项

目

报

告

之

文件夹复制

项

目

作者：刘俊杰

班级：物网171

日期：2019年6月10号

指导老师：李军

# 目录

[1 、项目目的 3](#_Toc477335449)

[2 、项目设备及软件 3](#_Toc477335450)

[3 、项目内容 3](#_Toc477335451)

[4 、项目原理 3](#_Toc477335452)

[5 、项目相应知识点 4](#_Toc477335453)

[6 、项目流程图 6](#_Toc477335454)

[7 、项目程序实现 7](#_Toc477335455)

[8 、项目步骤 7](#_Toc477335456)

[9 、项目结果 16](#_Toc477335457)

# 1、项目目的

**通过本课程设计巩固《ARM嵌入式应用程序设计》课程的教学成果，深入理解结构体、指针、链表、动态分配内存和文件操作的等C程序设计中的中高级技术，通过课程设计的综合训练，培养学生实际分析问题、编程和动手能力、提高学生利用C语言进行程序设计的能力，初步培养良好的编程习惯和编程风格，初步学习程序的文档的撰写方法。**

**掌握文件 I/O、递归的操作使用；**

**掌握 C 语言中的结构体、指针、链表、动态分配内存**

# 2 、项目设备及软件

**电脑一台，虚拟软件VM， redhat平台；**

# 3 、项目内容

**文件夹复制**

**1.指定文件夹复制到另一个文件夹**

**2.复制文件夹附带个人信息**

# 4 、项目原理

**基于文件I/O的知识。**

**1.使用opendir函数打开目录a，返回指向目录a的DIR结构体c。**

**2.调用readdir(c)函数读取目录a下所有文件（包括目录），返回指向目录a下所有文件的dirent结构体d。**

**3.遍历d，调用stat（d->name,stat \*e）来获取每个文件的详细信息,判断是否是目录**

**4.调用strcat()函数将取得的文件名与当前文件夹拼接成一个完整的路径**

**5.调用fopen()打开源文件和目标文件的文件流,调用fread()函数和fwrite()函数进行复制**

# 5、项目相应知识点

**1).fopen()打开文件**

**FILE \*fopen(char \*filename, char \*mode);**

**2).fclose()关闭文件**

**intfclose(FILE\*stream);**

**3).strcat()字符串拼接函数,把 src 所指向的字符串追加到 dest 所指向的字符串的结尾**

**char \*strcat(char \*dest, const char \*src)**

**4).strlen()计算字符串长度函数**

**size\_t strlen(char const\* str);**

**5).sizeof()计算数据类型的大小**

**Sizeof(data type);**

**6).time()以time\_t对象返回当前日历时间**

**time\_t time(time\_t \*t);**

**7).strtime()根据 format 中定义的格式化规则，格式化结构 timeptr 表示的时间，并把它存储在 str 中。**

**size\_t strftime(char \*str, size\_t maxsize, const char \*format, const struct tm \*timeptr)**

**8).fputs()把字符串写入到指定的流 stream 中，但不包括空字符**

**int fputs(const char \*str, FILE \*stream)**

**9).malloc()申请内存空间**

**void \*malloc(size\_t size);**

**10).memset可以方便的清空一个结构类型的变量或数组**

**void\*memset(void\*s,intc,size\_tn);**

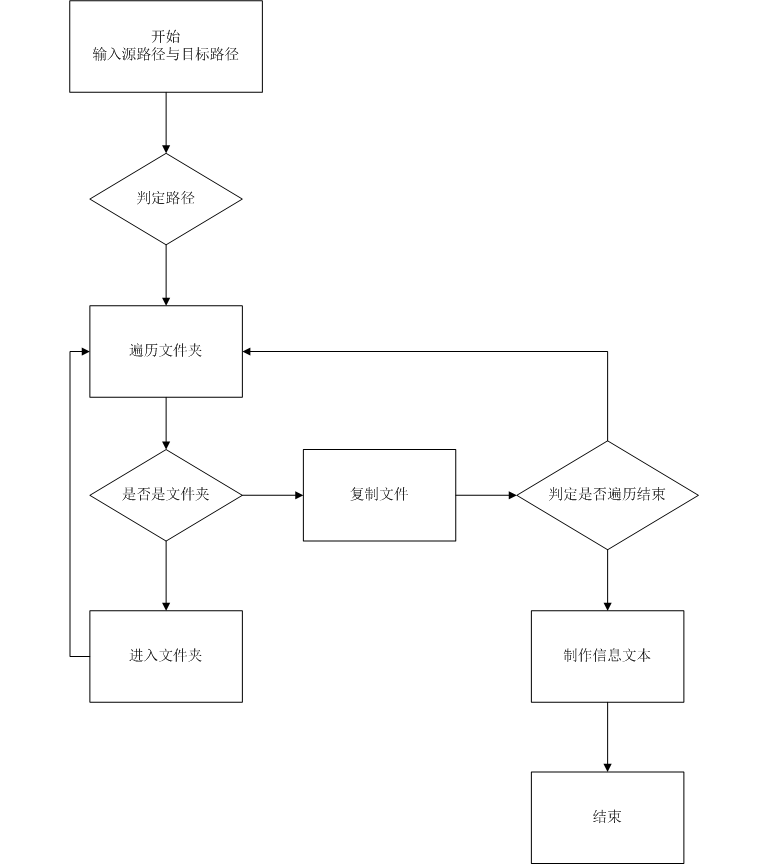
**11).opendir()用来打开参数name 指定的目录, 并返回DIR\*形态的目录流, 和open()类似, 接下来对目录的读取和搜索都要使用此返回值**

