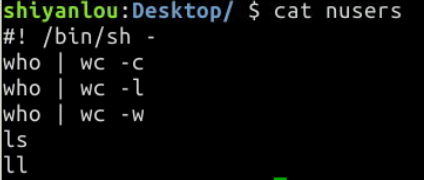
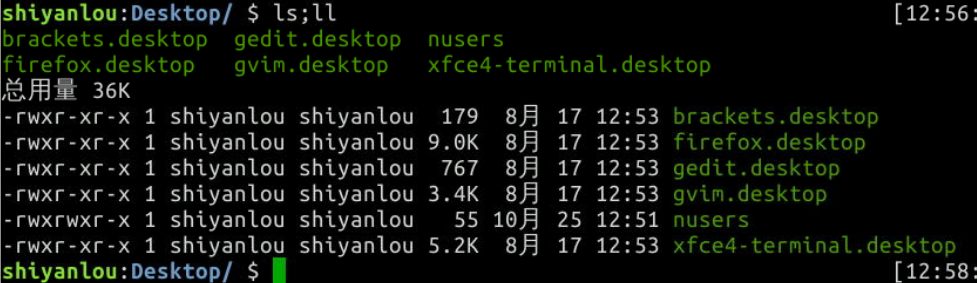


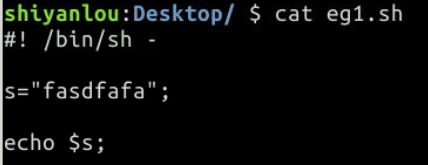
脚本里可以写多条命令



分号用来分隔同一行的多条命令：



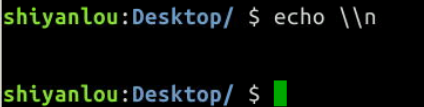
定义变量时不用$,调用时要使用$:



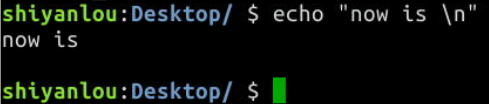
原始的echo就是输出：



Echo 会将特殊字符转义：





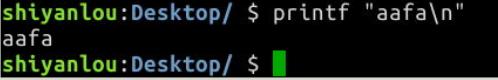


Printf

Printf 的语法：



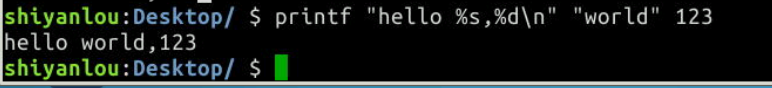
1. 输出简单的字符串，必须显示的指定换行符



1. 输出格式化的参数

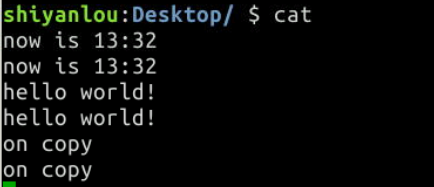
%s表示字符串，

%d表示数字

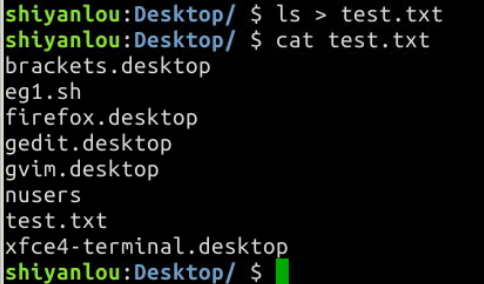


Cat:

