**实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生姓名 | 黄东 | 学号 | 16281255 | 指导老师 | 何永忠 |
| 实验地点 | 九教401 | 实验时间 | 2019.3.5 | 班级 | 安全1601 |

1. **实验项目名称：**

实验一: 操作系统初步

1. **实验环境**

Deepin15.6桌面版

1. **实验内容：**
2. 系统调用实验
3. 并发实验
4. 内存分配实验
5. 共享问题
6. **实验过程：**
7. **系统调用实验**

要求：

1. 参考下列网址中的程序。阅读分别运行用API接口函数getpid()直接调用和汇编中断调用两种方式调用Linux操作系统的同一个系统调用getpid的程序(请问getpid的系统调用号是多少？linux系统调用的中断向量号是多少？)。
2. 上机完成习题1.13。
3. 阅读pintos操作系统源代码，画出系统调用实现的流程图。



图1.1 getpid.c文件

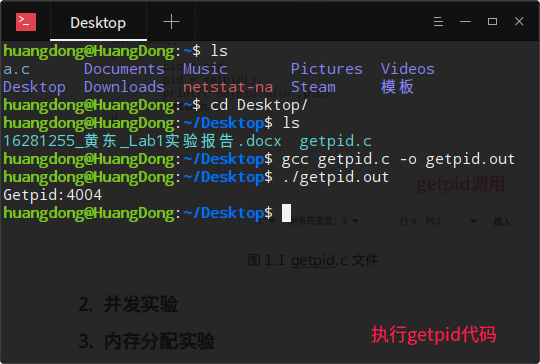


图1.2 执行getpid.c文件



图1.3 汇编getpid代码

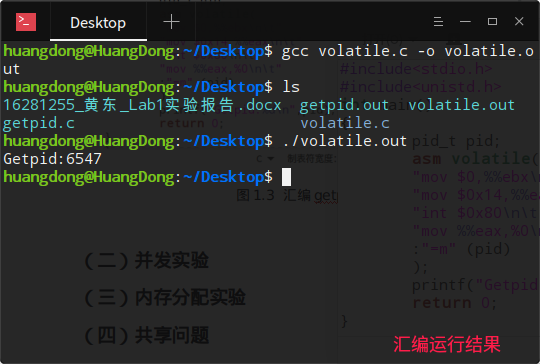


