第一章 引论

第一节 什么是操作系统(识记)

1. 计算机系统定义:是按用户的要求接收和存储信息,自动进行数据处理并输出结果信心的系统
2. 计算机系统构成:硬件系统和软件系统
3. 硬件系统组成:中央处理器(CPU),主存储器,辅助存储器,各种输入/输出设备

二.操作系统

1.操作系统定义:是一种管理计算机系统资源,控制程序执行,改善人机界面和为其它软件提供支持的系统软件操作系统的两个主要设计原则

2.能使得计算机系统使用方便.

3.能使得计算机高效的工作

第二节 操作系统的形成

1. 控制台:早期,程序的装入,调试以及控制程序的运行都是程序员通过控制台上的开关来实现
2. 原始汇编系统:用汇编语言编写的程序称为源程序,它不能直接在机器上执行,只有通过汇编语言解释程序把源程序转换成用机器指令序列表示的目标程序后才能在计算机上运行.
3. 设备驱动程序:是最原始的操作系统.是一种控制设备工作的程序
4. 管理程序:是初级的操作系统.是一种能对计算机硬件和软件进行管理和调度的程序
5. 操作系统:采用了 SPOOLING 的处理形式

SPOOLING 又称”斯普林”.从本质上说,SPOOLING 是把磁盘作为一个巨大的缓冲器.在一个计算问题开始之前,把计算所需要的程序和数据从读卡机或其它输入设备上预先输入到磁盘上读取程序和数据,同样,对于计算的结果也是先在磁盘上缓冲存放,待计算完成后,再从打印机上打印出该计算问题的所有计算结果

第三节 操作系统的基本类型

按照操作系统提供的服务进行分类,可分为批处理操作系统,分时操作系统,实时操作系统,网络操作系统,分布式操作系统,多机操作系统和嵌入式操作系统等.其中批处理操作系统,分时操作系统,实时操作系统是基本的操作系统

一 批处理操作系统

1. 定义:用户为作业准备好程序和数据后,再写一份控制作业执行的说明书.然后把作业说明书连同相应的程序和数据一起交给操作员.操作员将收到一批作业的有关信息输入到计算机系统中等待处理,由操作系统选择作业,并按其操作说明书的要求自动控制作业的执行.采用这种批量化处理作业的操作系统称为批处理操作系统.
2. 分类

- 批处理单道系统:一次只选择一个作业装入计算机系统的主存储器运行.

批处理多道系统:允许多个作业同时装入主存储器,使中央处理器轮流的执行各个作业,各个作业可以同时使用各自所需要的外围设备

3. 批处理多道系统优点

多道作业并行减少了处理器的空闲时间,既提高了处理器的利用率

作业调度可以按一定的组合选择装入主存储器的作业,只要搭配合理

作业执行过程中,不再访问低速的设备,而是直接从高速的磁盘上存取信息,从而缩短了作业执行时间,使单位时间内的处理能力得到提高

作业成批输入,自动选择和控制 i 作业执行,减少了人工操作时间和作业交接时间,有利于提高系统的吞吐率

- 分时操作系统

1. 定义:能使用户通过与计算机相连的终端来使用计算机系统,允许多个用户同时与计算机系统进行①系列的交互,并使得每个用户感到好像自己独占一台支持自己请求服务的计算机系统.具有这种功能的操作系统称为分时操作系统,简称分时系统
2. 分时技术:既把 CPU 时间划分成许多时间片,每个终端用户每次可以使用一个由时间片规定的 CPU 时间.这样,多个用户就轮流的使用 CPU 时间,如果某个用户在规定的一个时间片内还没有完成它的全部工作,这时也要把 CPU 让给其他用户,等待下一轮再使用一个时间片的时间,循环轮转,直至结束.
3. 分时系统主要特点:

同时性.允许多个终端用户同时使用一个计算机系统

独立性:用户在各自的终端上请求系统服务,彼此独立,互不干扰

及时性:对用户的请求能在较短的时间内给出应答

交互性:采用人机对话的方式工作

- 实时操作系统

定义:能使计算机系统接受到外部信息后及时处理,并且在严格的规定时间内处理结束,再给出反馈信号的操作系统称为实时操作系统,简称为实时系统

1. 设计实时系统注意点

要及时响应,快速处理

实时系统要求高可靠性和安全性,不强求系统资源的利用率

第四节 操作系统的发展

1. 单用户微机操作系统:是指早期的微型计算机上运行的操作系统每次只允许一个用户使用计算机
2. 网络操作系统:为计算机网络配置的操作系统称为网络操作系统.网络操作系统把计算机网络中各台计算机系统有机的联合起来,为用户提供一种统一,经济而有效的使用各台计算机系统的方法,可使各台计算机系统相互间传送数据,实现各台计算机系统之间的通信以及网络中各种资源的共享
3. 分布式操作系统:为分布式计算机系统配置的操作系统称为分布式操作系统.分布式操作系统能使系统中若干计算机相互协作完成一个共同的任务,或者说把一个计算问题可以分成若干个子计算,每个子计算可以在计算机系统中的各计算机上并行执行
4. 多机操作系统:为多处理器系统配置的操作系统称为多机操作系统

5. 嵌入式操作系统:是指运行在嵌入式系统中对各种部件,装置等资源进行统一协调,处理和控制的系统软件(主要特点是微型化和实时性)

第五节 Unix 操作系统简介

1. 诞生

Unix 的第一个版本 version 1 是 AT&TTA 公司下属的 Bell 实验室里两位程序员 Ken Thompson 和 Dennis Ritchie 凭兴趣和爱好 1969 年在一台闲置的 PDP-7 上开发的.

2. 特点

Unix 是一个交互式的分时操作系统

Unix 系统的源代码公开

第六节 操作系统的功能

1. 操作系统的功能:(从资源管理的角度来分)]

处理器管理:对 CPU 进行管理

存储管理:对主存储器进行管理

文件管理:通过对磁盘进行管理,实现对软件资源进行管理

设备管理:对各类输入.输出设备进行管理

2. 操作系统为用户提供的使用接口

程序员接口:通过”系统调用使用操作系统功能(开发者)

操作员接口:通过操作控制命令提出控制要求.

第二章 计算机系统结构简介

第一节 计算机系统结构

一 层次结构

1.计算机系统构成:硬件系统和软件系统

硬件系统构成:中央处理器(cpu),存储器,输入,输出控制系统和各种输入/输出设备

软件系统组成:系统软件,支撑软件,应用软件

2.层次结构:最内层是硬件系统,最外层是使用计算机系统的人,人与硬件系统之间是软件系统.软件系统又依次为系统软件-支撑软件应用软件

二.系统工作框架

1. 引导程序:进行系统初始化,把操作系统中的核心程序装入主存储器,并让操作系统的核心程序占用处理器执行.

2. 操作系统核心程序:完成自身的初始工作后开始等待用户从键盘或鼠标输入命令,每接受一条命令就对该命令进行处理

第二节 硬件环境

一. CPU 与外设的并行工作

在现代的通用计算机系统中,为提高计算机的工作效率,均允许中央处理器和外设并行工作.当执行到一条启动外设的指令时,就按指令中给定的参数启动指定的设备,并把控制移交给输入/输出控制系统,由输入/输出控制系统控制外围设备与主存储器之间的信息传送,外围设备独立工作,不再需要中央处理器干预,于是中央处理器可继续执行其它程序

二. 存储体系

1.寄存器:是处理器的组成部分,用来存放处理器的工作信息.存取速度快,但造价高.

- 通用寄存器:存放参加运算的操作数.指令的运算结构等
- 指令寄存器:存放当前从主存储器读出的指令
- 控制寄存器:存放控制信息以保证程序的正确执行和系统的安全
  - 程序状态字寄存器:存放当前程序执行时的状态.
  - 中断字寄存器:记录出现的事件
  - 基址寄存器:设定程序执行时可访问的主存空间的开始地址
  - 限长寄存器:设定程序执行时可访问的主存空间的长度

2.主存储器:以字节为单位进行编址.主存储器容量较大,能被处理器直接访问,但断电会丢失数据.

3. 高速缓冲存储器:也称 cache,位于处理器和主存储器之间起到缩短存储时间和缓冲存储的作用

4. 辅助存储器:最常用的辅助存储器有磁盘和磁带.优点是容量大且能永久保存信息,但不能被中央处理器直接访问.

三. 保护措施

一般是硬件提供保护手段和保护装置,操作系统利用这些设施配合硬件实现保护

1.指令分类

- 特权指令:不允许用户程序中直接执行的指令.如:启动 i/o,设置时钟,设置控制器等
- 非特权指令:允许用户程序中直接执行的指令

2.cpu 工作状态

- 管态:可执行包括特权指令在内的一切机器指令.一般是操作系统程序占用中央处理器时,cpu 处于管态
- 目态:不允许执行特权指令.一般是用户程序占用中央处理器时,CPU 处于目态.

3.存储保护:不同的存储管理方式有不同的实现保护方法,如可变分区存储管理方式中:基址寄存器的值<=访问地址<=基址寄存器的值+限长寄存器的值

第三节 操作系统结构

一 设计目标

- 正确性:能充分估计和把握各种不确定的情况,使操作系统不仅能保证正确性,且易于验证其正确性

- 高效性:减少操作系统的开销从而提高计算机系统的效率,尤其对常驻主存储器的核心程序部分更要精心设计
- 维护性:当系统发现错误或为提高效率而对算法进行调整等工作时,应使操作系统容易维护
- 移植性:移植性是指能否方便的把操作系统从一个硬件环境移植到另一个新的硬件环境之中.在结构设计时,应尽量减少与硬件直接有关的程序量,且将其独立封装.

二. 操作系统的层级结构

1. 设计方法:无序模块法,内核扩充法,层次结构法,管理设计法等
2. 层次结构法:最大特点是把整体问题局部化.一个大型复杂的操作系统被分解成若干单向依赖的层次,由各层的正确性来保证整个操作系统的正确性.采用层次结构不仅结构清晰,而且便于调试,有利于功能的增加,删减和修改
3. 操作系统层次结构:处理器管理要对中断事件进行处理,要为程序合理的分配中央处理器的工作事件,它是操作系统的核心程序,是与硬件直接有关的部分,因而把它放在最内层.以后的各层依次存放的是存储管理,设备管理和文件管理.即:硬件-处理器管理-存储管理-设备管理-文件管理

- 主要优点:有利于系统的设计和调试
- 主要困难:层次的划分和安排

三 .Unix 系统的结构

1.Unix 层次结构:内核层和外壳层

- 内核层:是 unix 操作系统的核心.它具有存储管理,文件管理,设备管理,进程管理以及为外壳层提供服务的系统调用等功能
- 外壳层:为用户提供各种操作命令和程序设计环境

2.外壳层组成:由 shell 解释程序,支持程序设计的各种语言,编译程序,解释程序,使用程序和系统库等组成.其中其它模块归 shell 解释程序调用,shell 解释程序用来接收用户输入的命令并进行执行.

3.内核层组成:内核程序用 C 语言和汇编语言编写.按编译方式可分为:汇编语言文件,C 语言文件和 C 语言全局变量文件.

4.程序运行环境:用户态和核心态.外壳层的程序在用户态运行,内核层的程序在核心态运行.用户态运行的程序称为用户程序,核心态运行的程序称为系统程序(外壳层的用户程序在执行时可通过系统调用来请求内核层的支持)

第四节 操作系统与用户的接口

一. 操作控制命令

- 联机用户:操作控制命令
- 批处理系统用户:作业控制语言,用来编制作业控制说明书

二. 系统调用

1. 系统调用定义:既系统功能调用程序,是指操作系统编制的许多不同功能的供程序执行中调用的子程序.
2. 执行模式:系统调用在管态下运行,用户程序在目态下运行,用户程序可以通过”访管指令:实现用户程序与系统调用程序之间的转换.(访管指令本身是一条在目态下执行的指令)
3. 系统调用分类:文件操作类,资源申请类,控制类,信息维护类.