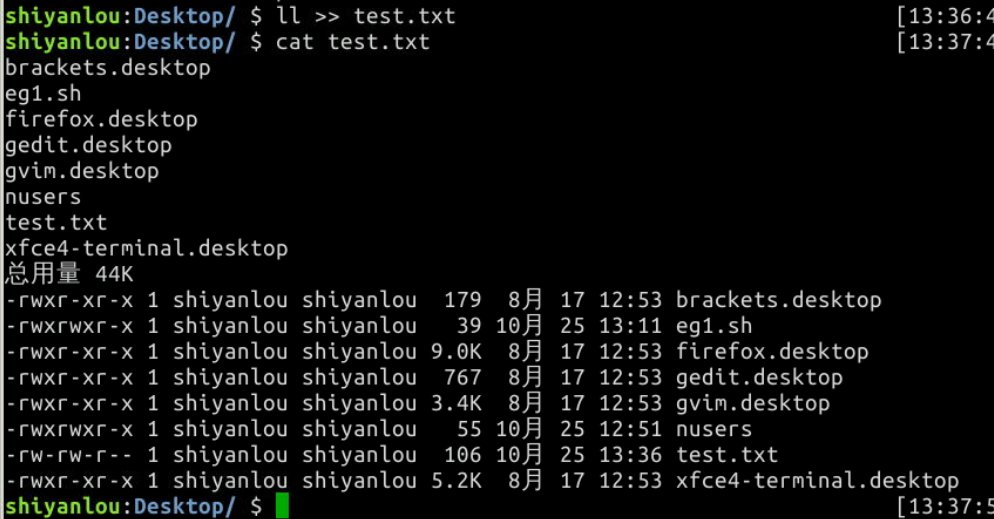
不指定参数，写一个输出一个：



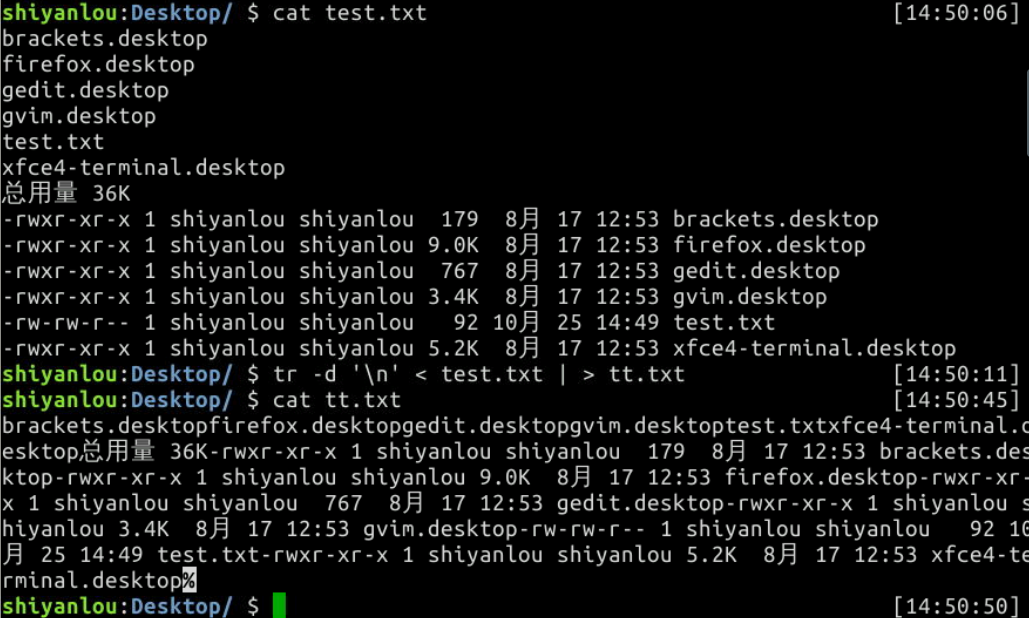
输出重定向：>



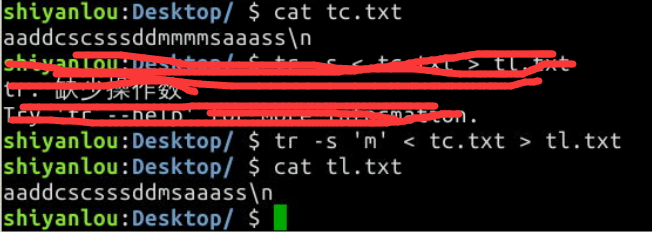
>>附加到文件：



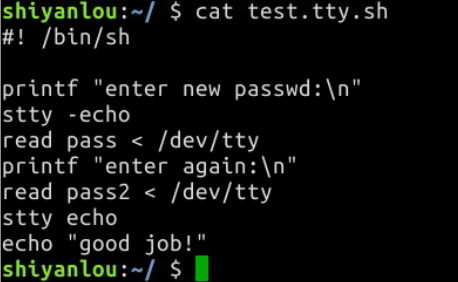
Tr命令：



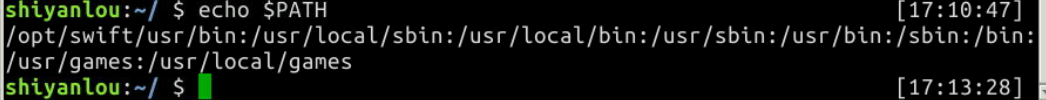
删除指定的叠字符：



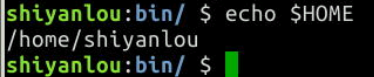
输入命令：



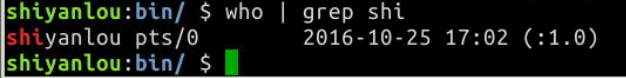
查看环境变量：其中本地执行路径/usr/local/bin可以用来执行用户的脚本，如安装软件等的脚本



