

苏州大学实验报告

院、系	计算机学院	年级专业	14 计科	姓名	沈家赞	学号	1427405017
课程名称	操作系统课程设计					成绩	
指导教师	王红玲	同组实验者	无	实验日期	2017 年 6 月 10 日		

实验名称

实验四 文件管理

一. 实验目的

文件系统是操作系统中管理取信息的机构，它具有“按名存取”的功能，不仅方便用户，而且能提高系统效率且安全可靠。为了使学生对各种文件操作命令的实质内容和执行过程有深入了解，指导学生在 Linux 环境下编写和调试一个简单的文件系统，达到模拟文件管理工作的目的。

二. 实验内容

使用 Linux 下 shell 编程或图形化编程，模拟实现一个简单的文件系统，包括文件目录操作和文件操作等。

【实验要求】

使用 Linux 下图形化编程环境，使用 C++ 语言模拟实现二级文件目录结构，第一级为主目录文件 MFD，第二级为用户文件。目录文件 UFD。

- (1) 设计一个有 m 个用户的文件系统，每个用户最多可保存一个文件；
- (2) 规定用户在一次运行中只能打开 K 个文件；
- (3) 系统能检查键入命令的正确性，出错时应能显示出错原因；
- (4) 对文件应能设置保护措施，如只能执行，允许读，允许写等；
- (5) 对文件的操作设计提供一套文件操作：

CREATE 建立文件

DELETE 删除文件

OPEN 打开文件

CLOSE 关闭文件

READ 读写文件

WRITE 写文件

三. 实验步骤和结果

【GUI 交互】

为了实现良好的交互效果，我决定采用 GUI 编程来实现比命令行更加简单易用、用户友好的使用体验。

Linux 环境 GUI 编程主流的 GUI 开发库有 GTK+ 以及有名的 Qt。考虑到 Qt 优良的跨平台特性，可以实现“一处编写，处处编译”且 Qt 在现实中的工程开发中广受欢迎，正是时下流行且领先的 GUI 开发库，我决定自学 Qt 并采用 Qt ceator 作为本次实验的开发平台。