图1.4 执行结果

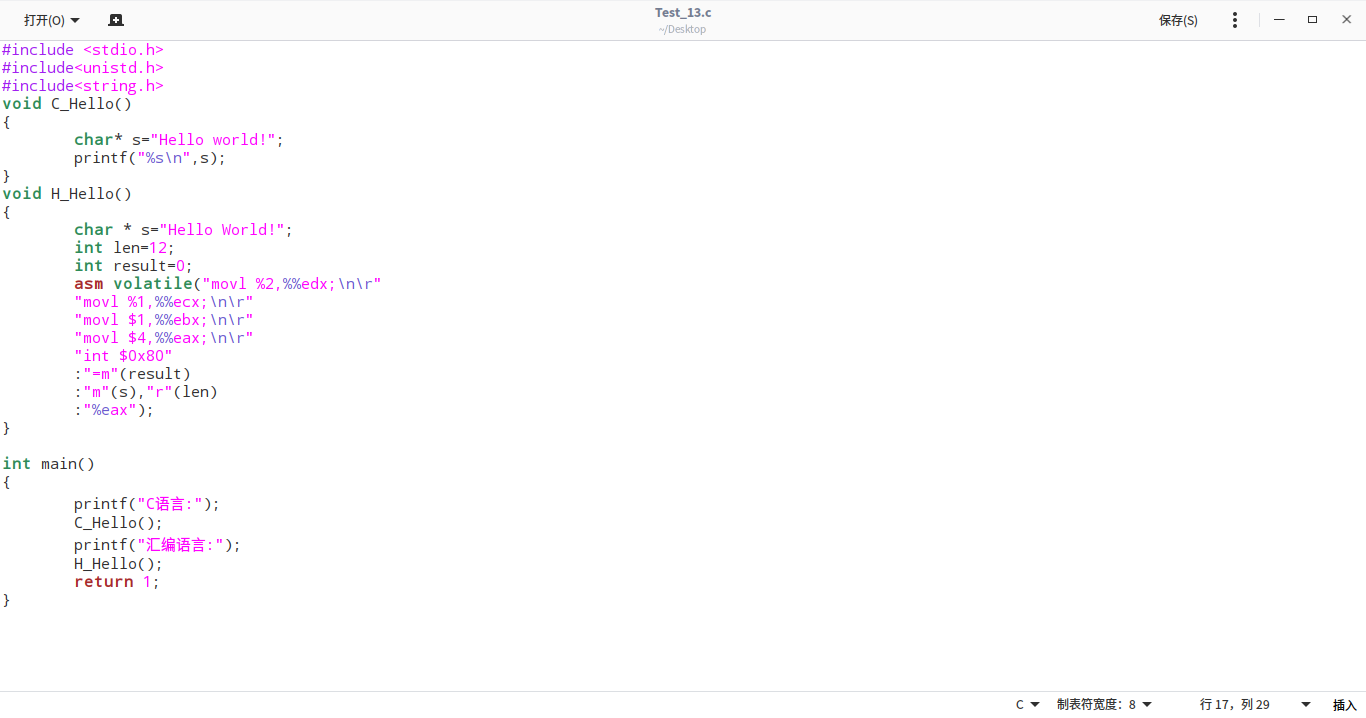


图1.5 习题13代码

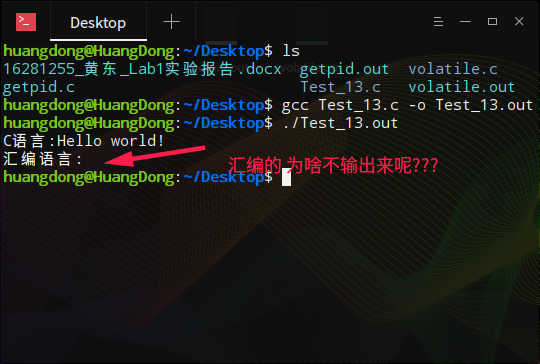


图1.6 执行习题13代码

**注释**:汇编代码在此参照同学,但是只看见皮毛,未学到精髓,无法更改后完美执行!

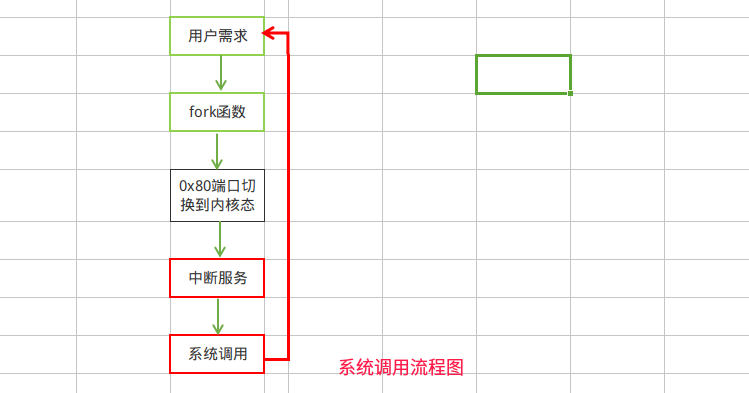


图1.7 系统调用流程

注释:Linux中没有Visio程序,所以借助WPS绘制流程图~

分析:

1. getpid的系统调用号是20；linux系统调用的中断向量号是0x80。

1. 习题13执行如上图1.6所示,然而实在不会汇编执行这个,后期加以补习
2. 系统调用流程见图1.7
3. **并发实验**

要求：

1. 编译运行该程序（cpu.c），观察输出结果，说明程序功能。

(编译命令： gcc -o cpu cpu.c –Wall)（执行命令：./cpu）

1. 再次按下面的运行并观察结果：执行命令：./cpu A & ; ./cpu B & ; ./cpu C & ; ./cpu D &程序cpu运行了几次？他们运行的顺序有何特点和规律？请结合操作系统的特征进行解释。

