

课程名称	软硬件接口程序设计	课程编号	A2130731
实验地点	重庆邮电大学综合实验楼	实验时间	2022. 11. 10
实验名称	实验六 系统调用		

一、 实验目的

练习并理解内核级文件操作，如何创建并管理文件以及如何阅读文件目录。

二、 实验内容

任务 1:

1. 通过给出的例子，对系统调用函数 `open` 进行测试。
2. 通过给出的例子，对系统调用函数 `close` 进行测试。
3. 通过给出的例子，对系统调用函数 `read` 进行测试。
4. 通过给出的例子，对系统调用函数 `write` 进行测试。

任务 2: 编写一个程序，随机生成介于 0 和 99 之间的 100 个数字（0 和 99 包括在内），计算数字的总和，打印总和，并使用 `open`, `close`, `write` 系统调用函数将数字写入新文件。具体要求有以下几点:

(1) 输入: 用户无需输入数字，使用 `rand` 和 `srand` 函数来生成数字。

(2) 输出: 打印总和，并创建一个名为 `numbers.xxxx` 的文件，`xxxx` 为文件中的数字总和。

任务 3: 编写一个程序，搜索当前工作目录中名称与“`numbers.xxxx`”匹配的文件。对于每个匹配的文件，打开该文件，读取该文件，计算文件中的整数和，并打印文件名和整数和。您可以假设文件将包含 100 个整数。

三、 实验过程原始记录(数据、图表等)

任务 1:

1:

```
root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab5# ./task1_1
fd = 3
```

图 1 运行程序



图 2 生成文件

2:

```
root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab5# ./task1_2
fd2 = 3
```

图 3 运行程序

3:

```
root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab5# ./task1_3
called read(3, c, 10)returned and 0 bytes were read.
Those bytes are as follows:
```

图 4 运行程序

4:

```
root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab5# ./task1_4
called write(3, "hello world\n", 12)returned and 12
```

图 5 运行程序

```
root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab5# ./task1_3
called read(3, c, 10)returned and 12 bytes were read.
Those bytes are as follows: hello world
```

图 6 再次运行 3 中程序

任务 2:

```
root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab5# ./task2
sum:4852
root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab5# ls
baz.txt  foo.txt  numbers.4852.txt  task1_1  task1_1.c  tas
```

图 7 生成文件

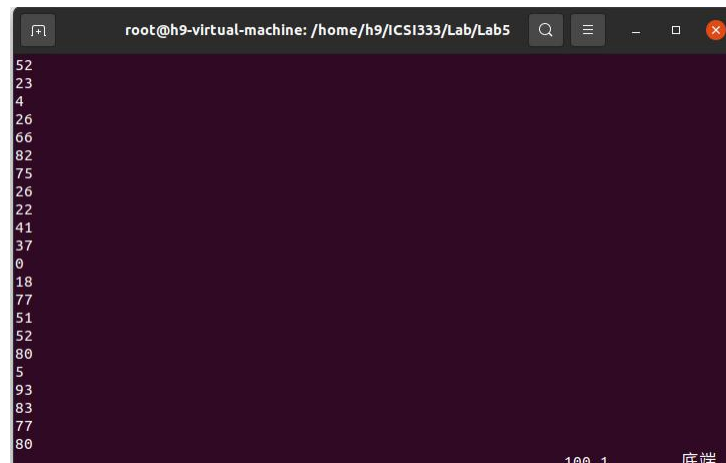


图 8 文件内容

任务 3:

```
root@h9-virtual-machine:/home/h9/ICSI333/Lab/Lab5# ./task3
Directory Content List!
Get the file:numbers.4852.txt
sum:4852
```

图 9 运行程序

四、实验结果及分析

实验结果可见原始数据记录。相应分析如下:

Task1:

1. 通过 `open` 函数打开文件，打开成功则返回文件标识符。因 0、1、2 已被标准输入、输出、错误占用，故标识符为 3。
2. 首先通过 `open` 打开 `foo.txt`，再根据标识符，通过 `close` 关闭文件。再次通过 `open` 打开 `baz.txt` 时，因已完成对 `foo` 的关闭，故返回标识符为 3。
3. 使用 `open` 打开 `foo.txt` 文件，再调用 `read` 对 `foo` 内容进行读取，并写入到创建的字符串。因此时 `foo.txt` 内容为空，故读取的字节为 0。
4. 使用 `open` 打开 `foo.txt`，使用 `write` 将 “hello world\n” 写入 `foo`，并打印展示写入的完成状态。此时再次运行 3 中的程序，对 `foo` 进行读取，可看到读出内容不再为空，而与写入的内容相对应。

Task2: 使用 `rand` 和 `srand`，同时利用取余，随机生成 100 个 0-99 的数字，将其写入数组，并计算总和。使用 `open`，根据计算的总和，创建符合要求格式的文件，再通过 `write` 将数字写入文件当中。

Task3: 首先使用 `opendir` 打开当前目录。打开成功，则通过 `while` 循环，通过 `readdir` 读取目录内容，当内容不为空时，将文件名和 `numbers` 进行匹配。若匹配成功则打印文件名，并利用 `fopen` 对数字读取，并计算数字和。计算完成和对和进行打印展示。

五、实验心得体会

- 1.掌握了系统调用中的 `open`, `close`, `read`, `write`。
- 2.掌握了系统调用中设计到的 `fd`, `flag`, `mode_t`。
- 3 掌握了如何通过系统调用或内核操作，对文件、目录进行相关操作。
- 4.`write` 写入限制类型 `void* buf`，不利于按行写入整数类型，未找到较好解决方案，利用 `int` 与 `char*`转换完成了任务。