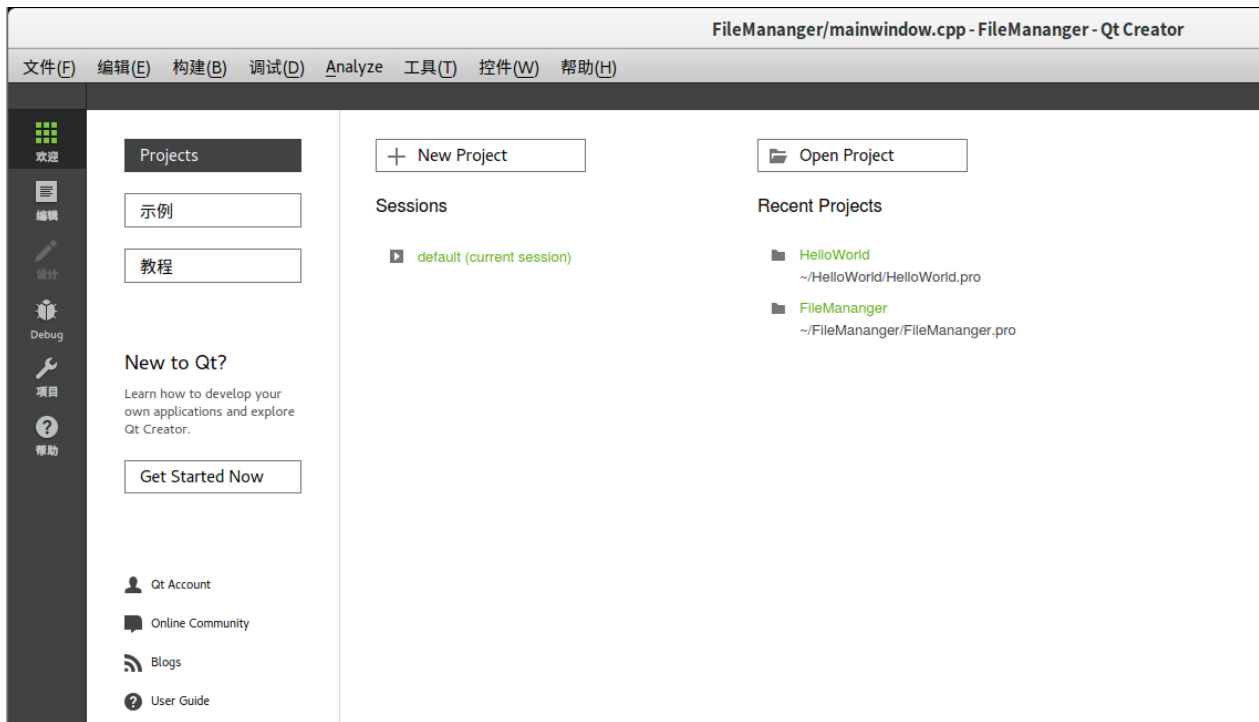
【Qt 5.6.2】

我采用了较新且稳定的 Qt5.6.2 版本，在 Ubuntu16.04 LTS 平台进行开发及测试。

Qt 官方网站： <https://www.qt.io/cn/>

注册账号，说明使用需求后，即可免费下载开源社区版的 Qt。此版本与商业版没有太大区别，主要是使用版权上仅允许开发非商业销售的或遵守开源协议的软件。

安装好后，Linux 环境下使用 Qt Creator 进行开发，Qt Creator 是一个轻盈已用的 IDE。界面如图：



Ubuntu 环境下的 Qt 安装好后还不能顺利编译调试，需要配置好相关编译环境和库环境，如：G++ 编译器，OpenGL 的链接库。内容较为繁琐。

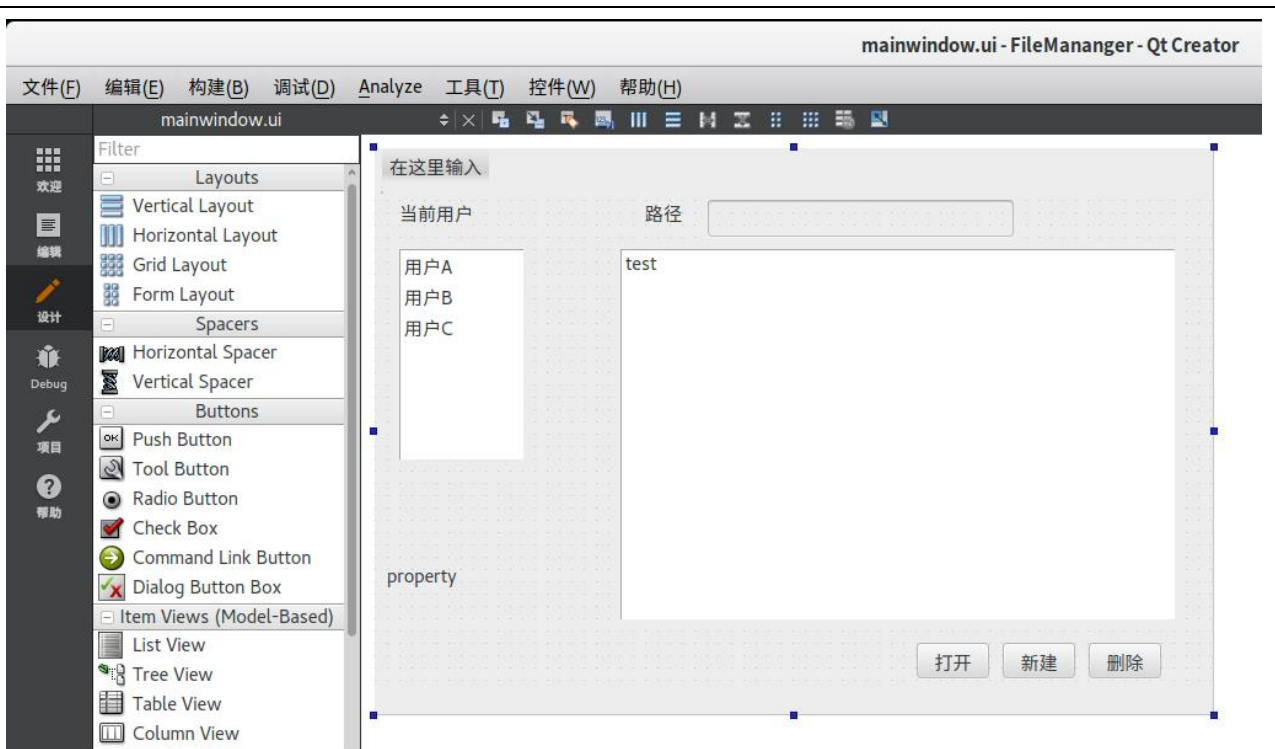
具体整理在 我的个人博客（heary.cn）文章：

Qt - Linux 下的 Qt 开发环境配置 (<https://heary.cn/index.php/archives/80/>)

