xm 拷贝



**操作系统课程设计报告**

**—需求分析**

**学 院： 计算机学院（国家示范性软件学院）**

**专 业： 计算机科学与技术**

**班 级： 2020211311**

**组 长： 梁正阳**

**组 号： 5**

**2023年3月17日**

目录

[一、进程管理 3](#_Toc130161033)

[1.数据流分析 3](#_Toc130161034)

[2.功能分析 3](#_Toc130161035)

[3.数据说明 3](#_Toc130161036)

[二、中断机制 4](#_Toc130161037)

[1.数据流分析 4](#_Toc130161038)

[2.功能分析 4](#_Toc130161039)

[三、设备管理 5](#_Toc130161040)

[1.数据流分析 5](#_Toc130161041)

[2.功能分析 5](#_Toc130161042)

[3.数据说明 6](#_Toc130161043)

[四、内存管理 7](#_Toc130161044)

[1.数据流分析 7](#_Toc130161045)

[2.功能分析 7](#_Toc130161046)

[3.数据说明 7](#_Toc130161047)

[五、文件管理 8](#_Toc130161048)

[1.数据流分析 8](#_Toc130161049)

[2.功能分析 8](#_Toc130161050)

[3.数据说明 9](#_Toc130161051)

[六、UI界面 11](#_Toc130161052)

[1.数据流分析 11](#_Toc130161053)

[2.功能分析 11](#_Toc130161054)

[七、时钟 12](#_Toc130161055)

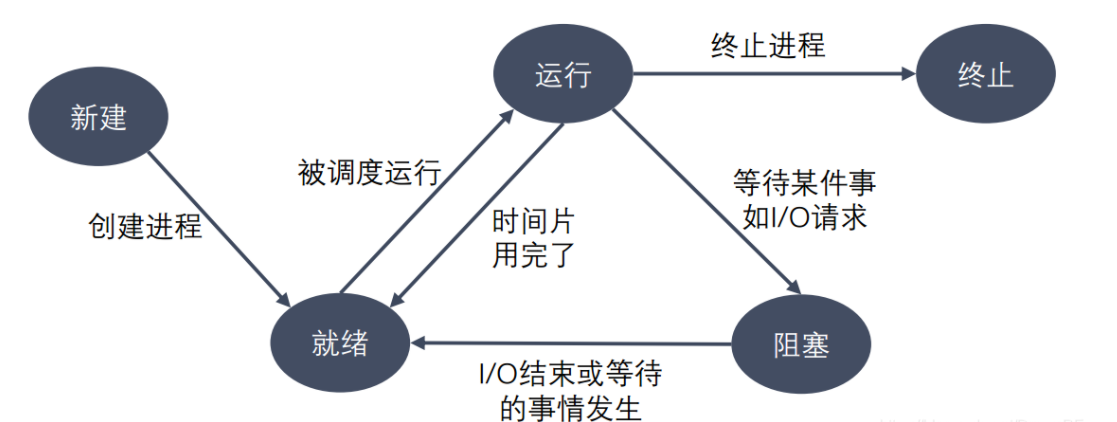
[1.数据流分析 12](#_Toc130161056)

[2.功能分析 12](#_Toc130161057)

[3.数据说明 12](#_Toc130161058)

# 一、进程管理

## 1.数据流分析



## 2.功能分析

1）维护进程的状态：操作系统在内存中自己维护，通过PCB对进程的运行状态进行管理。

2）进程控制与同步：控制进程就绪、执行、阻塞三种状态的基本功能，在时间片结束后依据算法对所有进程的状态进行更新。对临界资源进行访问时，必须依据操作系统同步机制等待占用进程释放资源后才可以竞争使用。

3）进程调度：根据进程调度算法，选择多级反馈队列调度，将队列中的进程有序的调度，在时间片结束后对运行队列进行更新。

多级队列采用多个优先队列，根据优先级进行排序，首先运行第一级队列中的进程，根据测试情况确定最佳时间片长度，时间片用完时向CPU发出信号，CPU接收到信号后进行进程切换，并同时判断进程是否运行完毕，根据情况进行进程状态的修改。

