北京郵電大學

Beijing University of Posts and Telecommunications

操作系统课设

模拟操作系统

需求分析

时间：2022/3/20

组号：8

目录

0、小组人员及分工

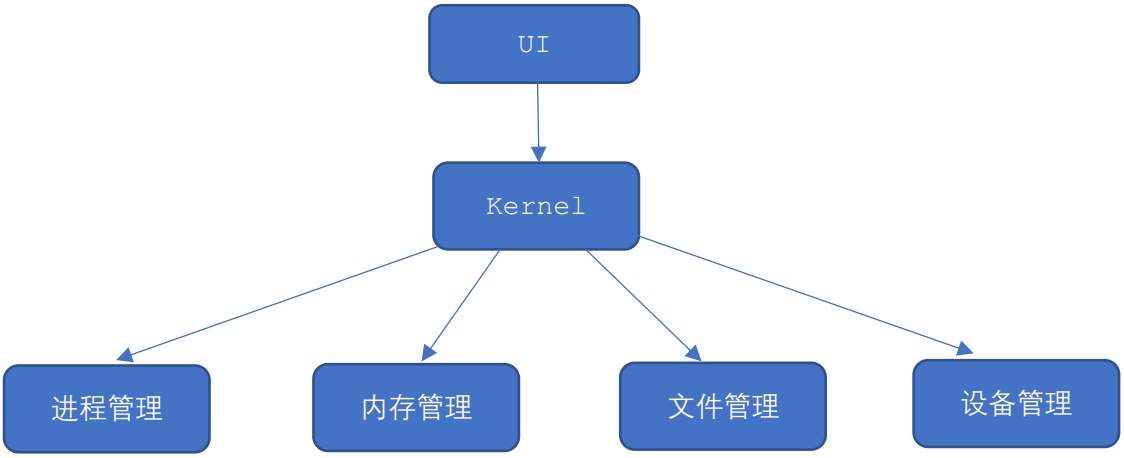
姓名	学号	分工	备注
张明昱	2019211590	进程管理	组长
董晓雨	2019211589	进程管理	组员
朱飞烟	2019211620	设备管理	
李佳奕	2019211588	内存管理	
邵逸辰	2019211591	文件管理	
张博	2019211623	UI	

1、总体框架

为实现设计一个模拟操作系统，根据操作系统的实际功能，设计划分功能模块：UI、kernel、process management、memory management、file management、IO management 六部分。

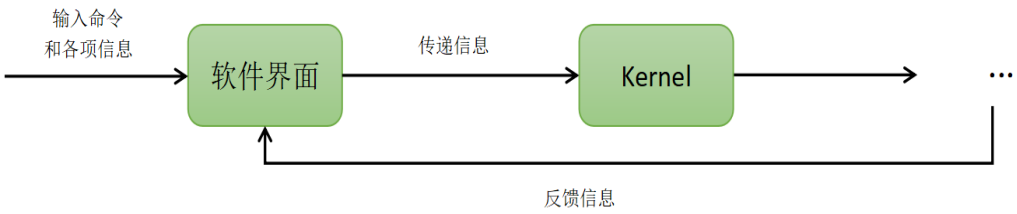
其中，UI 提供与用户的图形化界面交互；kernel 内核实现各模块功能的整体连接；process management（进程管理）进行操作系统中进程的管理和创建，包含控制实时时间的 timer 和对中断的处理等；memory management（内存管理）使用合适的算法完成进程存储空间的分配和回收功能；file management（文件管理）负责目录/文件的创建和删除以及空间分配和回收；IO management（设备管理）负责使用相应的算法管理 IO 设备的申请、分配、使用、释放。各模块细致的需求将在下面详细讲解。

