

进程线程 lab 说明文档

一、实验简介

本次 lab 的目标是创建进程运行环境,并能成功运行 user/hello 程序。请认真阅读 MIT JOS lab3 的实验说明文档(<https://pdos.csail.mit.edu/6.828/2018/labs/lab3/>),实现我们要求的代码并回答问题。大家需要在 JOS process_lab 分支基础上(在 fdus2021 文件夹中, git checkout process_lab)完成实验。

为了和课程授课顺序保持一致,我们操作系统课程的 lab 将内存管理部分安排在进程线程部分之后。因此本次实验不要求实现与内存分配及管理相关的 Exercise,但提前了解 JOS 的内存布局及内存管理相关函数的功能对理解整个 kernel 很有帮助。(如果时间充裕建议先完成内存管理部分(JOS lab2)的 exercise 再开始本实验,这样也可以减少期末季的压力)。

二、实验要求

A. 代码实现

- (1) Exercise2、Exercise4、Exercise 5、Exercise 6、Exercise7、Exercise 8

其中 Exercise2 中, env_setup_vm、region_alloc、load_icode 三个函数 TA 已经实现,但他们是实现其他函数的基础,所以仍然需要了解它们的功能。

(注: 全部正确实现后, make grade 能拿到 100 分)

- (2) 优化 kern/trapentry.S 和 kern/trap.c 中的代码,通过重构,减少 idt_init () 中重复出现的代码,使其同样的代码写一遍就好。并回答,这样做在这一系统中的目的?

请独立完成这部分内容,如果参考了网上的回答或者同学的代码,请在实验报告的相应位置说明。

B. 问题回答

- 1、回答 exercise4, exercise6 后面的四个问题。
- 2、请详细说出系统是如何实现从用户态到内核态的转换,什么时候切换,以及详细说出此系统的中断处理流程
- 3、IDT 和 GDT 存储的信息分别是什么?系统是如何初始化它们的?
- 4、_alltraps 的具体作用是什么?它做的事情在后面哪里用到了
- 5、用户环境是什么?此实验使用哪种数据结构来存储这些信息?此数据结构具体形式是什么样子的

三、建议阅读 (报告中不需要说明)

- (1) fdus2021 目录中 process_lab 分支相关代码
- (2) Exercise1 该部分实现已给出,请阅读并了解该部分功能
- (3) Exercise3 请至少阅读 Chapter 9 Exceptions and Interrupts

四、实验提交

[1] 实验代码文件夹(提交前请 make clean)

[2] 实验报告 (中文即可),需要包含如下内容:

- (1) 每个 exercise 代码实现思路,相关代码截图,最终运行 make grade 的截图
- (2) kern/trapentry.S 和 kern/trap.c 中的代码优化的实现思路,运行成功后的截图
- (3) 每个问题的回答
- (4) 实验反馈(可选,不算分),如 process_lab 分支中 TA 给出的实现中可能存在的 bug,

实验中遇到的未解决问题以及对之后 lab 的建议等。

[3] 实验截止日期: 2021.10.25 (未按时提交会适当扣分)

[4] 提交方式: 坚果云, 链接:

<https://workspace.jianguoyun.com/inbox/collect/231302c4134b48e2ad0c3f69581ba2ef/submit>

请将[1]、[2] 放在一个文件夹中, 命名为: 学号_姓名_lab2_任课老师名字, 并压缩为 zip 文件。例: 18307130xxx_小明_lab2_王飞.zip。

五、评分标准

TA 会根据代码、问题回答情况及实验报告给出最终成绩。总分: 100 分, 各部分占比如下:

- (1) mit exercise: 50 分 (根据代码和实现思路给分)
- (2) kern/trapentry.S 和 kern/trap.c 中的代码优化: 15 分 (根据代码和实现思路给分)
- (3) 问题回答: 30 (一个问题 6 分)
- (4) 报告格式、可读性: 5