

操作系统的主要特征是用户可脱机使用计算机、成批处理及多道程序运行,其优点是共享系统资源、系统资源利用率和作业吞吐量高,缺点是无交互性、作业周转时间长。分时系统采用时间片轮转方式使一台计算机为多个终端用户服务,具有交互性、多用户同时性和独立性等特征。实时系统应用于实时控制和实时信息处理领域,完成对信息及时地分析和处理,其主要特点是即时响应和高可靠性。通用操作系统可以同时兼有批处理、分时、实时处理和多重处理的功能,或其中两种以上的功能。个人计算机上的操作系统是联机的交互式单用户操作系统,随着多媒体技术的应用,个人计算机操作系统越来越要求成为具有高速数据处理能力的实时多任务操作系统。网络操作系统是通过计算机网络将多个计算机系统互联,实现信息交换、资源共享、可互操作和协作处理的系统。分布式操作系统是通过通信网络将物理上分布的具有自治功能的数据处理系统或计算机系统互联、实现信息交换和资源共享、协作完成任务的系统。以上这些都是操作系统的基本类型。

操作系统的基本功能包括处理机管理、存储管理、设备管理、信息管理以及用户接口。这些基本功能的目标是管理和控制计算机系统所有软硬件资源,为用户提供一个良好的工作环境和友好的接口。计算机最基本的功能是执行指令,指令的执行涉及处理机与内存之间的数据传输或处理机与外部设备之间的数据传输。指令执行过程中,若外部设备或计算机内部发来亟须处理的数据或其他紧急事件处理信号时,系统将进入中断处理过程。中断时处理机将保存当前的执行现场,并调用新的程序到处理机上执行。

对操作系统的研究有不同的观点,如操作系统是计算机资源管理者的观点、用户界面的观点以及进程管理的观点等,这些观点代表对操作系统不同角度的看法,都有助于理解、分析和设计操作系统。

习 题

- 1.1 什么是操作系统的基本功能?
- 1.2 什么是批处理、分时和实时系统? 它们各有什么特征?
- 1.3 多道程序设计(multiprogramming)和多重处理(multiprocessing)有何区别?
- 1.4 讨论操作系统可以从哪些角度出发,如何把它们统一起来。
- 1.5 写出 1.6 节中巡回置换算法的执行结果。
- 1.6 设计计算机操作系统时与哪些硬件器件有关?

系统调用是操作系统提供给编程人员的唯一接口。编程人员通过系统调用使用操作系统内核所提供的各种功能。系统调用的执行不同于一般用户程序的执行。系统调用执行是在核心态下执行系统子程序,而用户程序则是在用户态下执行。一般来说,操作系统提供的系统调用越多,功能也就越丰富,系统也就越复杂。

本章还简要介绍了 Linux 和 Windows 系统的用户界面。

习 题

- 2.1/ 什么是作业和作业步?
- 2.2/ 作业由哪几部分组成? 这几部分各有什么功能?
- 2.3/ 作业的输入方式有哪几种? 各有何特点?
- 2.4 试述 spooling 系统的工作原理。
- 2.5/ 操作系统为用户提供哪些接口? 它们的区别是什么?
- 2.6 作业控制方式有哪几种? 调查你周围的计算机的作业控制方式。
- 2.7/ 什么是系统调用? 系统调用与一般用户程序有什么区别? 与库函数和实用程序又有什么区别?
- 2.8/ 简述系统调用的实现过程。
- 2.9/ 为什么说分时系统没有作业的概念?
- 2.10 Linux 操作系统为用户提供哪些接口? 试举例说明。
- 2.11 在你周围装有 Linux 系统的计算机上,查看有关 Shell 的基本命令,并编写一个简单的 Shell 程序,完成一个已有数据文件的复制和打印。
- 2.12 用 Linux 文件读写的相关系统调用编写 copy 程序。
- 2.13 用 Windows 的 DLL 接口编写 copy 程序。