整体模块划分和模块间的相互调用关系如下：



2、UI

2.1 数据流分析



用户在软件界面输入进程、外设、文件、调度算法等各种必需信息以及不同的指令，再由前端进行初步处理后传递给内核，把对应参数传递给相应子程序；

同时在运行过程中，还需子程序以及内核反馈给前端相应信息，并且输出到界面。

2.2 功能分析

（1）手动控制程序的提交执行

需要输入必需信息，例如：指令，进程结构相关，PCB 相关，调度算法的选择等等。

（2）动态展示系统运行期间的快照（snapshot）

包括：各并发进程的状态（进程 ID，进程类型，到达时间，持续时间，进程状态，进程优先级等）以及各种队列的相关信息；

内存分配情况（内存大小，各进程所占内存块大小及其位置，系统空闲内存块大小及其位置--帧 ID、页 ID 等等）；

文件树，文件路径，文件类型

（3）各设备的状态以及设备队列情况

设备请求信息、设备现场信息

2.3 数据说明

（1）get_command：类

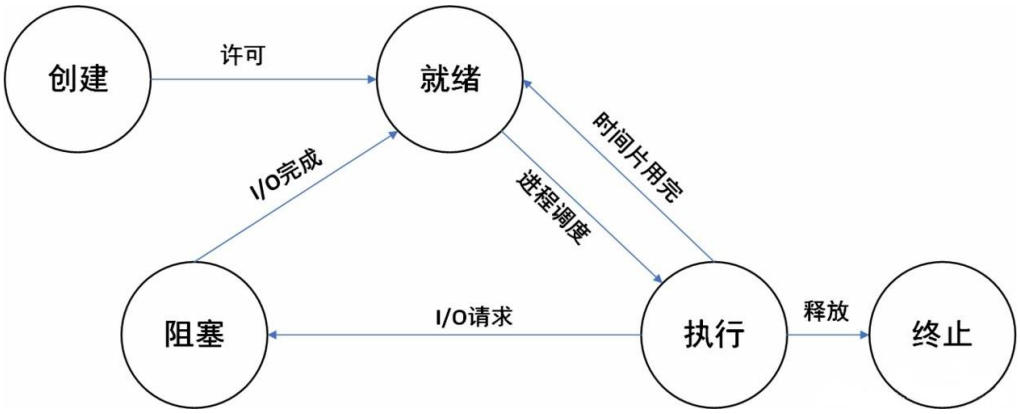
作用：对输入信息进行初步处理（识别、分割、检错）后交由内核处理；并且对命令进行缓存，使得用户的使用体验以及使用效率能够得到大幅提升。

（2）Print_system_info：类

作用：动态输出反馈信息，展示系统运行期间的快照，各设备的状态以及设备队列情况等

3、进程管理

3.1 数据流分析



在进程的生命周期中共包含五个状态：

- 创建：进程正在被创建。
- 执行：指令正在被执行。
- 阻塞：进程等待某个事件的发生（如 I/O 完成或收到信号）
- 就绪：进程等待分配处理器。
- 终止：进程完成执行。

当新建一个进程后，进程就由创建态转换到就绪态，等待分配处理器。通过进程调度算法将选中的进程由就绪态转变成执行状态，进程开始执行。进程处于执行状态时，若发生中断或时间片用尽则回到就绪状态；若产生 I/O 操作或等待事件，则从运行态转换到阻塞态，等待 I/O 操作或事件的完成，进而转换到就绪态。在程序运行的过程中，如果接收到退出指令，或者程序运行完毕，则转入终止状态，释放进程占用的资源。

3.2 功能分析

（1）新建进程：由系统在初始时刻创建，或者根据用户输入的进程信息——包括进程到达时间、持续时间、进程优先级、进程所需内存、I/O 请求等信息创建新的进程。每个进程用一个进程管理块 PCB 表示。

（2）实现进程状态转移：设置就绪队列 QueueReady、阻塞队列 QueueBlock 和执行队列 QueueRunning。当进程创建成功后将其加入到就绪队列 QueueReady。当 CPU 空闲时，在就绪队列 QueueReady 中选择一个进程将其加入到执行队列 QueueRunning 执行，并从就绪队列 QueueReady 中删除该进程。运行中的进程若产生 I/O 操作或等待事件，则将其迁移到阻塞队列 QueueBlock，等待 I/O 操作或事件的完成，重新进入到就绪队列 QueueReady。若进程执行完毕或接收到退出指令，将该进程从执行队列 QueueRunning 中删除，并更改该进程 PCB 的状态由 running 更改为 terminated，释放其占用的资源。

以上每次队列间的迁移都涉及到 PCB 中状态信息的改变。

（3）实现进程调度：支持 FCFS 算法、SJF 算法、优先级算法和 RR 算法实现 CPU 调度，支持抢占和非抢占。

（4）实现进程同步：支持使用信号量实现进程间的同步，共享内存。

3.3 数据说明

（1）PCB 进程控制块：类
进程编号 PID：用于识别进程的进程标识符；
进程状态 process_state：五个可能状态；
程序计数器 count：表示进程要执行的下个指令的地址；
寄存器 register：保存中断信息；
（2）PCB 是存放进程的管理和控制信息的数据结构，是进程存在的唯一标志。PCB 将以链表形式存储在系统中，它包含许多与一个特定进程相关的信息。

3.4 timer 时钟管理

任何系统都需要一个周期性的信号源，以供系统处理延时超时等与时间相关的事件，这个周期性的信号源就叫做时钟。在进程控制模块中同样需求时钟来统一系统中时间，辅助完成并发中断等功能。

3.5 中断处理

中断响应以及中断处理是进程模块要处理的另一个问题，由于某种事件的发生而导致程序流程的改变。产生中断的事件称为中断源。CPU 将对中断进行相应和处理，判断中断形成的原因并把进程的状态作出相应的改变。