第五节 Unix 的用户接口

一. shell 命令

1. 注册和注销
   - 注册:用户可通过 login 输入用户名和通过 password 输入口令,系统注册成功后在 shell 解释程序控制下,出现提示符(采用 C shell 提示符:%)以交互方式为用户服务.
   - 注销:输入 logout 或同时按下 crtl +D 键
2. 常见的 shell 命令
   - Mkdir:请求系统建立一个新的文件目录
   - Rmdir:请求系统删除一个空目录
   - Cd:切换当前的工作目录
   - Pwd:显示用户的当前目录
   - Ls:显示用户一个目录中的文件名.
   - Cp:复制一个文件
   - Mv:对文件重新命名
   - Rm:删除一个指定的文件
   - Cat:显示用 ascll 码编写的文本文件
   - More:分屏显示文件内容,按空格键显示下一屏
3. 后台执行的 shll 命令

- 方法:在请求后台执行的命令末尾输入字符”&”.
- 特点:Unix 把一个程序转入后台执行后,不等该程序执行完就显示可以输入新命令的提示符.因此,允许多个任务在后台执行,也允许后台任务和前台任务同时执行

4. shell 文件

- 定义:用 shell 命令编辑成的文件称为 shell 文件
- 执行 shell 文件:csh shell 文件名
- 把 shell 文件改成可执行文件:chmod+ x shell 文件名-以后就可直接在提示符后面直接输入文件名就可执行

二 Unix 系统调用

1. 常用的系统调用

- 有关文件操作的系统调用

Create:建立文件 open:打开文件

Read:读文件 write:写文件

Close:关闭文件 link:链接一个文件

Unlink:解除文件的链接 lseek:设定文件的读写位置

Chmod:改变对文件的访问权限 rename:更改文件名

- 有关控制类的系统调用

Fork:创建一个子进程 wait:父进程等待子进程终止

Exit:终止子进程的执行 exec:启动执行一个指定文件

- 有关信号与时间的系统调用:Unix 把出现的异常情况或异步事件以传送信号的方式进行

Kill:把信号传送给一个或几个相关的进程

Sigaction:声明准备接收信号的类型

Sigreturn:从信号返回,继续执行被信号中断的操作

Stime:设置日历时间 time:获取日历时间

Times:获取执行所花费的时间

2. trap 指令:是 unix 系统中的访管指令

3. 系统调用程序入口表

- 作用:实现对系统功能调用程序的统一管理和调度
- 构成:系统调用编号,系统调用所带参数个数,系统调用处理程序入口地址,系统调用名称.

4. 系统调用实现过程

  - 步骤一:当处理器执行到 trap 指令时便形成一个中断事件.此时将暂停当前用户程序的执行,而由 unix 系统内核的”trap 处理子程序来处理这个中断事件
  - 步骤二:trap 处理子程序根据 trap 指令中的系统调用编号查系统调用程序入口表,得到该系统调用所带的参数个数和相应的处理程序的入口地址.然后,把参数传送到内核的系统工作区,再按处理程序入口地址转向该系统调用的处理程序执行
  - 步骤三:当系统调用程序完成处理后,仍需返回到 trap 处理子程序,由 trap 处理子程序对被暂停的用户程序进行状态恢复等后续处理,再返回用户程序执行.

第三章 处理器管理

一.什么是多道程序设计

1. 定义:让多个计算问题同时装入一个计算机系统的主存储器并行执行,这种程序设计称为多道程序设计.这种计算机系统称为多道程序设计系统.

2. 注意事项

- 存储保护:必须提供必要的手段使得在主存储器中的各道程序只能访问自己的区域,避免相互干扰
- 程序浮动:是指程序可以随机的从主存储器的一个区域移动到另一个区域,程序被移动后,仍丝毫不影响它的执行(可集中分散的空闲区,提高主存空间的利用率)
- 资源的分配和调度:多道程序竞争使用处理器和各种资源时,多道程序设计的系统必须对各种资源按一定的策略进行分配和调度.

二.为什么要采用多道程序设计

1. 程序的顺序执行:处理器和外围设备,外围设备之间都得不到高效利用
2. 程序的并行执行:让程序的各个模块可独立执行,并行工作,从而发挥外围设备之间的并行能力
3. 多道并行执行:在一个程序各个模块并行工作的基础上,允许多道程序并行执行,进一步提高处理器与外围设备之间的并行工作能力,具体表项在:
   - 提高了处理器的利用率
   - 充分利用外围设备资源.
   - 发挥了处理器与外围设备之间的并行能力

三.采用多道程序设计应注意的问题

1. 可能延长程序执行时间:多道程序设计能提高资源使用效率,增加单位时间的算题量.但是对每个计算问题来说,从算题开始到全部完成所需的计算时间可能要延长
2. 并行工作道数与系统效率不成正比;并不是并行工作的道数越多,系统的效率就越高,而要根据系统配置的资源和用户对资源的要求而定
   - 主存储器空间的大小限制了可同时装入的程序数量
   - 外围设备的数量也是一个制约条件
   - 多个程序同时要求使用同一资源的情况也会经常发生

第二节 进程概述

一.进程的定义

1. 程序:具有独立功能的一组指令或一组语句的集合,或者说是指出处理器执行操作的步骤
2. 进程:是指一个程序在一个数据集上的一次执行
3. 程序和进程的区别:程序是静态的文本,进程动态的过程.进程包括程序和数据集.

二.为什么要引入进程

1. 提高资源的利用率:一个程序被分成若干个可独立执行的程序模块,每个可独立执行的程序模块的一次执行都可看作一个进程,通过进程的同步可提高资源的利用率.
2. 正确描述程序的执行情况:可以方便描述一个程序被执行多次时,各自的执行进度.

三. 进程的属性

1.进程的基本属性

- 进程的动态性
- 多个不同的进程可以包含相同的程序
- 进程可以并发执行
- 进程的三种基本状态 等待态 就绪态 运行态

2.进程的状态变化:运行态-等待态　等待态-就绪态　运行态--就绪态 就绪态-运行态

3.进程特性: 动态性,并发性,异步性

第三节 进程队列

一. 进程控制块

1.进程控制块作用:既 PCB,是进程存在的标识

2.进程控制块构成

- 标识信息:用来标识进程的存在和区分各个进程.进程名
- 说明信息:用于说明本进程的情况.包括:进程状态,等待原因,进程程序存放位置,进程数据存放位置
- 现场信息:用来当进程由于某种原因让出处理器时,记录与处理器有关的各种现场信息,包括:通用寄存器内容,控制寄存器内容,程序状态字寄存器内容
- 管理信息:用来对进程进行管理和调度的信息.包括 进程优先级,队列指针

二. 进程的创建和撤销

1.进程创建:当系统为一个程序分配一个工作区(存放程序处理的数据集)和建立一个进程控制块后就创建了一个进程,刚创建的进程其状态为就绪状态(若执行过程中还缺少资源可以再将其转为等待状态).

2.进程的撤销:当一个进程完成了特定的任务后,系统收回这个进程所占的工作区和取消该进程控制块,就撤销了该进程.

3.原语:是操作系统设计用来完成特定功能且不可中断的过程,包括 创建原语,撤销原语,阻塞原语,唤醒原语.

三 . 进程队列的链接

1.进程队列概念:为了管理方便,进程把处于相同状态的进程链接在一起,称为进程队列

2.进程队列分类

- 就绪队列:把若干个等待运行的进程(就绪)进程按一定的次序链接起来的队列.
- 等待队列:是指把若干个的等待资源或等待某些事件的进程按一定的次序链接起来的队列.

等待队列:是把若干个等待资源或等待某些事件的进程按一定的额次序链接起来的队列

3. 对列实现方法:只需将状态相同的进程控制块链接起来就可以.链接的方式包括单向链接和双向链接.

4. 队列管理:是指系统中负责进程入队和出队的工作

- 入队:是指一个进程进入到指定的队列
  - 从队首入队成为新的队首进程
  - 从队尾入队成为新的队尾进程
  - 插入到队列中某两个进程之间
- 出队:是指一个进程从所在的队列中退出,也存在三种情况

第四节 unix 系统中的进程

一.unix 进程的特点

Unix 中的进程执行用户程序时在用户态执行,执行操作系统程序时在核心态执行.在用户态执行的进程请求系统功能调用时,便转换到核心态执行操作系统程序,当一次系统调用结束时,该进程从核心态的执行返回到用户态执行用户程序

二.Unix 进程的组成

1. 进程控制块:

- 进程基本控制块:用来记录进程调度时必须使用的一些信息,常驻主存储器.把进程基本控制块的数据结构称为 proc 结构
  - 标识信息:包括用户标识(分为实际用户标识号和设置用户标识号)和进程标识.
  - 有关进程非常驻主存部分的信息:用来建立信息在主存与磁盘之间传送.包括:非常驻主存部分的=所在的地址,长度和一些必要的指针.
  - 有关进程调度的信息:包括:进程状态,标志,优先数以及调度有关的其他信息.
  - 其它信息:用于管理和控制的信息,如进程扩充控制块的地址,进程共享正文段和共享主存段的管理信息,进程接收的信号.
- 进程扩充控制块:随用户程序和数据装入主存储器或调出主存储器.把进程扩充控制块的数据结构称为 user 结构.包括:标识,现场保护,主存管理,文件读写,系统调用,进程控制与管理等.

2,正文段:是指 Unix 中可供多个进程共享的程序.系统中设置了一张正文表 TEXT[],用来指正该正文段在主存和磁盘上的位置,段的大小和调用该正文段的进程数等钱情况

3.数据段:包括进程执行的非共享程序和程序执行时用到的数据.

- 用户 zhai 区:是进程在用户态执行时的工作区,主要用于函数调用的参数传递,现场保护,存放返回地址,存放局部变量等.
- 用户数据区:存放进程执行中的非共享程序和用户工作数据.
- 系统工作区:
  - 核心 zhai:是进程在核心态执行时的工作区,主要用于函数调用的参数传递,现场保护,存返回地址,存放局部变量等
  - user 区:存放进程扩充控制块.

三 .Unix 进程的状态

运行状态,就绪状态,睡眠状态,创建状态,僵死状态.

四.unix 进程的创建和终止

1.unix 的进程树:0 号进程(也称交换进程,是系统启动后 unixde 核心程序完成初始化后创建的第一个进程,在核心态运行.用来进行进程调度和让进程在主存与磁盘上进行交换-1 号进程(页称初始化进程,由 0 号进程创建,在用户态运行,用来为终端用户请求注册时创建 login 进程-login 进程(用来处理用户的登录过程,登录成功后创建 shell 进程-shell 进程(等待用户输入命令).