本章小结

进程是操作系统中最重要、最基本的概念之一。它是系统分配资源的基本单位,是一个具有独立功能的程序段对某个数据集的一次执行活动。引入进程的概念是由操作系统的资源有限性、处理上的并行性以及系统用户的执行起始时间的随机性所决定的。一切仅具有静态特征的概念,例如程序,不能反映系统的上述特征,因此,导入了具有动态特征的进程概念。

进程具有动态性、并发性等特点。反映进程动态特性的是进程状态的变化。进程要经历创建、等待资源、就绪准备执行、执行和执行后释放资源并消亡等几个过程和状态。进程的状态转换要由不同的原语执行完成。进程的并发特性反映在进程对资源的竞争以及由资源竞争所引起的对进程执行速度的制约。这种制约可分为直接制约和间接制约。进程间的直接制约是被制约进程和制约进程之间存在着使用对方资源的需求,只有制约进程执行后,被制约进程才能继续往前推进。进程间的间接制约是被制约进程共享某个一次只能供一个进程使用的系统资源,只有得到该资源的进程才能继续往前推进,其他进程在获得资源进程执行期间不允许交叉执行。因此,直接制约进程之间具有固定的执行顺序,而间接制约的进程之间则没有固定的执行顺序。

进程间的间接制约可利用加锁法和 P、V 原语操作实现。进程间的直接制约既可用 P、V 原语实现,也可用其他互相传递信号的方式实现。

尽管进程是一个动态概念,但是,从处理机执行的观点来看,进程仍需要静态描述。一个进程的静态描述是处理机的一个执行环境,被称为进程上下文。进程上下文由以下部分组成:PCB(进程控制块)、正文段和数据段以及各种寄存器和堆栈中的值。寄存器中主要存放将要执行指令的逻辑地址、执行模式以及执行指令时所要用的各种调用和返回参数等。而堆栈中则存放 CPU 现场保护信息和各种资源控制管理信息等。

本章中所述的另一个重要的概念是进程通信。进程通信又可分为传送控制信号的低级通信和大量传送数据的高级通信。从通信方式来看,又可分为主从式、会话式、消息与邮箱方式以及共享虚存方式。

无论是互相通信的进程或是共享某些不同类型资源的进程,都可能因通信顺序不当或资源分配顺序不当而造成死锁。死锁是一种因各并发进程等待资源而永久不能向前推进的系统状态。死锁对操作系统是十分有害的,消除死锁的方法是预防、回避、检测与恢复 3 种。比较常用的死锁排除方法是检测与恢复方法。

线程是为了提高操作系统的执行效率而引入的,它是进程内的一段程序的基本调度单位。线程可分为用户级线程和内核级线程。用户级线程的管理全部由线程库完成,与操作系统内核无关。线程由寄存器、堆栈以及程序计数器等组成,同一进程的线程共享该进程的进程空间和其他所有资源。线程主要用于多机系统以及网络系统的操作系统中。

习 题

- 3.1 有人说,一个进程是由伪处理机执行的一个程序,这话对吗?为什么?
- 3.2 试比较进程和程序的区别。

3.3 我们说程序的并发执行将导致最终结果失去封闭性。这话对所有的程序都成立吗？试举例说明。

3.4 试比较作业和进程的区别。

3.5 UNIX System V 中,系统程序所对应的正文段未被考虑成进程上下文的一部分,为什么?

3.6 什么是临界区? 试举一个临界区的例子。

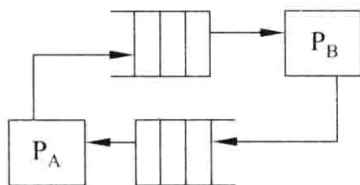
3.7 并发进程间的制约有哪两种? 引起制约的原因是什么?

3.8 什么是进程间的互斥? 什么是进程间的同步?