【图形界面设计】

通过 Qt Creator 即可快速建立新的 GUI 应用程序，双击后缀为 ui 的文件即可打开所见即所得的 GUI 设计界面。

原理： *.ui 文件实质上是一个 xml 格式文档，在 Qt 中不允许用户手动编辑，由 Qt Creator 根据开发者的排版设计统一格式化生成。



文件管理器的 GUI 界面设计如图所示：

根据实验要求，我准备了：

- (1) **用户窗口**：点击即可切换到用户的用户目录下；
- (2) **路径**：显示当前所在的路径，用户也可以修改路径，跳转至目标路径。
- (3) **文件窗口**：显示当前目录下的目录和文件。
 - 3-1) 单击，即可查看属性
 - 3-2) 双击，即可打开文件预览窗口
- (4) **属性框**：显示选中的文件或目录的实用属性（大小、类型、所属用户/组）
- (5) **按钮区**：实现文件或目录的 打开、新建、删除 功能。

【C++程序开发】

Qt 采用 C++实现编程开发。

在 GUI 交互方面，Qt 采用了特色的 signal&slot 机制，而非常见的传统 callback 或事件监听机制。信号与槽的机制就是把 GUI 交互时间作为信号，连带相关参数传输到信号的接收槽中进行进一步的事件处理。

具体原理及开发特性我在个人博客（heary.cn）中进行了整理：

signal&slot - Qt 的交互响应机制（<https://heary.cn/index.php/archives/81/>）

以我的程序为例：

（**注意**：slot 函数需要在 MainWindow 类中声明）

```

/*
 * signal & slot
 * 在 MainWindow 的构造函数中实现 signal 和 slot 的信号连接
 * slot 命名风格均定为: on<Widget><Event>
 */
// 列表框图
connect(ui->listWidget_users,          SIGNAL(itemClicked(QListWidgetItem*)),          this,
        SLOT(onUserClicked(QListWidgetItem*)));
connect(ui->listWidget_files,          SIGNAL(itemClicked(QListWidgetItem*)),          this,
        SLOT(onFileClicked(QListWidgetItem*)));
connect(ui->listWidget_files,          SIGNAL(itemDoubleClicked(QListWidgetItem*)),          this,
        SLOT(onFileDoubleClicked(QListWidgetItem*)));

// 路径框
connect(ui->lineEdit_path, SIGNAL(returnPressed()), this, SLOT(onPathReturned()));

// 功能按钮
connect(ui->pushButton_open, SIGNAL(clicked(bool)), this, SLOT(onBtnOpenClicked(bool)));
connect(ui->pushButton_new, SIGNAL(clicked(bool)), this, SLOT(onBtnNewClicked(bool)));
connect(ui->pushButton_delete, SIGNAL(clicked(bool)), this, SLOT(onBtnDeleteClicked(bool)));

