

# 操作系统第二次作业

BobAnkh

March 2021

## 1. 进程和程序的区别和联系是什么？

答：

### • 进程和程序的区别如下：

- 进程是动态的，程序是静态的：程序是有序代码的集合进程是程序的执行。通常进程不可在计算机之间迁移；而程序通常对应着文件、静态和可以复制；
- 进程是暂时的，是一个状态变化的过程，是动态地创建和消亡的，具有一定的生命周期，而程序则是一组代码的集合，它是永久存在的，可长期保存；
- 进程与程序的组成不同：程序由代码段组成（或认为是一组有序的指令集合），进程的组成则包括程序、数据和进程控制块（即进程状态信息）。

### • 进程和程序的联系如下：

- 通过多次执行，一个程序可对应多个进程；通过调用关系，一个进程可包括多个程序；
- 进程的运行实体是程序，离开程序的进程没有存在的意义，而进程是程序的一次执行过程。

## 2. 什么是进程控制块？它一般需要包含哪些信息？

答：

(1) 进程控制块是操作系统为了管理进程而设置的一个专门的数据结构，用它来记录进程的外部特征，描述进程的运动变化过程。

(2) 进程控制块一般需要包含以下信息：

### • 进程描述信息

- 进程标识符，唯一，通常是一个整数
- 进程名，通常基于可执行文件名
- 用户标识符
- 进程组

### • 进程控制信息

- 当前状态
- 优先级
- 代码执行入口地址

- 程序的外存地址
- 运行统计信息 (执行时间、页面调度)
- 进程间同步和通信
- 阻塞原因
- 资源占用信息
  - 虚拟地址空间的现状
  - 打开文件列表
- CPU 现场信息
  - 寄存器值 (通用、程序计数器 PC、状态 PSW, 地址包括栈指针)
  - 指向赋予该进程的段/页表的指针

### 3. 说明进程在三个基本状态之间转换的典型原因。

答：三个基本状态之间转换原因如下：

- 运行-> 阻塞
  - 当一进程必须等待如下事件时：对一资源的访问尚不能进行；初始化 I/O 且必须等待结果；等待某一进程提供输入 (IPC)
- 运行-> 就绪
  - 运行进程用完了时间片；
  - 一高优先级进程进入就绪态，运行进程被中断
- 就绪-> 运行
  - 调度程序选择该程序运行
- 阻塞-> 就绪
  - 阻塞进程所等待的事件发生

### 4. 在单 CPU 的操作系统中有 $n$ 个进程并发执行，那么处于运行、就绪和阻塞状态的进程个数是多少？(给出取值范围)

答：处于运行状态的进程最少 0 个，最多 1 个；处于就绪状态的进程最少 0 个，最多  $N-1$  个；处于阻塞状态的进程最少 0 个，最多  $N$  个。

### 5. 为什么引入挂起状态？

答：引入原因如下：

- 由于 I/O 操作比 CPU 计算慢得多，故可能会出现内存中所有进程都等待 I/O 的现象。即使运行多个程序，处理器在大多数时间仍处于空闲状态，利用率很低。为此可将内存中的一部分进程转移到外存中，再调入其他进程执行。
- 在内存资源不足时，需要把一些进程换出到外存。

## 6. 为什么要引入线程？ULT 不能在多个 CPU 上实现真正的并行，那么引入 ULT 有什么意义？

答：

### (1) 引入线程的动机：

- 进程作为资源分配单位，在创建、终止、切换的过程中系统需要付出较大的时空开销，所以系统中并发执行的进程数目不宜过多、切换频率不宜过高，这限制了系统的并发程度。
- 传统的操作系统中，进程是单线程的，而单线程的进程内部需要顺序执行，不能并发，不能充分利用多处理器和多核处理器计算机的优势
- 现实中有很多需要并发处理的任务，如字处理程序、Web 服务器

在上述条件的要求下，将进程的两个属性分开，由操作系统分别处理，进程只作为资源分配单位，而线程则被引入作为 CPU 调度单位。

### (2) 引入 ULT 的意义：

- 用户级线程切换不需要核心权限，不调用操作系统内核，因此开销较小、性能良好
- 用户级线程调度是应用程序特定的，可做针对性的优化，与操作系统的低级调度算法无关
- 无论操作系统内核是否支持线程，用户级线程都可以运行，适用性好

## 7. 为什么每个线程需要有自己的栈？对于内核级线程，线程栈为什么要分为用户栈和核心栈？对于用户级线程，线程栈是否也要分为用户栈和核心栈？

答：

- (1) 因为线程是独立执行的，函数调用时传递的参数、存放的局部变量、返回地址都需要保存在栈中，它执行时要从中取参数、局部变量，所以为了线程不互相干扰，堆栈必须要是独立的，此即解释了为什么每个线程需要有自己的栈。
- (2) 因为内核级线程可能需要进行用户态和内核态之间的相互切换，所以内核级线程的线程栈需要分为用户栈和核心栈两个栈。
- (3) 用户级线程不需要将线程栈分为用户栈和核心栈。因为用户级线程的调度程序和线程自己的运行都是在用户态中执行实现，与内核态无关，因此只需要一个用户栈。