打印家目录：



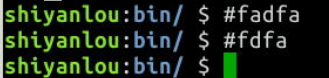
查找：grep



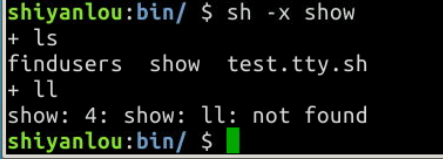
脚本参数查询：

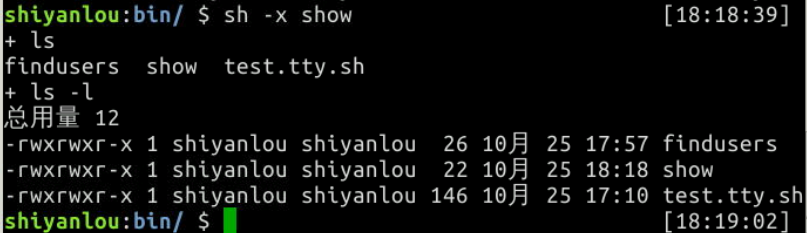


Linux和shell中的注释：#

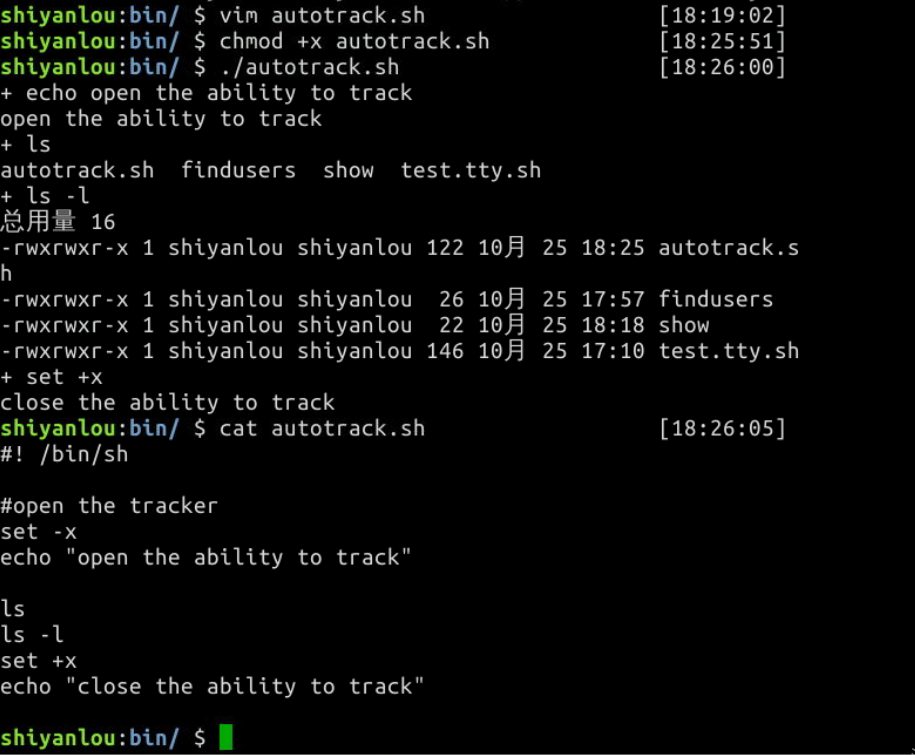


执行跟踪：



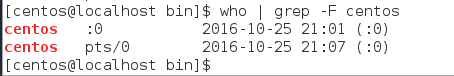


在脚本中打开和关闭自动跟踪功能：

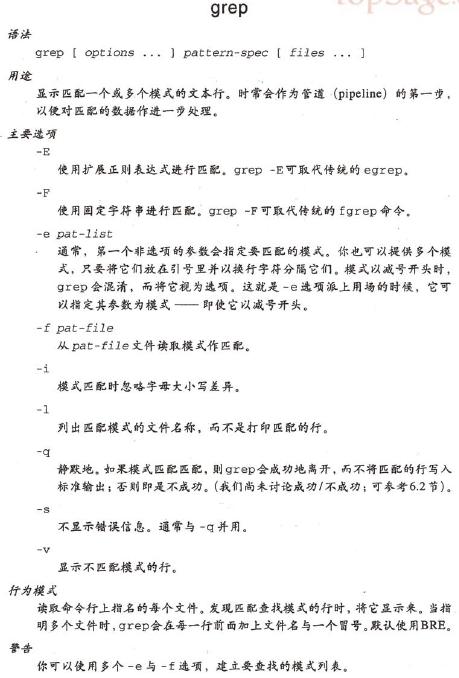


第三章

固定字符串查询：-F指定为固定字符串匹配，省略参数-F也一样



Grep语法：