2.进程的创建:在 unix 中,除了 0 号进程和 1 号进程外,其他的进程总是使用系统调用 fork 来创建新进程,形成父子进程.子进程时父进程的一个印像,除了进程的状态,标识和时间有关的控制项外,全部复制父进程的

proc,user,zhai,和数据区

- Fork 的主要工作
- 实现子进程可与父进程执行不同的程序段

3.进程的终止

- 系统调用 exit 的主要任务时把终止进程自被创建以来所占用的系统资源退还给系统.关闭该进程所有打开的文件,释放它对正文段的使用权,把它的 user 结构换出到磁盘对换区后收回时间段占用的主存空间,此后,把终止进程的状态改为”僵死状态”,向父进程发出信号,由父进程作善后处理.
- 系统调用 wait 要对用 exit 请求终止的进程作善后处理,当进程用系统调用 wait 等待其子进程终止时,wait 的任务是先查找处于僵死状态的子进程,若子进程尚未僵死,则让该进程等待,直到子进程成为僵死状态后被释放.进程被释放后,wait 继续执行,再从磁盘对换区把该子进程的 user 结构读入主存缓冲区,释放该 user 再对换区所占的空间,然后,把保存在 user 中的子进程的时间信息加入到本进程的 user 结构中,在释放主存缓冲区,把子进程在 proc[]中的表项清除.

五. unix 进程的换进换出

在 unix 中经常要发生进程在主存与磁盘之间的转换,我们把这项工作称为进程的换进换出,次项工作由交换进程(0 号进程)执行 sched 程序来完成,标志 runout 和 runin 是交换进程的睡眠标志,当磁盘对换区中没有要换进的进程时,标志 runout 置为 1.交换进程睡眠,直到对换区有要换进的进程时被唤醒,当磁盘对换区有就绪进程要换进,但没有足够的主存空间,也没有可换出的进程,则标志 runin 置为 1,交换进程睡眠,直到主存有进程可换出时被唤醒.

六 unix 进程的睡眠与唤醒

- 进程的睡眠: 一般说,进程总是在执行一个系统调用时被确定是否应书面.所以,进程的状态也总是从”核心态运行变成”在主存睡眠”,在确定一个进程需睡眠时,便调用 sleep 程序让进程进入睡眠状态,且将其链入睡眠队列.
- 进程的换醒:通过调用 wakeup 程序来唤醒等待相应事件的进程,被唤醒的进程从睡眠队列退出,状态别修改位就绪,在主存睡眠而被唤醒的进程链入在主存的就绪队列,在磁盘对换区睡眠的进程被唤醒后仍保留在对换区,链入就绪且换出队列.

第五节 中断技术

1.中断基本概念:一个进程占有处理器运行时,由于自身或外界的原因使运行被打断,让操作系统处理所出现的事件,到适当的时候再让被打断的进程继续运行,我们称该进程被中断了,引起中断的事件称为中断源.对出现的事件进行处理的程序称为中断处理程序.

2.中断类型:按中断事件的性质来分

- 强迫性中断事件:这类中断事件不是正在运行的进程所期待的,而是由于外部的请求或某些意外事故而使正在运行的进程被打断
- 硬件故障中断:由计算机故障造成的中断,如电源中断电压超过规定范围
- 程序性中断事件:由执行到程序的某条指令出现的问题引起的中断,如除数为 0
- 外部中断事件:由各种外部事件引起的中断,如用户从终端上输入了一条命令
- 输入,输出中断事件:由来自输入/输出控制系统的事件所引起的中断,如外围设备完成了一次信息传输操作
- 自愿中断事件:这是正在运行的进程所期望的中断事件,是正在运行的进程执行一条”访管指令”请求系统调用为其服务所引起的中断,也称为访管中断.

二.中断响应

1. 概念:若有中断事件发送,则暂停现行进程的执行,而让出操作系统的中断处理程序占用处理器,这一过程称为中断响应

2. 中断装置的具体工作

- 检查是否有中断事件发生
- 若有中断事件发生,则暂停现行进程的执行,且保护好被中断进程的断点以及其他一些信息,以便进程在适当的时候能继续执行
- 启动操作系统的中断处理程序工作

3,中断寄存器:用来记录强迫性中断事件的寄存器.中断装置只要检查中断字寄存器就可知道是否有中断事件发生.若中断字寄存器的内容为 0,则无中断事件发生,处理器继续执行下一条指令;若中断字寄存器的内容为非 0,则表示有中断事件发生.

4.程序状态字和程序状态字寄存器

- 程序状态字 psw:用来控制指令执行顺序并且保留和指示与程序有关的系统状态
- 程序状态字寄存器:在计算机系统中,对每个处理器设置一个用来存放当前运行进程的 psw 的寄存器,该寄存器称为程序状态字寄存器.

5.中断响应

- 当前 psw:存放在程序状态字寄存器中的 psw 是当前正在占用处理器的进程的 psw
- 新 psw:出现中断事件后,要由操作系统的中断处理程序占用处理器,让中断处理程序处理出现的中断事件.我们把中断处理程序的 psw 称为新 psw
- 旧 psw:中断处理程序在占用处理器前,必须把被中断进程的 psw 保护好,以便该进程在适当的时候按被中断时的情况继续执行,我们把保护号的被中断进程的 psw 称为旧 paw

三.中断事件的处理

1. 保护被中断进程的现场信息;把中断时的通用寄存器内容,控制寄存器内容以及被中断装置保存的旧 psw 保存到被中断进程的进程控制块中.

2. 分析中断原因:根据被中断时由中断装置保存的旧 psw 中的中断码可知发生该种中断的具体原因

3. 处理发生的中断事件:在多数情况下中断处理程序只需做一些保护现场,分析事件性质等原则性的处理,而具体的处理可适当的例行程序来完成.

四. 中断优先级和中断屏蔽

1. 中断优先级:一般来说,中断装置是按预定的顺序响应同时出现的中断事件,这个预定的顺序称为中断优先级别.一般情况下,优先级的高低顺序依次为硬件故障中断,自愿中断,程序性中断,外部中断,输入/输出中断.

2. 中断屏蔽:是指中断发生后,若出现优先级别比正在处理的中断级别高是否可再中断,形成中断的嵌套.一般中断屏蔽只屏蔽比自己级别低的中断事件.另外,资源中断时不能屏蔽的.

第六节 unix 系统的中断技术

一. 中断事件和异常情况

1.中断事件:如果出现的事件与正在运行的进程无关,则把这些事件称为中断事件.如/o 中断事件中断事件等.

2.异常情况:如果出现的事件与正在运行的进程有关,则把这些事件称为异常情况,如执行到 trap 指令,地址越界等

二.处理状态字

Unix 用一个由 32 位组成的字作为处理器状态字(记为 ps).16-31 位的内容为特权信息,只允许内核程序对它作修改;0-15 位的内容为非特权信息,允许任何一个进程按规定的含义自由设置和修改.

三.中断处理

1. 中断响应

- 把程序计数器寄存器中的内容(旧 pc)和处理器状态字寄存器中的内容(旧 ps)保存在指定的内部寄存器中.
- 按出现的事件查处理程序入口表,从中取出处理程序的入口地址(新 pc),把它送到程序计数器寄存器中.
- 把取出的处理器状态字(新 ps)送到处理器状态字寄存器中,从而操作系统的处理程序就占用了处理器.

2.中断处理过程

- 处理过程:现场保护-分析处理-恢复现场
- 中断事件与异常情况处理过程的区别;
  - 在对异常情况进行处理时,处理器的中断优先级一般是不改变的.而对中断事件进行处理时,则可以修改处理器的中断优先级

- 对异常情况的处理总是在产生异常情况的哪个进程的核心 zhai 上进行.而对中断事件的处理,则离开产生事件的进程而在系统的中断 zhai 上进行.

第七节　处理器调度

一. 处理器的两级调度

1. 作业调度:是指从输入＃中选取后备作业装入主存储器的工作(作业调度应遵循的必要条件:系统现有的尚未分配的资源可以满足被选作业的资源要求)
2. 进程调度:是指从就绪进程中选取一个进程,让它占用处理器的工作

作业调度和进程调度的关系:任何一个作业,只有先被作业调度选中才有机会去竞争处理器,然后仅当被进程调度选中时才能占用处理器.

二　批处理作业的调度算法

1. 设计调度算法原则
   - ■ 公平性:对用户公平,不能无故或无限制的拖延一个作业的执行
   - ■ 平衡资源使用:尽可能的使系统都处于忙碌
   - ■ 极大的流量:在单位时间内为尽可能多的作业服务,保证计算机系统的吞吐能力

2.周转事件:假定作业 i 进入输入＃的时间为 si.若它被选中执行,得到计算结果的时间为 ei,那么它的周转时间就定位为 ti=ei-si

3.作业调度算法

- ■ 先来先服务算法
- ■ 计算事件短的作业优先算法
- ■ 响应比高者优先算法:响应比=等待时间/计算时间
- ■ 均衡调度算法

三 进程调度算法

1. 进程切换:是指一个进程让出处理器由另一个进程占用处理器的过程.引起进程切换的事件

- ■ 一个进程从运行状态变成等待状态
- ■ 一个进程从运行状态变成就绪状态
- ■ 一个进程从等待状态变成就绪状态
- ■ 一个进程完成工作后被撤销

2. 进程调度算法

- ■ 先来先服务调度算法
- ■ 最高优先级调度算法
- ■ 时间片轮转调度算法
- ■ 分级调度算法:要求掌握调度的原则

四　unix 系统的进程调度算法

1. 进程调度算法:Unix 采用动态优先数调度算法.Unix 中每个进程都有一个优先数,进程的优先数随进程的执行情况而编号.就绪进程能占用处理器的优先级取决于进程的优先数,优先数越小则优先权越高

2. 进程优先权确定原则:

- 进入核心态运行的进程优先权高于在用户态运行的进程优先权
- 一个进程因用完了一个时间片而被剥夺处理器时,应降低该进程的优先权,以使其他进程有机会使用处理器
- 对进入睡眠的进程,系统将按照他们等待事件的轻重缓急赋予他们不同的优先权.
- 应相应降低累计使用处理器事件较长的进程的优先权,以减少这些进程找那用处理器的机会

3. 进程的优先数确定方法

- 设置法:用于即将进入睡眠状态的进程,当进程由于某个事件要睡眠时,根据事件的性质对该进程设置优先数.对因紧迫的事件而入眠的进程设置较小的优先数,一般为负值(其优先级就高);对等待的事件不会产生全局性影响的睡眠进程设置较大的优先数,一般为正值.
- 计算法:当进程转入用户态运行时,则用计算法来确定其优先数,系统对正在占用处理器的进程每个隔一个时钟周期,就会变大其优先数,优先权就相应降低,在进程切换时再次占用处理器的机会就减少.

4. 进程调度程序 SWTCH

- 引起进程调度的原因
  - 进程完成了预定的工作而终止
  - 进程因等待某些事件而进入睡眠状态
  - 进程用完了一个规定的时间片
  - 对捕获到的异常情况处理结束后
- SWTCH 主要任务:在主存就绪的进程中,选择一个优先数最小的进程;为被选中的进程恢复现场信息.

第四章 存储管理

第一节 概述

一. 信息的二级存储

利用辅助存储器提供的大容量存储空间,存放准备运行的程序和数据,当需要时或主存空间允许时,随时将它们读入主存储器.

二. 存储管理的功能

1. 主存储器空间划分

- 系统区:用来存放操作系统与硬件的借口信息,操作系统的管理信息和程序,标准子程序等.
- 用户区:用来存放用户的程序和数据

- 存储管理的对象:主存储器中的用户区(辅助存储器空间的管理属于文件系统的范畴)

2. 存储管理功能:主存空间的分配与回收,实现地址转换,主存空间的共享与保护;主存空间的扩充.