4）进程的创建与删除：

创建：申请一个空白的PCB，向PCB中填写用于控制和管理进程的信息，为进程分配运行时所必须的资源，待满足条件时将进程转入就绪状态并插入就绪队列。

删除：找到该PCB，若处于运行状态，剥夺其CPU，终止其全部子进程，将PCB清零，将该空白PCB和占用的资源返还给系统。

## 3.数据说明

1）PCB：进程在操作系统中的户口，用于表示并唯一确定这个进程。其包含了进程的全部信息，包括：进程id、进程状态信息、进程切换时需要保存和恢复的寄存器信息、描述虚拟地址空间信息、当前工作目录、与信号相关的信息、进程可以使用的资源上限等等信息。

2）进程表示符：PID：process identifier，每个进程都有一个唯一的标识符，内核通过这个标识符来识别不同的进程，同时，进程标识符PID也是内核提供给用户程序的接口，用户程序通过PID对进程发号施令。PID是32位的无符号整数，它被顺序编号：新创建进程的PID通常是前一个进程的PID加1。

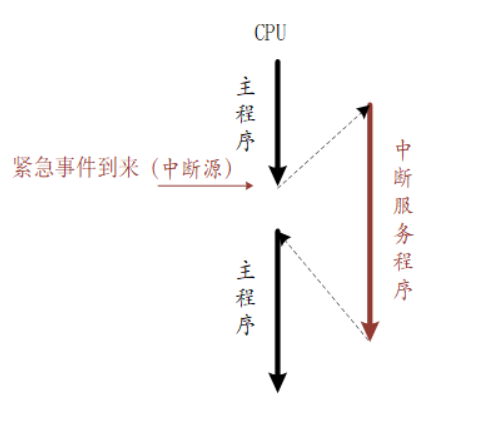
3）进程状态，进程执行时，它会根据具体情况改变状态。进程状态是调度和对换的依据。其中包含了可运行状态、等待状态、暂停状态、僵死状态等。

4）进程通信有关信息：IPC，为了使进程能在同一项任务上协调工作，进程之间必须能进行通信即交流数据。

5）进程链接信息：程序创建的进程具有父/子关系。因为一个进程能创建几个子进程，而子进程之间有兄弟关系，在PCB中有几个域来表示这种关系。

# 二、中断机制

## 1.数据流分析

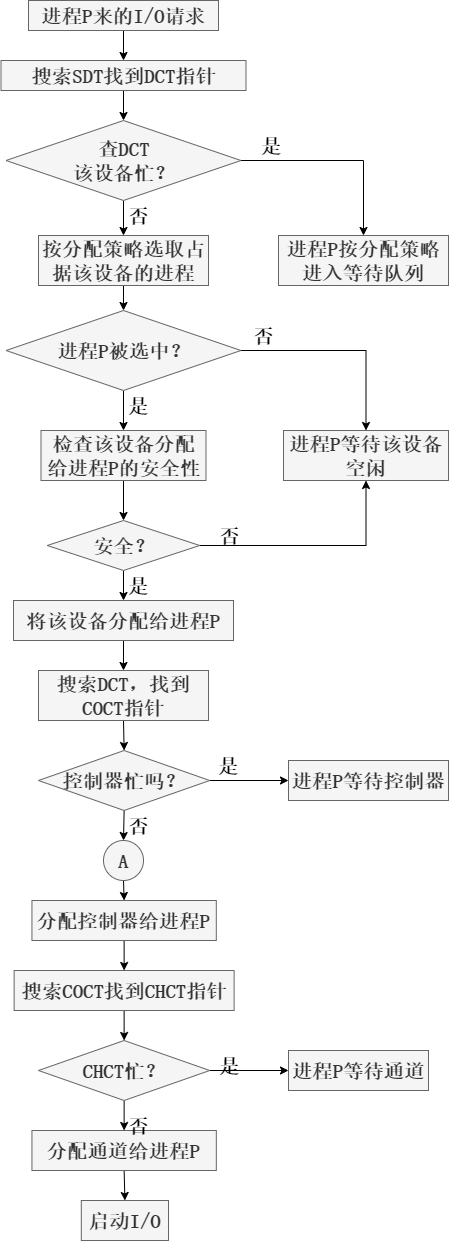
****

## 2.功能分析

由于当前程序需要处理某些情况，中断CPU当前进程的运行，转而运行其他程序来处理当前IO请求，处理完成后CPU返回暂停的进程中执行。采用信号机制以及多线程完成中断功能，子线程循环进行获取中断信号，一旦获取中断信号就中断当前进程，保存线程，进行中断程序的处理。

