

实验5：可变参数机制与内核线程 - 2023.04.14

- [实验5：可变参数机制与内核线程 - 2023.04.14](#)
 - [Assignment 1 printf的实现](#)
 - [Assignment 2 线程的实现](#)
 - [Assignment 3 时钟中断的处理](#)
 - [Assignment 4 调度算法的实现](#)
 - [提交要求](#)

Assignment 1 printf的实现

学习可变参数机制，然后实现printf，你可以在材料中的printf上进行改进，或者从头开始实现自己的printf函数。结果截图并说说你是怎么做的。

Assignment 2 线程的实现

自行设计PCB，可以添加更多的属性，如优先级等，然后根据你的PCB来实现线程，演示执行结果。

Assignment 3 时钟中断的处理

操作系统的线程能够并发执行的秘密在于我们需要中断线程的执行，保存当前线程的状态，然后调度下一个线程上处理机，最后使被调度上处理机的线程从之前被中断点处恢复执行。现在，同学们可以亲手揭开这个秘密。

编写若干个线程函数，使用gdb跟踪`c_time_interrupt_handler`、`asm_switch_thread`等函数，观察线程切换前后栈、寄存器、PC等变化，结合gdb、材料中“线程的调度”的内容来跟踪并说明下面两个过程。

- 一个新创建的线程是如何被调度然后开始执行的。
- 一个正在执行的线程是如何被中断然后被换下处理器的，以及换上处理机后又是如何从被中断点开始执行的。

通过上面这个练习，同学们应该能够进一步理解操作系统是如何实现线程的并发执行的。

Assignment 4 调度算法的实现

在材料中，我们已经学习了如何使用时间片轮转算法来实现线程调度。但线程调度算法不止一种，例如

- 先来先服务。
- 最短作业（进程）优先。
- 响应比最高者优先算法。
- 优先级调度算法。

- 多级反馈队列调度算法。

此外，我们的调度算法还可以是抢占式的。

现在，同学们需要将线程调度算法修改为上面提到的算法或者是同学们自己设计的算法。然后，同学们需要自行编写测试样例来呈现你的算法实现的正确性和基本逻辑。最后，将结果截图并说说你是怎么做的。

参考资料: <https://zhuanlan.zhihu.com/p/97071815>

Tips:

- 先来先服务最简单。
- 有些调度算法的实现可能需要用到中断。

提交要求

1. 截止日期: 2023年5月5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5-学号-姓名
4. 内容要求: 请大家根据上述要求, 学习实验网站上lab5的内容, 并完成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。