**DIR \* opendir(const char \* name);**

**12).mkdir()创建目录,0777权限所有者所在组都有权限进行读、写、执行操作**

**int mkdir(const char \*pathname, mode\_t mode);**

# 6、项目流程图



**图1 文件复制的实现**

# 7、项目程序实现

# **#include<stdio.h>**

# **#include<time.h>**

# **#include<stdlib.h>**

# **#include<string.h>**

# **#include<dirent.h>//输出文件信息---只在Linux平台**

# **#include<sys/stat.h>//判断是否目录**

# **#include<fcntl.h>**

# **void readme(char \*source\_path,char \*destination\_path)**

# **{**

# **FILE \*fd;**

# **char\* result;**

# **strcat(destination\_path,"/readme.txt");**

# **if ((fd = fopen(destination\_path,"a+")) == NULL) {**

# **printf("readme.txt文件创建失败!");**

# **exit(1);**

# **}**

# **time\_t rawtime;**

# **struct tm \*info;**

# **char buffer[80];**

# **time( &rawtime );**

# **info = localtime( &rawtime );**

# **strftime(buffer, 80, "%Y-%m-%d %H:%M:%S", info);**

# **strcat(buffer,"\n复制此文件夹的人:LJJ\n此文件夹来自:");**

# **strcat(buffer,source\_path);**

# **fputs(buffer,fd);**

# **printf("\n已在文件夹内readme.txt写入有关信息\n");**

# **fclose(fd);**

# **}**

# **int is\_dir(char\* path){//判断是否是目录**

# **struct stat st;**

# **stat(path,&st);**

# **if(S\_ISDIR(st.st\_mode)){**

# **return 1;**

# **}**

# **else{**

# **return 0;**

# **}**

# **}**

# **/\*字符串处理函数\*/**

# **int endwith(char\* s,char c){//用于判断字符串结尾是否为“/”**

# **if(s[strlen(s)-1]==c){**