3.9 试比较 P、V 原语法和加锁法实现进程间互斥的区别。

3.10 设在 3.6 节中所描述的生产者-消费者问题中,其缓冲部分由 m 个长度相等的有界缓冲区组成,且每次传输数据长度等于有界缓冲区长度,生产者和消费者可对缓冲区同时操作。重新描述发送过程 $\text{deposit}(\text{data})$ 和接收过程 $\text{remove}(\text{data})$ 。

3.11 两个进程 P_A 和 P_B 通过两个 FIFO 缓冲区队列连接(如下图所示),每个缓冲区长度等于传送消息长度。进程 P_A 和 P_B 之间的通信满足如下条件:



(1) 至少有一个空缓冲区存在时,相应的发送进程才能发送一个消息。

(2) 当缓冲队列中至少存在一个非空缓冲区时,相应的接收进程才能接收一个消息。

试描述发送过程 $\text{send}(i, m)$ 和接收过程 $\text{receive}(i, m)$ 。这里 i 代表缓冲队列。

3.12 在和控制台通信的例子中,设操作员不仅回答用户进程所提出的问题,而且还能独立地向各用户进程发出指示。对于这些指示,操作员不要求用户进程回答,但它们享有比其他消息优先传送的优先度。即如果 inbuf 中有指示存在,系统不能进行下一次通信会话。试按上述要求重新描述 CCP、KCP 和 DCP。

3.13 编写一个程序使用系统调用 fork 生成 3 个子进程,并使用系统调用 pipe 创建一个管道,使得这 3 个子进程和父进程公用同一管道进行信息通信。

3.14 设有 5 个哲学家,共享一张放有 5 把椅子的桌子,每人分得一把椅子。但是,桌子上总共只有 5 支筷子,在每人两边分开各放一支。哲学家们在肚子饥饿时才试图分两次从两边拾起筷子就餐。条件如下:

(1) 只有拿到两支筷子时,哲学家才能吃饭。

(2) 如果筷子已在他人手上,则该哲学家必须等待到他人吃完之后才能拿到筷子。

(3) 任一哲学家在自己未拿到两支筷子吃饭之前,决不放下自己手中的筷子。

试解答以下问题:

(1) 描述一个保证不会出现两个邻座同时要求吃饭的通信算法。

(2) 描述一个既没有两邻座同时吃饭,又没有人饿死(永远拿不到筷子)的算法。

(3) 在什么情况下,5 个哲学家全部吃不上饭?

3.15 什么是线程? 试述线程与进程的区别。

3.16 使用库函数 $\text{clone}()$ 与 $\text{creat-thread}()$ 在 Linux 环境下创建两种不同执行模式的线程程序。

已经证明,对于 $n(n \geq 1)$ 个周期的不同任务来说,设每个周期为 T_i ,其相应任务的执行时间为 C_i ,则使用频率单调调度算法的充分条件是

$$\frac{C_1}{T_1} + \frac{C_2}{T_2} + \dots + \frac{C_i}{T_i} + \dots + \frac{C_n}{T_n} \leq n(2^{\frac{1}{n}} - 1)$$

例如,对于由 3 个周期组成的实时任务序列来说,其执行时间与周期之比应是

$$\frac{C_1}{T_1} + \frac{C_2}{T_2} + \frac{C_3}{T_3} \leq 3(2^{\frac{1}{3}} - 1) = 0.799$$

