

论脚本在本司应用

-----分享自吴林军 2018 年 8 月 24 日 星期五

为什么，我要在普遍搞自动化，智能化平台的时期，还弄上古时代的脚本事情叽歪呢。因为还是业务要求的。

我是这样认为的，公司先有了业务，才有的运维，才有了 DBA。

只有活挑人，哪有人挑活的，除非你想跳槽。

简单介绍下我们的业务

我们业务主要集中在政府部门。而且在某个领域可以算是垄断，项目分布在一定区域，比如说一个城市，或者一个省份，现在的多个项目在不同行政系统内部。

业务系统多，但是都没有大的流量。你们讨论的并发，我这都没有。

各个机构的信息化建设差异大，业务部署多种多样。

维护人员水平参差不齐。实施阶段有现场运维，项目进入维护阶段就有运维部门统一接收了！

1890 年左右瑞士人发明了一种可以在小刀上集合了多种工具，方便不同的使用场景方便快捷的使用，解决痛点，后来在欧洲的军人喜好这种多功能小刀，就有了瑞士军刀。切、割、钻、锯、削、简单、耐用便携。

其实在运维岗中也有我们的瑞士军刀

那就是脚本。脚本是用文字命令编辑可以通过解释执行的文本文件。任何用脚本语言编辑的可执行的文本都是脚本。脚本语言丰富多样。特定的场景中，脚本一出，问题解决，有时候就是查个错误日志，或者就是简单的 SQL 语句，就让你判断问题的思路，豁然开朗。做运维或者 DBA，哪个手里不都是一大把的脚本合集。也就有了运维的瑞士军刀，脚本。因为脚本的好用，也就成就了一些人见任务要脚本的脚本小子。

我的经历大致上是以下几个阶段

第一阶段，纯手工制作--自己吭哧吭哧的写，碰上卡壳是才有的事情。这个阶段熟练写查询，碰上复杂的情况。自己一个人面壁了。总觉得的自己看过的脚本都会了，自己写的时候，脑袋一片空白，几次喝过大酒以后，忘记了如何开始，这个时间持续还很长。有次项目库 HANG 了，一个脚本跑起后，把自己晾在那了。

进入第二阶段，西天取经--到处求医问药，不问如何实现，只求有结果。收集各个应用脚本，不求质量，拿到后，简单跑一下，只要能跑通，就留下。集聚了大量的脚本文件。发来发现大多数都用不上的，或者功能重复的。真正需要解决问题的时候才进入了第三阶段。

第三阶段，敢爱敢恨--吃过囫圇吞枣的亏，拿来就就用的，小问题搞出大事件。总是有新手干了不该干的事情，才知道悔改。复制剪切是此阶段的主要工作。搞出一些拼接的东西。

第四阶段才是正常阶段 跳进黄河--分门别类汇总，对比不同情况下的脚本因素，完整交互选项，授权是给操作员的。走在第四个阶段后，我仍不敢马虎大意，如行走在江湖中，低调谨慎。我觉得脚本是一个运维一个 DBA 的成色的体现。脚本里有这个人的办事思维，办事流程，办事角度等等，多年的脚本风格，也就成就了他以后开发的风格。我也深有感触。有些人永远都写不好脚本。对于一个好运维或者是 Dba 这样的脚本才是安全可用的：

- 1.指定其运行解释器，更换到其他的运行环境保障起最大的可用性。

- 2.关于注释的，超过五行的脚本，即使是自己写的，一个月后估计自己都不知道什么逻辑了。加上作者，时间，用法，特殊说明等。

- 3.注意编排方式，多换行，严格控制单行字符数目。不美观的结果是人类无法理解。

- 4.任何一个转交给他人执行的脚本，都要经过测试。
- 5.清除掉写到脚本里面的用户信息，我见过开发大胆的把库的字符连接串带上。
- 6.学好脚本的途径是，由比较优劣 到去其糟粕，融会贯通三大部分。
- 7.对于不同人员进行不同的权限分配。或者执行要求。

从拿到别人的脚本开始，查看比较重要的逻辑实现方式，语句风格，编排版等。慢慢的积累了大量的可用脚本，自信性也不断增加了，然而收集大量的脚本不代表自身的拥有。聪明的办法是从中取优，一个多页脚本可能实现的特别的功能也就是那么三五句。记得这其中重要部分就完全把握重点部分。对于不同的应用脚本，要进行分门类存放，积累了一部分后可以放到云服务器上，让瑞士军刀发挥高速快捷的作业。整理好优秀的实现逻辑，完成新的需求组合，去掉原有的不相关的成份，在优秀的基础建立起来的再差也不会差到哪里去！遇到特别优秀的脚本，优化后的版本，首先尝试这默写课文似的的，自己写一边，然后对照查漏补缺。自我实现一遍。进阶的办法是用其他的语言再写一遍。建荣同学分享出来 `dba_lite` 时，当时里面脚本还是很完整的，可能默写原因删除了一部分。我用 `python` 重新写，现在 `go` 也不错，有兴趣的同学可以自己来一遍。写好脚本，就如专注于基础原理一样，耐得住寂寞，放的下心态，勤劳的用手。你就是那个解决问题的人，信的过人。

不论二线、三线的城市，解决业务问题，维护业务稳定运行，你就是那个被需要的人！哪个行业不是一样呢，坚持，恒定，慎言。你终将成为你想要成为的那个人！