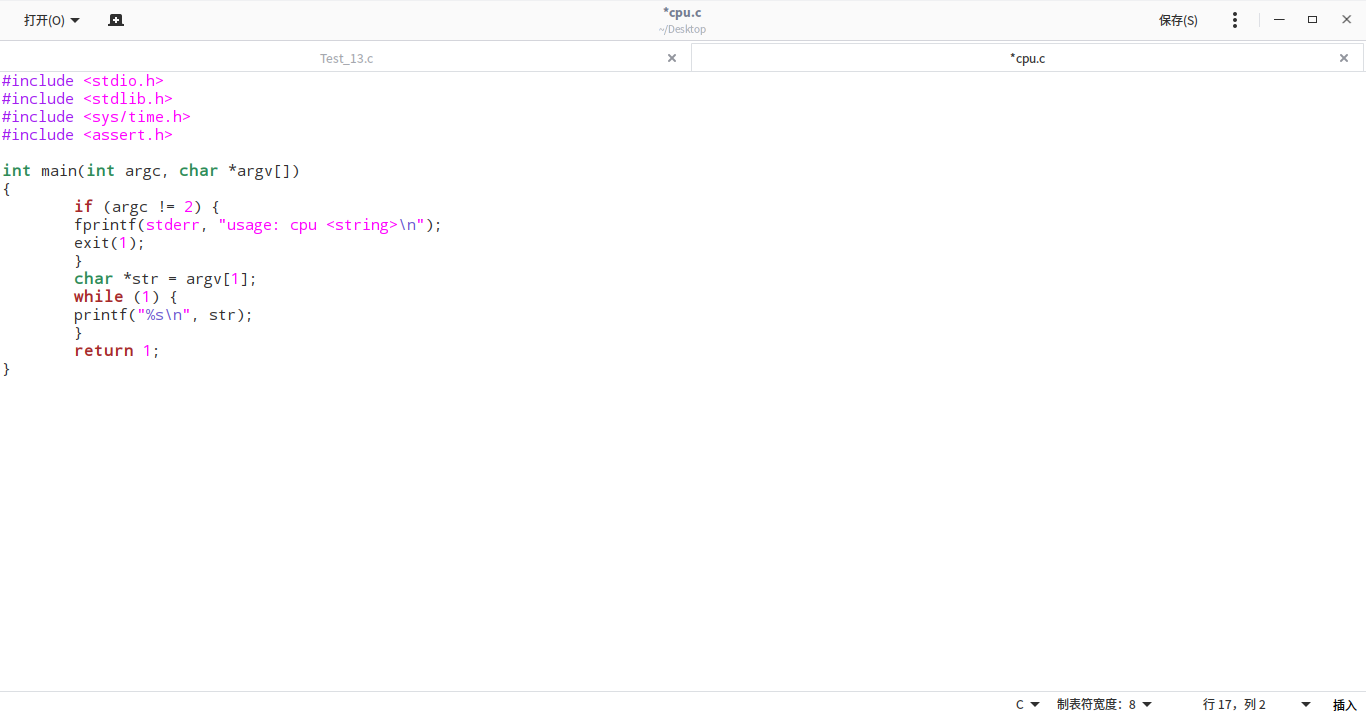


图2.1 编写cpuc文件



图2.2 执行cpu.c文件

分析:

实验代码中当运行指令”./cpu A”(A,B,C,D)时,会一直单一的输出A/B/C/D;当运行指令”./cpu A & ./cpu B & ./cpu C & ./cpu D”时,则会随机出现A B C D,我分析是系统多个CPU会被随机或者其他方式分配给程序使用.

1. **内存分配实验**

要求：

1、阅读并编译运行该程序(mem.c)，观察输出结果，说明程序功能。(命令： gcc -o mem mem.c –Wall)

2、再次按下面的命令运行并观察结果。两个分别运行的程序分配的内存地址是否相同？是否共享同一块物理内存区域？为什么？命令：./mem &;