# 三、设备管理

## 1.数据流分析



## 2.功能分析

1）向用户提供使用模拟外部设备的方便、统一的接口，按照用户的要求和设备的类型，控制设备工作，完成用户的输入输出请求。

2）充分利用中断技术、通道技术和缓冲技术，提高CPU与设备、设备与设备间的并行工作能力，充分利用设备资源，提高外部设备的使用效率。

3）设备申请、分配、使用和释放。保证在多道程序环境下，当多个进程竞争使用设备时，按照一定的策略分配和管理设备，以使系统能有条不紊地工作，并在设备使用完成后予以释放。

## 3.数据说明

1）独享设备：这类设备在一段时间内最多只能有一个进程占有并使用它。

2）共享设备：这类设备允许多个进程共享，即多个进程的I/O传输可以交叉，在宏观上多个进程同时使用。例如，多个进程可以交替地从磁盘上读写信息。

3）虚拟设备：在一类设备上模拟另一类设备的技术称为虚拟设备技术。通过虚拟技术将一台独占设备变换成若干台逻辑设备。

4）设备控制表（DCT）：反映设备的特性、设备和I/O控制器的连接情况。包括设备标识、使用状态和等待使用该设备的进程队列等。

系统中每个设备都必须有一张DCT，且在系统生成时或在该设备和系统连接时创建，但表中的内容则根据系统执行情况而被动态地修改。

DCT包括以下内容：

·设备标识符，设备标识符用来区别设备。

·设备类型，反映设备的特性，例如是终端设备、块设备或字符设备等。

·设备地址或设备号，每个设备都有相应的地址或设备号。这个地址既可以是和内存统一编址的，也可以是单独编址的。

·设备状态，指设备是处理工作还是空闲中。

·等待队列指针，等待使用该设备的进程组成等待队列，其队首和队尾指针存放在DCT中

·I/O控制器表指针，指向与该设备相连接的I/O控制器的I/O控制器表。

5）控制器控制表（COCT）：每个控制器一张，它反映I/O控制器的使用状态，以及和通道的连接情况等。

6）通道控制表（CHCT）：每个通道一张。CHCT包括通道标识符、通道忙／闲标识、等待获得该通道的进程等待队列的队首指针与队尾指针等。

7）系统设备表（SDT）：整个系统只有一张，记录已被连接到系统中的所有物理设备的情况，并为每个物理设备设一表目项。

SDT的每个表目项包括的内容有：

·DCT指针，该指针指向有关设备的设备控制表。

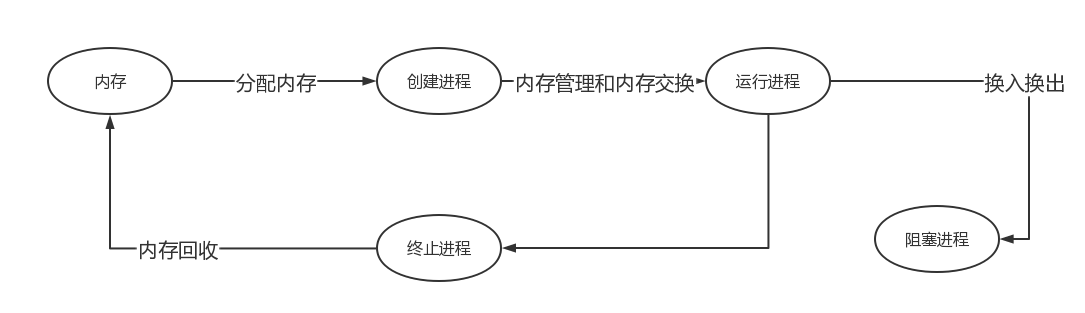
·正在使用设备的进程标识。

·设备类型和设备标识符，该项意义与DCT中相同。

SDT的主要意义在于反映系统中设备资源的状态，即系统中有多少设备，有多少是空闲的，而又有多少已分配给了哪些进程。

# 四、内存管理

## 1.数据流分析



## 2.功能分析

1）整体模拟操作系统的内存管理设计，与进程调度和文件调度相结合，并为其提供对应的接口来实现其对内存的操作。使得各个进程能够在内存中有序地存放，并且能够有逻辑地存入和取出。