4、内存管理

4.1 数据流分析

说明：从内核接受到分配或回收内存的指令，并匹配合适算法，随后分配内存并将该部分内存标记为不可用，并更改内存使用情况。

4.2 功能分析

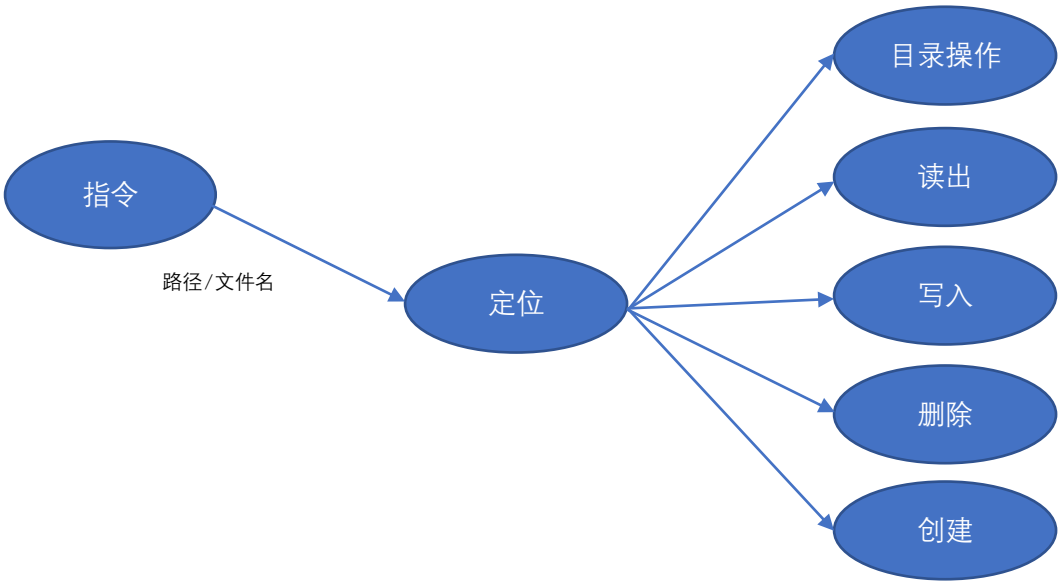
（1）根据核心下达的分配或回收指令，对内存进行管理
（2）实现采用连续分配与页式分配
（3）选择合适算法实现虚拟内存分配
（4）实现核心算法：best-fit 或 worst-fit 算法，页式分配，页面替换算法等
（5）实现监测功能，用另外一组数据结构保存最近 N 次操作的内存变化和分配情况，并根据内核态的指令打印内存分配情况

4.3 其它说明

（1）在能力允许的情况下，设计过程中尽量根据实际情况尝试多种分配算法
（2）尝试通过多线程实现对内存分配与释放情况的跟踪

5、文件管理

5.1 数据流分析



5.2 功能分析

（1）访问目录/文件：根据路径/文件名进行文件定位
（2）创建文件：创建新文件，分配内存
（3）打开文件：判断文件是否可用，并打开
（4）关闭文件：文件结束使用后，关闭文件，安全存储
（5）读取文件：文件打开后可顺序读文件中的记录，读长度表示本次操作需要读取的记录个数
（6）写入文件：文件的顺序写入和随机写入
（7）删除文件：把请求撤消的文件从拥护文件目录表中除名，收回该文件站用的存储区域。

5.3 数据说明

（1）iNode 结构体：文件的索引节点
  a） 文件类型：目录/普通文件
  b） 文件大小（字节数）
  c） 文件数据位置
  d） 文件状态（防止同时写入）
（2）dir 目录：结构体
  a） 文件名
  b） iNode 编号
（3）file_handle 文件句柄：结构体
  a） iNode 编号
  b） 读指针

c） 写指针

6、设备管理

6.1 数据流分析

某进程申请设备资源，将其请求信息放入设备请求列表，利用调度算法分配资源，等待具体任务完成后，将分配的资源释放，回收设备以便下次申请分配。

6.2 功能分析

（1）申请：对发出该请求的进程做出响应，记录所申请的设备，以及其数据内容等信息；
（2）分配：按照设备类型和相应的分配算法（如 FCFS、优先算法等）决定将 I/O 设备分配给哪一个进程，通过重新安排队列顺序以改善系统总体效率和应用程序的平均响应时间；
（3）使用：进程对于分配到的设备进行具体的操作。
（4）释放：当进程使用完对应的设备后，释放所占有的设备，系统进行回收修改对应的数据结构，以便下次分配使用。

6.3 数据说明

（1）Device：结构体；
作用：储存设备对象的基本信息；
其中包含信息：设备名称，设备传输速率，设备状态是否忙，请求队列；
（2）DeviceRequest：结构体；
作用：储存设备请求的信息；
其中包含的信息：发出该请求的进程，所请求的设备，请求内容，I/O 操作需要的时间（若是 I/O 操作进程），目标文件和文件操作（若请求设备是磁盘）。