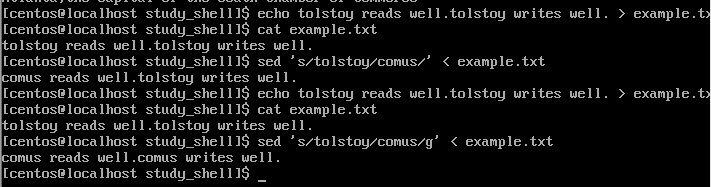
得到又符合型整数的最大值

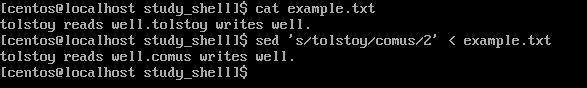




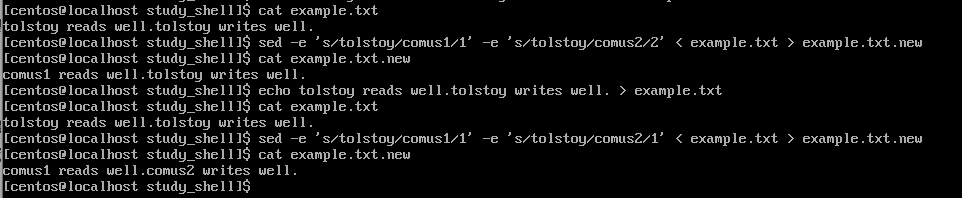
Sed方法替换文本：

模式修饰符：

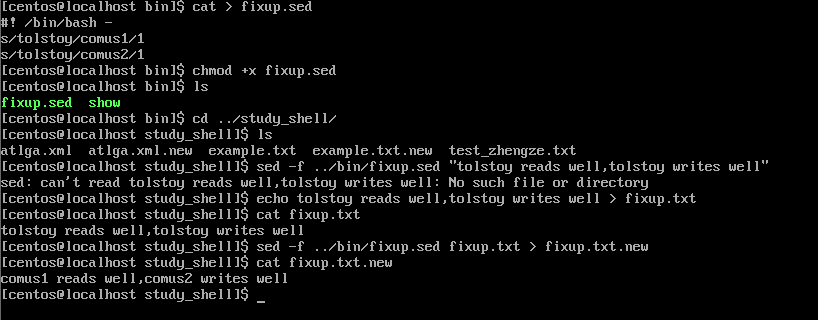
替换指定某一个：



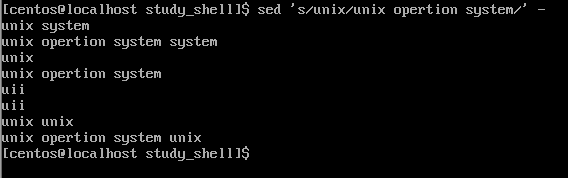
同时执行多个文本替换命令：从下面的结果可以看出，替换命令是一个一个执行的（执行完第一个才执行第二个，因此替换完第一个tolstoy，剩余的就称为第一个了）



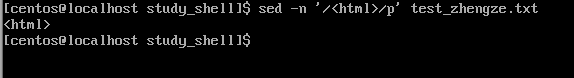
-f选项执行替换的shell脚本：