第二节 重定位

一 . 绝对地址和相对地址

1. 绝对地址:是指主存空间的地址编号.绝对地址对应的主存空间称为物理地址空间
2. 相对地址:是指用户程序中使用的地址.相对地址对应的存储空间称为逻辑地址空间.(用户作业都从 0 地址开始往下编写

二.重定位的方式

1. 重定位:也称地址转换,是指把相对地址转换称绝对地址的工作(分为静态重定位和动态重定位)
2. 静态重定位:地址转换的时刻在入主存储器时
3. 动态重定位:地址转换的时刻在入处理器.
4. 动态重定位支持程序浮动,既作业执行时,被改变了存放区域的作业仍然能正确执行.而采用静态重定位时,由于装入主存储器的作业信息已经都是用绝对地址指示,故作业在执行过程中不能移动位置.

第三节 单用户连续存储管理

一,存储空间的分配

1. 存储空间分配方法:处理器中设置一个界限寄存器,存放当前可供用户使用的主存区域的起始地址.作业入主存储器时总是存放到由界限寄存器指示的起始地址开始往下存储.
2. 重定位:采用静态重定位方式进行地址转换,既在作业装入主存时,由装入程序完成地址转换.装入程序只要把界限寄存器的值加到相对地址上就可完成地址转换.
3. 存储保护:主存最大地址>=绝对地址>=界限地址,成立即可执行,否则有地址错误,形成”地址越”的程序性中断事件.
4. 缺点
   - 当作业执行中出现了某个等待事件时,处理器就处于空闲状态,不能被利用.
   - 一个作业独占主存中的用户区,当主存中有空闲区域时,也不能被其它作业利用,降低了主存空间的利用率.
   - 外围设备也不能充分被利用.

二.覆盖技术

1.将作业划分称若干段,其中有一个主段是作业执行过程中经常要用到的信息,而其它段是不会同时工作的

2.主段驻留区:其它段轮流入覆盖区

三 对换技术

在分时系统中,单用户连续存储管理可用兑换方式让多个用户的作业轮流进入主存储器执行,系统中必须要有一个大容量的高速辅助缓冲器,多个用户的作业信息都被保留在磁盘上,把一个作业先装入主存储器让它执行.当执行中出现等待事件或用完成一个时间片时,把该作业从主存储器换出,再把由调度程序选中的另一个作业换入到主存储器中.

第四节 固定分区存储管理

一,基本原理

将主存储器中可分配的用户区域预先划分成若干个连续区,每个分区的大小可以相同,也可以不同.每个分区可用来装入一个作业,但不允许在一个分区同时装入多个作业.

二 主存空间的分配与回收

系统设置一张”分区分配表”,用来说明各分区的分配和使用情况.表中指出各分区的起始地址和长度,并为每个分区设置一个标志位.当标志位为0时表示分区空闲,当标志位为非0是表示分区被占用

三. 地址转换和存储保护

1. 地址转换:采用静态重定位方式
2. 存储保护:处理器设置一对寄存器,既下限寄存器和上限寄存器用来存放当前进程所对应的下限地址和上限地址分别送入下限寄存器和上限寄存器.下限地址<=绝对地址<=上限地址,成立则执行.否则产生地址越界中断

第五节 可变分区存储管理

一. 主存空间的分配与回收

1. 分区的划分:系统初始启动时,主存储器中除操作系统占用部分外,把整个用户区看做一个大的空闲区.当有作业要装入主存储器时,根据作业对主存空间的需要量,从空闲区中划出一个与作业长度一致的分区来装入作业,剩余部分仍为空闲区.
2. 主存空间的分配算法:分区分配表由两张表格组成,一张是”已分配区表”,另一张是”空闲区表”.常用的分配算法:

- 最先适应分配算法
- 最优适应分配算法:按作业要求从所有的空闲中挑选一个能满足作业要求的最小空闲区,这样可保证不去分割一个更大的区域,使装入大作业时比较容易得到满足.
- 最坏适应分配算法:这种算法总是挑选一个最大的空间分割一部分给作业使用,使剩下的部分不至于太小,仍可供分配使用.

3. 主存空间的回收算法:归还区有下邻空闲区;归还区有上邻空闲区;归还区既有上邻空闲区又有下邻空闲区;归还区既无上邻区又无下邻空闲区.

二,地址转换和存储保护

1. 地址转换:采用动态重定位.作业执行过程中,每当取出一条指令后,就把该指令中的相对地址与基址寄存器的内容相加得到绝对地址.
2. 存储保护:基址寄存器内容<=绝对地址<=限长寄存器内容,成立则执行,否则产生”地址越界”中断

三, 移动技术

1. 移动:把作业从一个存储区域移动到另一个存储区域的工作称为移动.
2. 目的:集中分散的空闲去,便于作业动态扩充主存.
3. 注意事项:移动会增加系统开销,移动是有条件的.

第六节 页式存储管理

一. 基本原理

1. 基本原理:把主存储器分成大小相等的许多区,每个区称为一块.与次对应,编程程序的相对地址也分成页,页的大小与块的大小相等.
2. 相对地址构成:页号和页内地址.

二. 存储空间分配与回收

可用一张主存分配表来记录已分配的块和尚未分配的块以及当前剩余的空闲块数.由于块的大小是固定的,所以主存分配表可简化为一张”位示图”,用0表示对应块为空闲,1则为占用.

三. 页表和地址转换

1. 页表:每个作业一张页表,用来指出相对地址中页号和主存中块号的对应关系.
2. 地址转换:采用动态重定位方式.每执行一条指令时,都要由地址转换机构按相对地址

中的页号查页表,得到该页对应的主存块号,再按相对地址中的业内地址换算储欲访问的主存单元的绝对地址,计算

办法:

- 绝对地址=块号*块长+业内地址
- 将块号作为绝对地址的高位,将业内地址作为绝对地址的低位,即可形成绝对地址(因为分块和分页大小是一致的).

3. 块表:
   - 页表缺点:需两次访问主存.第一次按页号读出页表中对应的块号;第二次按计算出来的绝对地址访问主存.
   - 快表:是指存放在高速缓冲存储器中的部分页表.把存放块表的高速缓冲存储器称为相联存储器.

第七节　虚拟存储管理

一, 什么是虚拟存储器

如果能做到只装入作业的部分信息就可以让作业开始执行,那么当主存空间小于作业需求量时,系统就可以接受该作业,进而也就可以允许逻辑地址空间大于实际的主存空间,这样就带来了好处 ,第一,使主存空间能充分的利用;第二 从用户的角度来看,好像计算机系统提供了容量很大的主存储器,虚拟存储器实际上是夸大主存容量而采用的一种管理技巧.

二. 虚拟存储器的工作原理

1,工作原理:把作业信息保留在磁盘上,当要求装入时,只将其中一部分先装入主存储器,作业执行过程中,若要访问的信息不在主存中,则在设法把这些信息装入主存.

2.程序特点

- 程序执行时有些部分是彼此互斥的,即在程序的一次执行中,执行了这部分就不会区执行另一部分
- 程序的执行往往具有局部性,即在一段时间里可能循环执行某些指令或多次访问某一部分的数据

3. 实现虚拟存储器的关键问题
   - 怎样知道当前哪些信息已在主存储器中,哪些信息尚未装入主存储器中
   - 如果作业要访问的信息不在主存储器中,怎样找到这些信息并把它们装到主存储器?
   - 在把欲访问的信息装入主存储器时,发现主存中已无空闲块又该怎么办

三 页式虚拟存储器的实现

1. 实现原理:将作业的全部信息作为副本存放在磁盘上,作业调度选中一个作业时,至少把作业的第一页信息装入主存储器.在作业执行过程中欲访问不在主存储器中的页时,再把它们装入.为次,页表需更改,至少应包括页号,标志,主存块号,磁盘上的位置

2. 页面调度:
   - 页面调度概念:是指采用某种算法选择一页暂时调出,把它存放到磁盘上区,让出主存空间,用来存放当前要使用的页面的这已过程称为页面调度.
   - 抖动:也成颠簸,是指刚被调出的页面又立即要用,因而又要把它调入;而调入不久又被调出,调出不久又被再次调入.如此反复,是调度非常频繁,以至于使大部分时间都花费在来回调度上
   - 页面调度算法

➢ 最佳调度算法:OPT　总是把以后不再访问的页或距当前最长时间后再访问的页先调出.是一种理想算法,用来用做衡量其他算法的标准.

- 先进先出调度算法:FIFO 总是把先进入主存储器的页面调出.
- 最近最久未使用调度算法:LRU 距当前最长时间内没有使用过的页面先调出.
- 最近最不经常使用调度算法:LFU 在最近一段时间内使用次数最少的页面先调出.

3. 缺页中断率

- 定义:如果作业执行中访问页面的总次数为 A,其中有 F 次访问的页面尚未转入主存,故产生了 F 次缺页中断.现定义缺页中断率 F:F=F/A
- 影响缺页中断率的因素:分配给作业的主存块数;页面大小;程序的编制方法;页面调度算法.

四.多级页表

1. 原理:建立页表时,第一级是页面组表(称为一级页表,每个作业一张) ,第二级是组内页面表(称为二级页表,没组一张).一级页表指出二级页表的存放地址,二级页表指出每个页在主存块中的块号.
2. 地址转换方法:采用二级页表结构的系统总是把页表保存在辅助存储器中,程序执行时只需把一级页表先转入主存储器.进行地址转换时,按相对地址中的页号 I 查一级页表,找出对应的表项,再根据表项中的标志位可以知道对应的二级页表是否已在主存中,若已在主存,则可按页号 ii 查二级页表中的表项,得到页所在位置(已在主存或尚未装入主存).若二级页表尚未装入主存,则应先将其装入,再按页号 II 查找页所在位置.若页已在主存中,则根据对应的主存块号和相对地址中的业内地址得到当前要访问的主存绝对地址,否则需将该页先调入主存再进行地址转换.
3. 优缺点:有利于主存空间的利用,但会增加访问主存的次数,会影响指令执行速度.

第八节 unix 系统的页式虚拟存储管理

一.unix 的虚拟地址结构

1.虚拟地址:Unix 采用页式虚拟存储管理,并把编程序时用的地址称为虚拟地址.

2.地址空间划分:系统区间,程序区间,控制区段.操作系统在系统区段运行,正文段和非共享程序都在程序区段,控制区段用来存放用户栈,核心栈,usr 区等.系统区段中的程序和数据常驻内存,其余两个区段中的信息可随执行情况在主存和磁盘兑换区之间换进换出.

3.虚拟地址结构:最高两位表示访问的区段,最低 9 位表示业内地址(称为字节偏移量)中间 21 位表示页号(称为虚拟页号)

二.Unix 的页表和地址转换

1. 页表结构:

- “v”为有效位:如果此位为 1,表示这个虚拟页已经在主存,其主存位置由物理页号指定;如果此位为 0,则进程在访问该页时硬件将产生缺页中断,由操作系统进行处理.
- “M”为修改位,指出该页是否被修改过.若 M 为 1,表示该页被修改过,被修改过的页面被调出时必须将其写回到磁盘上,否则不需要重写到磁盘上.
- “prot”四位是访问权限指示位.进行地址转换时都要核查这四位.如果一个进程超出了规定的访问权限,硬件将产生一个”非法访问”中断.

物理页号:既主存块号.

2. 地址转换:

硬件为每个区段设置一对页表寄存器,分别用来存放该区段页表的起始地址和长度.在进行地址转换时,首先判断虚拟地址区段,然后找到该区段的页表寄存器,按寄存器中指示的地址就可找到该区段的页表,只要虚拟地址在指定的长度范围内就能在页表中找到该页的登

记项.检查登记项中的有效位,若为 1,则该页已在主存,把等级项中的物理页号作为绝对地址的高地址部分,把虚拟地址中的字节偏移量作为绝对地址的低地址部分.

