

# 实验4：混合编程与中断处理 - 2023.03.31

[Assignment 1 混合编程的基本思路](#)

[Assignment 2 使用C/C++编写内核](#)

[Assignment 3 中断的处理](#)

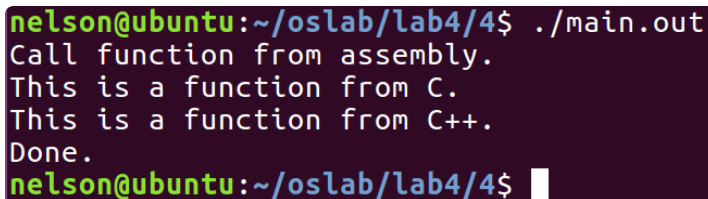
[Assignment 4 时钟中断的处理](#)

[提交要求](#)

## Assignment 1 混合编程的基本思路

复现指导书中“一个混合编程的例子”部分。要求：

1. 将原例子中最后一行的输出"Done"（参考下图）改为"Done by 你的学号 你的姓名首字母";



```
nelson@ubuntu:~/oslab/lab4/4$ ./main.out
Call function from assembly.
This is a function from C.
This is a function from C++.
Done.
nelson@ubuntu:~/oslab/lab4/4$
```

2. 结合具体的代码说明C代码调用汇编函数的语法和汇编代码调用C函数的语法。  
例如，结合关键代码说明 `global`、`extern` 关键字的作用，为什么C++的函数前需要加上 `extern "C"` 等，保存结果截图并说说你是怎么做的；
3. 学习make的使用，并用make来构建项目，保存结果截图并说说你是怎么做的。

## Assignment 2 使用C/C++编写内核

复现指导书中“内核的加载”部分，在进入 `setup_kernel` 函数后，将输出 `Hello World` 改为输出你的学号+姓名首字母，保存结果截图并说说你是怎么做的。

## Assignment 3 中断的处理

复现指导书中“初始化IDT”部分，你可以更改默认的中断处理函数为你编写的函数，然后触发之，结果截图并说说你是怎么做的。**要求：调用处理函数时输出个人特征信息。**

## Assignment 4 时钟中断的处理

复现指导书中“8259A编程——实时钟中断的处理”部分，要求：

仿照该章节中使用C语言来实现时钟中断的例子，利用 `C/C++`、`InterruptManager`、`STDIO` 和你自己封装的类来实现你的时钟中断处理过程，保存结果截图并说说你的思路和做法。

**注意：**不可以使用纯汇编的方式来实现。

## 提交要求

1. **截止日期：**2023年4月21日00: 00
2. **提交邮箱：**sysu\_os2023@163.com
3. **邮件主题、压缩包、报告文件命名：**实验4-学号-姓名
4. **内容要求：**请大家根据上述要求，完成Assignment 1~Assignment 4，按规范撰写实验报告，并提交.asm汇编源码/C/C++源码。实验结果截图放到实验报告中，截图中尽量体现个人信息。