如果进程执行时间与周期比之和大于 $n(2^{1/n} - 1)$,则用户所要求的时限无法保证。

本章小结

CPU 是计算机系统中十分重要的资源,本章主要介绍处理机的调度目标、策略以及评价方法等。

因为处理机调度程序不可能选择全部驻留在外存的进程,因此,在调度一个进程占有处理机之前,系统必须按某种策略把外存中处于后备状态的作业选择出来,并创建进程和分配内存,为进程执行准备必需的资源。这一步称为作业调度或高级调度。作业调度的目标是尽量做到公平合理,能执行尽可能多的作业、尽可能快的响应时间以及高的设备利用率等。任一调度算法要同时满足这些调度目标是不可能的。大多数操作系统都是根据用户需要而采用兼顾某些目标的方法。比较常用的作业调度算法有 FCFS 方法、SJF(最短作业优先)法和 HRN(最高响应比)法等。这几种方法各有特点。其中 FCFS 法系统开销小,且对每个作业来说按其到达顺序被依次调度。FCFS 法不利于短作业。SJF 法可得到最大系统吞吐量,即每天处理的作业个数最多。但是 SJF 法有可能使长作业永远没有机会执行。HRN 法是介于 FCFS 法和 SJF 法之间的一种方法。

除了作业调度之外,还介绍了一种称为交换调度的中级调度。在有的系统中,把那些处于等待状态或就绪状态的进程换出内存,而把那些等待事件已经发生或已在外存交换区中等待了较长时间的进程换入内存。

只有在进程被建立起来并且已获得足够的资源之后,系统才使用进程调度策略把处理机分配给选出进程。因此,处理机的调度涉及 3 个层次的调度。进程调度的主要任务是选择一个合适的进程占据处理机。系统的要求不一样,进程调度方法变化较大。比较常用的有 RR(轮转)法、FCFS(先来先服务)法、优先级法和 SRR(线性优先级)法等。其中轮转法主要用于分时系统,它具有较好的响应时间,且对每个进程来说都具有较好的公平性。FCFS 法不利于执行时间短的进程,而 SRR 法则是介于 FCFS 法和 RR 法之间的一种进程调度方法。

习 题

- 4.1 什么是分级调度? 分时系统中有作业调度的概念吗? 如果没有,为什么?
- 4.2 试述作业调度的主要功能。
- 4.3 作业调度的性能评价标准有哪些? 这些性能评价标准在任何情况下都能反映调度策

- 略的优劣吗?
- 4.4 进程调度的功能有哪些?
- 4.5 进程调度的时机有哪几种?
- 4.6 假设有 4 道作业,它们的提交时刻及执行时间由下表给出:

作业号	提交时刻/hh:mm	执行时间/hr
1	10:00	2
2	10:20	1
3	10:40	0.5
4	10:50	0.3

计算在单道程序环境下,采用先来先服务调度算法和最短作业优先调度算法时的平均周转时间和平均带权周转时间,并指出它们的调度顺序。

- 4.7 设某进程所需要的服务时间 $t=k \times q$,其中, k 为时间片的个数, q 为时间片长度且为常数。当 t 为一定值时,令 q 趋于 0,则有 k 趋于无穷,从而服务时间为 t 的进程响应时间 T 为 t 的连续函数。对应于时间片的轮转调度方式(RR)、先来先服务方式(FCFS)和线性优先级调度方式(SRR),其响应时间函数分别为

$$T_{rr}(t) = t * \mu / (\mu - \lambda)$$

$$T_{fc}(t) = 1 / (\mu - \lambda)$$

$$T_{sr}(t) = 1 / (\mu - \lambda) - (1 - t * \mu) / (\mu - \lambda')$$

其中, $\lambda' = ((1 - b) / a) * \lambda = r * \lambda$ 。

取 $(\lambda, \mu) = (50, 100)$ 和 $(\lambda, \mu) = (80, 100)$,分别改变 r 的值,画出 $T_{rr}(t)$ 、 $T_{fc}(t)$ 和 $T_{sr}(t)$ 的时间变化图。

- 4.8 什么是多处理机系统? 并行处理系统、计算机网络、分布式系统和多处理机系统的操作系统之间有何区别?
- 4.9 什么是实时调度? 它与非实时调度有何区别?
- 4.10 写出表 4.1 所示周期性任务调度用的时限调度算法。
- 4.11 设周期性任务 P_1, P_2, P_3 的周期 T_1, T_2, T_3 分别为 100, 150, 350, 执行时间分别为 20, 40, 100, 问: 是否可用频率单调调度算法进行调度?

