

操作系统程序作业报告

姓名: 禹泽海 李博昊 学号: 2021302111 2021302009

一、程序要求及信息

用 API 编程实现对某目录下特定类型文件进行递归计数

格式: `rmdir -t type dir`

程序语言: C 语言

程序运行环境: ubuntu 22.04.1

运行用户: yzh2021302111(这里 ubuntu 用户用我的名字和学号作为用户名称)

程序代码:

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <dirent.h>
#include <sys/stat.h>

// 递归统计指定扩展名的文件数量
void countFiles(char *path, char *extension, int *count) {
    struct dirent *entry;
    DIR *dp = opendir(path); // 打开目录

    if (dp == NULL) {
        perror("opendir");
        exit(EXIT_FAILURE);
    }

    // 遍历目录下的所有内容
    while ((entry = readdir(dp))) {
        if (entry->d_type == DT_DIR) {
            // 跳过当前目录和父目录
            if (strcmp(entry->d_name, ".") != 0 && strcmp(entry->d_name, "..") !=
0) {
                char new_path[1024];
                snprintf(new_path, sizeof(new_path), "%s/%s", path,
entry->d_name);
                countFiles(new_path, extension, count); // 递归进入子目录
            }
        } else {
            char *file_extension = strrchr(entry->d_name, '.'); // 获取文件扩展
名
            if (file_extension != NULL && strcmp(file_extension + 1, extension) ==
```

```

0) {
    (*count)++;
}
}

closedir(dp); // 关闭目录
}

int main(int argc, char *argv[]) {
    if (argc != 4) {
        printf("Usage: %s -t <extension> <directory>\n", argv[0]);
        return EXIT_FAILURE;
    }

    if (strcmp(argv[1], "-t") != 0) {
        printf("Invalid option. Use -t to specify the file extension.\n");
        return EXIT_FAILURE;
    }

    char *extension = argv[2];
    char *directory = argv[3];

    struct stat path_stat;
    stat(directory, &path_stat); // 获取目录信息

    if (!S_ISDIR(path_stat.st_mode)) {
        printf("Error: %s 不是目录.\n", directory);
        return EXIT_FAILURE;
    }

    int count = 0;
    countFiles(directory, extension, &count); // 开始统计文件数量

    printf("目录 %s 下扩展名为 .%s 的文件数量为: %d\n", directory, extension,
count);

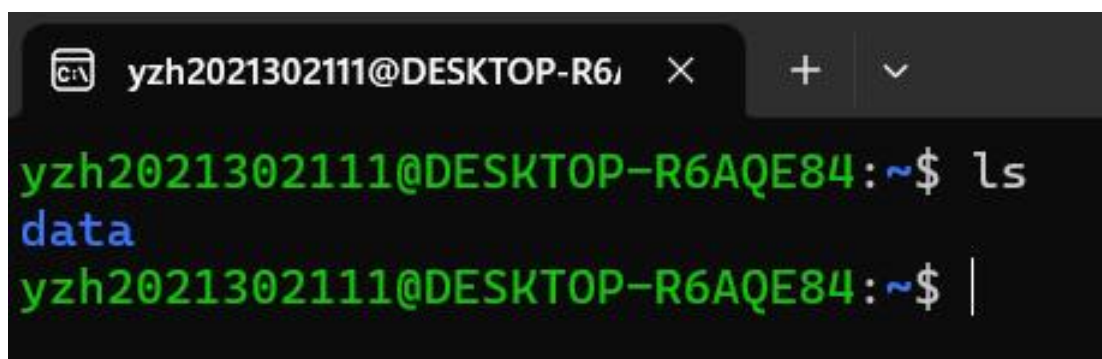
    return EXIT_SUCCESS;
}

```

二、代码程序运行全过程：

Step1

首先启动 ubuntu，在 home/yzh2021302111 用户目录下查询文件

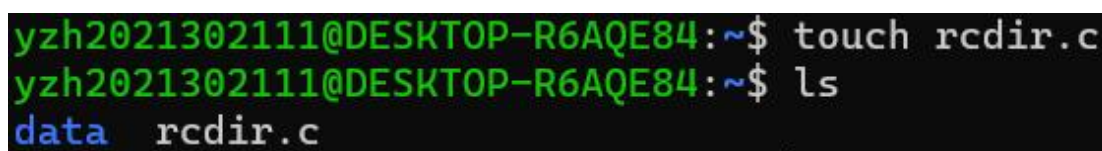


