

加分项：页替换算法和多级页表

中国科学院大学 操作系统研讨课

2017.11.29

1. 介绍

本次任务的加分项有两个：页替换算法和多级页表。

1.1. 需要了解的部分

- 常用页面替换算法
- 多级页表结构

2. 初始代码

2.1. 文件介绍

- Makefile: 编译文件。
- bootblock.s: 内核启动程序，请使用作业一中自己写的代码。
- Createimage.c: 生成内核镜像的Linux工具，请使用作业一中自己写的代码。
- Entry.S: 中断处理函数。
- kernel.c: 内核最先执行的文件，放在内核的起始处。
- scheduler.c: 调度器，实现`task`的调度。
- syslib.S: 系统调用函数
- syslib.c: 系统调用接口
- interrupt.c: 系统调用和中断处理相关的函数。
- queue.c: 队列处理函数，提供了队列操作的一些接口。
- print*.c: 提供一些输出函数，可以用于调试以及显示信息。
- sync.c: 一些同步操作。
- mbox.c: 邮箱的操作，本次任务需要实现。
- util.c: 提供了一些输出函数，可以用于调试以及显示信息。
- file.c: 将所有任务放在File的机构体中，供do_spawn查找使用
- ramdisk.c: 提供操作File的一些接口
- settest: 设置需要测试的样例。
- *.h: 相应.c/.S文件的头文件。
- Memory.c: 需要实现的关于虚拟内存的函数。
- Memory.h: 需要实现的关于虚拟内存的一些定义。

2.2. 获取

课程网站。

2.3. 运行

Makefile 文件提供编译功能。

make 编译命令

make clean 对编译产生的文件进行清除

sudo dd if=image of=/dev/sdb 将产生的 image 写进 SD 卡中
在 minicom 中执行 *loadboot* 运行程序

3. 任务

3.1. 设计和评审

帮助学生发现设计的错误，及时完成任务。学生需要对这次的作业进行全面考虑，在实现代码之前有清晰的思路。学生讲解设计思路时可以用不同的形式，如伪代码、流程图等，建议使用 PPT。

3.1.1. 设计介绍

任务一和任务二中的页表都是线性页表，都是单级的，而实际系统中通常会使用多级页表来寻址。另外，任务二中的页替换使用的是 FIFO 算法，通常实际系统中会使用 Clock 或 LRU 算法。因此，本次作业中有两个加分项，第一个加分项是实现一个除 FIFO 以外的页替换算法；第二个加分项是实现两级页表。

3.2. 开发

3.2.1. 要求

加分项 1：实现除 FIFO 以外一种页面替换算法，例如 Clock，LRU 等，具体算法自行选择。

加分项 2：实现两级页表。

3.2.2. 注意事项

请认真阅读任务一和任务二的注意事项。

- 进程虚址空间大小

实现两级页表要求进程地址空间较大，请考虑如何扩大测试用例进程使用的内存空间。

4. 测试

请在任务一和任务二运行的基础上自行设计测试进程用例，能展示（1）实现的页替换算法在缺页数上低于 **FIFO** 算法（加分项任务一）；（2）测试用例进程能正常运行，并且实际地址空间使用两级页表所需物理页少于（至少等于）使用单级页表所需物理页（加分项二）。