并。Join“将两个文件里指定栏位同样的行连接起来”，即依照两个文件里共有的某一列，将相应的行拼接成一行，head命令用于显示文件的开头的内容。在默认情况下，head命令显示文件的头10行内容。Tail从指定点开始将文件写到标准输出。使用tail命令的-f选项可以方便的查阅正在改变的日志文件，tail -f filename会把filename里最尾部的内容显示在屏幕上，并且不但刷新，使你看到最新的文件内容。Grep 显示匹配一个或多个模式的文本行，时常会作为管道后的第一步，以便对匹配上的数据做进一步处理。Look指令用于英文单字的查询。您仅需给予它欲查询的字首字符串，它会显示所有开头字符串符合该条件的单字。Sed、awk用于文本编辑，wc命令的功能为统计指定文件中的字节数、字数、行数，并将统计结果显示输出。命令功能是统计指定文件中的字节数、字数、行数，并将统计结果显示输出，tr主要用于压缩重复字符，删除文件中的控制字符以及进行字符转换操作。Col命令是一个标准输入文本过滤器，它从标注输入设备读取文本内容，并把内容显示到标注输出设备。在许多UNIX说明文件里，都有RLF控制字符。当我们运用shell特殊字符>和>>，把说明文件的内容输出成纯文本文件时，控制字符会变成乱码，col命令则能有效滤除这些控制字符。nl命令读取 file 参数（缺省情况下标准输入），计算输入中的行号，将计算过的行号写入标准输出。在输出中，nl命令根据您在命令行中指定的标志来计算左边的行。输入文本必须写在逻辑页中。每个逻辑页有头、主体和页脚节（可以有空节）。除非使用-p选项，nl 命令在每个逻辑页开始的地方重新设置行号。可以单独为头、主体和页脚节设置行计算标志（例如，头和页脚行可以被计算然而文本行不能）。其默认的结果与cat -n有点不太一样， nl 可以将行号做比较多的显示设计，包括位数与是否自动补齐0等等的功能。Pr，打印输出，将较大的文件分割成多个页面，并为每个页面添加标题。Iconv用于文件编码转换。

12文件和归档命令，tar用来压缩和解压文件。tar本身不具有压缩功能。他是调用压缩功能实现的，ar - 创建静态库.a文件，rpm命令是RPM软件包的管理工具。rpm原本是Red Hat Linux发行版专门用来管理Linux各项套件的程序，由于它遵循GPL规则且功能强大方便，因而广受欢迎。逐渐受到其他发行版的采用。RPM套件管理方式的出现，让Linux易于安装，升级，间接提升了Linux的适用度。cpio 可将文件或目录从文件库获取出来或将散列文件拷贝到文件库。bzip2命令用于创建和管理（包括解压缩）“.bz2”格式的压缩包。compress命令使用“Lempress-Ziv”编码压缩数据文件。compress是个历史悠久的压缩程序，文件经它压缩后，其名称后面会多出".Z"的扩展名。当要解压缩时，可执行uncompress指令。事实上uncompress是指向compress的符号连接，因此不论是压缩或解压缩，都可通过compress指令单独完成。whereis命令只能用于搜索程序名，而且只搜索二进制文件（参数-b）、man说明文件（参数-m）和源代码文件（参数-s）。如果省略参数，则返回所有信息。which指令会在环境变量$PATH设置的目录里查找符合条件的文件。whatis命令是用于查询一个命令执行什么功能，并将查询结果打印到终端上。Getfacl获取目录或文件访问控制列表。Readlink获取当前脚本的绝对路径

13比较命令 diff命令在最简单的情况下，比较给定的两个文件的不同。如果使用“-”代替“文件”参数，则要比较的内容将来自标准输入。diff命令是以逐行的方式，比较文本文件的异同处。如果该命令指定进行目录的比较，则将会比较该目录中具有相同文件名的文件，而不会对其子目录文件进行任何比较操作，cmp命令用于比较两个文件是否有差异。当相互比较的两个文件完全一样时，则该指令不会显示任何信息。若发现有所差异，预设会标示出第一个不同之处的字符和列数编号。若不指定任何文件名称或是所给予的文件名为"-"，则cmp指令会从标准输入设备读取数据。comm命令可以用于两个文件之间的比较，它有一些选项可以用来调整输出，以便执行交集、求差、以及差集操作。