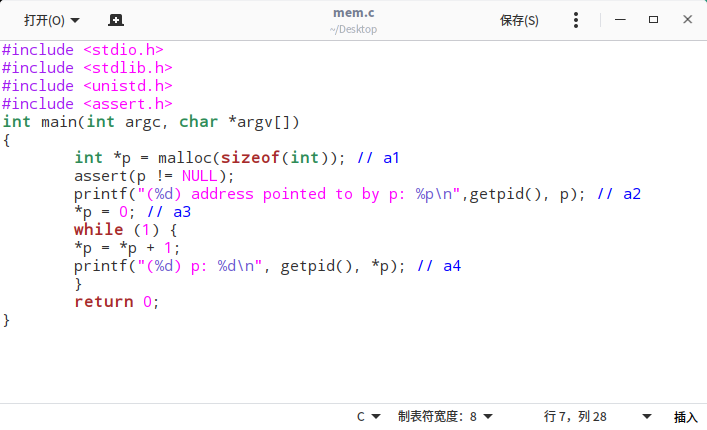


图3.1 编写mem.c文件

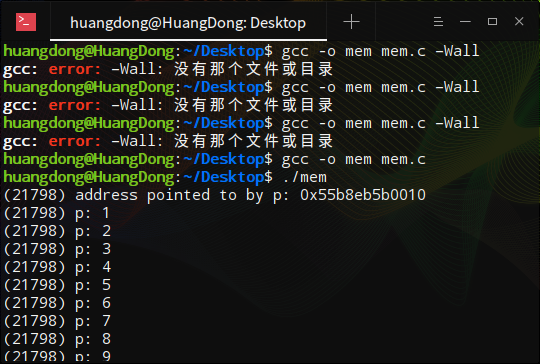


图3.2 执行mem文件

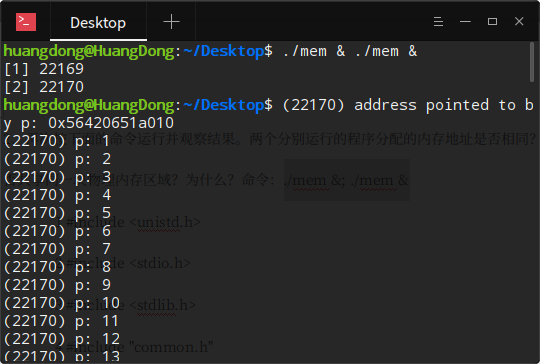


图3.3 执行mem文件

分析:

程序功能是创建一个进程后用指针p指向并输出该进程的内存空间地址，然后顺序输出进程号和内存地址。由结果知两个程序分配的内存地址不相同。

1. **共享问题**

要求：

1. 阅读并编译运行该程序，观察输出结果，说明程序功能。（编译命令：gcc -o thread thread.c -Wall –pthread）（执行命令1：./thread 1000）
2. 尝试其他输入参数并执行，并总结执行结果的有何规律？你能尝试解释它吗？（例如执行命令2：./thread 100000）（或者其他参数。）
3. 提示：哪些变量是各个线程共享的，线程并发执行时访问共享变量会不会导致意想不到的问题。



图4.1 编写thread.c文件



图4.2 执行thread文件

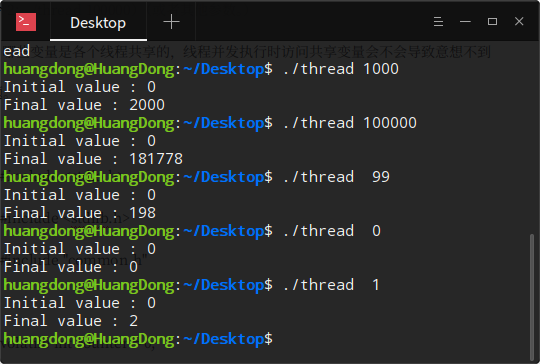


图4.3 thread数据测试

分析:

这个程序会输出两个线程进行前的结果，再将两个线程结果输出.后续尝试了很多组参数之后，发现最终结果都是两个线程结果。由于全局变量是共享的，所以两个线程访问同一个counter数据，在执行worker函数时，两个线程会将上一个线程的执行结果代入，所以直观看来程序输出结果为输出参数的2倍。但是结果有界限,达到某个值后便会到极限,所以达不到2倍关系.(后续参照其他同学的数据后才知道线程会发生冲突,这样就会产生脏数据,使其”2倍”关系发生微变化).

1. **总结反思：**

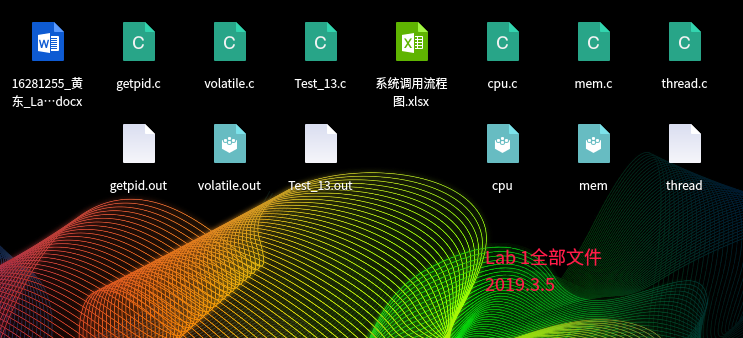


图5.1 实验文件展示

通过本次实验，我对操作系统给应用程序提供服务的方式–系统调用有所了解，理解系统调用与普通函数调用不同之处在于系统调用会在系统内核态中执行后再返回到用户态。而汇编代码中，系统调用号通过eax寄存器传递给内核，然后启动0x80端口后，规定返回值送eax寄存器。软中断就是进程-内核中的设备驱动程序，是一种需要内核为正在运行的进程去做一些事情（通常为I/O）的请求。

本次实验在C/汇编代码和Linux系统命令中也了解更多,首先是在代码抄写过程中总结到代码没看懂就会在运行中不能发现错误,例如:实验三和四中缺少头文件,就会导致调用的函数无法使用.再就是Linux命令中会有不同的执行方式,采取不同的命令会达到一样的效果,例如:编译运行.c文件时会有不同的处理方式.

最后,我把本次实验打包放在了Github上.