```
yzh2021302111@DESKTOP-R6AQE84:~$ ls
data
yzh2021302111@DESKTOP-R6AQE84:~$ |
```

观察到只有名为 data 的文件夹存在，并没有其他文件

Step2

使用 touch 命令在当前目录下创建名为 rcdirc.c 的 c 文件，可通过 ls 查询目录检查是否创建成功



```
yzh2021302111@DESKTOP-R6AQE84:~$ touch rcdirc.c
yzh2021302111@DESKTOP-R6AQE84:~$ ls
data  rcdirc.c
```

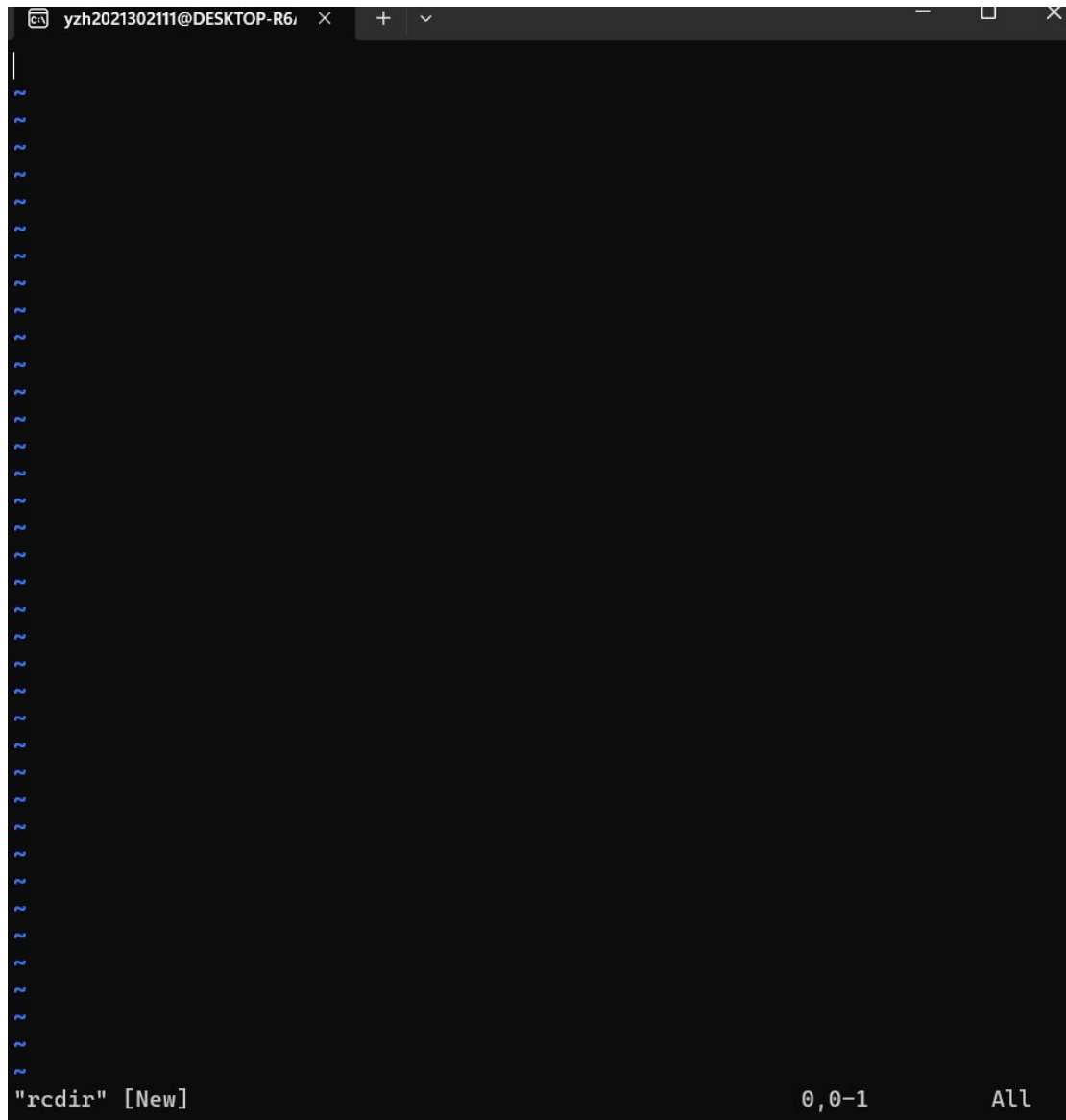
Step3

创建后即可通过 vim 命令对 c 文件进行编辑，输入命令并按下回车