三 unix 的页面调度

1. 优化措施

- 一个正在与外围设备交换信息的页面或一个正在被装入的页面是不能被替换的
- 页面页面调度采用二次机会页面替换算法
- 为了装入一个新页面而要调出一页时,要检查被调出页的修改位标志,若该页被修改过,则调出时必须把该页的内容写会磁盘上,否则就不必写回磁盘,以减少输入输出传送.
- 系统中有一个 2 号进程,unix 把它称为页面守护进程,它的作用是保证有足够的空闲物理页可供使用,一般它都处于睡眠状态

2.2 次机会页面替换算法实现要点

3.页面进程守护进程职责

第五章 文件管理

第一节 文件和文件系统

1.文件:是指逻辑上具有完整意义的信息集合,每个文件都要用一个名字作标识,称为文件名.
2.文件分类:文件标准不一样,文件可分为不同的类型.
二 .系统的组成部分
1.文件系统的目的:对文件统一管理,目的是方便用户且保证文件的安全可靠,面向用户,文件系统主要实现”按名存取”
2.文件系组成部分:文件目录.文件的组织,文件存储空间的管理,文件操作,文件的安全措施
3.文件操作:为了保证文件系统能正确的存储和检索文件,系统规定了一个文件上可执行的操作,这些可执行的操作称为文件操作.基本操作有建立文件,打开文件,读文件,写文件,关闭文件和删除文件

第二节 文件的存储介质

1. 存储介质:是指用来记录信息的磁带,硬磁盘组,软磁盘片,光盘,卡片等
2. 存储设备:是指可安装存储介质的设备如磁带机,磁盘驱动器.卡片机.
3. 卷:我们把存储介质的物理单位定义为卷,如一盘磁带,一张软盘片等.
4. 物理记录:也称为块,是指存储介质上可连续存储信息的一个区域,块是住存储器与存储设备进行信息交换的物理单位.
5. 磁盘机:是一种按地址直接存取的存储设备.信息定位方式:柱面号,磁头号,扇区号.每个参数均从 0 开始

第三节 文件的存取方式

一.存取方式分类
1. 顺序存取:是指对文件中的信按顺序一次进行读写的存取方式
2. 随机存取:是指对文件中的信息不一定是按顺序读写,而是可以按任意的次序随机的读写的存取方式.
二.存取方式选择
1. 与文件的使用方式有关:文件的性质决定了文件的使用,也就决定了存取方式的选择.如源程序文件必须按字符顺序进行存取,数据库的访问则采用随机存取方式

2.　与存储介质的特性有关:磁带机适合顺序存取的存储设备;磁盘机既可采用顺序存取方式,又可采用随机存取方式.

第四节 文件目录

一 .一级目录结构

1. 基本思想:把一卷存储介质上的所有文件都登记在一个文件目录中.

2. 要求:在文件目录中登记的各个文件都有不同的文件名.

二. 二级目录文件

1.　用户文件目录:是二级目录结构中为每个用户设置的一张目录表.

2.　主文件目录:是一张用来登记各个用户的目录表存放地址的总目录表.

3.　优点:

- 采用二级目录结构后,即使不同的用户在为各自的文件命名时取了相同的名字也不会引起混乱.
- 采用二级目录结构可使不同的用户共享某个文件,这只要在各用户的文件目录表中使某个目录项指向共享文件存放的物理位置即可.

三. 树形目录结构

1.　多级目录结构:也称树形目录结构

2.　绝对路径:路径名可以从根目录开始到该文件的通路上所有各级子目录及该文件名顺序拼起来组成.各子目录与文件名之间用\隔开

3.　相对路径:访问文件时,从当前目录开始设置路径

4.　树形目录结构优点:解决了重名问题;有利于文件分类;提高检索文件的速度,能进行存取权限的控制.

四 文件目录的管理

1.　目录文件:通常把文件目录页作为文件保存在存储介质上,由文件目录组成的文件称为目录文件.

2.　文件目录的管理:文件系统可以根据用户的要求从目录文件中找出用户的当前目录,把当前目录读入主存储器,这样既不占用太多的主存空间,又可减少搜索目录的时间

第五节　文件的组织

一. 文件的逻辑结构

1.文件的逻辑结构:用户从使用的角度来组织文件,用户把能观察到的且可以处理的信息根据使用构造文件,这种构造方式是独立与物理环境的,所以称为文件的逻辑结构.

2.逻辑文件:是指用户组织的文件.统称问文件

3.逻辑文件分类:

- 流式文件:是指用户对文件中的信息不再划分可独立的单位,整个文件是由依次的一串信息组成.
- 记录式文件:是指用户对文件中的信息按逻辑上独立的含义再划分信息单位.区分概念:逻辑记录,逻辑记录号,主键,次键.

二. 文件的存储结构

1.文件的存储结构:文件系统从文件的存储和检索的角度来组织文件,文件系统根据存储设备的特性,文件的存取方式来决定以怎样的形式把把用户文件存放到存储介质上,在存储介质上的文件构造方式称为文件的存储结构

2.物理文件:是指存放在存储介质上的文件

3.磁带文件的组织:

- 组织方式:磁带机是一种顺序存取设备,因此组织在磁带上的文件都采用顺序结构
- 磁带文件的组成

- 文件头标:用来标识一个文件和说明文件的属性
- 文件信息:是用户逻辑文件中的信息
- 文件尾标:用来表示一个文件的信息结束
- 文件与文件之间用一个带标隔开,用两个带标标识磁带上的有效信息到此结束.

4.磁盘文件的组织:

●顺序结构

➢优点:顺序结构适合顺序存取,其优点是存取信息的速度快,存取文件不必每次去查找信息的存放位置,只要记住当前块号,则该文件的后继信息一定在下一块中,减少了检索时间

➢缺点:磁盘存储空间的利用率不高,对输出文件很难估计需要多少磁盘块;影响文件的扩展

●链接结构

➢链接结构:顺序的逻辑记录被存放在不相邻的磁盘块上,再用指针把这些磁盘块按逻辑记录的顺序链接起来,便形成了文件的链接结构

➢链接文件:也称串联文件,是指采用链接结构的文件

●注意事项

➢对链接文件采用顺序存取方式是高效的采用随机存取方式是低效的.

➢在插入或删除一个记录时,若某一块中的指针需修改,则应先把该块内容读到主存储器,再修改指针,然后把修改后的该块信息按原地址重新写回磁盘

➢每一个磁盘块中既要存放文件信息,又要存放用于管理的指针,这会增加文件需要占有的磁盘块数

➢读写磁盘上的信息以块为单位,当读出一块信息后,应把其中的指针分离,仅把属于逻辑文件的信息传送给用户,以保证用户使用文件信息的正确性

➢在存取文件时,如果某个指针丢失或被破坏,则错误的指针可能指向其它文件而导致混乱.(可采用双指针)

●索引结构:每一个文件建立一张索引表,把指示每个逻辑记录存放位置的指针集中在索引表中

➢索引文件:是指采用索引结构的文件

➢特点:索引结构既适合顺序存取记录,又可方便的按任意次序随机存取记录,且容易实现记录的增,删,和插入.但采用索引结构必须增加索引表占用的空间和读写索引表的事件

5.存取方式与存储结构之间的关系

三. 记录的成组与分解

1. 记录的成组:把若干个逻辑记录合成一组存入一块的工作称为记录的成组,每块中的逻辑记录个数称为块因子.

2. 记录的分解:从一组逻辑记录中把一个逻辑记录分离出来的操作

3. 优缺点

●优点:提高存储空间的利用率,减少启动外设的次数,提高系统的工作效率

●缺点:是以设立主存缓冲区和操作系统增加成组与分解操作的功能为代价

第六节 **磁盘存储空间的管理**

一. 位示图法

二. 空闲块表法

系统为每个磁盘建立一张空闲块表,表中每个登记项记录一组连续空闲块的首块号和块数.空闲快数为 0 的登记项为无效登记项

三. 空闲块链法

把所有的磁盘空闲块用指针链接在一起构成空闲块链.分配空间时从链中取出空闲块,归还空间时把归还块加到链中,对磁盘空间块可用单块连接法连接起来.每一个空闲块中都设置一个

指向个空闲块的指针,最后一个空闲块中的指针为 0

第七节 基本文件操作及其使用

一. 基本文件操作

要求掌握相应的参数和主要工作

1. 建立操作
2. 打开操作
3. 读操作
4. 写操作
5. 关闭操作
6. 删除操作

二 文件操作的使用

1,读一个文件信息,打开文件-读文件-关闭文件

2.写一个文件信息:建立文件-写文件-关闭文件

3.删除一个文件:关闭文件-删除文件

第八节 文件的安全性

一 文件的保护

1. 防止天灾人祸造成的破坏:采用建立多个副本的办法
2. 防止系统故障造成的破坏:采用建立副本和定时转储的办法
3. 防止文件共享时造成的破坏:采用不允许同时使用;或允许同时使用但限制对文件使用的权限
4. 防止计算机病毒的侵害:采用针对各种病毒涉及相应的杀毒软件,还可在二进制文件的目录中设置一般用户只能读的权限

二 文件的保密

1,隐藏文件目录

2.设置口令

3.使用密码

第九节 unix 系统的文件管理

一. Unix 的文件和文件系统

1. 文件的逻辑结构:是一串顺序的字符流组成的流式文件
2. 文件的存储结构:采用索引结构方式,既把文件按一定的长度分块后存放到磁盘,并建立索引表
3. 文件的分类:普通文件,目录文件和设备文件
4. 文件系统:分成基本文件系统和可装卸的子文件系统两部分

二. Unix 的文件结构

1. 磁盘空间划分:分成 512 个字节的许多块
2. 文件结构:采用多级索引结构.规定每个文件的索引表使用 13 个登记项,前 10 个登记项直接指出存放文件信息的磁盘块号,如果 10 个磁盘块不够容纳该文件信息,则利用第 11 分登记项指向一个磁盘快.对于大型文件还可利用第 12 和第 13 两个登记项作为二级和三级间接索引.

三. Unix 的文件目录

1. 目录结构:unix 采用树形目录结构
2. 目录文件存取权限

- 读:可以读该目录
- 写:可以增,删,这个目录中的目录项,从而改变目录内容

- 执行:可以为寻找一个文件而搜索这个目录

四 unix 的索引节点

1. 索引节点概念:也称为 i 节点或 i_node,是把目录项中用于对文件进行控制和管理的信息分离出来,单独组成一个数据结构

2. 磁盘块分类

- 存放索引节点的索引节点区
- 存放文件信息的文件存储区

3. 索引节点区管理:对索引节点区中的磁盘块进行编号,其中每一个磁盘块可以用来存放一个索引节点.索引节点区中的第一个磁盘块存放文件系统的根目录文件的索引节点,它的索引节点号就为 1.其余的磁盘块岁文件的建立和删除进行分配和回收.

4. 磁盘索引节点:存放在磁盘上索引节点区中的索引节点称为磁盘索引节点,磁盘索引节点实际上是描述文件的第一个抽象数据结构,其中的 di addr 指向文件的索引表,从而可以找到文件存储的具体位置.

5. 活动索引节点表:Unix 在主存开辟一个索引节点缓冲区,用来建立活动索引表 inode[],inode[]有 100 个表项,能暂存 100 个索引节点.

6. 目录索引:文件目录中的每个目录项由 16 个字符组成,其中 14 个字符为文件名,是该文件的外部标识,另两个字符是文件的索引节点号,它是该文件的内部标识.因此,目录索引的流程为:按文件名查找目录项-取得对应的索引节点号区找相应的磁盘索引节点-将磁盘索引节点复制到-活动索引节点表中-通过活动索引节点可以找到索引表-按照索引表中的对应情况存取磁盘上的相应信息.