```

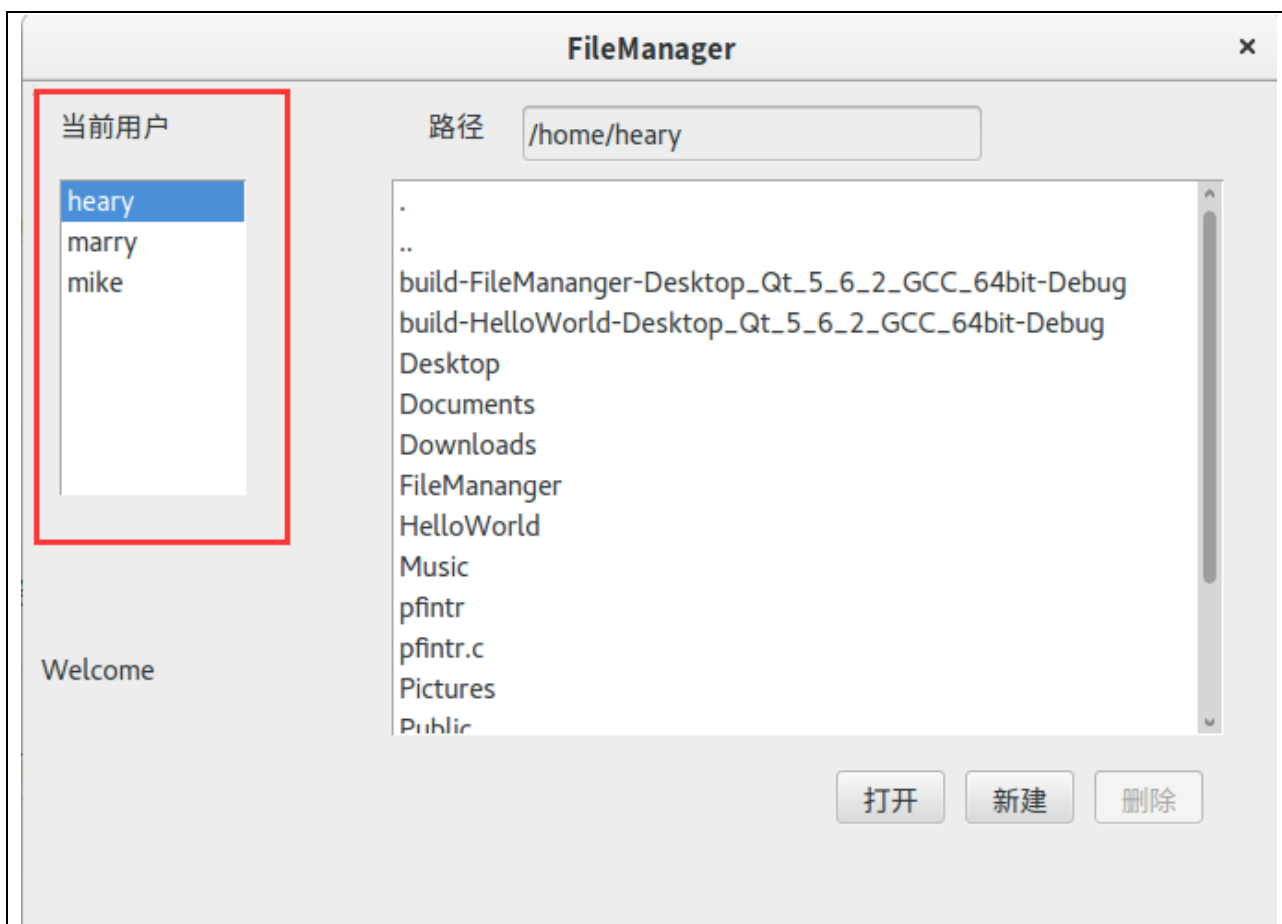
根据此次开发的内容，我把函数命名风格统一成了 on<Widget><Event>的形式，便于分工和编写。

