 AWK是Unix下的一款功能强大的文本格式化和抽取工具。利用这个工具，可以对复杂的文本文件进行整理，提取其中的全部或者部分数据，按照需要的格式予以显示。需要说明的是，AWK的强大功能只针对纯文本文件。对于带有很多不可显示字符的二进制数据文件，单凭AWK就无能为力了。这时我们需要其他工具的帮助。 在Unix下，还有一个工具叫做OD，其全称是“ display files in octal format ”，也就是说它能将各种文件以8进制的方式显示出来。如果设置不同的选项，它还能将文件以16进制方式显示。 此外为了方便处理，我们还需要用到另外一个工具,sed。这也是一个Unix下的传统文本处理工具。在这里我们主要用到它的文本替换功能。 通过组合以上三种工具，我们就可以完成我们用AWK处理二进制数据文件的任务了。 笔者手中有一个数据文件，FXT,其数据结构如表 1所示。 Table 1 起始位长度说明08账号87金额153操作员号 如果用普通的文本编辑器打开这个数据文件，看到是一串数字和一堆难以理解的字符。根本就无法分辨金额是多少。为了方便读者理解这个文件，我们用od来查看这个文件 (见List1)。 List 1 # od -An -v -tx1 FXT 32 35 38 35 36 30 30 39 00 00 05 00 00 00 0c 31 30 31 0a 32 35 38 30 30 32 33 34 90 20 20 80 20 20 0d 31 30 32 0a … 稍微解释一下Od的命令参数意义。-An表示不在每行左边显示偏移量；-v 表示每行都要显示；-tx1表示输出时以16进制方式输出，一次一个字节。 根据数据结构定义，我们可以看出前面8

个字节（32 35 38 35 36 30 30 39）代表账号，而且账号部分是由可显示的ASCII码组成的，翻译后的结果就是25856009；接下来7个字节（00 00 05 00 00 00 0c）代表金额。最后的c代表credit，就是贷方的意思。它所代表的实际金额是500,000.00。紧接着的3个字节代表操作员号，也是由可显示的 ASCII码组成的。0a是换行符的ASCII代码，表示一条记录结束。 可以看出，正是由于金额部分是由不可显示的ASCII码组成，导致了无法用常规方法来提取数据文件中的数据。 那么应该如何利用以上的工具来处理这类数据文件，并且按照可以理解的方式来生成新的纯文本的数据文件呢？ OD 已经将整个数据文件清楚地显示出来了。它输出的格式不符合我们的要求。比如说，本来在一行的记录，先在给分成了几行；本来连在一起的字符，现在中间出现了空格。这样AWK就不好处理了。所以，为了使AWK能够方便的处理，我们在正式提取数据之前，必须生成AWK可以处理的中间文件。 从List1可以看出，OD在显示文件时，每一行前都有一个制表符，而且原来记录之间的换行符也变成了对应的ASCII码0a。那么我们的任务就要去掉制表符，而且要恢复正确的记录间的换行符。这一步可以通过以下命令完成。# od -v -An -tx1 fxt |sed s/ //|sed s/0a/,/|awk -f org.awk 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