本章小结

本章介绍了各种常用的内存管理方法，它们是分区式管理、页式管理、段式管理和段页式管理。内存管理的核心问题是如何解决内存和外存的统一，以及它们之间的数据交换问题。内存和外存的统一管理使得内存的利用率得到提高，用户程序不再受内存可用区大小的限制。与此相关联，内存管理要解决内存扩充、内存的分配与释放、虚拟地址到内存物理地址的变换、内存保护与共享、内外存之间数据交换的控制等问题。

表 5.1 系统地对几种存储管理方法所提供的功能和所需硬件支持作了比较。

表 5.1 各种存储方法比较

方法 功 能	单一 连续区	分 区 式		页 式		段 式	段页式
		固定分区	可变分区	静态	动态		
适用环境	单道	多道		多道		多道	多道
虚拟空间	一维	一维		一维		二维	二维
重定位方式	静态	静态	动态	动态		动态	动态
分配方式	静态分配连续区	静态或动态分配连续区		静态或动态分配以页为单位的非连续区		动态分配以段为单位的非连续区	动态分配以页为单位的非连续区
释放	执行完成后全部释放	执行完成后全部释放	分区释放	执行完成后释放	淘汰与执行完成后释放	淘汰与执行完成后释放	淘汰与执行完成后释放
保护	越界保护或没有	越界保护与保护键		越界保护与控制权保护		越界保护与控制权保护	越界保护与控制权保护
内存扩充	覆盖与交换技术	覆盖与交换技术		覆盖与交换技术	外存、内存统一管理的虚存	外存、内存统一管理的虚存	外存、内存统一管理的虚存
共享	不能	不能		较难		方便	方便
硬件支持	保护用寄存器	保护用寄存器	保护用寄存器加重定位机构	地址变换机构 中断机构 保护机构		段式地址变换机构， 保护与中断，动态 连接机构	段式地址变换机构， 保护与中断，动态 连接机构

习 题

- 5.1 存储管理的主要功能是什么？
- 5.2 什么是虚拟存储器？其特点是什么？
- 5.3 实现地址重定位的方法有哪几类？形式化地描述动态重定位过程。
- 5.4 常用的内存信息保护方法有哪几种？它们各自的特点是什么？
- 5.5 如果把 DOS 的执行模式改为保护模式，起码应作怎样的修改？

- 5.6/ 动态分区式管理的常用内存分配算法有哪几种？比较它们各自的优缺点。
- 5.7 5.2 节讨论的分区式管理可以实现虚存吗？如果不能，需怎样修改？试设计一个分区式管理实现虚存的程序流程图。如果能，试说明理由。
- 5.8/ 什么是覆盖？什么是交换？覆盖和交换的区别是什么？
- 5.9/ 什么是页式管理？静态页式管理可以实现虚存吗？
- 5.10/ 什么是请求页式管理？试设计和描述一个请求页式管理时的内存页面分配和回收算法(包括缺页处理部分)。
- 5.11 请求页式管理中有哪几种常用的页面置换算法？试比较它们的优缺点。
- 5.12/ 什么是 Belady 现象？试找出一个 Belady 现象的例子。
- 5.13 描述一个包括页面分配与回收、页面置换和存储保护的请求页式存储管理系统。
- 5.14/ 什么是段式管理？它与页式管理有何区别？
- 5.15 段式管理可以实现虚存吗？如果可以，简述实现方法。
- 5.16/ 为什么要提出段页式管理？它与段式管理及页式管理有何区别？
- 5.17 为什么说段页式管理时的虚地址仍是二维的？
- 5.18 段页式管理的主要缺点是什么？有什么改进办法？
- 5.19/ 什么是局部性原理？什么是抖动？你有什么办法减少系统的抖动现象？