【多用户切换】

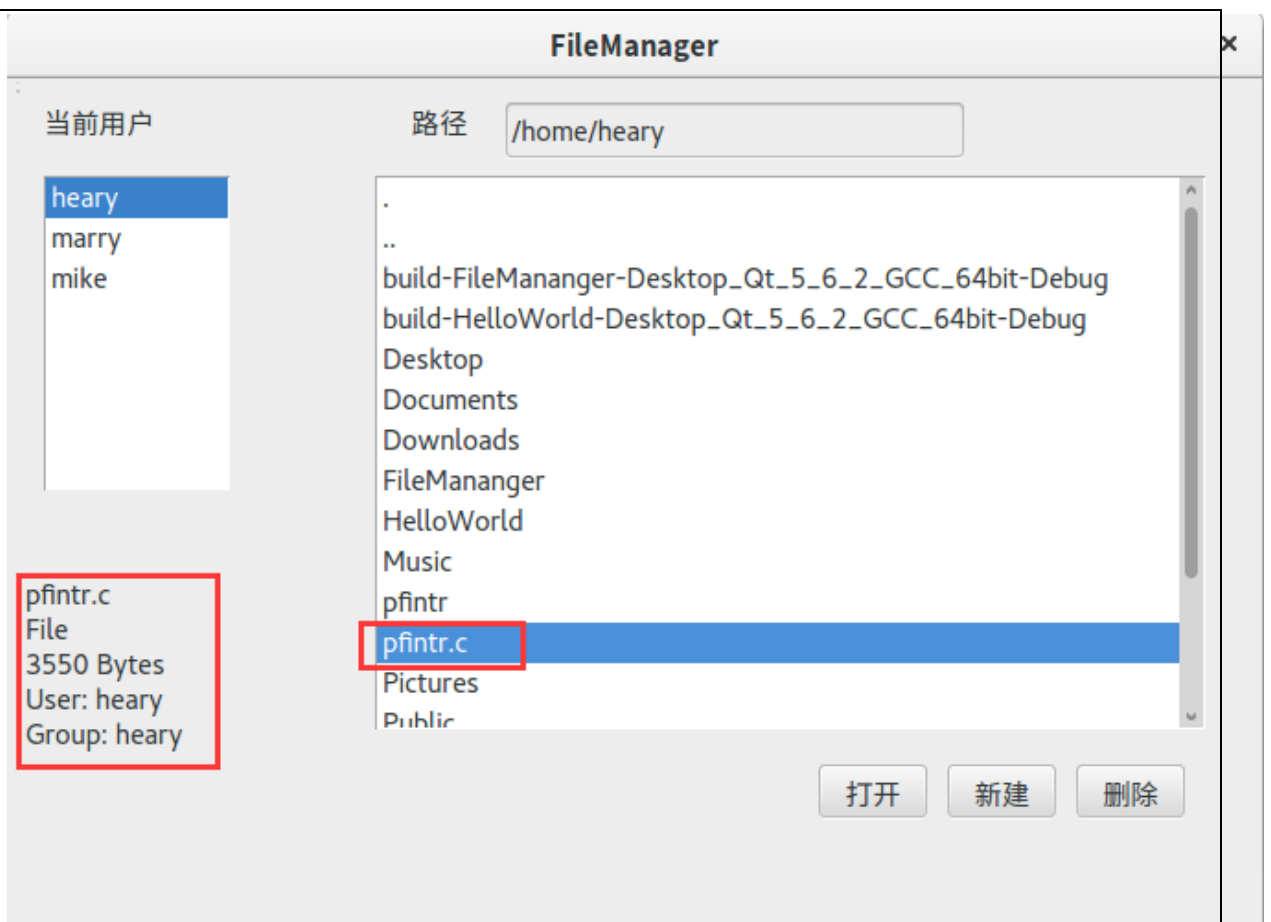
根据实验要求，我实现了多用户的文件目录切换。

本程序依据 Linux 的多用户特性，读取并解析出/home 目录下拥有文件目录的用户。

例如，点击用户窗口中，heary 用户，即自动跳转到 heary 用户的文件主目录——/home/heary/效果如图：



【查看属性】



鼠标单击任意文件或目录，即可在左下角的属性显示区域看到此文件的基本属性，包含：

- (1) 文件或目录名
- (2) 类型：文件或目录
- (3) 大小：文件所占用的字节数
- (4) 文件所属用户：Linux 权限机制下的文件所属用户
- (5) 文件所属组：Linux 权限机制下的文件所属组

