

系统调用流程：

作用：

步骤：ecall trap 流程：权限切换，保存状态，切换至内核，处理错误原因，返回继续执行。

修改进程加速：

目的：加快内核与用户之间的信息传输

步骤，让进程拥有内核页表的映射；在内核页表中映射用户态页表的物理地址；

难点，在释放进程资源的时候，内核页表只释放映射的用户那部分。

追踪函数调用栈；

步骤：寄存器中存有当前栈地址，前8字为函数返回地址，下一个8字为函数调用者的栈地址

响应时钟中断：

目的：设计了系统的定时任务程序，模拟了设备中断处理流程

步骤 两个系统调用 一个设置时钟以及处理程序，一个从处理程序中返回；

在pcb中保存信息；当时间片用完就会触发时钟中断；此时计数增加，当满足时间要求是，调用处理程序，在调用前，需要保存当前的陷阱帧和 PC，

返回程序恢复pc和陷阱帧。

多线程优化：

目的：同步问题

步骤：用到了睡眠锁，sleep；时间达成唤醒wakeup

懒分配：

写时拷贝：

目的，提高系统效率

步骤：fork时子进程共享父进程的物理页，将物理页设置为cow模式，将权限设定为只读；当进程中存在某个需要写入物理页时，触发页错误。此时再去申请物理页，复制保存的物理页，将权限设定为可写可读。

难点：在释放某个子进程时，有的页面被引用，有的页面是独享的，不能释放，解决对物理页设置引用计数。

实现CPU专属内存链表，降低内存分配时的锁争用；

步骤：为每个cpu设置空闲链表，只有当当前cpu的链表使用完，才可以申请其他cpu的资源；其中用锁来解决互斥；

通过减少锁的粒度来加速：

步骤：将buffer pool 分为多个哈希桶，先查询属于哪个哈希桶，再在哈希桶中进行遍历。于此同时记录时间撮最小的buffer并且没被引用的。找到就返回增加引用计数，若没找到就替换最久未被使用的。

坑：一定要判断引用计数是不是0，再进行替换。

改进 文件系统实现对大型文件的存储支持，以及对软链接文件的支持。

更改inode结构，构建三级结构