2）内存的分配：当进程创建时，检查剩余内存是否足够，若不够，是否要采用交换或者覆盖。管理储存方式，使用连续分配方式或者页式存储将程序存放在内存当中。

3）进程交换：将外存中的进程交换与内存中不常用的进程进行交换，从而符合其对内存空间大小的要求。可以将暂时不能执行的程序（进程）送到外存中，从而获得空闲内存空间来装入新程序（进程），或读人保存在外存中而处于就绪状态的程序。

4）内存的回收：进程终止时，将分配给其的内存空间回收，便于后续分配给其他的进程使用。修改对应的变量和结构。

5）动态分区可以使用各种算法，如首次适应算法或者最佳适应算法，来进行内存分配。

## 3.数据说明

1）物理内存：设置物理内存的大小，将进程存放在物理内存中。又包括连续存储和非连续存储的存储方法。

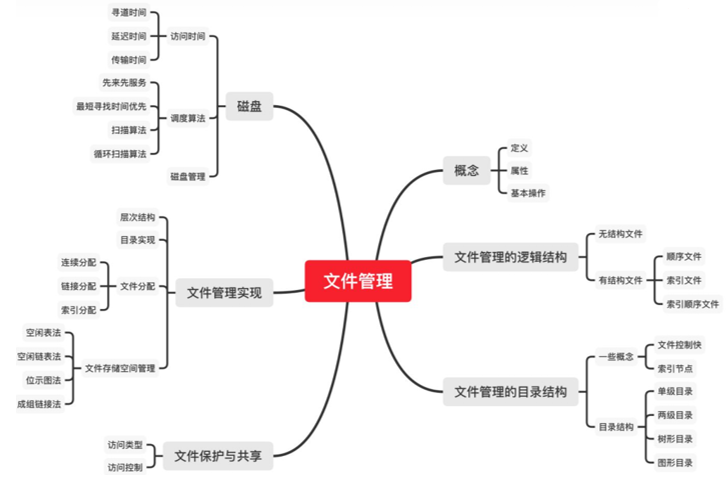
2）虚拟内存，每个进程都有其对应的虚拟内存，虚拟内存和物理内存之间有映射关系，通过虚拟内存可以访问到对应的物理内存。

3）如果采用段式管理，则需要一个段表，其中记录段号，虚拟地址由段号和段内偏移量构成。总的来说，段式存储管理的优点是：没有内碎片，外碎片可以通过内存紧缩来消除；便于实现内存共享。缺点与页式存储管理的缺点相同，进程必须全部装入内存。

4）如果采用页式管理，则需要一个页表，其中记录虚拟页号和物理页号,一个程序不必连续存放。便于改变程序占用空间的大小，主要指随着程序运行，动态生成的数据增多，所要求的地址空间相应增长，要求程序全部装入内存，没有足够的内存，程序就不能执行。

# 五、文件管理

## 1.数据流分析



## 2.功能分析

1）整体模拟操作系统的文件管理设计，与进程调度、内存管理、磁盘管理相结合，并为其提供对应的接口来实现其对文件的操作。使得各个进程能够在高效、有序地增删目录与文件、读写文件，并且能够清晰的模拟对磁盘（存储空间）的分配和回收。

2）目录实现：

在读文件前，必须先打开文件。打开文件时，操作系统利用路径名找到相应目录项，目录项中提供了查找文件磁盘块所需要的信息。

目录管理：通过树形结构（多级目录结构）来解决和实现。从用户的角度看，目录在用户所需要的文件名和文件之间提供一种映射，所以目录管理要实现“按名存取”；

目录的基本操作有：搜索文件、创建文件、删除文件、显示目录、修改目录

3）文件实现

文件的实现就是研究文件的物理结构，即文件数据在物理存储设备上是如何分布和组织的。一是文件的分配方式，讲的是对磁盘非空闲块的管理；二是文件存储空间管理，是对磁盘空闲块的管理。