五.Unix 的打开文件表

1. 系统打开文件表 flie[];反应对被打开文件进行操作的动态信息,主要用于对文件读写操作的控制,共有 100 个表项.

2. 进程打开文件表:用来记录每个进程打开的文件情况.Unix 系统规定每个进程最多同时打开 15 个进程.

六 .Unix 的文件操作

1. 建立文件 create
2. 打开文件 open
3. 读文件 read
4. 写文件 write
5. 关闭文件 close
6. 系统调用 link:为一个文件再取一个新文件名
7. 系统调用 unlink:删除文件的一个文件名

七. unix 的文件存储空间管理

Unix 把磁盘上的用来存放文件信息的磁盘块采用成组链接的方法进行管理.成组链接法是把空闲块分成若干组,把指向一组中个空闲块的指针集中在一起.unix 规定定,每 100 个空闲块为一组,每组的第一个空闲块中登记下一组空闲块的磁盘物理块号和空闲块数,最后不足 100 块的那部分磁盘物理块号及块数记入专用块中.

第六章 设备管理

第一节 设备管理的功能

1,外围设备分类

存储设备:磁盘机,磁带机

输入输出:显示器,输入机,打印机

2.输入/输出操作:是指主存储器与外围设备直接的信息传送操作

对存储性设备:输入/输出操作的信息传送单位为"块"
对输入输出性设备:输入.输出操作的信息传送单位为"字符:
3.设备管理的主要功能
实现对外围设备的分配与回收
实现对外围设备的启动
实现对磁盘的驱动调度
处理外围设备的中断事件
实现虚拟设备

第二节 外围设备的分类

1.独占设备:是指在作业执行期间只允许一个作业独占使用的设备,如输入机,打印机等
2.可共享设备:是指可让若干个作业同时使用的设备.如磁盘机等
独占设备的管理

第三节 **独占设备的管理**

1.设备的绝对号:为了对设备进行管理.计算机系统对每一台设备都要进行登记,且为每一台设备确定一个编号以便区分和识别,这个确定的编号称为设备的绝对号
2.设备的相对好:由用户对自己需要使用的若干太同类设备给出的编号称为设备的相对号
二 .独占设备的分配
1.设备申请的指定方式

- 指定设备的绝对号:系统就必须把对应的设备分配给作业,且不可替代
- 指定设备类和相对号:可以实现用户编制程序时使用的设备与实际能占用的设备无关

2.设备的独立性:是指用户程序中使用由:设备类,相对号"定义的逻辑设备,可以实现用户编制程序时使用的设备与实际能占用的设备无关,该种使用设备的特性称为设备的独立性..
3.设备独立性优点:
1.系统只要从指定的那一类设备中找出"好的且尚未分配的"设备来进行分配.
2.万一用户使用的设备出了故障,系统就可以从同类设备中找另一台"好的且尚未分配"的设备来替换
4.独占设备的分配:系统可设置设备分配表,指出系统所配置的独占设备类型,数量以及分配和使用的情况等,该表由两部分组成;

- 设备类表:每一台独占设备在设备类表中占一个登记项.表项包括设备类,拥有的总台数,现存台数,设备表始址.
- 设备表:每一台设备在设表中占一个等级项.表项包括绝对号.好/坏.已/未分配.占用作业名,相对号

第四节 **磁盘的驱动调度**

一.访问磁盘的操作时间
1.输出/输入操作所需花费的时间

- 寻找时间:磁头在移动臂带动下到指定柱面所需的时间
- 延迟时间:指定扇区旋转到磁头位置所需的时间
- 传送时间:由指定的磁头把磁道上的信息读到主存储器或把主存储器中信息写到磁道上所需的时间.由于块的长度一样,因此传送时间是固定的

2.驱动调度

- 移臂调度:是指根据等待访问者指定的柱面位置来决定次序的调度,目标是尽可能的减少寻找时间
- 旋转调度:是指根据延迟时间来决定执行次序的调度,目标是尽可能的减少延迟时间

二.移臂调度

1.先来先服务调度算法
2.最短寻找时间优先调度算法
3.电梯调度算法
4.单向扫描调度算法
三.旋转调度
1.若干请求者要访问同一磁头下的不同扇区,旋转调度总是先对到达读写磁头位置下的扇区进行信息传送
2.若干请求者要访问不同磁头下的不同编号的扇区,旋转调度总是对先到达读写磁头位置下的扇区进行信息传送
3.若干请求者要访问不同磁头下具有相同编号的扇区:旋转调度根据磁头号可从任意选择一个磁头进行读/写操作,其余的请求者必须等磁盘再次把扇区旋转到磁头位置时才有可能被选中
信息的优化分布:可减少驱动调度

第五节 输入/输出操作的实现

一.通道结构和通道程序
1.通道结构:也称输入/输出处理机,时计算机系统中用来连接外围设备,并能独立工作的系统
2.通道的连接:一个中央处理器可以连接多个通道;一个通道可以连接多个设备控制器;一个设备控制器可以连接同类型的多台设备
3.通道的基本工作模式:当有输入输出请求时,中央处理器先执行"启动 i/o"指令,启动指定通道上的指定设备,当启动成功,通道按规定的要去通过设备控制器控制外围设备进行操作.这时中央处理器就可执行其它任务,并与通道并行工作,直到输入输出操作完成,由通道发出操作结束的"i/o 中断:时中央处理器
才暂停当前的工作,转去处理 i/o 中断事件.
4.通道命令 CCw:每一条通道命令规定了设备的一种操作.操作系统可以用若干条通道命令来规定通道执行一次输入/输出操作应做的工作.若干条通道命令组成了一个通道程序.
5.通道地址字 CAW:用来存放通道程序首地址的主存固定单元.通道系统可以依据该地址去取相应的通道命令
6.通道状态字 csw:用来存放通道和设备执行情况的主存固定单元

- 通道命令地址:用来指示下一条通道命令在主存中的地址,因为通道地址字只存放通道程序的首地址
- 设备状态:用来记录设备控制器和设备的工作情况
- 通道状态:用来记录通道的工作情况
- 剩余字节个数:是指最后一次执行的那条通道命令在操作结束后还剩余多少字节未传输

二.外围设备的启动
1.输出/输出操作的过程

- 准备阶段
- 启动 i/o 阶段
- 结束处理阶段

2.设备处理的一致性:在具有通道结构的计算机系统中,从启动外围设备到完成输入/输出操作都没有考虑不同类型的物理设备的特性,都采用了统一的方法在进行处理.这种不考虑设备的具体物理
特性(实际上设备特性已隐含在通道程序中)的处理方法成为设备处理的一致性.
三.I/O 中断事件的处理
1.作用:I/o 中断是通道和中央处理器协调工作的一种手段
2.I/o 中断事件分类

- 操作正常结束
- 操作异常结束

A.设备故障
B 设备特殊

第六节 缓冲技术

1.缓冲技术
缓冲技术:是指操作系统中,把利用缓冲区来缓解处理器与外围设备之间工作速度不匹配的矛盾而采用的技术
2.单缓冲技术:是指操作系统在主存储器的系统区中设立一个缓冲区
3.双缓冲技术:是利用两个缓冲区来完成输入输出操作的工作,两个缓冲区交替使用
4.缓冲池技术
操作系统可以在主存中设置一组缓冲区,把这一组缓冲区称为缓冲池.缓冲池中的各缓冲区是系统的公共资源,可供各进程共享,并由操作系统统一分配和管理

第七节 虚拟设备

一 .为什么要提供虚拟设备
1.虚拟设备概念:操作系统利用共享设备来模拟独占设备的工作,当系统只有一台输入设备和一台输出设备的情况下,可允许两个以上的作业并行执行,并且让每个作业都感觉到获得了供自己独占
使用的输入设备和输出设备,我们说,操作系统采用的这种技术为用户提供了"虚拟设备"
2.为什么要提供虚拟设备,我们已经知道像输入机,打印机等独占使用的设备采用静态分配方式,既不能充分利用设备,又不利于提高系统效率,主要表现为

- 占有输入机和打印机的作业,只有一部分时间在使用他们,其余时间这些设备处于空闲状态,在设备空闲时不允许其它作业去使用他们,因此不能有效利用这些设备.
- 当系统只配有一台输入机和一台打印机时,就不能接受两个以上要求使用输入机和打印机的作业同时执行,不利于多道并行工作
- 这些独占设备大多是低速设备,在作业执行中往往由于等待这些设备的信息传输而延长了作业的执行时间

二.虚拟设备的实现
1.基本条件:实现虚拟设备必须由一定的硬件和软件条件为基础,对硬件来说,必须配置大容量的磁盘,要有中断装置和通道,具有中央处理器与通道并行工作的能力,对操作系统来说,应采用多道程序设计技术
2.实现原理:把一批作业的全部信息通过输入设备预先传送到磁盘上等待处理,在多道程序设计系统中,可以从磁盘上选择若干个作业同时装入主存储器,并让他们同时执行,由于作业的信息已全部在磁盘上,
故作业执行时不必再启动输入机读信息,而可以从共享的磁盘上读取各自的信息,把作业产生的结果也存放到磁盘上,而不直接启动打印机输出,直到一个作业得到全部结果而执行结束时,才把该作业的结果从打印机输出
3.实现技术

- 输入＃和输出＃

➢ 输入＃:存放作业的初始信息
➢ 输出＃:存放作业的执行结果机文缩写为 sppooling,

- 预处理程序:任务是把作业流中每个作业的初始信息传送到输入保存以备作业执行时使用
- ＃管理程:＃管理程序又分成＃管理读程序和＃管路写程序两个功能,当作业请求从输入机上读文件信息时,就把任务转交给＃管理读程序.当它从输入＃读出信息供作业执行时

使用.当作业请求从打印机上输出结果时,就把任务转交给＃管理写程序,由它把产生的结果保存到输出＃中

- 缓输出程序:缓输出程序负责查看输出＃中是否有待输出的结果信息.若有,则启动打印机把结果文件打印输出

3.实现技术

- 数据结构
  - 作业表:用来登记输入＃中的各个作业的作业名,作业状态,作业拥有的文件数,以及预输入表和缓输出表的位置等
  - 预输入表:每个作业都有一张预输入表,用来登记改作业的初始信息中的各个文件,指出哥文件的文件名,传输文件信息时使用的设备类型,文件的长度以及文件的存放位置等
  - 缓输出表:对每个作业设置一张缓输出表,用来登记该作业产生的结果文件,作业产生的结果也按链接结构组织成文件存放在输出中
  - 输入＃中作业的状态

1. 输入状态:预输入程序启动了输入机中正在把作业的信息传输到输入
2. 收容状态:该作业的信息已经存放在输入＃中,但尚未被选中执行
3. 执行状态:作业已被选中并装入主存储器开始执行
4. 完成状态:作业已执行结束,其执行结果在输出＃中等待打印输出

第八节 unix 的设备管理

一.unix 的设备和设备文件

1.块设备和字符设备:按设备与主存之间信息交换的物理单位来分

- 块设备:也称存储设备,已块为单位与主存交换信息,是用来存储信息的设备.如磁盘机,磁带机等
- 字符设备:也称输入/输出设备.以字节为单位与主存交换信息,是用来接受外部信息或把处理好的 信息传向外部的设备,如终端,打印机等.