# **return 1;**

# **}**

# **else{**

# **return 0;**

# **}**

# **}**

# **char\* str\_contact(char\* str1,char\* str2){//字符串连接**

# **char\* result;**

# **result=(char\*)malloc(strlen(str1)+strlen(str2)+1);//str+\0;**

# **if(!result){//如果内存动态分配失败**

# **printf("字符串连接时，内存动态分配失败\n");**

# **exit(1);**

# **}**

# **strcat(result,str1);**

# **strcat(result,str2);//字符串拼接**

# **return result;**

# **}**

# **/\*复制函数\*/**

# **void copy\_file(char\* source\_path,char \*destination\_path){//复制文件**

# **char buffer[1024];**

# **FILE \*in,\*out;//定义两个文件流，分别用于文件的读取和写入int len;**

# **if((in=fopen(source\_path,"r"))==NULL){//打开源文件的文件流**

# **printf("源文件打开失败！\n");**

# **exit(1);**

# **}**

# **if((out=fopen(destination\_path,"w"))==NULL){//打开目标文件的文件流**

# **printf("目标文件创建失败！\n");**

# **exit(1);**

# **}**

# **int len;//len为fread读到的字节长**

# **while((len=fread(buffer,1,1024,in))>0){//从源文件中读取数据并放到缓冲区中**

# **fwrite(buffer,1,len,out);//将缓冲区的数据写到目标文件中**

# **}**

# **fclose(out);**

# **fclose(in);**

# **}**

# **void copy\_folder(char\* source\_path,char\* destination\_path){//复制文件夹**

# **if(!opendir(destination\_path)){**

# **if (mkdir(destination\_path,0777))//如果不存在就用mkdir函数来创建**

# **printf("创建文件夹失败！");**

# **}**

# **char \*path;**

# **path=(char\*)malloc(512);//path直接申请512的位置的空间用于目录的拼接**