文件的基本操作：文件的打开与关闭、创建文件。写文件。读文件、删除文件。

4）存储空间的分配和回收：

分配磁盘块:与内存管理中的动态分区分配很类似，为一个文件分配连续的存储空间。可采用首次适应、最佳适应、最坏适应等算法来决定要为文件分配哪个区间。

回收磁盘块:与内存管理中的动态分区分配很类似，当回收某个存储区时需要有四种情况①回收区的前后都没有相邻空闲区;②回收区的前后都是空闲区;③回收区前面是空闲区;④回收区后面是空闲区。需要注意表项的合并问题。

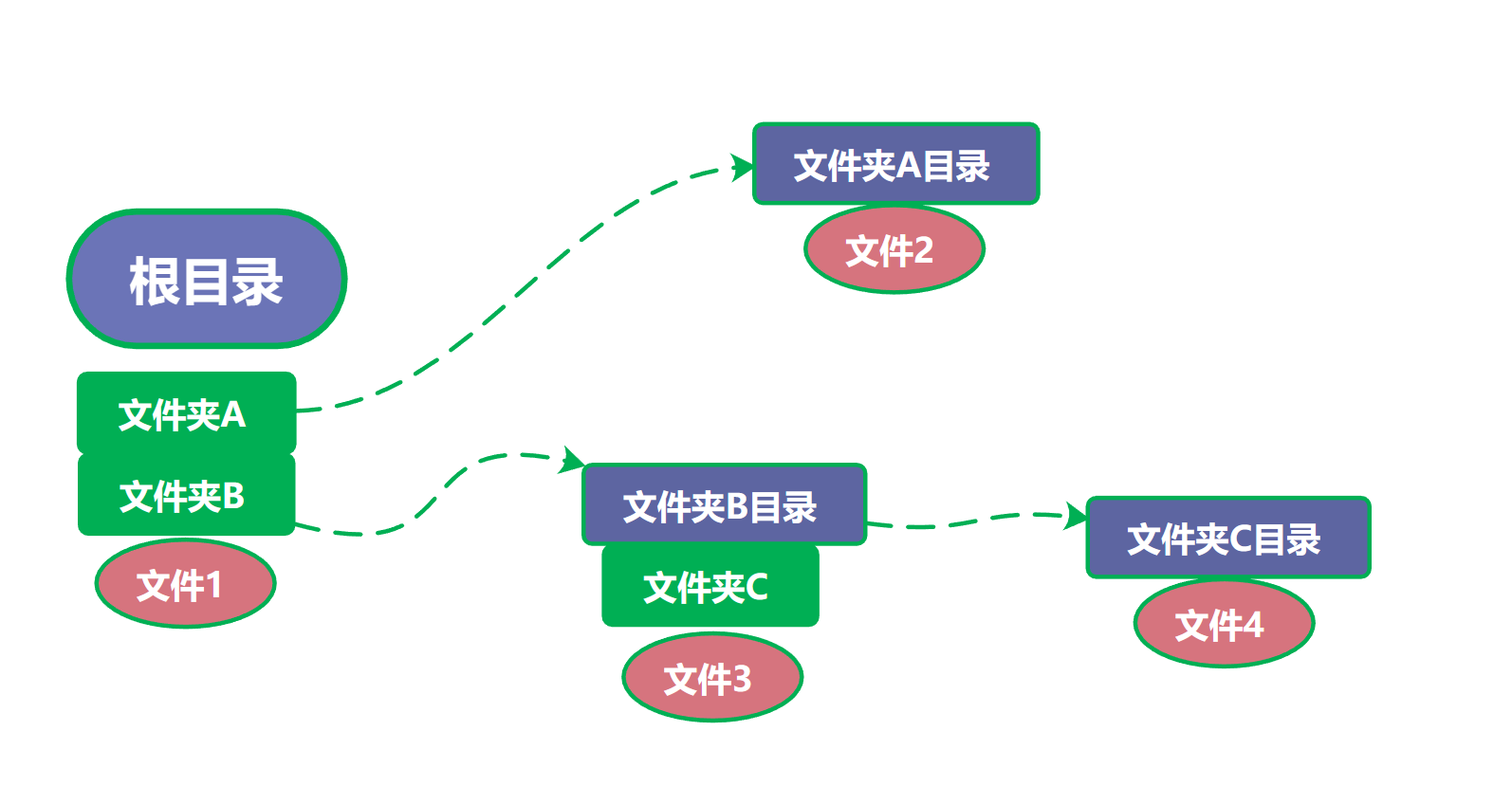
## 3.数据说明

1）多级目录结构：用户要访问某个文件时，用文件的路径名标识文件，文件路径名是个字符串，由根目录出发到所找文件通路上所有目录名与数据文件名用分隔符“/”链接而成。从根目录出发的路径称为绝对路径，从当前目录出发的路径称为相对路径。