- 主设备号:是 unix 对每一类设备给出的一个编号
- 次设备号:是为了标识某一台具体设备,unix 对每一台设备给出的一个编号
- 请求设备工作时,必须给出主设备号和次设备号.系统根据主设备号决定哪个驱动程序工作.驱动程序再根据设备号确定控制哪些具体的设备工作

3.设备文件:

unix 把设备当作特别文件,每个设备特别文件也都有一个索引节点.索引节点中的文件类型被定义为"块"或"字符",一则区别于其它文件,二则用已确定该索引节点代表的是块设备文件还是字符设备文件.设备文件登记在根目录下的 dev 子目录中.把设备定义成文件后,就可以对设备象文件那样操作,对设备的使用也是通过系统调用 open,close,write 来进行.这些系统调用的内部实现与普通文件的系统调用的实现相似,但要增加一些与设备有关的处理过程

二.unix 的块设备缓冲技术

1.unix 采用的缓存技术类型

系统设置一批缓冲区,构成系统缓冲区池.对写操作,驱动程序分配一个缓冲区,先把信息从用户空间复制到缓冲区,再启动设备把缓冲区中信息保存起来,对读操作,启动设备后把指定的信息传送到一个缓冲区中,再将
信息复制到指定的用户空间.这样做的目的是为了减少启动设备的次数.

2.缓冲区构成

缓冲区数据区:用于存放文件信息

缓冲区控制块:用于缓冲区的管理,主要管理信息

- 状态标志 b_flags;描述该缓冲区当前的状态

- 队列指针:共有四个队列指针.包括:av forw:空闲缓冲区队列前向指针;av back:空闲缓冲区队列后向指针;b_forw:设备缓冲区队列前向指针;b_back;设备缓冲区队列后向指针.
- 设备号 dev:缓冲区数据区信息所属设备的设备号,包括主设备号和次设备号.
- 字节数 b bcount:请求传送的字节数
- 块号: b blkno:指定设备上的块号.
- 存放地址 b addr:存放信息的主存起始地址

3.缓冲区队列

- 空闲缓冲区队列:空闲缓冲区队列又称空闲 ac 链.它是由指针 av_forw 和 av_back 形成双向链接队列.该队列中所以缓冲区的状态标志 BUSY 位为 0.表示都是空闲缓冲区.系统初始启动时将所以缓冲区

按序号挂在 av 链上

- 设备缓冲区队列:设备缓冲区队列又称设备 b 链.它是由指针 b_forw 和 b_back 形成的双向链接队列.该队列链接设备正在使用或已经使用过的缓冲区.可为每个设备设立一个设备缓冲区队列.

4.缓冲区的管理原则

(1)当需要一个缓冲区时，总是从空闲缓冲区队列的队首取一个缓冲区。一个被使用过的缓冲区释放时，总是排入空闲缓冲区队列的队尾。

（2）一个缓冲区被分配用于读写某磁盘块信息时，这个缓冲区就从空闲缓冲区队列退出，并链入该设备的设备缓冲区队列，该缓冲区的状态标志中应置上 BUSY（表示缓冲区正在"忙"）标志。

（3）当缓冲区中的信息传送到用户空间后，或当用户信息写到了缓冲区且对缓冲区置了 DELWR（延迟写标志，表示缓冲区中内容尚未写到磁盘上）标志后，这样的缓冲区可以释放。此时便清除缓冲区中的 BUSY 标志，把它链到空闲缓冲区队尾，同时仍把它保留在该设备的设备缓冲区队列中。这里首先要注意的是，虽然缓冲区被链入空闲缓冲区队列，但缓冲区中的信息仍是有效的。

（4）分配缓冲区时，总是摘取空闲缓冲区队列中的第一个缓冲区。但如果该缓冲区有 DELWR 标志（延迟写标志），则表示该缓冲区信息尚未写回磁盘，暂不能分配。必须将它从空闲缓冲区队列退出,且提出 I/O 请求,由设备驱动程序把该缓冲区内容写到相应设备的指定磁盘块上。待输入输出操作完成后，清除该缓冲区的 DELWR 标志且将它重新链入空闲缓冲区队列的队尾，同时仍把它保留在原设备缓冲区队列中,其目的是当需要时仍然不必启动设备而可直接使用该块中的信息。

（5）如果一个缓冲区既在空闲缓冲区队列中，又在设备缓冲区队列中，而该缓冲区被分配移作它用，则要让它从空闲缓冲区队列和原设备缓冲区队列退出，且链入新的设备缓冲区队列。

5、缓冲区的检索：由设备管理中的缓冲区检索模块 getblk 完成。

（1）根据设备号找到该设备的设备缓冲区队列。如果能找到一个缓冲区，其缓冲区控制块中的 b_blkno 与文件系统提供的磁盘块号相同,则说明该磁盘块所需的缓冲区已经存在,因此不必再分配缓冲区.但是要根据该缓冲区是否正在忙而区别对待

(2)如果在设备缓冲区队列中找不到相应的缓冲区,那么就从空闲缓冲区队列中分配一个缓冲区.

第七章　进程同步与进程通信

一、进程的顺序性

1、进程的顺序性：是指进程在顺序处理器上的执行是严格按序的，即按照程序规定的操作顺序，只有在前一个操作结束后才能开始后继操作。

2、进程顺序执行的特性：

（1）封闭性：进程执行的结果只取决于进程本身，不受外界影响。也就是说，进程执行的结果与其执行的速度无关。

（2）可再现性：进程重复执行时，必定获得同样的结果。

二、进程的并发性

1、并发性：在一个进程的工作没有全部完成之前，另一个进程就可以开始工作，我们说这些进程是可同时执行的，或称它们具有并发性，并且把可同时执行的进程称为并发进程。

2.并发进程间的关系

1.无关:如果一个进程的执行不影响其它进程的执行结果,也不依赖其它进程的进展情况,则他们是各自独立的,就说这些进程相互直接是无关的

2.有交互:如果一个进程的执行要依赖其它进程的进展情况或者可能影响其它进程的执行结果,则说这些进程相互之间是交互的,有交互的进程并发执行时,执行结果与其执行的相对速度有关

第二节　与时间有关的错误

造成进程执行不正确的因素是与进程占用处理器的时间,执行的速度以及外界的影响有关这些因素都与时间有关,所以把它们称为与时间有关的错误

临界区

- 1、临界区：是指并发进程中与共享变量有关的程序段。
- 2、相关临界区：是指并发进程中涉及相同变量的那些临界区。
- ➔说明：如果有进程在临界区执行时，不让另一个进程进入相关的临界区执行，就不会形成多个进程对相同的共享变量交叉访问，于是就可避免出现与时间有关的错误。
- 3、相关临界区的管理要求：
- （1） 一次最多一个进程能够进入临界区。当有进程在临界区执行时，其它想进入临界区执行的进程必须等待。
- （2）不能让一个进程无限制地在临界区执行，即任何一个进入临界区的进程必须在有限的时间内退出临界区。

（3）不能强迫一个进程无限制地等待进入它的临界区，即有进程退出临界区时应让一个等待进入临界区的进程进入它的临界区执行

二 pv 操作(不可中断的过程,即原语)

1.p 操作:占有资源的过程

将信号量 S 减去 1,若结果小于 0,则把调用 P(S)的进程置成等待信号量 S 的状态

2.V 操作:释放资源的过程

将信号量 S 加 1,若结果不大于 0,则释放一个等待信号量 S 的进程.

3.用 PV 操作管理临界区

```
...
P(s)
{临界区 Ci};
V(S);
...
```

第四节 进程的互斥

一.用 pv 操作实现进程的互斥
进程的互斥:是指当有若干进程都要使用某一共享资源时,任何时刻最多只允许一个进程去使用该资源,其它要使用它的进程必须等待,直到该资源的占有者释放了该资源
二.读者/写者问题
1.共享文件
在计算机系统中,把可供多个进程使用的文件称为共享文件
我们把想读文件信息的进程称为读者,把想修改文件内容的进程称为写着
2.共享文件的管理方式
- 不允许多个进程同时使用共享文件:这种管理方式限制了每次只有一个进程可以使用文件
- 允许多个进程同时使用共享文件

第五节 进程的同步

一.协作
- 进程 1 把一个记录存入缓冲区后,应向进程 2 发送"缓冲区中有等待处理的记录"的消息
- 进程 2 从缓冲区中取 1 等待,直到消息到达出一个记录后,应向进程 1 发送"缓冲区中的记录已取走"的消息
- 进程 1 只有在得到进程 2 发送来的"缓冲区中的记录已取走"的消息后,才把下一个记录存入缓冲区否则进程 1 等待,直到消息到达进程 2 只有在得到进程 1 发送来得"缓冲区中有等待处的记录"的消息后,才能从缓存区取出记录并加工否则进程 2 等待,直到消息到达

二.用 PV 操作实现进程的同步
1.进程的同步:是指并发进程之间存在一种制约关系,一个进程的执行依赖另一个进程的消息
当一个进程没有得到另一个进程的消息时应等待,直到消息到达才被唤醒
2.同步机制:要实现进程的同步就必须提供一种机制这种机制应能把其它进程所需的消息发送出去,也能测试自己需要的
消息是否到达,把能实现进程同步的机制称为同步机制
3.同步与互斥概括:进程的同步与进程的互斥都涉及到并发进程访问共享资源的问题.从进程互斥和进程同步的讨论中,我们看到,进程的互斥实际上是进程同步的一种特殊情况,实现进程互斥时用 P 操作测试是否可以使用共享资源,这相当于测试"资源可使用"的消息是否到达;用 V 操作归还共享资源,这相当于发送了"共享资源已空闲"的消息.因此互斥使用资源的进程之间实际上也存在一个进程等待另一个进程发送消息的制约关系
所以,经常把进程的互斥与进程的同步统称为进程的同步,把用来解决进程互斥与进程同步的机制(如 PV)操作统称为同步机制
4.同步与互斥的混合问题:若一组涉及共享资源的并发进程执行时不仅要等待指定的消息到达,而且还必须考虑资源的互斥使用,那么,就既要实现进程的同步,又要实现进程的互斥进程通信

第六节 进程通信

一,通信机制
1.进程通信:是指通过专门的通信机制实现进程间交换大量信息的通信方式.
2.通信机制:采用高级通信方式时,进程间用信件来交换信息,一个正在执行的进程可以在任何时刻向其它进程发送信件,一个正在执行的进程也可以在任何时刻向其它进程索取信件.
3.信件构成:
- 发送者名:为发送信件进程的进程名
- 信息(或信息存放的地址和长度):指要传送给某一进程的信息,若信息量大,则可把信息

存放在某个缓冲区中,在信件中指出缓冲区的起始地址和信息长度

- 等/不等回信:表示信件发送者是否等信件接受者的回信
- 回信存放地址:若需要等回信,应指明回信的存放地址

4.通信原语

- 发送原语:应给出两个参数,一个是信件或者信件存放地址;另一个是信送到哪里
- 接受原语:应给出两个参数,一个是从哪里取信,另一个是取出的信存放在哪里

5.通信方式

- 直接通信方式:固定在一对进程之间进行的通信
- 间接通信方式:以信箱为媒体来进行的通信

二 .间接通信

1.信箱构成:

- 信箱说明:包括可存信件数,已有信件数,可存信件的指针等参数
- 信箱体;存放信件的区域

2.间接通信的通信规则:

- 若发送信件时信箱已满,则应把发送信件的进程置成"等信箱"状态,直到信箱有空时才被释放
- 若取信件时信箱中无信,则应把接收信件的进程置成:等信件:状态,直到信箱中有信件时才被释放

3.间接通信中通信原语的功能及实现要求

1.send(N,M)

功能:把信件 M 送到指定的信箱 N 中

实现要求:查指定信箱 N,若信箱未满,则按"可存信件的指针"把信件 M 存入信箱且释放"等信者;若信箱已满,则把发送进程置为"等信箱"状态.