【打开文件】

打开文件，可以看到预览的文件内容。打开目录，即跳转到目标目录。

操作方式为：

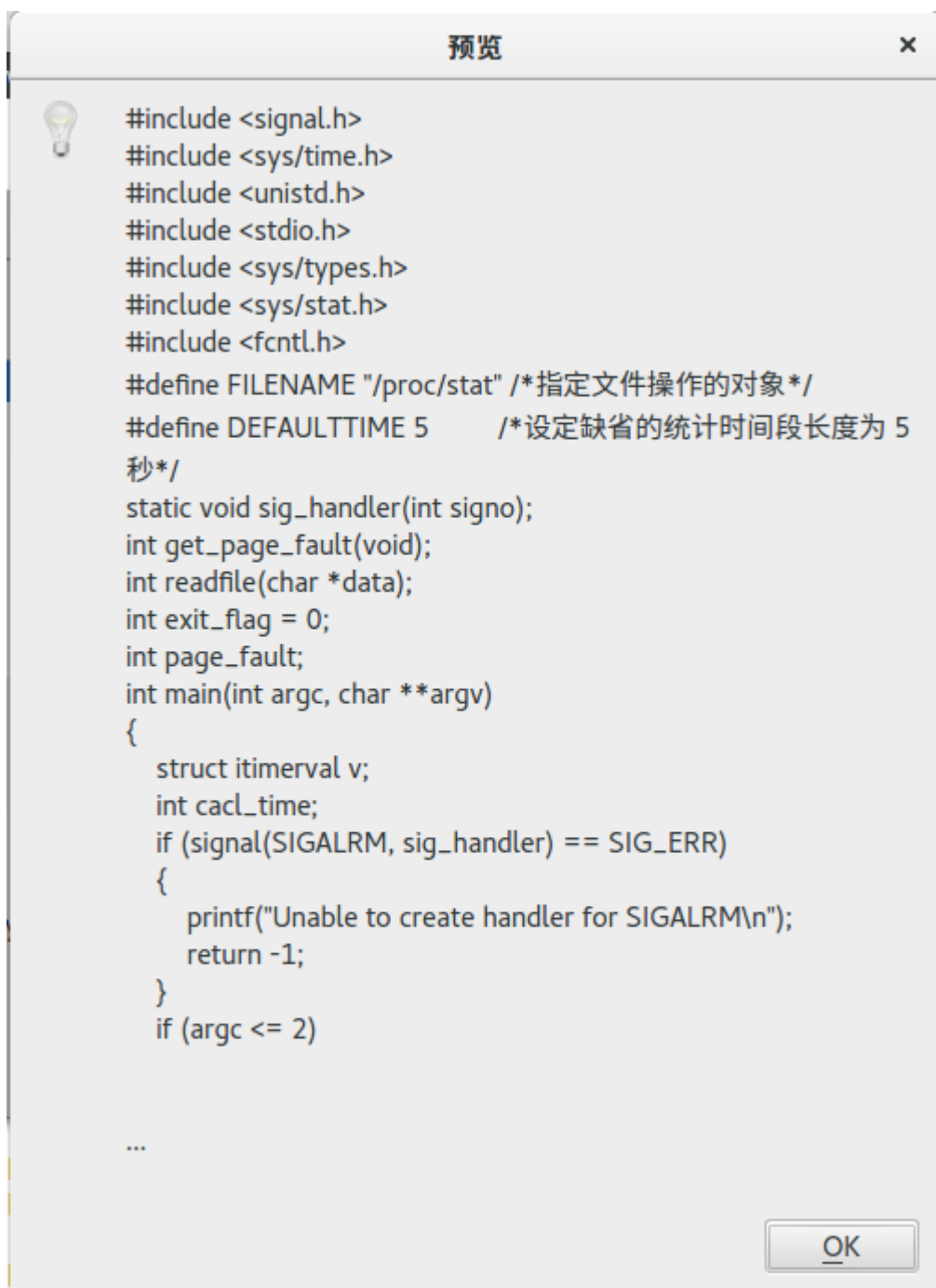
- (1) 选中文件或目录后，点击打开按钮；
- (2) 直接双击打开文件或目录。

