**进程管理**

1. 进程的基本概念

1.程序的顺序执行及其特征

(1)顺序性：(2)封闭性：程序运行时独占全机资源，资源状态只有本程序才能改变他，其执行结果不受外界影响 (3) 可再现性：

2.前趋图(Precedence Graph)是一个有向无循环图，记为DAG(Directed Acyclic Graph)，用于描述进程之间执行的前后关系。

如果(Pi, Pj)∈→,可写成Pi→Pj，称Pi是Pj的直接前趋，而称Pj是Pi的直接后继。在前趋图中，把没有前趋的结点称为初始结点(Initial Node)，把没有后继的结点称为终止结点.

应当注意，前趋图中必须不存在循环

3.程序的并发执行及其特征

1. 间断性 2)失去封闭性 3)不可再现性

4.进程的特征与状态

1)结构特征 2)动态性 3)并发性

4)独立性 5)异步性

进程是进程实体的运行过程，是系统进行资源分配和调度的一个独立单位

5.进程的三种基本状态

1)就绪(Ready)状态 2) 执行状态 3) 阻塞状态



1. 挂起状态(a.终端用户的请求.b父进程请求.c负荷调节的需要.d.操作系统的需要。)



6.进程控制块

进程控制块的作用是使一个在多道程序环境下不能独立运行的程序(含数据)，成为一个能独立运行的基本单位，一个能与其它进程并发执行的进程。或者说，OS是根据PCB来对并发执行的进程进行控制和管理的。

1. 进程控制块中的信息
2. 进程标识符:用于惟一地标识一个进程

1)内部标识符。在所有的操作系统中，都为每一个进程赋予一个

一的数字标识符，它通常是一个进程的序号。 设置内部标识符主要是为了方便系统使用。

2)外部标识符。它由创建者提供，通常是由字母、数字组成，往往是由用户(进程)在访问该进程时使用。

b.处理机状态

c.进程调度信息

① 进程状态，指明进程的当前状态， 作为进程调度和对换时的依据；② 进程优先级，用于描述进程使用处理机的优先级别的一个整数， 优先级高的进程应优先获得处理机； ③ 进程调度所需的其它信息，它们与所采用的进程调度算法有关，比如，进程已等待CPU的时间总和、 进程已执行的时间总和等；④ 事件，是指进程由执行状态转变为阻塞状态所等待发生的事件，即阻塞原因。

d.进程控制信息

1. 进程控制块的组织方式
2. 链接方式



1. 索引方式



1. 进程控制
2. 进程的创建(Creation of Progress)

(1)申请空白PCB。

(2) 为新进程分配资源。

(3) 初始化进程控制块。

(4) 将新进程插入就绪队列。

2.进程的终止(Termination of Process)

(1) 根据被终止进程的标识符，从PCB集合中检索出该进程的PCB，从中读出该进程的状态。

(2) 若被终止进程正处于执行状态，应立即终止该进程的执行，并置调度标志为真，用于指示该进程被终止后应重新进行调度。

(3) 若该进程还有子孙进程，还应将其所有子孙进程予以终止，以防他们成为不可控的进程。

(4) 将被终止进程所拥有的全部资源，或者归还给其父进程， 或者归还给系统。 (5) 将被终止进程(它的PCB)从所在队列(或链表)中移出， 等待其他程序来搜集信息。

1. 进程的阻塞与唤醒

正在执行的进程，当发现上述某事件时，由于无法继续执行，于是进程便通过调用阻塞原语block()把自己阻塞。可见，进程的阻塞是进程自身的一种主动行为

唤醒原语wakeup()执行的过程是：首先把被阻塞的进程从等待该事件的阻塞队列中移出，将其PCB中的现行状态由阻塞改为就绪，然后再将该PCB插入到就绪队列中。

1. 进程的挂起与激活

当出现了引起进程挂起的事件时，比如，用户进程请求将自己挂起，或父进程请求将自己的某个子进程挂起， 系统将利用挂起原语suspend( )将指定进程或处于阻塞状态的进程挂起。

激活原语active()先将进程从外存调入内存，检查该进程的现行状态，若是静止就绪，便将之改为活动就绪；若为静止阻塞便将之改为活动阻塞。

三．进程同步

1.两种形式的制约关系

(1)间接相互制约关系:共享临界资源时发生

(2)直接相互制约关系:进程间有某种前后顺序执行关系

2.临界资源(Critical Resouce)

生产者-消费者(producer-consumer)问题是一个著名的进程同步问题.