2.receive(N,Z)

功能:从指定信箱 N 中取出一份信,存到指定的地址 Z 中.

实现要求:查指定信箱 N.若信箱中有信,则取出一份信存于 Z 中,且释放"等信箱:者;若信箱中无信,则把接收信件进程置为"等信件"状态.

第七节 Unix 中的进程同步与进程通信

一 Unix 中的进程同步

1,用 wait 和 exit 实现同步;

Unix 的进程在用户态执行时,可使用系统调用 fork 创建新进程,被创建的进程与创建者构成了父子进程的关系.因为父进程创建子进程的目的是要子进程去完成一些特定的任务,所以父进程要等待子进程的执行,于是用 wait 让自己等待.等待期间,子进程按父进程的要求执行任务,直到任务完成时,用系统调用 exit 来终止自己,且释放父进程.父进程被释放后,继续后台操作.所以,当进程在用户态执行用户程序是,系统调用 wait 和 exit 是实现同步的主要手段

2.用 sleep 和 wakeup 实现同步

unix 中,在用户态执行的进程请求操作系统为其服务时,就转换到和心态执行操作系统程序.进程在核心态执行时会出现各种等待事件.

无论哪种原因引起了等待事件,都要使用系统调用 sleep 让进程进入睡眠状态,当等待结束时,使用系统调用 wakeup 来唤醒进程,使被唤醒的进程结束等待

而称为就绪状态,所以,当进程在核心态执行系统程序时,由系统调用 sleep 和 wakeup 实现同步

二.unix 中的进程通信

1.管道机制:允许进程间按先进先出的方法传送消息.

- 无名管道 pipo:适用于一个用户有同一祖先的父子进程间的通信.

所谓无名管道实际上是连接进程间的可共享文件,称为 pipe 文件.一个进程可以通过也称为 pipe 的系统调用来建立一个无名管道,然后其子进程可与该进程共享该管道,对 pipe 文件进行读写操作.一个进程可以吧信息写入 pipe 文件,而另一个进程可从 pipe 文件中读取信息,但这种读写操作需由管道机制加上一些读写同步来控制.

shell 中用符号"|"表示对 pipe 文件的操作,故管道机制的一般形式是:p1|p2,该操作的作分量的输出信息是管道 pipe 的输入,而管道 pipe 的输出则是该操作的右分量的输入信息

- 命名管道 FIF0:适有名管道文件.用于不同用户的进程间的通信.

命名管冠有文件名的管道文件.道实际上是一个冠有文件名的管道文件,用户既可以使用 shell 命令中的 mknod 命令来产生一个有名管道文件,也可以在程序中使用系统调用 mknod 来创建一个
有名管道文件.命名管道的使用方式与无名管道的使用方式不同.对命名管道的使用就像对普通文件的使用一样,要通过文件操作来使用.首先必须建立,读写之前先打开,通信结束后要关闭,利用有名
管道文件进行通信时,通信的发送者用"只写"方式打开,通信的接受者用"只读"方式"打开.对被打开的有名管道文件,进程可按打开的方式对该文件进行读或写.在读写的过程中管道机制要对读写操作进行同步控制,以保证信息传输的正确性.

2.消息缓冲机制:

1.消息缓冲机制:Unix 中的消息缓冲机制是利用缓冲区来传输消息的,由系统统一管理一组缓冲区,其中每一个缓冲区都可用来存放一个消息.当一个进程要发送消息时,首先向系统申请一个缓冲区,然后再把组织好的消息存入缓冲区;再把存有
消息的缓冲区连接到消息队列中.所有这些工作可通过调用消息缓冲机制所提供的系统调用来完成.接受消息的进程也可通过系统调用从消息列表中取出消息,从缓冲区取出消息后,就应释放该缓冲区.

2.缓冲区的管理信息:发送消息的进程命,消息长度,需传输的消息正文,下一个消息缓冲区指针.

第八节 线程概念

1.线程概念:是进程中可以独立执行的子任务

2.线程属性

- 每个线程有一个唯一的标识符和一张线程描述表.线程描述表记录了线程执行时的寄存器和宅等现场状态.
- 不同的线程可以执行相同的程序,即同一个服务程序被不同的用户调用时,操作系统为它们创建成不同的线程
- 同一进程中的各个线程共享分配给进程存地址空间
- 线程时处理器的独立调度单位,多个线程可以并发执行
- 一个线程被创建后,便开始了它的生命期,直到终止,线程在生命期内会经历等待态,就绪态,和运行态等

3.线程和进程的区别:进程时资源分配单位,而线程是调度和执行单位.每个进程都有自己的主存空间,同一个进程中的各线程共享该进程的贮存空间,进程中的所有线程对进程的整个主存空间都有存取权限

1.进程缺点

每个进程要占用一个进程控制快和一个私有的主存区域,开销较大
进程之间的通信必须由通信机制来完成,速度较慢
进程增多会给调度和控制带来复杂性,增加了死锁的机会

2.线程优点

1.创建线程无须另外分配资源,因而创建线程的速度比创建进程的速度快,且系统开销小

线程间的通信在同一地址空间中进行,故不需要额外的通信机制,使通信更简单,信息传递速度也更快.

线程能独立执行,能充分利用和发挥处理器与外围设备并行工作的能力

第八章 死锁

第一节 死锁的形成

1、死锁概念：若系统中存在一组进程（两个或多个），它们中每个进程都占用了某种资源，又都在等待已被该组进程中的其它进程占用的资源，如果这种等待永远不能结束，则说系统出现了死锁，或者说这组进程处于死锁状态。

2、死锁的形成：

（1）与进程对资源的需求有关；

（2）与进程的执行速度有关；

（3）与资源的分配策略有关。

第二节 死锁的特征

一、死锁的必要条件：（即死锁产生一定会成立的条件）

1、互斥地使用资源：每个资源每次只能给一个进程使用。

2、占有且等待资源：一个进程占有了某些资源后又申请新资源而得不到满足时，处于等待资源的状态，且不释放已占资源。

3、不可抢夺资源：任何一个进程不能抢夺另一个进程所占的资源。

4、循环等待资源：相互等待已被其它进程占用的资源。

➔必要条件说明：以上四个条件仅仅是必要条件而不是充分条件，即只要发生死锁，则这四个条件一定同时成立，如果其中的一个或几个条件不成立，则一定没有死锁。但反之不然，即若这四个条件同时成立，系统未必就有死锁存在。

二、资源分配图：（是判断死锁产生与否的方法）

第三节 死锁防止

一、互斥条件

要使互斥使用资源的条件不成立，唯一的办法是允许进程共享资源。但部分硬件设备的物理特性是改变不了的，所以要想破坏“互斥使用资源”这个条件经常是行不通的。

二、占有并等待条件

1、静态分配资源：是指进程必须在开始执行前就申请自己所要的全部资源，仅当系统能满足进程的全部资源申请要求且把资源分配给进程后，该进程才开始执行。

2、释放已占资源：仅当进程没有占用资源时，才允许它去申请资源。因此，如果进程已经占用了某些资源而又要再申请资源，那么按此策略的要求，它应先归还所占的资源，归还后才允许申请新资源。（如进程要占用磁带机，又要申请打印机，可以先启动磁带机工作，完成后先释放磁带机再申请打印机。由于申请者是在归还资源后才申请新资源，故不会出现占有了部分资源再等待其它资源的现象。）

三、不可抢夺条件

为了使这个条件不成立，我们可以约定如下：如果一个进程已占有

了某些资源又要申请新资源 R，而 R 已被另一进程 P 占用因而必须等待时，则系统可以抢夺进程 P 已占用的资源 R。具体做法如下：

（1）若进程 A 申请的资源 R 尚未被占用，则系统可把资源 R 分配给进程 A。

（2）若进程 A 申请的资源 R 已被进程 B 占用，则查看进程 B 的状态。如果进程 B 处于等待另一个资源的状态，那么就抢夺进程 B 已占的资源 R，并把 R 分配给进程 A，适当的时候再把 R 归还给进程 B 使用；否则让进程 A 处于等待资源 R 的状态。

（3）一个等待资源的进程只有在得到自己所申请的新资源和所有被其它进程抢夺去的老资源后，才能继续执行

四、循环等待条件

对资源采用按序分配的策略可使循环等待资源的条件不成立。按序分配资源是指对系统中所有资源排一个顺序，对每一个资源给出一个确定的编号，规定任何一个进程申请两个以上资源时，总是先申请编号小的资源，再申请编号大的资源

第四节　死锁的避免

一、安全状态

1、安全状态概念：如果操作系统能保证所有的进程在有限的时间内得到需要的全部资源，则称系统处于安全状态，否则说系统是不安全的。显然，处于安全状态的系统不会发生死锁，而处于不安全状态的系统可能会发生死锁。

2、资源分配算法：在分配资源时，只要系统能保持处于安全状态，就可避免死锁的发生。故每当有进程提出分配资源的请求时，系统应分析各进程已占资源数、尚需资源数和系统中可以分配的剩余资源数，确定是否处于安全状态。如果分配后系统仍然能维持安全状态，则可为该进程分配资源，否则就暂不为申请者分配资源，直到其它进程归还资源后再考虑它的分配问题。

二、银行家算法

1、当一个用户对资金的最大需求量不超过银行家现有的资金时，就可接纳该用户；

2、用户可以分期贷款，但贷款总数不能超过最大需求量；

3、当银行家现有的资金不能满足用户的尚需贷款数时，可以推迟支付，但总能使用户在有限的时间里得到贷款；

4、当用户得到所需的全部资金后，一定能在有限时间里归还所有的资金。

第五节 死锁的检测

一、死锁的检测方法

1、每类资源中只有一个资源：

如果每类资源中只有一个资源，则可以设置两张表格来记录进程使用和等待资源的情况。一张为占用表，记录进程占用资源的情况。另一张为等待表，记录进程正在等待资源的情况。任一进程申请资源时，若该资源空闲，则把该资源分配给申请者，且在占用表中登记，否则把申请者登入等待表中。死锁检测程序定时地检测这两张表。如果发现有循环等待资源的进程，则表明有死锁出现。

2、资源类中含有若干个资源：

（1）初始检测：找出资源已经满足的进程，即不再申请资源的进程。若有这样的进程，则它们一定能在有限的时间内执行结束，且归还所占的资源。所以可以把它们所占的资源与系统中还剩余的资源加在一起作为可分配的资源，同时对这些进程置上标志。

一、死锁的检测方法

2、资源类中含有若干个资源：

（2）循环检测：检测所有无标志的进程，找出一个尚需资源量不超过系统中的可分配资源量的进程。若能找到，则只要把资源分配给该进程，就一定能在有限时间内收回它所占的全部资源。故可把该进程已占的资源添加到可分配的资源中，同时为该进程置上一个标志。重复执行第二步，直到所有进程均有标志，或无标志的进程尚需资源量均超过可分配的资源量。

（3）结束检测：若进程均有标志，表示当前不存在永远等待资源的进程，也即系统不处于死锁状态。若存在无标志的进程，表示系统当前已有死锁形成，这些无标志的进程就是一组处于死锁状态的进程。检测结束，解除检测时所设置的所有标志。

1.终止进程

终止涉及死锁的所有进程

一次终止一个进程

2.抢夺资源

从涉及死锁的一个或几个进程中抢夺资源

把抢夺的资源再分配给卷入死锁的其他

进程,直到死锁解除

注意:

抢夺哪些进程的哪些资源

被抢夺着的恢复

进程的"饿死"

重点:分析与时间有关的错误;用 PV 操作实现进程的互斥与同步

用信箱实现进程通信