

编程小作业 2: mmap 初尝试

计算机系统 (2)

作业截止时间: 2022 年 4 月 13 日 12:00

更新: 2022 年 3 月 29 日

作为系统课程, 首先需要学会使用操作系统以及其服务。本系列编程实践小作业的目的是希望大家知道操作系统提供了哪些服务, 如何快速使用他们。在这个完成过程中, 需要学会查阅各种手册, 包括但不限于 man 页面等。

0.1 任务 1: 通过 mmap 修改文件

mmap 是一种内存映射文件的方法, 即将一个文件或者其它对象映射到进程的地址空间, 实现文件磁盘地址和进程虚拟地址空间中一段虚拟地址的一一对映关系。实现这样的映射关系后, 进程就可以采用指针的方式读写操作这一段内存。

在这个任务中, 你需要使用 mmap 来实现通过共享映射方式的文件修改。任务的期望目标是实践如何使用 mmap, 对 mmap 的 API 有所了解。你只能按照注释的要求添加代码, 而不可以对代码有任何的删减或修改。

当你完成必要的填空之后, 你可以通过在 task1 目录下执行 make 以编译程序, 并且在同一个目录下执行 make run 以测试你的结果。

本任务的评分方式是通过 Code Review 方法进行评分。

0.2 任务 2: 通过共享映射实现两个进程之间的通信

进程间通信（Inter-process communication）有许多种方法，其中一种是共享内存，即两个进程读写同一段内存来实现信息交换。

在这个任务中，你需要实现两个进程将同一个文件映射到自己的地址空间，进程 A 先运行，每隔 1 秒读取映射区域，5 秒后运行进程 B，修改映射区域，从进程 A 中观察映射区域内容变化。

当你完成必要的填空之后，你可以通过在 task2 目录下执行 make 以编译程序，并且在同一个目录下执行 make run 以测试你的结果。

本任务的评分方式是通过 Code Review 方法进行评分。

0.3 任务 3: 通过匿名映射实现父子进程通信

进程间通信不一定需要写入文件。我们可以使用匿名映射（anonymous mapping）使得多进程运行时访问同一段内存，而不进行任何文件操作。

在这个任务中，你需要实现在当前进程中创建一个子进程，用匿名映射的方法从父进程向子进程传输一条消息，再从子进程向父进程传输一条消息。

当你完成必要的填空之后，你可以通过在 task3 目录下执行 make 以编译程序，并且在同一个目录下执行 make run 以测试你的结果。

本任务的评分方式是通过 Code Review 方法进行评分。

0.4 任务 4: 共享映射读写与传统系统调用读写 (read, write) 读写速度比较

当进行一般的文件读操作时，磁盘空间的文件需要先拷贝到内核态，再从内核态拷贝到用户态，一共是两次拷贝操作；mmap 将文件从磁盘直接映射到用户态内存，只需要一次拷贝。写操作同理。

在这个任务中,你需要实现 4 个函数,分别测量使用共享映射读写/系统调用顺序读写 64MB 文件所需的时间。而对于这个 64MB 的读写,我们采用了多个不同的 `<step>`, 每次读/写 `<step>` bytes, 来观察不同读写粒度两种读写方式的性能差异。

当你完成必要的填空之后,你可以通过在 `task4` 目录下执行 `make` 以编译程序,并且在同一个目录下执行 `make run-write` 以测试文件写入部分,执行 `make run-read` 以测试文件读取部分。

本任务的评分方式是通过 Code Review 方法进行评分。

注: `wsl` 的文件系统和 `linux` 可能有所差异,有条件建议在 `linux` 环境下测量。