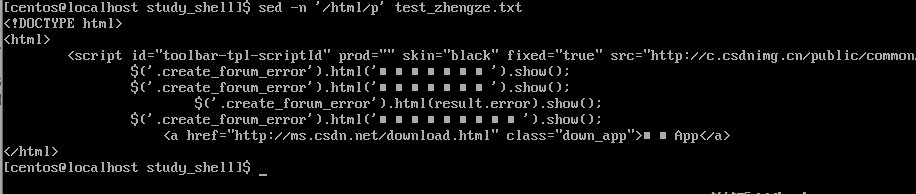


一个一个一次读入替换：

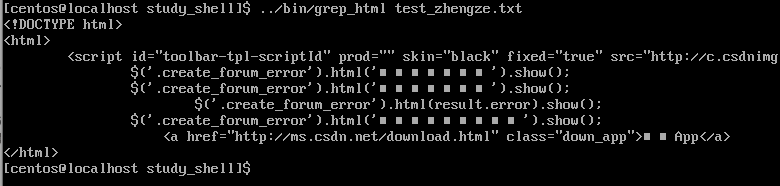


-n选项改变了sed的默认行为：

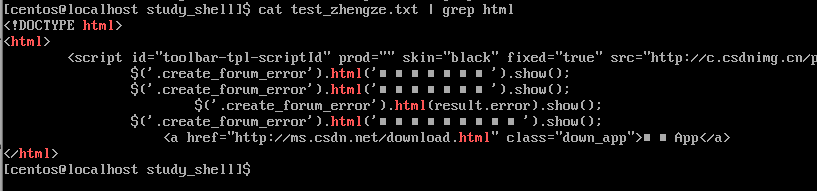




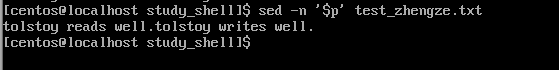
将命令写入文件：



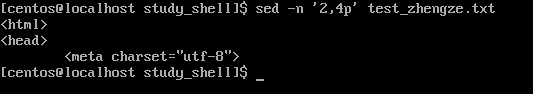
Grep执行类似的查询操作：



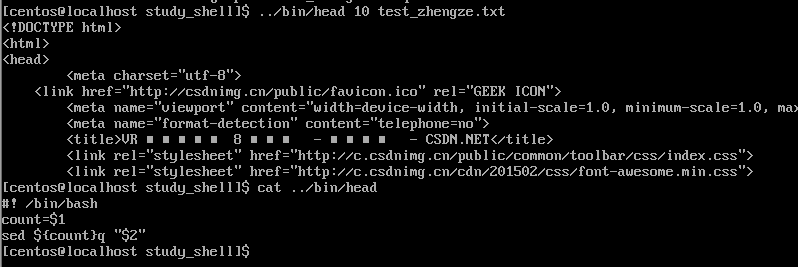
打印最后一行：



打印指定行：如下为2-4行



脚本实现打印前10行：



Grep命令后面的自动作为输入的文本