实验五：UNIX V6++文件系统

# 实验目的

结合课程所学知识，通过在 UNIX V6++实验环境中编写使用文件管理相关的系统调用或库函数的应用程序，进一步了解 UNIX 文件管理的工作过程。

# 实验要求

（1）（1 分）给代码 1 添加注释，详细解释每一步操作的含义。

（2）（1 分）按上述过程，分别编辑、编译并运行 filetest 程序，截图展示程序的输出结果。

（3）（1 分）分别解释 creat 和 open 两个库函数中第二个参数的意义。

（4）（1 分）解释程序在作出图 8 的修改后，输出变为图 9 的原因。

# 实验准备

在 UNIX V6++的/lib/file.c 文件中了解 UNIX V6++支持的所有和文件管理有关的库函数。

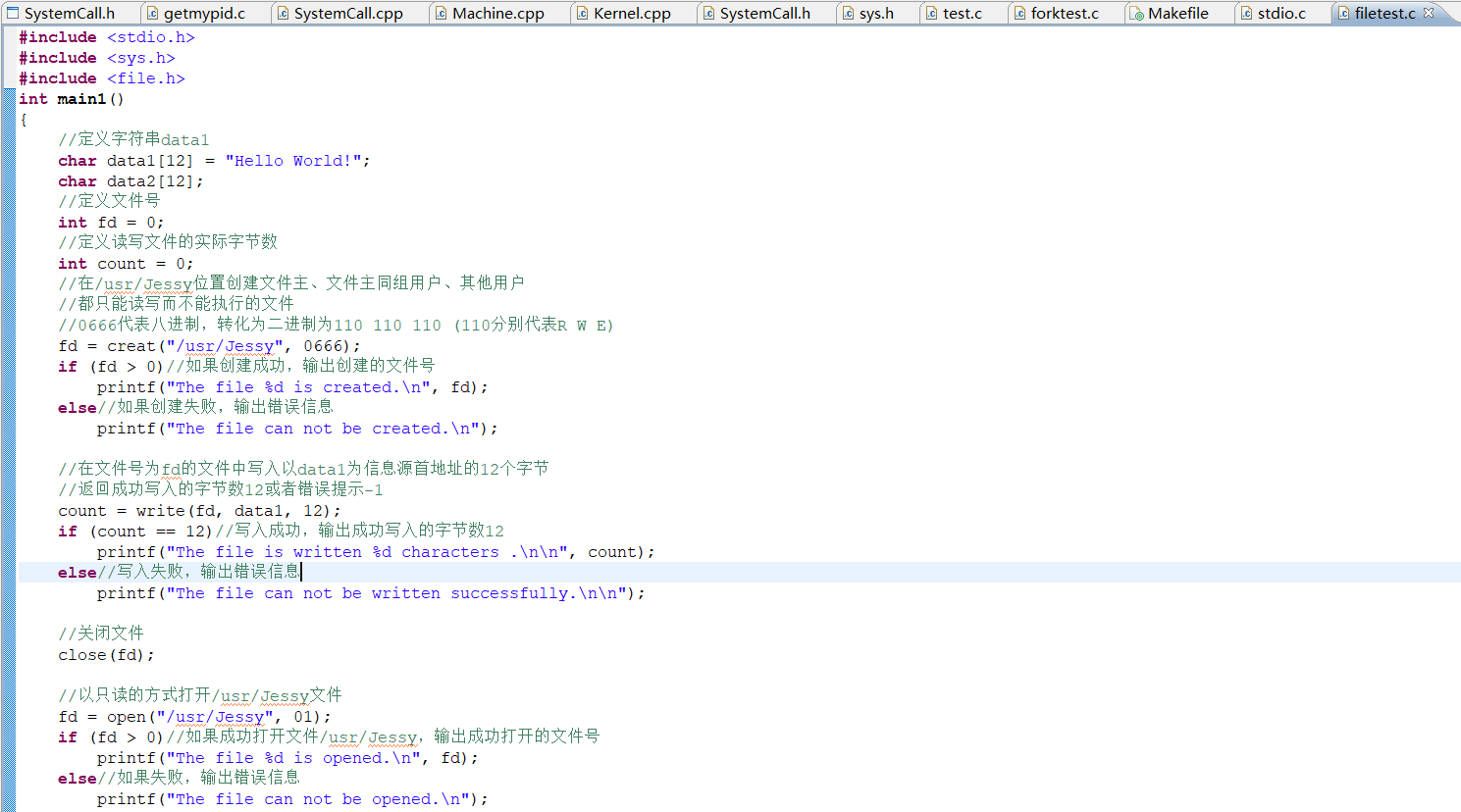
# 实验内容

## 一、给代码 1 添加注释，详细解释每一步操作的含义

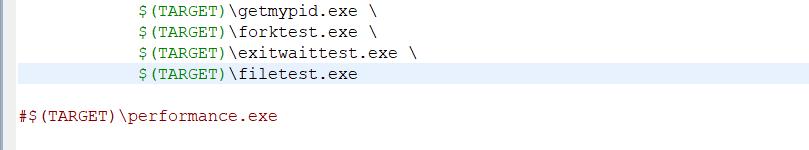
|  |
| --- |
| #include <stdio.h>  #include <sys.h>  #include <file.h>  int main1**()**  **{**  //定义字符串data1  char data1**[**12**]** **=** "Hello World!"**;**  char data2**[**12**];**  //定义文件号  int fd **=** 0**;**  //定义读写文件的实际字节数  int count **=** 0**;**  //在/usr/Jessy位置创建文件主、文件主同组用户、其他用户  //都只能读写而不能执行的文件  //0666代表八进制，转化为二进制为110 110 110 (110分别代表R W E)  fd **=** creat**(**"/usr/Jessy"**,** 0666**);**  **if** **(**fd **>** 0**)**//如果创建成功，输出创建的文件号  printf**(**"The file %d is created.\n"**,** fd**);**  **else**//如果创建失败，输出错误信息  printf**(**"The file can not be created.