```
yzh2021302111@DESKTOP-R6AQE84:~$ vim rcdirc.c
```

出现以下界面，此界面可通过键盘 ESC 控制是否进入编辑模式



接下来进入编辑模式写入 c 语言代码

```
        (*count)++;
    }
}

closedir(dp); // 关闭目录
}

int main(int argc, char *argv[])
{
    if (argc != 4) {
        printf("Usage: %s -t <extension> <directory>\n", argv[0]);
        return EXIT_FAILURE;
    }

    if (strcmp(argv[1], "-t") != 0) {
        printf("Invalid option. Use -t to specify the file extension.\n");
        return EXIT_FAILURE;
    }

    char *extension = argv[2];
    char *directory = argv[3];

    struct stat path_stat;
    stat(directory, &path_stat); // 获取目录信息

    if (!S_ISDIR(path_stat.st_mode)) {
        printf("Error: %s 不是目录.\n", directory);
        return EXIT_FAILURE;
    }

    int count = 0;
    countFiles(directory, extension, &count); // 开始统计文件数量

    printf("目录 %s 下扩展名为 .%s 的文件数量为: %d\n", directory, extension, count);

    return EXIT_SUCCESS;
}
```

65,1 Bot

编辑完成后退出编辑模式，并输入指令 `:wq` 执行写入保存并退出文件的功能，即可回到进入 vim 之前的窗口

```
yzh2021302111@DESKTOP-R6A: ~$ ls
data
yzh2021302111@DESKTOP-R6A: ~$ touch rcdire.c
yzh2021302111@DESKTOP-R6A: ~$ ls
data  rcdire.c
yzh2021302111@DESKTOP-R6A: ~$ vim rcdire
yzh2021302111@DESKTOP-R6A: ~$ |
```

Step4 (debug)

此时 rcdirc.c 文件中已存在代码，需要去编译，ubuntu 中的 gcc 编译器已被我成功下载并适配版本

22.04 ubuntu 很可能运行 gcc 编译时出现无法识别头文件，需要换源，我在上周调试程序的时候发现了这一问题，最终将源换成了 22.04 清华源，编译 rcdirc.c 的问题得以解决

以下是当时换源和出错时保留的部分截图：

首先下载好 gcc 后执行编译找不到头文件的 error 情况

```
yzh2021302111@DESKTOP-R6AQE84:~$ gcc --version
gcc (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

yzh2021302111@DESKTOP-R6AQE84:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.1 LTS
Release:        22.04
Codename:       jammy

yzh2021302111@DESKTOP-R6AQE84:~$ ls
ctest.c  rcdirc.c

yzh2021302111@DESKTOP-R6AQE84:~$ gcc ctest.c
ctest.c:1:10: fatal error: stdio.h: No such file or directory
#include <stdio.h>
        ^~~~~~
compilation terminated.
yzh2021302111@DESKTOP-R6AQE84:~$
```

其次是查询博文后想要换源，刚换完 22.04 清华源后尝试执行程序并成功的截图

```
Setting up libstdc++-11-dev:amd64 (11.4.0-1ubuntu1~22.04) ...
Setting up libc-devtools (2.35-0ubuntu3.3) ...
Setting up g++-11 (11.4.0-1ubuntu1~22.04) ...
Setting up g++ (4:11.2.0-1ubuntu1) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.9ubuntu3) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
yzh2021302111@DESKTOP-R6AQE84:~$ gcc --version
gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0
Copyright (C) 2021 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

yzh2021302111@DESKTOP-R6AQE84:~$ gcc ctest.c
yzh2021302111@DESKTOP-R6AQE84:~$ ls
a.out  ctest.c  rcdirc.c

yzh2021302111@DESKTOP-R6AQE84:~$ ./a.out
hello world!

yzh2021302111@DESKTOP-R6AQE84:~$ gcc rcdirc.c -o rcdirc
yzh2021302111@DESKTOP-R6AQE84:~$ ls
a.out  ctest.c  rcdirc  rcdirc.c

yzh2021302111@DESKTOP-R6AQE84:~$ ./rcdirc
Usage: ./rcdirc -t <file_extension> <directory>
yzh2021302111@DESKTOP-R6AQE84:~$ vim rcdirc.c
yzh2021302111@DESKTOP-R6AQE84:~$ ./rcdirc -t c /home/yzh2021302111
```

Step5 (debug)

Gcc 编译成功后，就可以用 ./ 命令执行 out 文件，并用代码中指定的指令形式执行递归计数查询文件类型数目 gcc rcdir -o rcdir 命令将编译结果 out 文件命名为 rcdir

为了方便计数，我们可以先创建几个文件

