



《Linux 分析与应用》课程实验报告

实验名称	课程项目：ls2		
姓 名	姜晨	学 号	09379062
专 业	10 计算机 A	实验时间	2012.12.11
<h3>一、项目简介</h3> <p>根据所学内容，编写一个 Unix/Linux 程序：ls2.c</p>			
<h3>二、功能要求</h3> <p>1) ls2 可以根据是否跟有命令选项（-l）分别以简要/详细两种方式显示某一 （组）目录下的文件以及子目录的信息，如：</p> <p>ls2 -l /tmp 表示以详细方式列出/tmp 下的文件/ 目录信息</p> <p>ls2 /mydir /tmp 表示以简要方式列出/mydir 与/tmp 两个目录下 的文件/ 目录信息</p> <p>2) 简要方式下，文件/ 目录信息的显示内容只包含文件（或目录）名字，但 显示版面必须做分栏处理，且各名字的排列按字母序（升或降）进行排 序。</p> <p>3) 详细方式下，每行显示一个文件（或目录）的详细信息，显示信息内容 与格式应与常规使用的“ls -l”命令的输出格式一致，即，每行分别依次 显示：文件类型，权限，用户名，组名，链接数，文件大小，以及修改 时间等。</p> <p>4) 在两种方式下，所有隐含文件(以圆点开头的文件名)不得被显示</p>			
<h3>三、实验环境</h3> <p>平台：Ubuntu 12.04.1 LTS</p> <p>编译器：gcc version 4.6.3</p> <p>编辑器：vim7.3</p>			



四、处理流程

本次实验是重写一个简单的扫描目录程序，老师给了 5 个源文件:arglist.c, ls1.c, stat1.c, stat2.c, getopt_long.c。通过 arglist.c，理解了 C 程序参数的传递；通过 ls1.c，知道了与目录操作有关的函数是放在头文件 dirent.h 里，通过目录流的结构体指针 (DIR *)来完成对目录的各种操作，这个和学过的文件流是很类似的。通过 stat1.c, stat2.c，熟悉了显示文件详细信息的接口函数;通过 getopt_long.c，知道了 linux 程序处理选项的标准方法。

所以，我先用 getopt_long 函数对程序中可能出现的-l 显示选项进行处理，将得到的参数传递给自定义的 do_ls 函数来显示目录下的文件信息。在核心函数 do_ls 中建立一个目录流指针*dir_ptr，打开传递进来的目录参数，再用一个 dirent 类型的结构体数组接收目录数据项，然后根据是否有-l 选项来显示，当指定了-l 选项，显示文件的详细信息；没有指定时，分栏递增显示文件和目录名。

五、算法描述

总体并没有太大复杂度，主要理解目录流操作，和目录相关的库函数的命名方法和函数参数接口。

其中对程序选项的处理值得注意，getopt_long 函数能够处理出现在程序参数中任意位置的选项，实际上是重写了 av 数组，把所有非选项参数都集中在一起，然后返回 optind 位置。这比我们自己写一个比较字符串中是否有"-l"的粗暴做法要严谨得多。

六、实验心得

自学 Linux 也有了一年了，平常基本上都是写写 shell, python 之类的脚本语言，没有真正接触到 linux 系统编程。而要进行 linux 系统编程，非 C 语言不可。

通过这次作业，复习和加深了对 C 语言的理解及应用。虽然最终写出来的程序不过两百行，但是，中间为了更灵活的调用一些库函数，翻阅了很多资料和手册，还专门查阅了系统调用和库函数的区别：库函数是运行在用户空间的，而系统调用需要进入内核空间，根据程序的规模需求，我们要适当选择系统调用和库函数，不能一概而论。

平常一直用 vim 写程序，这次为了更好的补全，也去 vim 官网多添加了几个插件，用 vim 写 C 程序也很爽。学 linux 就是从工具开始的，linux 和 vim 一样都是能极大改善程序员习惯的好工具，我也更加坚定了要从底层学好并掌握 linux 工具的开发。

以前都没有想过重写我们常用的 linux 命令，虽然这次是写一个简单的 ls,却学到了很多，相信为以后写任何 linux 下的 C 程序都开了个好头。