程序将自动识别选中项目是文件还是目录，并做出反应：

- (1) 若目标为目录，则跳转至打开的目标目录；
- (2) 若目标是文件，则打开新窗口，显示此文件的内容预览。

下图展示的是文件打开预览的窗口：

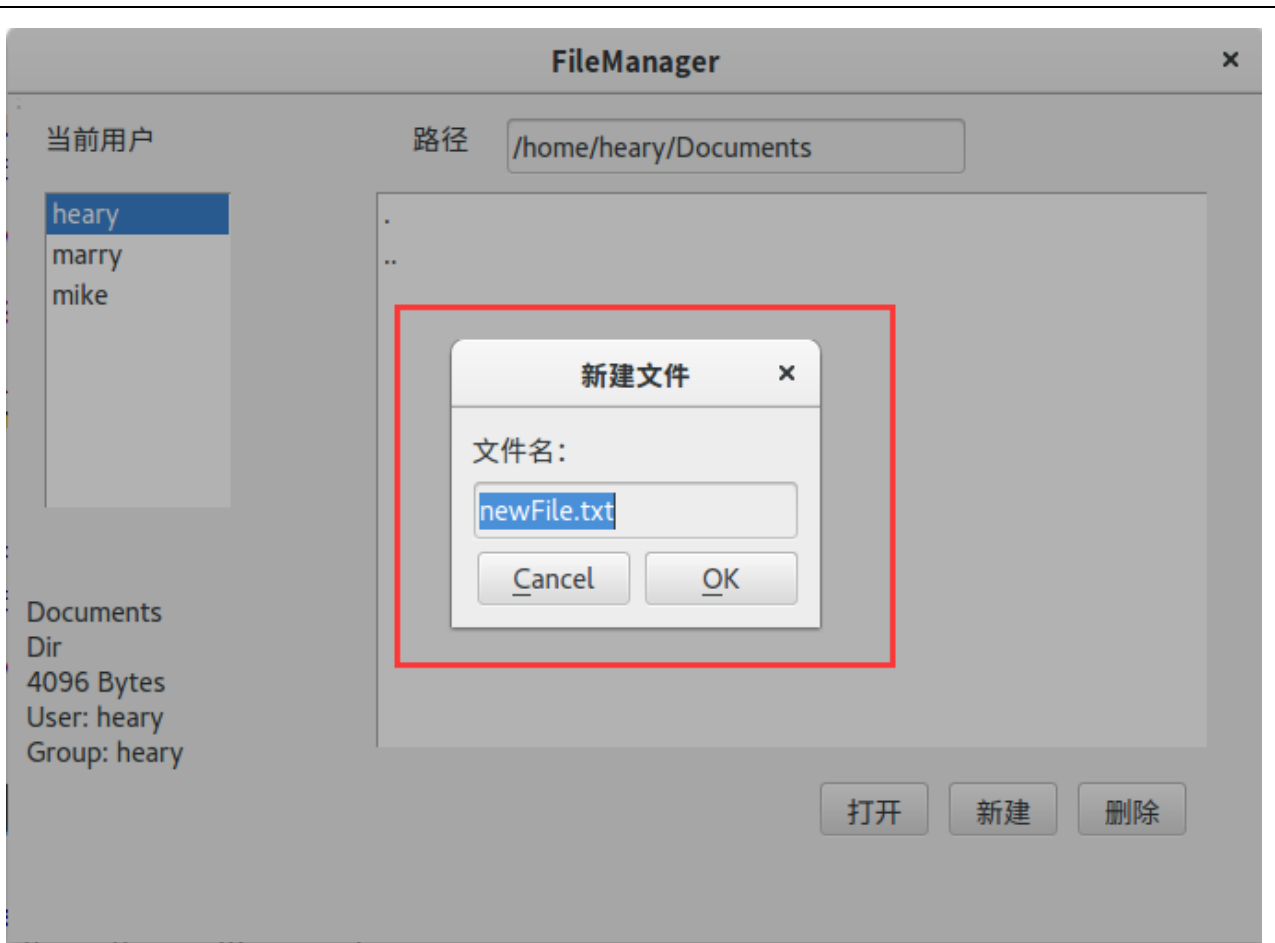
（打开的是 `pfintr.c` 文件，这是一段 C 代码）



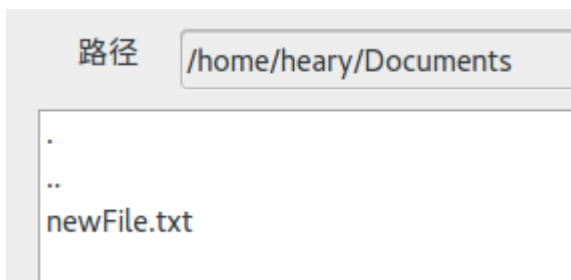
```
#include <signal.h>
#include <sys/time.h>
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#define FILENAME "/proc/stat" /*指定文件操作的对象*/
#define DEFAULTTIME 5 /*设定缺省的统计时间段长度为 5 秒*/
static void sig_handler(int signo);
int get_page_fault(void);
int readfile(char *data);
int exit_flag = 0;
int page_fault;
int main(int argc, char **argv)
{
    struct itimerval v;
    int cacl_time;
    if (signal(SIGALRM, sig_handler) == SIG_ERR)
    {
        printf("Unable to create handler for SIGALRM\n");
        return -1;
    }
    if (argc <= 2)
```