\n"**);**  //在文件号为fd的文件中写入以data1为信息源首地址的12个字节  //返回成功写入的字节数12或者错误提示-1  count **=** write**(**fd**,** data1**,** 12**);**  **if** **(**count **==** 12**)**//写入成功，输出成功写入的字节数12  printf**(**"The file is written %d characters .\n\n"**,** count**);**  **else**//写入失败，输出错误信息  printf**(**"The file can not be written successfully.\n\n"**);**  //关闭文件  close**(**fd**);**  //以只读的方式打开/usr/Jessy文件  fd **=** open**(**"/usr/Jessy"**,** 01**);**  **if** **(**fd **>** 0**)**//如果成功打开文件/usr/Jessy，输出成功打开的文件号  printf**(**"The file %d is opened.\n"**,** fd**);**  **else**//如果失败，输出错误信息  printf**(**"The file can not be opened.\n"**);**  //将文件号为fd文件的前12个字节写入以data2为目的源首地址的内存  //返回成功读出的字节数12或者错误提示-1  count **=** read**(**fd**,** data2**,** 12**);**  //输出读出文件的字节数和字符串  printf**(**"%d characters are read from file %d: %s.\n"**,** count**,** fd**,** data2**);**  **return** 1**;**  **}** |

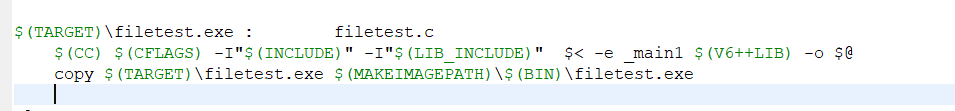
## 二、按上述过程，分别编辑、编译并运行 filetest 程序，截图展示程序的输出结果。