14终端控制命令 tput 命令将通过 terminfo 数据库对您的终端会话进行初始化和操作。通过使用 tput，您可以更改几项终端功能，如移动或更改光标、更改文本属性，以及清除终端屏幕的特定区域。reset命令其实和 tset 是一同个命令，它的用途是设定终端机的状态。一般而言，这个命令会自动的从环境变数、命令列或是其它的组态档决定目前终端机的型态。如果指定型态是 '?' 的话，这个程序会要求使用者输入终端机的型别。clear命令用于清除屏幕。resize设置终端机视窗的大小。

15数学命令 factor把整数拆分为质因子。bc命令是一种支持任意精度的交互执行的计算器语言。bash内置了对整数四则运算的支持，但是并不支持浮点运算，而bc命令可以很方便的进行浮点运算，当然整数运算也不再话下，dc相比与bc要复杂，但是简单操作还是比较简单。简单的说dc是一直压栈操作，和bc一样，它也可以交互使用，或者与echo一起配合使用。

16系统管理命令 user显示所有登陆的用户，groups显示当前用户以及所属的分组，chown命令改变某个文件或目录的所有者和所属的组，该命令可以向某个用户授权，使该用户变成指定文件的所有者或者改变文件所属的组。用户可以是用户或者是用户D，用户组可以是组名或组id。文件名可以使由空格分开的文件列表，在文件名中可以包含通配符，useradd和userdel分别表示增加和删除用户，usermod命令用于修改用户的基本信息。usermod命令不允许你改变正在线上的使用者帐号名称。当usermod命令用来改变user id，必须确认这名user没在电脑上执行任何程序。你需手动更改使用者的crontab档。也需手动更改使用者的at工作档。采用NIS server须在server上更动相关的NIS设定。id命令用于显示用户的ID，以及所属群组的ID，who显示所用登录系统的用户，su命令用于变更为其他使用者的身份，除 root 外，需要键入该使用者的密码，sudo命令是以root的身份执行command命令，但是环境变量还是当前用户的，执行目录也仍然是当前目录，passwd修改或者设置用户密码，last命令用于显示系统开机以来获是从每月初登入者的讯息，

17控制命令 ps命令最常用的还是用于监控后台进程的工作情况，因为后台进程是不和屏幕键盘这些标准输入/输出设备进行通信的，所以如果需要检测其情况，便可以使用ps命令了。pkill命令可以按照进程名杀死进程。pkill和killall应用方法差不多，也是直接杀死运行中的程序；如果您想杀掉单个进程，请用kill来杀掉。top命令是Linux下常用的性能分析工具，能够实时显示系统中各个进程的资源占用状况，类似于Windows的任务管理器，nohup 命令运行由 Command 参数和任何相关的 Arg 参数指定的命令，忽略所有挂断（SIGHUP）信号。在注销后使用 nohup 命令运行后台中的程序。要运行后台中的 nohup 命令，添加 & （ 表示”and”的符号）到命令的尾部。pidof用于找出正在运行的程序的进程PID，程序可以是一个二进制执行程序，也可以是一个shell脚本。如果是找出java程序的进程PID，pidof就无能为力了，可以使用ps -ef|grep java或jps -l来查看java进程的信息。通常找出进程PID的目的是确认程序是否在运行、或者为了把它杀掉、或者发送一个信号给它。

第五部分更加深入的对脚本进行分析，通过大量的举例分别阐述了I/O重定向操作，父子进程，复杂函数，局域变量，别名的使用，列表结构（可以有效替换掉if/else嵌套结构），数组用法，间接引用，网络编程，Bug调试，set –[option]操作，规范化编程，最后还介绍了交互式和非交互式shell编程，shell包装，方案的测试和比较，自身脚本的递归调用，脚本的润色处理，脚本的优化，脚本处理的小技巧，安全问题，移植性问题以及windows下的shell脚本编程。