**实验七：文件系统报告（阐述设计程序的思路）**

**difftree.sh:**

首先审题：编写的difftree命令是需要在命令行方式下进行的，所以需要编写Linux脚本（.sh文件）。然后就是编写的思考过程，先将整体代码展示如下：





首先第一行定义了Linux脚本的“头文件”-- #!/bin/bash，用来指示用bash这个shell来解析此脚本。然后注释出此脚本的目的以及关键性内容，紧接着第一个if循环来判断是否传入了两个参数（还没有判断参数的合法性），在Linux中用 $# 表示在命令行所输出的参数个数，因此条件就是判断 $# 是否等于2即可，如果不是则打印一行错误信息，告诉用户所输入的参数不是两个。

参数dir1和dir2用来记录传入的两个参数$1和$2，注意这里传入的参数自己要求是必须带上完整文件路径的。然后就是判断参数的合法性：Linux中检查目录是否存在使用 -d 命令，后面加上带路径的目录名称，如果目录不存在则打印一行错误信息，告诉用户输入的目录不存在。

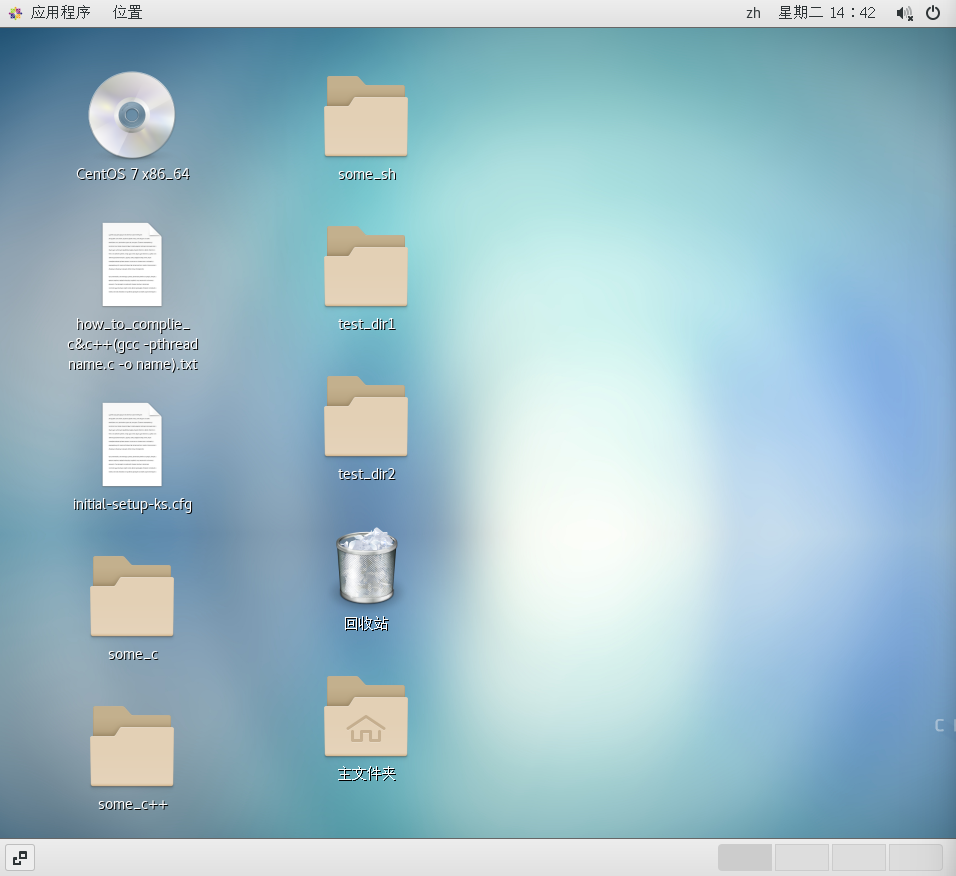
开始比较目录结构，需要调用的是Linux中的diff命令，dirr命令中参数 -q 仅报告文件是否有差异，不显示具体差异内容，-r 选项指定递归比较s。因此直接调用diff -qr命令即可，将结果保留在变量diff\_output中。接着的if语句使用了Linux内置参数$?，它保存的是上一条语句执行之后的退出状态，如果为0表示命令正确执行，非0则表示命令出错。因此判断执行diff命令的$?是否为0可以知道dir1和dir2两个目录中的结构是否相同，若结构不同则还需要打印出不同的文件信息。

但Linux的diff命令只能输出比较的两个目录中不同的部分，我们还需要判断目录是否有相同的文件（这里的相同指的是文件内容的100%一致性，因为自己测试过即使两个文件只是多了几行空白也会显示不完全相同）。