除了进程 0 之外, Linux 的所有进程都是由父进程创建生成的。在父进程创建生成子进程时, 系统要为孩子进程生成自己的上下文, 这就需要调用 6.6 节所述的存储管理程序为其分配 `task_struct` 结构和进程页表等。如果内存中无足够的空间, 或超出了用户资源限额, 则无法生成子进程。在子进程创建成功之后, 马上就看到了子进程和父进程并发执行, 抢占处理机的现象。至于谁先抢得处理机, 要依靠调度程序决定。Linux 的调度策略是基于优先级的, 即选取优先级最高者(优先数最小者) 占据处理机。但是, 优先级的计算又是与时间片有关的。因此 Linux 的调度是基于优先级加时间片的。

另外, Linux 系统没有作业的概念, 用户必须在一个进程的控制之下和系统进行会话, 所输入的数据也必须先送到内存工作区之后, 再由进程决定是否写入外存文件系统保存。

进程通信一直是早期 UNIX 系统较弱的一部分, 但 IPC 机构弥补了这一点。Linux 的 IPC 提供了消息、信号量及共享存储区等大量信息传递方法。不过, IPC 机构只能在一台机器内使用, 不能扩展到网络通信上, 这是它最大的弱点。

Linux 的存储管理策略是请求调页。请求调页系统只把进程的部分页面放入内存, 其他大部分页面在需要时再请求调入。因此, 进程大小不再受内存大小的限制。

系统将一个进程的虚拟空间分为不同的虚拟区域, 每个区对应一个 `vm_area_struct` 结构, 用户进程通过 `mmap` 和 `munmap` 系统调用申请和释放虚拟区域。

习 题

- 6.1 简述 Linux 系统进程的概念。
- 6.2 Linux 进程上下文由哪几部分组成? 为什么说核心程序不是进程上下文的一部分? 进程页表也在核心区, 它们也不是进程上下文的一部分吗?
- 6.3 假定在用户态下执行的某个进程用完了它的时间片, 由于时钟中断的原因, 核心调度一个新进程去执行。请形式化地描述出新、旧进程的上下文切换过程。
- 6.4 Linux 的调度策略是什么? 调度时应该封锁中断吗? 如果不封锁, 会发生什么问题?
- 6.5 试述进程 0 的作用。
- 6.6 Linux 在哪几种情况下发生调度?
- 6.7 编写一个程序, 利用 `fork` 调用创建一个子进程, 并让该子进程执行一个可执行文件。
- 6.8 什么是软中断?
- 6.9 进程在什么时候处理它接收到的软中断信号? 进程接收到软中断信号后放在什么地方?
- 6.10 Shell 符号 `>>` 将输出追加到一个指定的文件中, 如果指定文件不存在, 则该命令创建该文件并将输出写入其中; 否则, 它打开该文件并在该文件中数据尾部接着写入。编写实现 `>>` 的 C 语言代码。
- 6.11 编写一个程序, 比较使用共享存储区和消息机制进行数据传输的速度。
- 6.12 形式化地描述 Linux 中消息机制的通信原理。
- 6.13 Linux 存储管理策略中交换和请求调页方式有何区别?
- 6.14 在图 6.20 所示请求调页的调入处理过程中, 有可能出现空闲页面链表中的页面内容不同于外存设备上的页面内容的情况, 此时应取哪一个页面调入内存? 为什么?
- 6.15 简要总结 Linux 进程管理与存储管理部分的联系。