优点：可以很方便的对文件进行分类，层次结构清晰，能够有效地进行文件地管理和保护

缺点：查找文件时，需要按照路径名逐级访问中间结点，增加了访盘次数，影响了查询速度；不方便对文件共享。

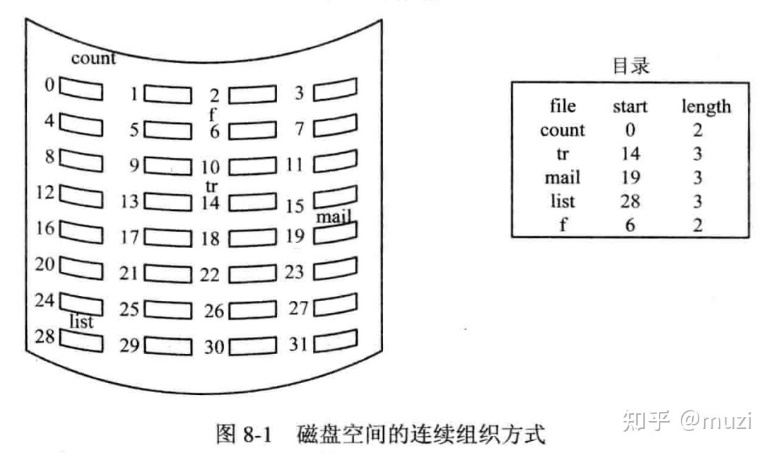
多级目录的形式



2）文件控制块（FCB）：文件控制块是用来存放控制文件需要的各种信息的数据结构，以实现“按名存取”。FCB的有序集合称为文件目录，一个FCB就是一个文件目录项。

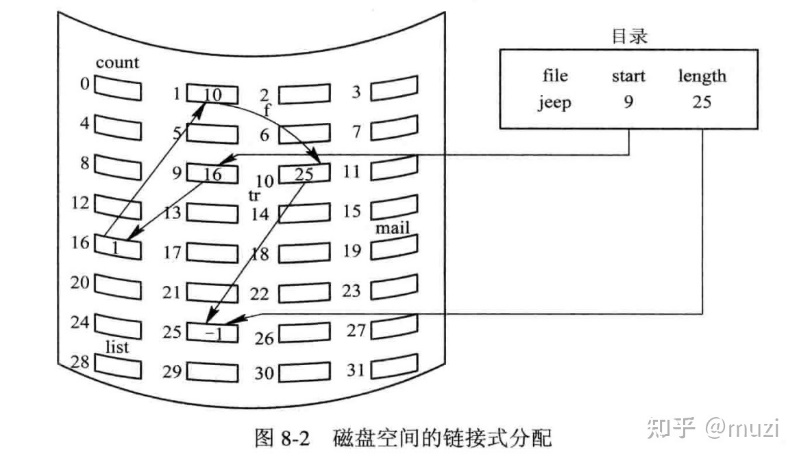
文件分配方式：

连续分配：连续分配方法要求每个文件在磁盘上占有一组连续的块。磁盘地址定义了磁盘上的一个线性排序。这种排序使作业访问磁盘时需要的寻道数和寻道时间最小。



链接分配：

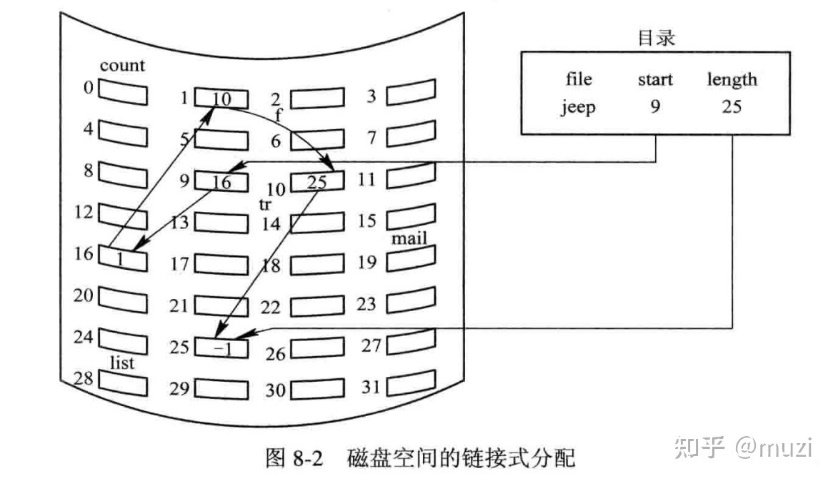
链接分配采用离散分配的方式，消除了外部碎片，显著提高了磁盘空间利用率；又因为根据文件的当前需求为其分配必须的块，当文件增长时，可以动态地再为它分配盘块



索引分配：

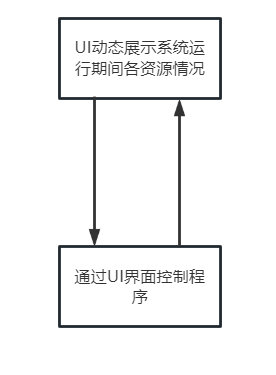
索引分配解决了链接分配不能有效支持直接访问的问题，它把每个文件的所有的盘块号都集中放在一起构成索引表。

索引分配允许文件离散地分配在各个磁盘块中，系统会为每个文件建立一张索引表，索引表中记录了文件的各个逻辑块对应的物理块。索引表存放的磁盘块称为索引块。文件数据存放的磁盘块称为数据块。