# **path=str\_contact(path,source\_path);**

# **struct dirent\* filename;**

# **DIR\* dp=opendir(path);//用DIR指针指向这个文件夹**

# **while(filename=readdir(dp)){//遍历DIR指针指向文件数组**

# **memset(path,0,sizeof(path));**

# **path=str\_contact(path,source\_path);**

# **//如果source\_path,destination\_path以路径分隔符结尾，那么source\_path/,destination\_path/直接作路径即可**

# **//否则要在source\_path,destination\_path后面补个路径分隔符再加文件名**

# **char \*file\_source\_path;**

# **file\_source\_path=(char\*)malloc(512);**

# **if(!endwith(source\_path,'/')){**

# **file\_source\_path=str\_contact(file\_source\_path,source\_path);**

# **file\_source\_path=str\_contact(source\_path,"/");**

# **}**

# **else{**

# **file\_source\_path=str\_contact(file\_source\_path,source\_path);**

# **}**

# **char \*file\_destination\_path;**

# **file\_destination\_path=(char\*)malloc(512);**

# **if(!endwith(destination\_path,'/')){**

# **file\_destination\_path=str\_contact(file\_destination\_path,destination\_path);**

# **file\_destination\_path=str\_contact(destination\_path,"/");**

# **}**

# **else{**

# **file\_destination\_path=str\_contact(file\_destination\_path,destination\_path);**

# **}**

# **//取文件名与当前文件夹拼接成一个完整的路径**

# **file\_source\_path=str\_contact(file\_source\_path,filename->d\_name);**

# **file\_destination\_path=str\_contact(file\_destination\_path,filename->d\_name);**

# **if(is\_dir(file\_source\_path)){//如果是目录**

# **if(!endwith(file\_source\_path,'.')){//同时并不以.