【新建文件】

选择合适的目录，点击新建，即可创建新文件。



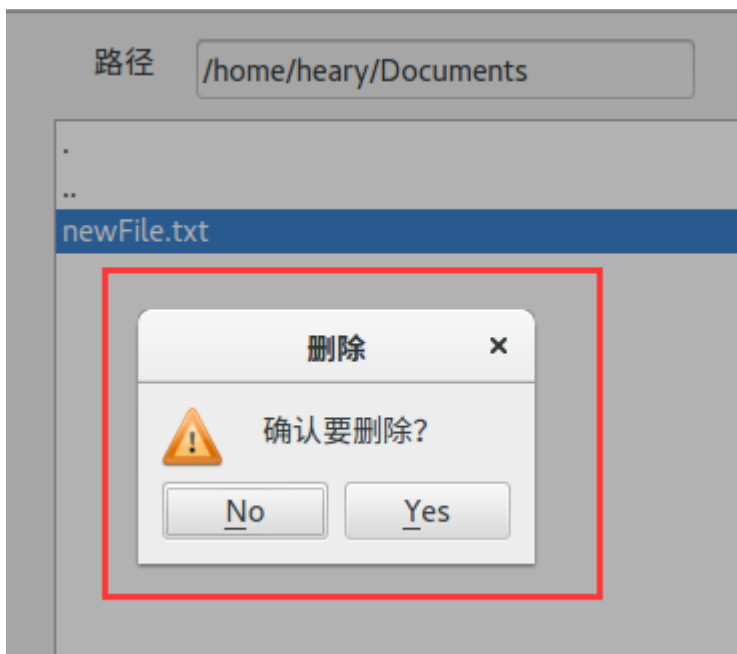
在弹出的新建文件对话框中，输入新建文件的文件名，单击 OK 即可创建新文件。（文件名默认为 newFile.txt）。



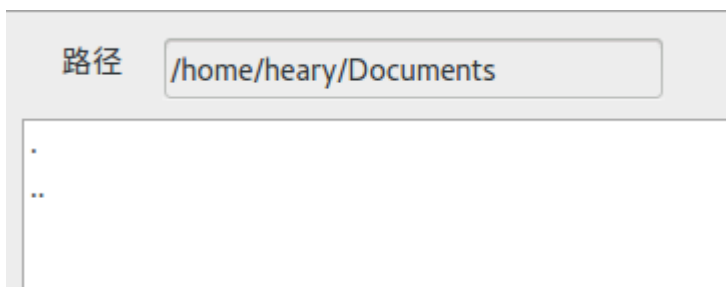
可以看到，当前目录下成功创建了新文件，newFile.txt

【删除文件】

本着“数据无价”的原则，删除文件作为较为特殊且具有风险性的操作，我在程序中设计为点击删除键后，不会直接删除文件，而是弹出删除文件的确认对话框。确认后才删除文件。



需要在删除对话框中点击 Yes，确认删除操作，才会删除文件。



如图，文件已成功删除。

【小结】

< 完整的程序代码篇幅较长，随附件给出，不在实验报告中贴出 >

本程序实现了实验要求的文件打开、新建、删除、关闭、以及多用户多目录切换等功能，并结合 Qt 的 GUI 开发平台，实现了一些交互细节和快捷操作：如，直观的单击选中查看属性，双击打开文件或目录。

四. 实验总结

通过本次实验，我以开发 Linux 文件管理器为契机，自学了 Qt 编程，并实现了实验内容要求的文件打开、新建、删除、关闭、以及多用户多目录切换的功能，并借助 Qt GUI 开发的优势，实现了直观且方便的单击选中查看属性、双击打开文件预览或进入目录等功能。

在本次实验的学习过程中，我不仅进一步学习了 Linux 环境下的文件管理机制，还自学并练习了 Qt 在 Linux 环境下的 GUI 程序设计