# 六、UI界面

## 1.数据流分析



## 2.功能分析

1)用图形界面展示多道程序并发执行的过程

图形界面需要接收各模块返回的数据信息，直观、明确、清晰地表达出操作系统现在 各个资源的占用情况。在多道程序执行过程中，UI界面需要展示出多道程序并发执行的执行过程，比如多道程序使用何种算法进行调度，可以通过UI界面清晰地看出其调度过程。

2)手动控制程序的提交执行

可以通过UI界面对各模块进行操作，需要设计对进程操作的界面有增加、删除、修改、查询，对目录的界面有搜索文件、创建文件、删除文件、显示目录、修改目录等等。

3)动态展示系统运行期间的快照（snapshot），包括：

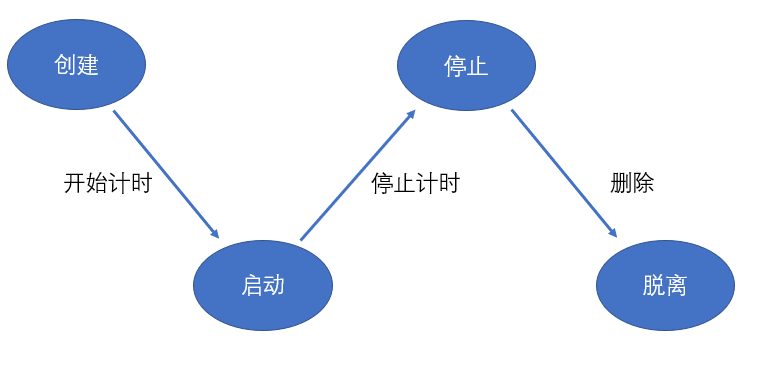
 各并发进程的状态

 内存分配情况，显示各进程占用的内存块及其位置、系统空闲内存块及其位置

 各设备的状态及设备队列情况

# 七、时钟

## 1.数据流分析

****

## 2.功能分析

1）维护系统时间：系统在初始化时获取到当前时间值之后，直至下一次重新启动前，不再通过硬件获取时间，通过时钟模块定期产生tick中断维护系统时间。

2）进程调度：时钟模块在时钟中断发生时，需要对进程时间片进行更新，每一个进程时间片结束时，要根据更新完的进程时间片作为主要依据进行进程调度。

## 3.数据说明

1）RTC：实时时钟，在PC被切断电源后，RTC还在继续工作，PC重新启动时，通过RTC获取当前时间。

2）Tick：系统滴答，时钟节拍。以固定频率进行中断，为操作系统提供调度。