**1.论述程序、进程、线程的区别与联系**

**进程与程序的区别联系：**

**进程：**一个正在计算机上执行的程序实例 ，其结构为：进程＝进程控制块＋程序段＋数据段，进程是程序在一个数据集合上运行的过程，它是系统进行资源分配和调度的一个独立单位引入进程的目的是使多道程序能够正确地并发执行；

程序是静态实体，进程具有动态性；

进程与程序之间不存在一一对应关系，一个程序可以对应多个进程，但一个进程也可能对应多个程序。

**进程和线程的区别与联系：**

进程是程序在一个数据集合上运行的过程，它是系统进行资源分配和调度的一个独立单位；线程是进程中的一个实体，是独立调度和分派的基本单位，同一进程内的线程共享进程的资源。

**2.论述进程控制块(PCB)的作用、存储内容**

PCB是操作系统感知进程存在的唯一标志，其内容主要包括

（1）进程标示符：唯一地标识一个进程，分为

* + - 内部标识符
    - 外部标识符
    - 父进程标识符、用户标识符

（2）处理机状态信息：主要是由处理器的各种寄存器中的内容组成的，包括

* + - 通用寄存器（用户可视寄存器）
    - 指令计数器
    - 程序状态字PSW
    - 栈指针

（3）进程调度信息：与进程调度和进程对换有关的信息，包括

* + - 进程状态
    - 进程优先级
    - 进程调度所需的其它信息
    - 事件

（4）其它信息，如

* + - 程序和数据的地址
    - 进程同步和通信机制
    - 资源清单

**3.以图为例，分析五状态进程模型**

新创建的进程，被系统接纳后变为就绪状态；得到调度以后从就绪变成执行状态；如果执行完毕则终止；如果执行过程中因为时间片用完则从执行变成就绪状态，等待下次调度；如果执行过程中因为等待事件发生，则从执行变成阻塞态，事件发生时从阻塞变到就绪态，等待再次调度。

**4.为什么引入挂起状态，其与阻塞状态的本质区别是什么**

内存中分没有就绪进程或内存空间非常紧张时，系统将一个或多个进程的全部或部分程序和数据从内存中换出到**外存**，以腾出部分内存空间，进程被换到外存后，其状态变为挂起状态。挂起状态和阻塞状态的本质区别是挂起进程不再参与CPU的竞争，而处于阻塞状态的进程变为就绪状态后就参与CPU的竞争。

（挂起状态和阻塞状态的区别在于对系统资源的占用不同，阻塞的进程仍处于内存中，而挂起的进程被换出到外存。）

**5.论述进程切换与模式切换的区别。什么情况下发生进程切换，什么时候发生模式切换。**

进程切换：是指调度另一个就绪进程占用处理器执行，进程切换的原因（什么情况下发生进程切换），包括时钟中断、内存失效、I/O事件、陷阱、系统调用等，进程切换通常会伴随模式切换（因为要切换到内核模式下完成进程调度）；

模式切换是用户模式和内核模式之间的切换，模式切换不一定会引起进程切换。模式切换的原因：系统调用或中断

**6.什么是中断？中断的处理过程如何？**

中断是指程序运行过程中，出现某些意外情况需操作系统干预时，操作系统暂停正在运行的程序并转入中断处理程序，处理完毕后又返回原被暂停的程序继续运行

出现中断时，系统会作如下工作：

•将程序计数器置为中断处理程序的开始地址

•从用户模式切换到内核模式，以便中断处理能执行特权指令

**7.** **fork()函数的作用，并举例说明？**

Fork()函数的作用是创建新进程。

**main()**

**{**

**pid\_t val;**

**printf(“PID…\n”);**

**val=fork();**

**if(val!=0)**

**printf(“parent…\n”);**

**else**

**printf(“****child…\n”);**

**}**

创建了一个新进程，新进程的打印输出为“child…”

**9.比较不同类型线程的优缺点。**

用户级：

优点：

线程切换不需要内核模式特权

·调度策略因应用程序不同而不同

·可以运行在任何操作系统上

缺点：

当用户级线程执行系统调用时，不仅阻塞当前线程，还将引起同一进程中的其他线程阻塞

不能利用多处理器技术

内核级：

优点：内核可以把同一个进程内的多个线程调度到多处理器上，提高并发性；

当一个线程阻塞时，内核可以调度同一进程内的其他线程；

内核例程本身也可以是多线程的，提高了内核的运行效率

缺点：把控制权从一个线程传递到相同进程内的另一个线程时，需要切换到内核模式（即线程切换涉及模式切换）

混合方法：

线程创建在用户空间完成（开销小，不需要模式切换）

线程调度和同步也可由应用程序完成（开销小，不需要模式切换）

一个应用程序中的多个线程被映射到一些（小于或等于用户级线程数）内核线程

某个线程阻塞时，可以不引起整个进程阻塞