

# 基本概念回顾

1 进程的定义:

2 进程的特征:

3 进程的三种基本状态:

4 进程的组成:

5 进程控制块的作用

6 CPU 的两种执行状态:

7 原语定义:

8 进程控制原语定义:

9 进程互斥定义:

10 临界资源定义:

11 临界区定义:

12 进程同步机制的准则:

13 进程同步的定义:

14 记录型信号量及 P、V 操作的定义:

15 信号量的作用:

16 进程通信的定义:

17 进程通信的类型:

18 线程的定义:

19 管程的定义:

20 实现管程的三个关键问题:

# 参考答案

## 1 进程的定义：

进程是一个具有一定独立功能的程序关于某个数据集合的一次运行活动。



## 2 进程的特征：

- (1) **动态性**：进程的实质是程序的一次执行过程，有一定的生命期。
- (2) **并发性**：指多个进程实体同存于内存中，能在一段时间内同时运行。
- (3) **独立性**：进程是一个能独立运行的基本单位，同时也是申请拥有系统资源的独立单位。
- (4) **异步性**：由于进程间共享资源和协同合作时带来了相互制约关系，造成进程执行的间断性。进程各自以独立的、不可预知的速度向前推进。
- (5) **结构特征**：为了描述进程的运动变化过程，使之能独立运行，应为每个进程配置一个进程控制块 PCB。



## 3 进程的三种基本状态：

- (1) **就绪(Ready) 状态**：当进程已分配到除 CPU 以外的所有必要的资源，只要获得处理机便可立即执行，这时的进程状态称为就绪状态。
- (2) **执行(Running) 状态**：当进程已获得处理机，其程序正在处理机上执行，此时的进程状态称为执行状态。
- (3) **阻塞(Blocked) 状态**：正在执行的进程，由于等待某个事件发生而无法执行时，便放弃处理机而处于等待状态。



## 4 进程的组成：

- (1) **进程控制块 PCB**：每一个进程均有一个进程控制块 PCB。用户进程被创建时，系统为它申请和构造一个相应的 PCB。
- (2) **程序段**：程序段是进程中能被进程调度程序调度在 CPU 上执行的程序代码段，它能实现相应的特定功能。

(3) **数据段**：一个进程的数据段，可以是进程对应的程序加工处理的原始数据，也可以是程序执行时产生的中间或最终结果数据。



## 5 进程控制块的作用：

PCB 通常是系统内存占用区中的一个连续存区，它存放着操作系统用于描述进程情况及控制进程运行所需的**全部信息**，它使一个在多道程序环境下不能独立运行的程序成为一个能独立运行的基本单位，一个能与其他进程并发执行的进程。



## 6 CPU 的两种执行状态：

**系统态**：又称核心态或管态。它具有较高的特权，能执行一切硬件指令，访问所有寄存器和内存储区。

**用户态**：又称目态。这是具有较低特权的执行状态，只能执行规定的指令，访问指定的寄存器和内存储区。



## 7 原语定义：

是操作系统内核中由若干条指令构成，用于完成特定功能的一个过程，该过程在执行时是不可中断的。



## 8 进程控制原语定义：

进程控制原语是对进程生命期控制和进程状态转换的原语，基于进程的基本状态它们是创建进程原语、撤销进程原语、挂起进程原语、激活进程原语、阻塞进程原语和唤醒进程原语等。



## 9 进程互斥定义：

所谓进程互斥是指当有若干进程都要使用某一共享资源时，最多允许一个进程使用，而其他要使用该资源的进程必须等待，直到占用该资源的进程释放了该资源为止。



## 10 临界资源定义：

操作系统中将一次仅允许一个进程访问的资源称为临界资源。



## 11 临界区定义：

操作系统中把每个进程中访问临界资源的那段代码段称为临界区。



## 12 进程同步机制的准则：

①空闲让进：当无进程处于临界区时，必须让一个要求进入它的临界区的进程立即进入，以提高临界资源的利用率。

②忙则等待：当已有进程处于临界区时，其他试图进入自己临界区的进程必须等待，以保证它们互斥地进入临界区。

③让权等待：对于等待进入临界区的进程而言，它必须立即释放处理机，以避免进程“忙等”而降低 CPU 的效率。

④有限等待：对要求进入临界区的进程，应在有限时间内进入，以免陷入“死等”。



## 13 进程同步的定义：

所谓进程同步是指多个合作进程为了完成同一个任务，它们在执行速度上必须相互协调。



## 14 记录型信号量及 P、V 操作的定义：

记录型信号量是一个记录型的数据结构，包含两个数据项，一个是计数值域，另一个是等待该信号量的进程队列首指针域。

P(S) 和 V(S) 操作原语描述如下：

```
void p(S)
struct semaphore S;
{
    S.value=S.value -1;
    if (S.value<0 ) block(S.P);
}
```

```
void v(S)
struct semaphore S;
{
S.value=S.value+1;
if (S.value<=0) wakeup(S.P);
}
```



## 15 信号量的作用：

可用信号量来解决  $n$  个进程互斥进入各自的临界区对临界资源访问的问题，也可以用信号量控制进程同步间的推进顺序等。



## 16 进程通信的定义：

所谓进程通信是指进程间的信息交换。



## 17 进程通信的类型：

(1) 共享存储器系统：为了传送大量数据，在存储器中划出一块共享存储区，诸进程可通过对共享存储区进行读或写数据实现通信。

(2) 消息传递系统：

直接通信方式（消息缓冲）：发送进程可将消息直接发送给接收进程。

间接通信方式（信箱）：发送进程将消息发送到指定信箱中，而接收进程从信箱中取得消息。

(3) 管道通信系统：所谓管道是指用于连接一个读进程和一个写进程，以实现它们间通信的共享文件，又称 pipe 文件。



## 18 线程的定义：

线程是进程中可独立调度执行的子任务，一个进程可以有一个或多个线程，它们共享所属进程所拥有的资源。



## 19 管程的定义：

一个管程定义了一个数据结构和能为并发进程调用的在该数据结构上的一组操作过程,这组操作过程能同步进程和改变管程中的数据。



## 20 实现管程的三个关键问题：

(1) 互斥：管程必须能有效地实现互斥。管程是被动成份，而进程是主动成份。当进程需要访问管程中的临界资源的时候，可调用管程中的有关入口过程。但当几个进程都需调用某一管程的同一个或不同的入口过程时，仅允许一个进程调用进入管程，而其他调用者必须等待。

(2) 同步：在管程中必须设置两个同步操作原语 `wait` 和 `signal`。当进程通过管程请求访问共享数据而未能满足时，管程便调用 `wait` 原语使该进程阻塞等待，当另一进程访问完该共享数据且释放后，管程便调用 `signal` 原语，唤醒等待队列中的队首进程。

(3) 条件变量：为了区别等待的不同原因，管程又引入了条件变量。条件变量的类型标识符为 `condition`。不同的条件变量，对应不同原因的进程阻塞等待队列，初始时空。条件变量上能作 `wait` 和 `signal` 原语操作，若条件变量名为 `nonbusy`，则调用同步原语的形式为 `wait(nonbusy)` 和 `signal(nonbusy)`。