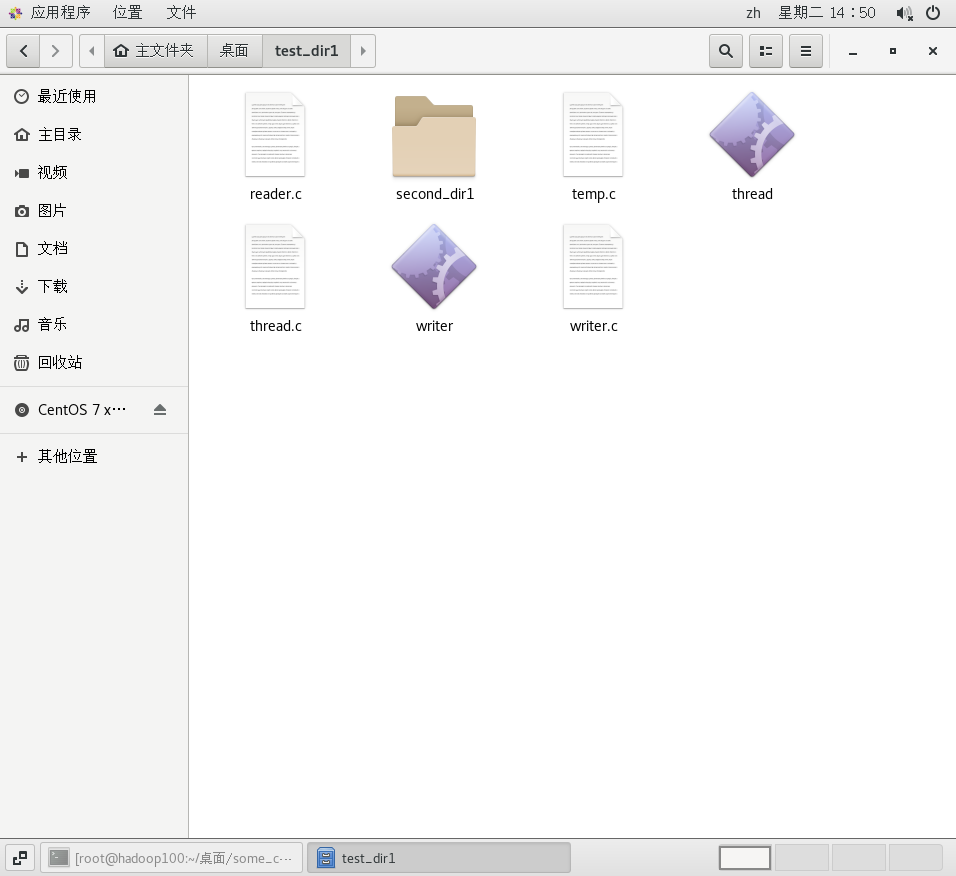
这里重点说明while循环判断文件是否完全相同的部分：首先看到在最后一行done后面用文件重定向将find命令的内容定向给了while循环。这里的意思是将find得到的结果作为while循环的输入，参数说明：-type f 选项指定只匹配文件，而不是目录或其他类型的文件，-print0 是 find 命令的一个选项，它指示 find 命令以空字符作为分隔符打印出文件路径。接着说明循环内：条件是从find命令的输出中逐行读取文件路径,并存储在变量file中,IFS= 确保读取的整行文本被保留，而read -r用于禁用特殊字符的转义，-d ''用于指定以空字符作为行分隔符。首先将目录1中每个文件的完整路径名去除其文件名，将其保留在relative\_path，指明这是相对路径，接着再拼接上目录2中的每一个文件，得到目录2下相应的文件路径。大if循环是判断首先是否有同名文件，如果不同名则不用判断两个文件内容是否相同，因为这两个文件的名称属性已经不一样了。如果名称相同，再调用Linux中diff命令判断这两个文件是否完全一致，一致或者不一致都需要输出信息告诉用户这两个文件只是同名还是完全相同。