结尾，因为Linux在所有文件夹都有一个.文件夹用于连接上一级目录**

# **copy\_folder(file\_source\_path,file\_destination\_path);//进行递归调用，相当于进入这个文件夹进行复制**

# **}**

# **}**

# **else{**

# **copy\_file(file\_source\_path,file\_destination\_path);//否则按照单一文件的复制方法进行复制。**

# 

# **printf(" 复 制 %s 到 %s 成 功!\n",file\_source\_path,file\_destination\_path);**

# **}**

# **}**

# **}**

# **/\*主函数\*/**

# **int main(int argc,char \*argv[]){**

# **if(argv[1]==NULL||argv[2]==NULL){**

# **printf("请输入两个文件夹路径，第一个为源，第二个为目的！\n");**

# **exit(1);**

# **}**

# **printf("\n");**

# **char\* source\_path=argv[1];//取用户输入的第一个参数**

# **char\* destination\_path=argv[2];//取用户输入的第二个参数**

# **DIR\* source=opendir(source\_path);**

# **DIR\* destination=opendir(destination\_path);**

# **if(!source)**

# **printf("你输入的一个参数不是文件夹！\n");**

# **if(!destination)**

# **printf("输入的第二个文件夹不存在---自动创建\n");**

# **printf("进入文件拷贝:\n");**

# **copy\_folder(source\_path,destination\_path);//进行文件夹的拷贝**

# **readme(source\_path,destination\_path);**

# **return 0;**

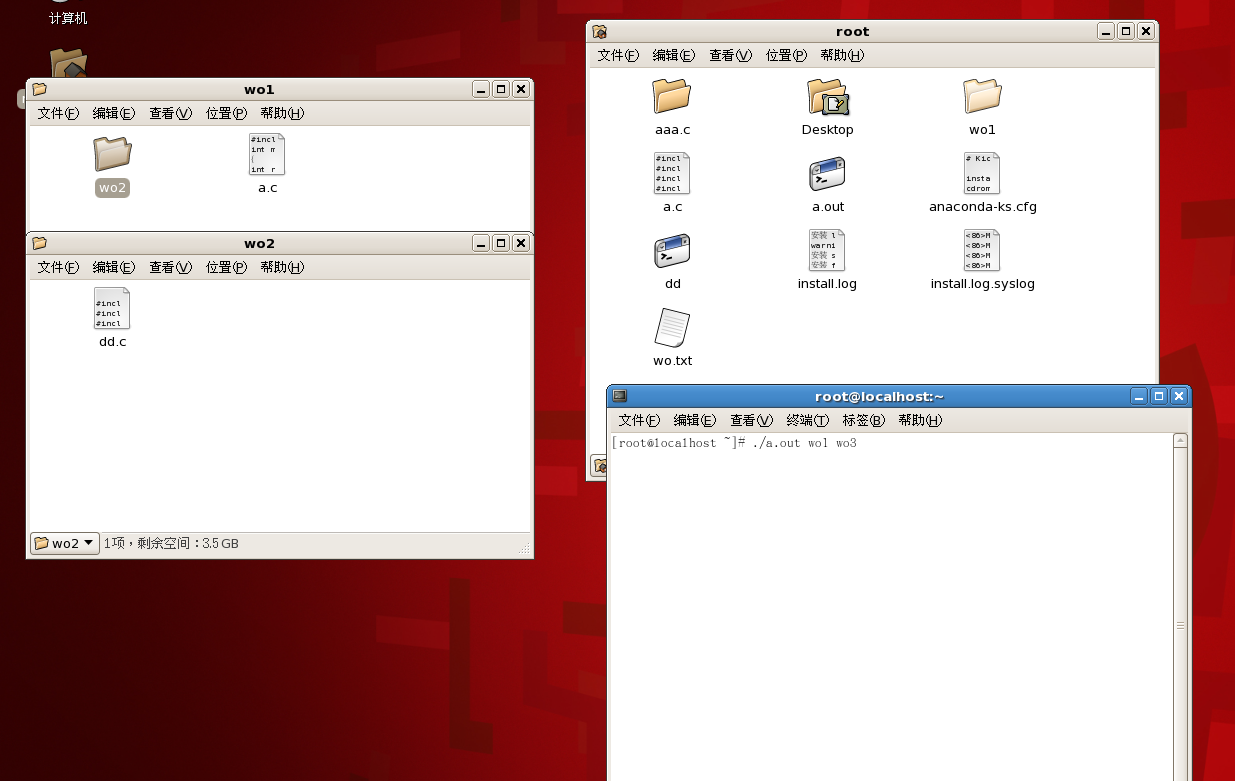
# **}**

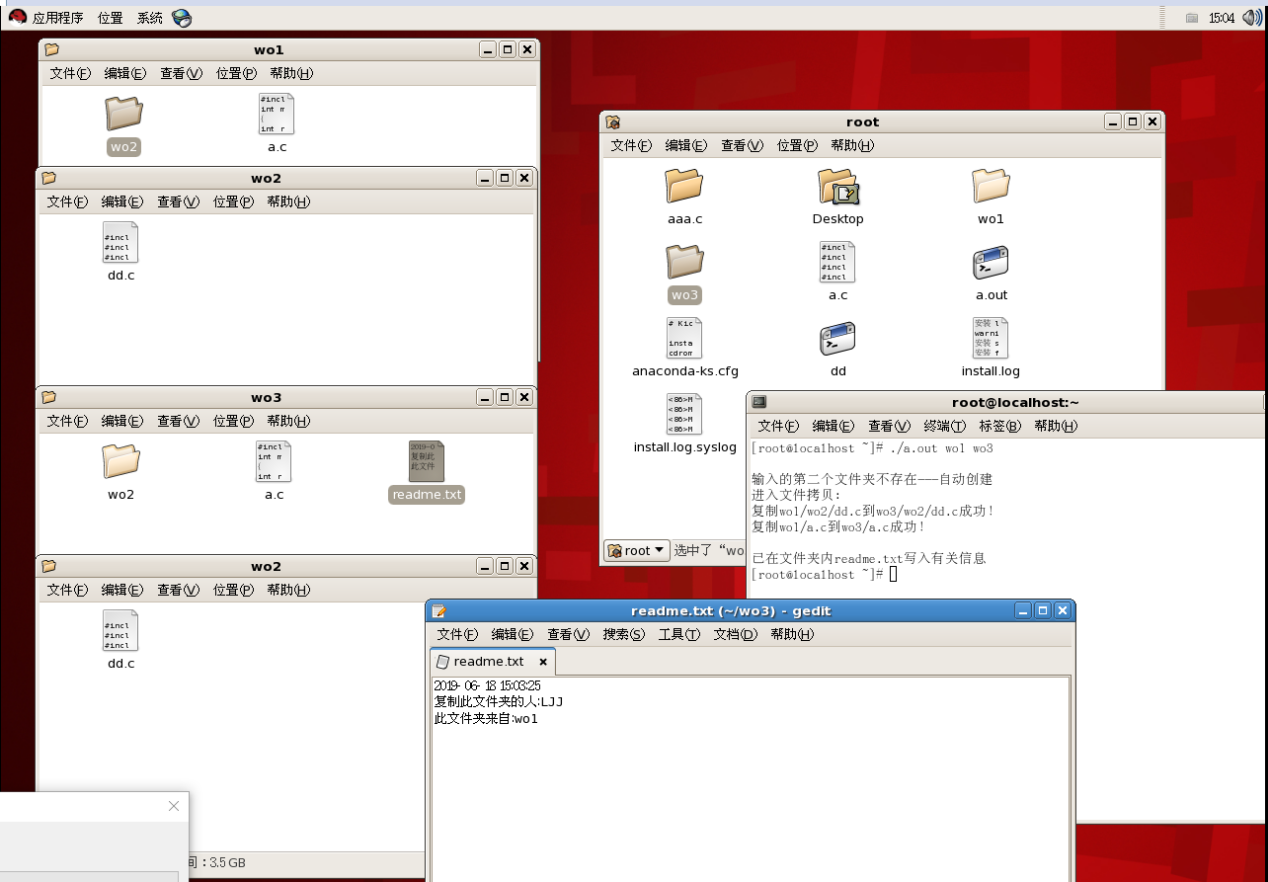


# 8 、项目步骤

1. **准备好项目环境。检查电脑是否运行正常，虚拟机软件软件是否运行良好**
2. **进入 LINUX 系统，运行代码，输入两个文件路径**

9 、项目结果

复制前



复制后