对文件的存取控制是和文件共享、保护和保密紧密相关的。存取控制可采用存取控制矩阵、存取控制表、口令和密码的方法进行存取验证,以确定用户权限。

最后,本章介绍了文件系统的使用方法和文件系统的层次模型。

习 题

- 8.1 什么是文件和文件系统? 文件系统有哪些功能?
- 8.2 文件一般根据什么分类? 可以分为哪几类?
- 8.3 什么是文件的逻辑结构? 什么是记录?
- 8.4 设文件的结构为多重结构和转置结构的组合,且定义函数 $\text{decode}(K, x)$ 和 $\text{retrive}(K, x)$ 如下:
- 函数 $\text{decode}(K, x)$ 为关键字 K 的搜索函数。其中 K 为待搜索关键字, x 为顺序指针。当被搜索文件为多重结构时,指向关键字 K 的指针只有一个,即 $e(K)$,此时 $x = \text{nil}$,且 $\text{decode}(K, x)$ 返回指针 $e(K)$ 。当被搜索文件为转置结构时,一般指向关键字 K 的指针有 n 个,即 $e_1(K), \dots, e_n(K)$, n 为包含关键字 K 的记录个数。此时,若 $x = \text{nil}$, $\text{decode}(K, x)$ 返回 $e_1(K)$;若 $x = e_n(K)$,则 $\text{decode}(K, x)$ 返回 nil ;否则,若 $x = e_i(K)$, $1 \leq i < n$,则 $\text{decode}(K, x)$ 返回 $e_{i+1}(K)$ 。
- 函数 $\text{retrive}(K, x)$ 是记录搜索函数。其中 K 为指向关键字的指针, x 为记录顺序指针。如果 $x = \text{nil}$ 则 $\text{retrive}(K, x)$ 返回被搜索记录队列的第一个记录的逻辑地址;若 x 等于该记录队列的最后一个记录的话,则 $\text{retrive}(K, x)$ 返回 nil ;否则, $\text{retrive}(K, x)$ 返回 x 的下一个记录的逻辑地址。
- 试问:
- (1) 如果 $\text{decode}(K, x)$ 采用线性搜索法,怎样描述 $\text{decode}(K, x)$?
- (2) 如果 $\text{retrive}(K, x)$ 采用二分搜索法,应怎样排列记录队列? 怎样描述函数 $\text{retrive}(K, x)$?
- (3) 设函数 $\text{search}(K)$ 给出含有关键字 K 的所有记录的逻辑地址,试描述 $\text{search}(K)$ 。
- 8.5 设散列函数 $h(n) = (676 \times l_1 + 26 \times l_2 + l_3) \pmod{t}$, $t = 11$, l_i 为关键字 n 的第 i 个英文字母序号。关键字表长为 11,关键字为英文字母。先按线性散列法,再按平方散列法计算将任意 7 个关键字放入链表中所用的计算次数。这里令 $a = 2, c = 1$ 。
- 8.6 设关键字表按英文字母顺序排列,且两关键字项之间的距离为 d ,写出 $n = 3$ 的三分搜索算法。
- 8.7 文件的物理结构有哪几种? 为什么说串联文件结构不适于随机存取?
- 8.8 设索引表长度为 13,其中 0~9 项为直接寻址方式,后 3 项为间接寻址方式,试描述出给定文件长度 n (块数)后的索引方式寻址算法。
- 8.9 常用的文件存储设备的管理方法有哪些? 试述主要优缺点。
- 8.10 试述成组链法的基本原理,并描述成组链法的分配与释放过程。
- 8.11 什么是文件目录? 文件目录中包含哪些信息?
- 8.12 二级目录和多级目录的好处是什么? 符号文件目录表和基本文件目录表是二级目录吗?

- 8.13 文件存取控制方式有哪几种？试比较它们各自的优缺点。
- 8.14 设文件 SQRT 由连续结构的定长记录组成，每个记录的长度为 500B，每个物理块长 1000B，且物理结构也为连续结构和采用直接存取方式；试按照图 8.23 所示的文件系统模型，写出系统调用 `Read(SQRT, 5, 15000)` 的各层执行结果。其中，SQRT 为文件名，5 为记录号，15000 为内存地址。

