210521李育腾（进程引入作业）。

1. OS为什么要引入进程？（1分）

操作系统引入进程的主要目的是为了提高系统的并发性和资源利用率。在单任务的操作系统中，每次只能运行一个程序，而进程的引入可以让多个程序同时运行，从而提高了系统的并发性和响应性。引入进程也能够更好地利用计算机系统的资源。每个进程都有自己的代码、数据和资源，操作系统可以通过调度算法将资源分配给不同的进程，从而实现资源的共享和高效利用。

进程还可以提高系统的稳定性和可靠性。每个进程都是独立的，即使某个进程出现错误或崩溃，其他进程仍然可以正常运行，不会对整个系统造成影响。此外，进程之间也可以进行通信和协作，从而实现更加复杂的功能。

1. 到底什么是进程（严格定义以及打比喻）？（1分）

进程（Process）是操作系统中的一种资源分配单位，是指正在运行的程序的实例。进程具有独立的内存空间、指令计数器、寄存器、文件描述符等资源，能够并发地运行、通信和协作。

可以将进程比作一家工厂，这家工厂生产的产品就相当于进程所运行的程序。每个工厂都有自己的生产线、原材料、工人和机器设备等资源，这些资源只能被该工厂独享。同样，每个进程也有自己的内存空间、寄存器、文件描述符等资源，只能被该进程独占。

多个工厂可以同时运作，各自生产不同的产品，这些工厂之间需要进行协作和通信才能完成整个生产过程。同样，多个进程也可以并发地运行，各自完成不同的任务，这些进程之间也需要进行通信和协作才能完成整个系统的功能。

需要注意的是，进程是动态的，它可以被创建、销毁、挂起、恢复等。操作系统通过进程管理机制，对进程进行管理和调度，从而实现多任务的并发执行。

1. 为什么说动态性和并发性是进程的主要特征，其他特征如何体现，进程和程序之间到底是什么样的关系（要有真切的体会，不是拷贝而来）？（1分）

动态性和并发性是进程的主要特征，因为进程是在操作系统的管理下被创建、调度、挂起、恢复、销毁等，它们的状态随时会发生变化，而且多个进程可以同时运行，相互之间不会干扰。这种特性使得进程能够更加灵活、高效地利用计算机资源，提高系统的运行效率。

程序是静态的代码和数据的集合，而进程是程序在运行时的实例。一个程序可以被多个进程实例化，每个实例都有自己的内存空间和运行状态，它们之间是独立的，互不干扰。进程是操作系统对程序资源的一种抽象，操作系统通过进程管理机制对进程进行管理和调度，实现多任务的并发执行。