编辑filetest程序：

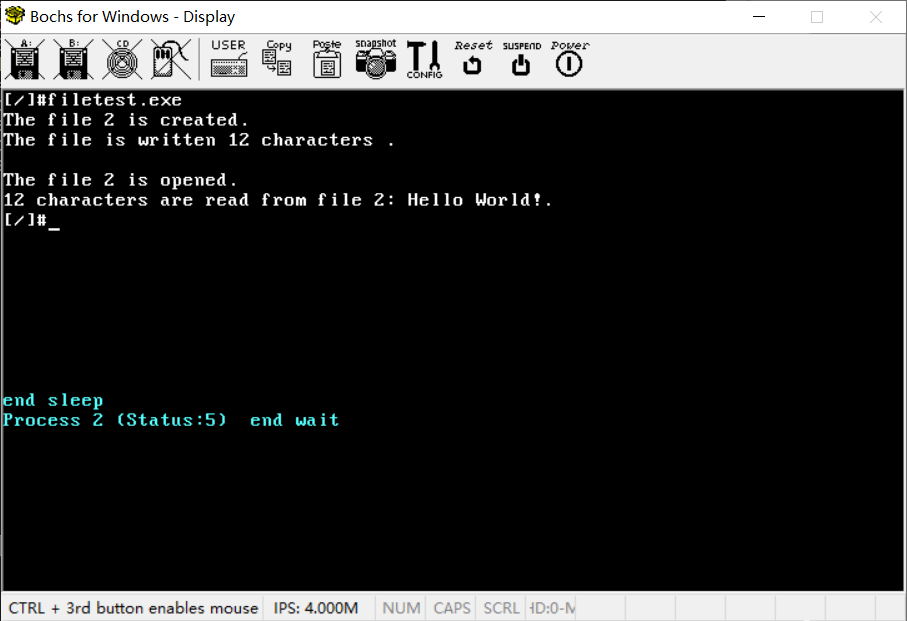


编译filetest程序：





运行filetest程序：



## 三、分别解释 creat 和 open 两个库函数中第二个参数的意义。

|  |
| --- |
| int creat**(**char**\*** pathname**,** unsigned int mode**);**  int open**(**char**\*** pathname**,** unsigned int mode**);** |

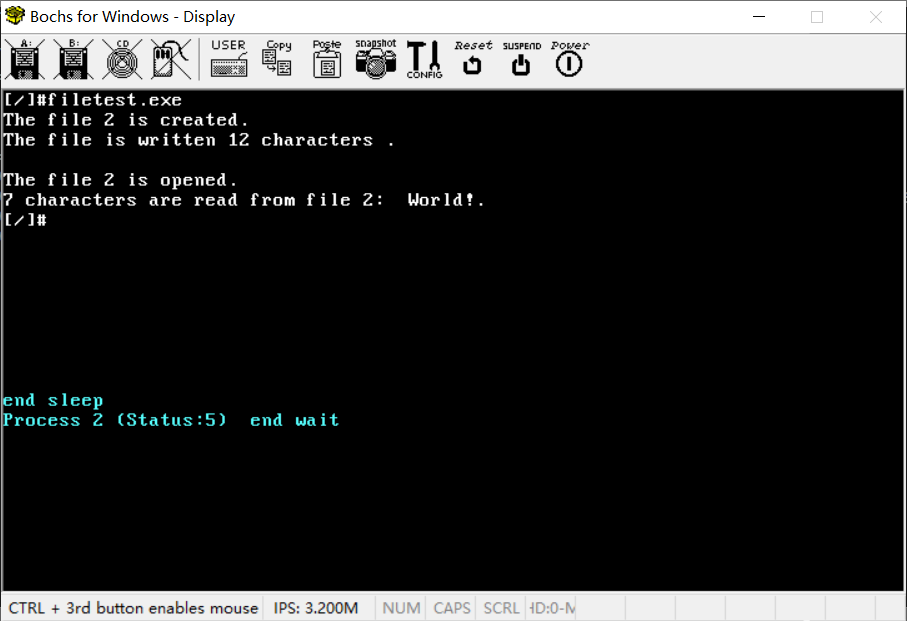
creat库函数第二个参数是创建模式，需要指定文件主，同组用户，其他用户的工作方式，二进制表示形式最低9位代表的含义如下：

|  |  |  |
| --- | --- | --- |
| 文件主 | 文件主同组用户 | 其他用户 |
| R W E | R W E | R W E |

其中R为1代表可读，0代表不可读；W为1代表可写，0代表不可写；E为1代表可执行，0代表不可执行。

open库函数第二个参数是打开模式，即打开文件采用读、写还是读写的方式，其中mode = 0代表写模式，mode = 1代表读模式，mode = 2代表读写模式。

## 四、解释程序在作出图 8 的修改后，输出变为图 9 的原因。



程序作出修改后，仅仅在读取文件前增加了一行seek（fd，5，0）的处理。

seek函数定义：int seek（int fd，unsigned int offset，unsigned int ptrname），主要功能为改变文件读写位置，其中fd为打开的文件号，如果ptrname == 0，则读写位置设置为offset，如果ptrname == 1，则读写位置加offset（可正可负），如果ptrname == 2，则读写位置调整为文件长度加offset，如果ptrname > 2，为3～5，意义同0~2，但长度单位从一个字节变为512个字节。

所以seek（fd，5，0）函数的功能是将读写位置设置为5（字节），即fd文件“Hello World!”中的空格字符，此时fd文件中不足12个字节，因此从空格处读取到最后，即为“ World!”，总计7个字符。