3.同步机制应遵循的规则

(1)空闲让进:当临界区无进程时，让新进程进入。

(2)忙则等待:当临界区有进程时，让新进程等待。

(3)有限等待:进程在临界区内的时间应该一定。

(4)让权等待:当进程长时间无法进入临界区时，应该释放此进程。

4.信号量机制

(1)整型信号量

wait (s):

if (s<=0) wait;

else s=s-1;

//临界区

signal (s):

s=s+1;

(2)记录型信号量

在整型信号量机制中的wait操作，只要是信号量S≤0， 就会不断地测试。因此，该机制并未遵循“让权等待”的准则， 而是使进程处于“忙等”的状态。记录型信号量机制，则是一种不存在“忙等”现象的进程同步机制。

typedef struct {

int value; //临界资源数目

list L; //进程链表

}semaphore;

相应地，wait(s)和signal(s)操作可描述为：

semaphore s;

wait(s):

s.value=s.value-1;

if (s.value＜0) block(process,s.L);

//临界区

signal(s):

s.value=s.value+1;

if (s.value≤0) wakeup(process,sL);

1. AND型信号量

AND同步机制的基本思想是：将进程在整个运行过程中需要的所有资源，一次性全部地分配给进程，待进程使用完后再一起释放。只要尚有一个资源未能分配给进程，其它所有可能为之分配的资源，也不分配给他。

为此，在wait操作中，增加了一个“AND”条件，故称为AND同步，或称为同时wait操作，即swait

swait(s1, s2, …, sn):

if (si≥1 .... sn≥1){

for (int i=1 .. n){

si=si-1;

}

}

else{

place the process in the waiting queue

place first si＜1 in the process state

}

//临界区

signal(s1,s2,...,sn):

for(int i =1...n ){

si=si+1;

remove all the process in the waiting queue

place si into the ready queue.

}

1. 信号量集

s:资源总数 t:资源下限量，当低于它，都不给予分配 d:资源需求值

swait(s1, t1, d1, …, sn, tn, dn):

if si≥t1 .... sn≥tn){

for(int i=1 ... n ){

si=si-di;

}

}

else{

place the executing process in the waiting queue

place first si＜ti in the process state 

}



//临界区

signal(s1, d1, …, sn, dn)

for (int i=1 ... n ){

si =si+di;

remove all the process in the waiting queue

place si into the ready queue

}

1. 进程通信
2. 进程通信的类型

a.共享存储器系统(Shared-Memory System)

(1)基于共享数据结构的通信方式

只适于传输少量数据

(2)基于共享存储区的通信方式

适于传输大量数据

b.消息传递系统(Message passing system)

不论是单机系统、多机系统，还是计算机网络，消息传递机制都是用得最广泛的一种进程间通信的机制。在消息传递系统中，进程间的数据交换，是以格式化的消息(message)为单位的；在计算机网络中，又把message称为报文。

c.管道(Pipe)通信

所谓“管道”，是指用于连接一个读进程和一个写进程以实现他们之间通信的一个共享文件，又名pipe文件。

1. 消息传递通信的实现方法

a,直接通信方式

这是指发送进程利用OS所提供的发送命令，直接把消息发送给目标进程。此时，要求发送进程和接收进程都以显式方式提供对方的标识符。通常，系统提供下述两条通信命令(原语)：

Send(Receiver, message); 发送一个消息给接收进程；

Receive(Sender, message); 接收Sender发来的消息；

在某些情况下，接收进程可与多个发送进程通信，因此，它不可能事先指定发送进程。因此，在接收进程接收消息的原语中的源进程参数，是完成通信后的返回值

Receive (id, message);

b.间接通信方式

(1)信箱的创建和撤消。进程可利用信箱创建原语来建立一个新信箱。创建者进程应给出信箱名字、信箱属性(公用、私用或共享)；对于共享信箱， 还应给出共享者的名字。当进程不再需要读信箱时，可用信箱撤消原语将之撤消。

(2) 消息的发送和接收。当进程之间要利用信箱进行通信时，必须使用共享信箱，并利用系统提供的下述通信原语进行通信。

Send(mailbox, message); 将一个消息发送到指定信箱；

Receive(mailbox, message); 从指定信箱中接收一个消息

a.私用信箱

用户进程可为自己建立一个新信箱，并作为该进程的一部分。信箱的拥有者有权从信箱中读取消息，其他用户则只能将自己构成的消息发送到该信箱中.

b.公用信箱

它由操作系统创建，并提供给系统中的所有核准进程使用。核准进程既可把消息发送到该信箱中，也可从信箱中读取发送给自己的消息。。

c.共享信箱

它由某进程创建，在创建时或创建后，指明它是可共享的，同时须指出共享进程(用户)的名字。信箱的拥有者和共享者，都有权从信箱中取走发送给自己的消息。

1. 消息传递系统
2. 通信链路(communication link)
3. 由发送进程在通信之前，用显式的“建立连接”命令(原语)请求系统为之建立一条通信链路；在链路使用完后，也用显式方式拆除链路。这种方式主要用于计算机网络中
4. 发送进程无须明确提出建立链路的请求，只须利用系统提供的发送命令(原语)，系统会自动地为之建立一条链路。这种方式主要用于单机系统中