1. 为什么要引入进程的状态？状态之间的转换说明了什么？（1分）

引入进程的状态是为了描述进程在运行过程中所处的不同状态，进而方便操作系统进行管理和调度

状态之间的转化表明了动态性和并发性是进程的主要特征，指的是进程的状态会随着时间的推移而不断变化，并且多个进程可以同时运行

1. Linux在描述进程状态上有什么特色，你有什么什么启发？（1分）

Linux在描述进程状态时，将进程状态分为了以下五种状态：

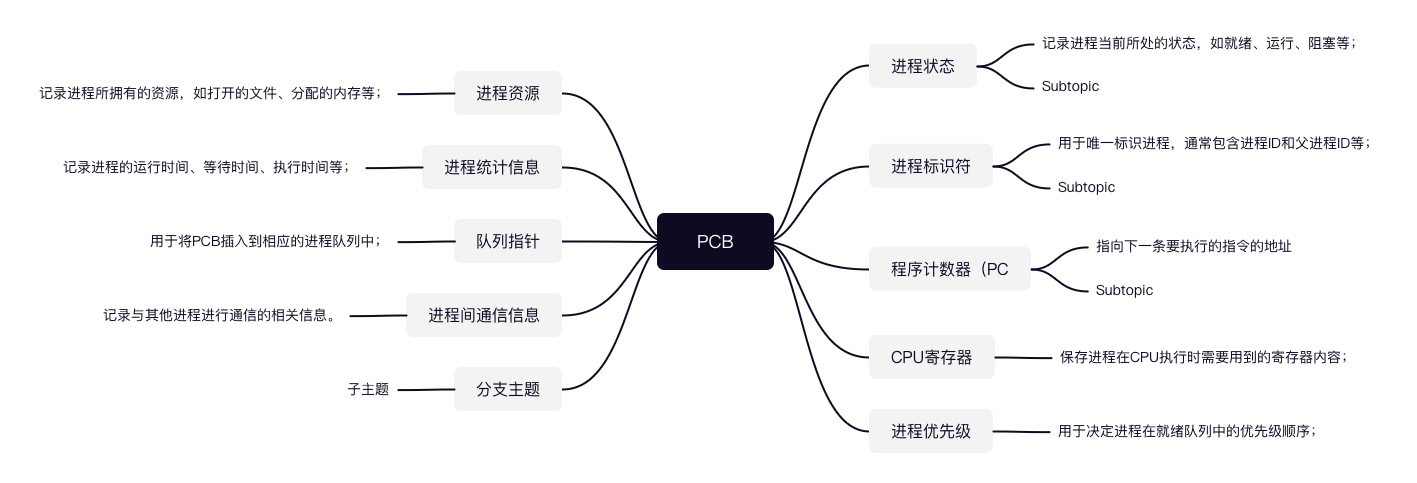
1. Running：正在运行或在运行队列中等待CPU调度；
2. Interruptible sleep：进程在等待某个条件的发生，一旦条件满足，就会被唤醒；
3. Uninterruptible sleep：进程在等待某个条件的发生，但是这个等待过程中是不能被中断的；
4. Stopped：进程被暂停执行；
5. Zombie：进程已经退出，但其父进程还没有对其进行善后处理。

这种状态描述方式相对于传统的阻塞状态、就绪状态、执行状态，更加细致、灵活，能够更准确地反映出进程的状态变化，有助于更好地实现进程管理。

从这种状态描述方式中，我们可以得到的启示是，在设计进程状态时，需要考虑到进程在不同的场景下所处的状态，并对这些状态进行细致、全面的描述。另外，在进行状态转换时，也需要考虑进程可能遇到的不同情况，保证状态转换的正确性和可靠性

1. 用思维导图对进程控制块进行全面的描述，不需要死记硬背，能理解几条就是几条（3分）

PCB是操作系统中用于管理进程的一种数据结构，它保存了进程的各种信息，以便操作系统能够对进程进行管理和调度



总之，PCB是操作系统中非常重要的一个概念，用于描述和管理进程的各种属性和状态。PCB的详细描述有助于我们更好地理解进程管理的实现机制，从而更好地进行操作系统的设计和开发。

1. 进入 https://elixir.bootlin.com/linux/latest/source/include/linux/sched.h#L629 查看Linux的task\_struct（即PCB）结构，指出哪一行表示什么，比如，第726行tasks字段，就表示把PCB组织成一个双向链表结构，等等，找的越多，并且你真的知道他们含义越好（3分）

。

在Linux的task\_struct结构定义中，第629行代码是 struct thread\_info \*thread\_info;，它表示进程的线程信息。线程信息包含了进程运行时所需的所有信息，如进程的内核栈、进程ID、进程的调度优先级等等。该指针指向一个单独的数据结构，该结构是一个单独的线程信息块，其中包含了这个进程/线程的所有状态信息。

在Linux内核中，task\_struct结构表示进程控制块（PCB），其中的每个成员都记录着一个进程的相关信息。根据Linux内核版本不同，task\_struct结构的定义可能会有所不同。

在最新版本的Linux内核（v5.15）中，task\_struct结构的定义可以在以下文件中找到：include/linux/sched.h

其中，第1170行到1197行的代码表示了task\_struct结构的定义。

例如，其中的第1179行代码volatile long state;表示进程的状态，第1194行代码struct mm\_struct \*mm;表示进程的内存管理结构体。其他的成员则包括进程ID，进程优先级，进程调度信息，进程打开的文件列表等等。每个成员都记录着一个进程的相关信息。