

利用While解决文件等待问题Linux认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/619/2021_2022__E5_88_A9_E7_94_A8Whil_c103_619635.htm

前几天有个朋友向笔者求助，他说他在Unix系统管理的时候遇到了一个问题。他在Unix系统中部署了一个数据库系统。现在他希望能够对这个数据库实现异地备份。他的解决思路是，先利用数据库自身的功能进行本地备份(数据库没有自动进行异地备份的功能)，然后再将这个备份文件复制到异地去。如果备份是确定的，即按照计划来进行备份，如每天晚上12点备份，这是好解决的。只需要利用Unix系统自带的CRON任务计划功能即可以解决。但是现在的问题是，有时候这个备份具有不确定性。如数据库管理员有时候也需要进行临时的备份，为此他希望对于这些临时备份也能够自动实现异地备份。所以，此时利用cron任务计划功能就无能为力了。

一、解决思路。针对这些临时性的备份任务，如果少的话，那么可以对其进行手工异地备份。然后利用cron任务计划执行有计划的异地备份工作。如果临时性备份比较频繁，而用户又确实对异地备份的功能比较急切的话，那么就只能够采用其他折中的处理方式。笔者下面的这个解决思路，希望能够帮助解决这个问题。这个思路比较简单，即让操作系统隔一段固定的时间(如10分钟)就去检查一下是否有生成了备份文件。如果生成了的话，则系统就将这个备份文件移动(不是复制)到异地进行备份。不过笔者这里要介绍的是另外一种解决方式，即利用while循环来解决这个文件等待的问题。其实，这个文件等待也不是这个系统工程师遇到的问题，而是具有一定的普遍性。如一

个应用程序需要从一个文件中读取数据，而这个文件又是另一个应用程序或者同一个应用程序的不同进程所产生的。为此，必须在另外一个程序或者进程创建完这个文件之后，原先的应用程序才能够得到顺利的运行。所以，如果从程序设计的角度，就是需要使用某个脚本，去侦测这个特定的文件是否已经成功创建。如果创建的话，执行某个命令，让其自动调用某个程序。如在实现无人值守安装时，需要一个参数文件。此时系统工程师可以编写一个脚本，先判断这个参数文件是否存在。如果存在的话，则马上进行安装。如果不存在的话，则进行等待。直到这个配置文件正常建立，然后启动安装进程(此时系统工程师不需要重新启动脚本程序)。

二、具体实现。

以上介绍的这个思路，主要是利用while循环来实现的。利用while循环，操作系统会重复的执行一组命令，直到控制命令返回一个假false的状态值。如果要实现上面这个无人值守的安装进程，可以通过如下的脚本来实现。

```
While  
[! r profile.txt] #循环条件语句 do #循环操作语句 sleep 30 done  
./setup.sh
```

在编写这个脚本的时候，各个部分的含义是什么，又有哪些注意事项呢?具体来说，系统工程师需要掌握下面的内容。这些是写好一个while循环结构最基本的东西。首先，需要直到常见循环条件语句的书写。一般来说，任何Unix命令或者test内容都可以作为循环条件。也就是说，只要掌握了Unix常用的命令，既可以写好循环语句。话虽然这么说，但是有时候写这个循环语句的时候，系统工程师往往需要有一定的创意，或者说需要有经验的积累。确实，最终的循环语句都是基本的操作系统命令所组成的。但是有时候往往需要对他们进行灵活的组合。如笔者上面用的这个循环条件语

句，相对来说比较简单，但是并不一定每个系统工程师都可以像得到。笔者这个! r profile.txt循环条件语句的含义，就是不能够读某个文件。一般情况下建立的文件当前用户都是可读的。如果系统判断不可读的话，那么百分之百可以确定是这个文件不存在(只要权限没有调整过)。换句话说，有时候系统工程师往往需要换一种角度去考虑问题。如按照这个案例，需要将“文件不存在”转换为“文件不可读”。此时这个循环条件语句就容易写了。可惜的是，不少系统工程师缺少的就是这种转换的能力。这部进行需要踏实的基础，而且还需要一定的创新能力以及相当的工作经验的积累。为此笔者建议，要用好Unix操作系统或者Linux操作系统，那么在平时的工作中就要有意识的锻炼自己这种能力。其次需要了解操作语句的编写技巧。有时候为了实现某些特定的需求，往往在while循环结构内部的操作语句部分，不会写一些实际的操作语句。而是将操作语句写在循环的外部。如上面这个案例中，笔者在内部的操作语句部分，使用了语句sleep 30。这是什么意思呢?如果结合整个While循环结构，他的意思就是，如果条件满足的话(profile配置文件不可读，即不存在)，则每个30秒执行以下这个循环。如果条件不满足的话(profile配置文件可读，即存在)，则跳出这个循环，然后执行后续的语句。也就是说，此时笔者将真正要执行的操作(启动某个安装进程)，放在了循环的外部。而在循环内部使用sleep 30，用来控制循环执行的时间。当然，在这个循环内部也可以写一些实际性的操作。如在屏幕上或者安装日志中显示一些“找不到配制文件”等类似的错误信息。往往这些类似友好的提示信息，可以提高程序的友好性。类似的处理机制还有很多。

系统工程师需要在日常的工作中注意总结。另外，如果多看看其他工程师编写的脚本程序，往往能够从中获取不少有价值性的内容。模仿往往是在最短时间内增强自己业务能力的捷径。最后就是需要注意书写格式与规范方面的内容。在编写循环结构时，无论是while循环还是for循环，如果能够有意识的采用TAB键或者空格键，可以明显的提高代码的可读性。如循环内部结构的执行语句最好能够跟do或者done等关键字分行写，并且采用缩进的格式。虽然即时不遵守这些规则，脚本程序也没有实质性的影响。但是，对后续维护阅读的时候，会增加不少的困难。特别是当循环内部语句比较多的时候。所以对于这些代码编写规范，系统工程师最好还是需要遵守。毕竟对于一些复杂的功能，需要团队合作才能够实现。如果大家都能够遵守同一的书写规范，那么无疑可以提高团队的协作型。还有就是需要注意语法方面的内容，不do后面没有分号结尾符等等。这些语法性的错误系统在执行时会告知，所以问题不大。只是会增加返工的几率与工作量而已。

三、利用while实现无限循环。

如上面那个数据库异地备份的案例，其实若能够通过无限循环来实现可能会更加的合适。如让操作系统每隔30分钟来执行一次循环。循环内不能的操作就是先利用if语句来判断备份文件是否在30分钟内更新过。如果有更新的话，就将其复制或者移动到远程的服务器上。不管有没有更新，这个循环的话，每30分钟就需要执行一次。此时就需要用到无限循环。利用while循环条件的含义，这个无限循环很容易实现，只需要循环条件语句设置为true即可。即while true . do 操作语句 .done &符号即可，就可以告诉操作系统将这个无限循环放入到后台运行。不过在编写

无限循环的时候，需要注意死循环的发生。如循环内部的某个操作比较耗时。在这个操作还没有完成之前，新的循环又开始了。这久而久之，必将耗尽系统资源。为此，在程序内部最好还是需要设置一些条件，如CPU等资源的使用程度，来强制结束无限循环。笔者认为这是比较安全的处理方式，可以有效的避免无限循环影响操作系统性能。更多优质资料尽在百考试题论坛 百考试题在线题库 linux认证更多详细资料 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com