```
yzh2021302111@DESKTOP-R6AQE84:~$ ls
exp1.doc exp1.txt exp2.txt rcdir
yzh2021302111@DESKTOP-R6AQE84:~$ ./rcdir -t txt /home/yzh2021302111
Number of .txt files in /home/yzh2021302111: 2
yzh2021302111@DESKTOP-R6AQE84:~$ ./rcdir -t doc /home/yzh2021302111
Number of .doc files in /home/yzh2021302111: 1
yzh2021302111@DESKTOP-R6AQE84:~$ ./rcdir -t c /home/yzh2021302111
Number of .c files in /home/yzh2021302111: 4
```

按照指令即可得到文件数量，但是我发现检测 c 文件的数目并不对，为此我去文件夹下查看了 ubuntu 的 home/yzh2021302111 文件夹下，发现并非命令行显示的只有 data 文件夹和 rcdir 相关文件，而是有很多配置文件

Linux > Ubuntu > home > yzh2021302111					在 yzh20213021... 🔍	
名称	修改日期	类型	大小			
📁 .local	2023/4/2 23:14	文件夹				
📁 data	2023/9/14 14:15	文件夹				
📄 .bash_history	2023/9/20 22:27	BASH_HISTORY ...	6 KB			
📄 .bash_logout	2023/3/18 9:06	Bash Logout 源...	1 KB			
📄 .bashrc	2023/3/18 9:06	Bash RC 源文件	4 KB			
📄 .motd_shown	2023/9/21 8:41	MOTD_SHOWN ...	0 KB			
📄 .profile	2023/3/18 9:06	Profile 源文件	1 KB			
📄 .sudo_as_admin_successful	2023/4/2 18:35	SUDO_AS_ADMI...	0 KB			
📄 .viminfo	2023/9/21 9:48	VIMINFO 文件	7 KB			

这影响到了程序对于文件数目的检测，并且直接在用户根目录下运行程序并不是一个很好的选择，这里我决定在新建的 program1 文件夹下创建并运行程序，在子文件夹 test1 中对文件数目进行计数，检验程序的正确性。

Step6

按照 step5 改良方案创建子文件夹并编译

```
yzh2021302111@DESKTOP-R6AQE84:~$ mkdir program1
yzh2021302111@DESKTOP-R6AQE84:~$ ls
data  program1
yzh2021302111@DESKTOP-R6AQE84:~$ cd program1
yzh2021302111@DESKTOP-R6AQE84:~/program1$ touch test.c
yzh2021302111@DESKTOP-R6AQE84:~/program1$ vim test.c
yzh2021302111@DESKTOP-R6AQE84:~/program1$ gcc test.c -o test
yzh2021302111@DESKTOP-R6AQE84:~/program1$ ls
test  test.c
yzh2021302111@DESKTOP-R6AQE84:~/program1$ ./test
Usage: ./test -t <extension> <directory>
yzh2021302111@DESKTOP-R6AQE84:~/program1$ |
```

创建统计数目的文件夹

```
yzh2021302111@DESKTOP-R6AQE84:~/program1$ mkdir test1
yzh2021302111@DESKTOP-R6AQE84:~/program1$ cd test1
yzh2021302111@DESKTOP-R6AQE84:~/program1/test1$ touch a
yzh2021302111@DESKTOP-R6AQE84:~/program1/test1$ touch a.c
yzh2021302111@DESKTOP-R6AQE84:~/program1/test1$ touch b.c
yzh2021302111@DESKTOP-R6AQE84:~/program1/test1$ ls
a  a.c  b.c
yzh2021302111@DESKTOP-R6AQE84:~/program1/test1$ cd ..
```

运行成功:

```
yzh2021302111@DESKTOP-R6AQE84:~/program1$ ./test -t c /home/yzh2021302111/program1/test1
目录 /home/yzh2021302111/program1/test1 下扩展名为 .c 的文件数量为: 2
yzh2021302111@DESKTOP-R6AQE84:~/program1$ |
```

至此解决所有问题，程序运行结束