断,而且能够并行操作的设备台数也受到中断处理时间的限制,中断次数增多导致数据丢失。DMA 方式和通道方式较好地解决了上述问题。这两种方式采用了外设和内存直接交换数据的方式。只有在一段数据传送结束时,这两种方式才发出中断信号要求 CPU 做善后处理,从而大大减少了 CPU 的工作负担。DMA 方式与通道控制方式的区别是,DMA 方式要求 CPU 执行设备驱动程序启动设备,给出存放数据的内存始址以及操作方式和传送字节长度等;而通道控制方式则是在 CPU 发出 I/O 启动命令之后,由通道指令来完成这些工作。

中断及其处理是设备管理中的一个重要部分。本章在介绍中断的基本概念的同时,对陷阱和软中断也做了相应的介绍和比较。另外,还介绍和描述了中断处理的基本过程。

缓冲是为了匹配设备和 CPU 的处理速度,以及为了进一步减少中断次数和解决 DMA 方式或通道方式时的瓶颈问题引入的。缓冲有硬缓冲和软缓冲之分。本章还介绍了对缓冲池的管理和操作。由于缓冲区是临界资源,所以对缓冲区或缓冲队列的操作必须互斥。

然后,介绍了设备分配的原则和算法。设备分配应保证设备有高的利用率和避免产生死锁。进程只有在得到了设备、I/O 控制器和通道(通道控制方式时)之后,才能进行 I/O 操作。另外,用户进程给出的 I/O 请求中包含逻辑设备号,设备管理程序必须将其变换成实际的物理设备。I/O 请求命令中的其他参数将被用来编制通道指令程序或由设备开关表选择设备驱动程序。

I/O 控制过程是对整个 I/O 操作的控制,包括对用户进程 I/O 请求命令的处理,进行设备分配和缓冲区分配,启动通道指令程序或驱动程序进行真正的 I/O 操作,以及分析中断原因和响应中断等。

习 题

- 9.1 设备管理的目标和功能是什么?
- 9.2 数据传送控制方式有哪几种? 试比较它们各自的优缺点。
- 9.3 什么是通道? 试画出通道控制方式时的 CPU、通道和设备的工作流程图。
- 9.4 什么叫中断? 什么叫中断处理? 什么叫中断响应?
- 9.5 什么叫关中断? 什么叫开中断? 什么叫中断屏蔽?
- 9.6 什么是陷阱? 什么是软中断? 试述中断、陷阱和软中断之间的异同。
- 9.7 描述中断控制方式时的 CPU 动作过程。
- 9.8 什么是缓冲? 为什么要引入缓冲?
- 9.9 设在对缓冲队列 em、in 和 out 进行管理时,采用最近最少使用算法存取缓冲区,即在一个缓冲区分配给进程之后,只要不是所有其他的缓冲区都在更近的时间内被使用过了,则该缓冲区不再分配出去。试描述过程 take_buf(type,number)和 add_buf(type,number)。
- 9.10 试述对缓冲队列 em、in 和 out 采用最近最少使用算法对改善 I/O 操作性能有什么好处。
- 9.11 用于设备分配的数据结构有哪些? 它们之间的关系是什么?

- 9.12 设计一个设备分配的安全检查程序,以保证把某台设备分配给某进程时不会出现死锁。
- 9.13 什么是 I/O 控制? 它的主要任务是什么?
- 9.14 I/O 控制可用哪几种方式实现? 各有什么优缺点?
- 9.15 设备驱动程序是什么? 为什么要有设备驱动程序? 用户进程怎样使用设备驱动程序?

来访问。

习 题

- 10.1 Linux 文件系统的特点是什么？简单描述一个典型的 Linux 系统的目录结构。
- 10.2 Linux 的文件类型有几种？分别是什么？
- 10.3 什么是 VFS？它有什么作用？其通用数据模型是什么？
- 10.4 VFS 包括哪些系统调用？分别简述其功能。
- 10.5 简述文件系统的注册、挂装以及卸载过程。
- 10.6 ext2 文件系统的数据块寻址是如何实现的？
- 10.7 Linux 系统中的设备可分为几种？分别是什么？