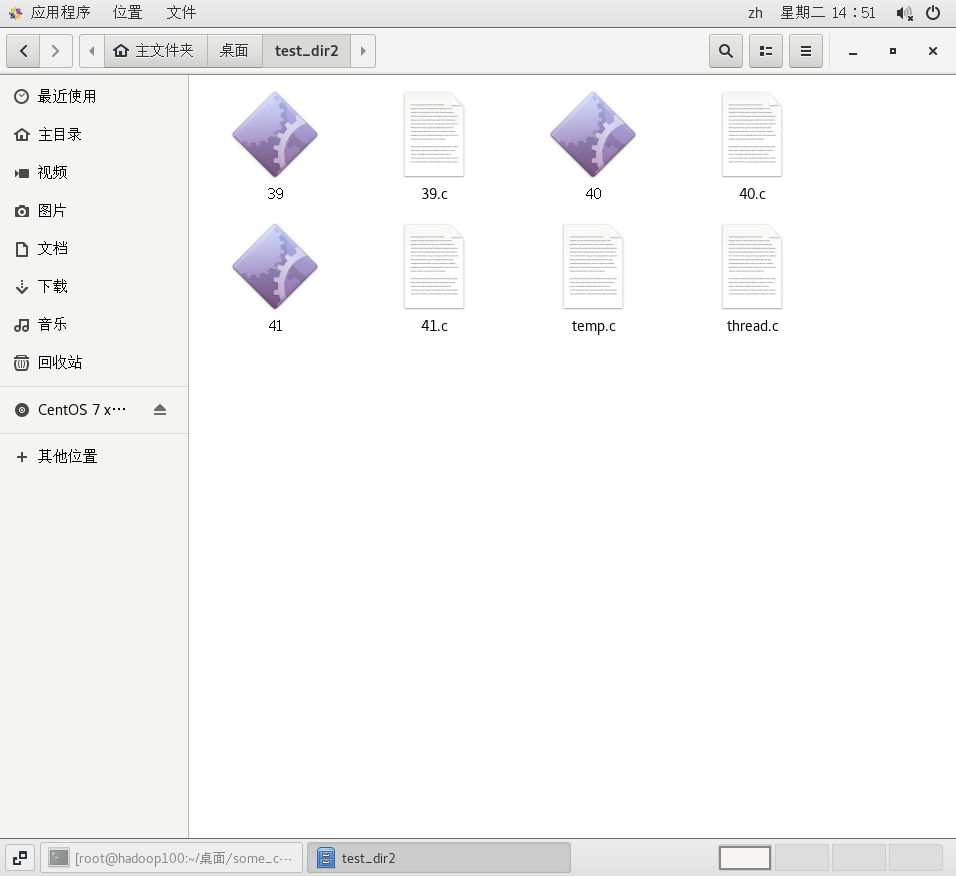
到此代码部分解释完毕，接下来是结果输出展示：

首先我在桌面创建了两个文件夹，命名为test\_dir1和test\_dir2。

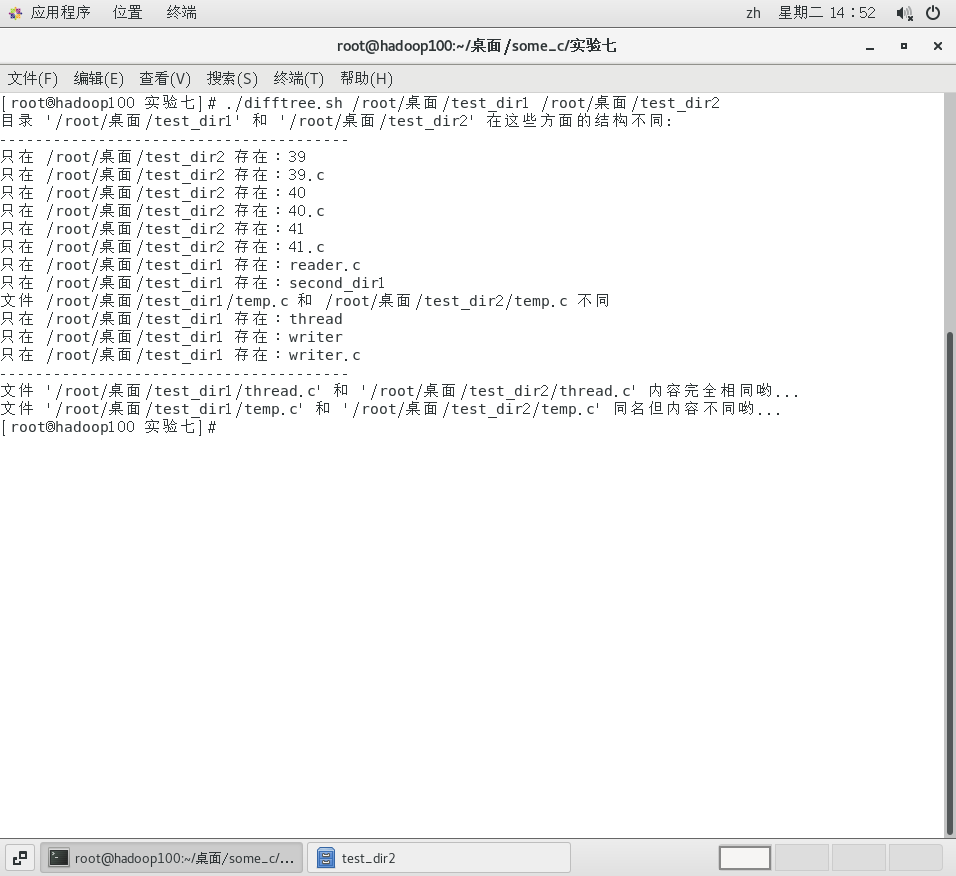


然后我分别在两个文件夹中创建了一些文件（把之前实验的代码复制过去测试），为了增加测试性，在两个目录中的thread.c文件是一模一样的（直接复制两次到两个文件中），而temp.c文件的区别仅仅是一个文件中多换了一行空白，还在目录1中还创建了一个子目录second\_dir1：





接着就是在命令行运行dirrtree了，效果如下：



可以看到difftree脚本命令可以正确判断出哪些文件只在目录1中出现，哪些文件只在目录2中出现，哪些文件是在两个文件中同名的，哪些文件在两个文件中的内容完全一样等。