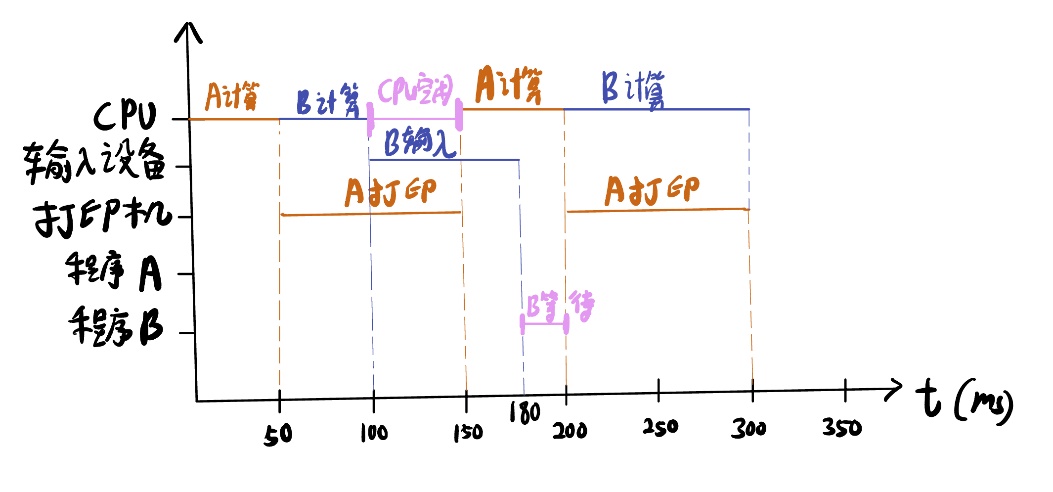
1. 在一个计算机系统中，有一台输入设备和一台打印机。现有两道程序投入运行，并且程序A先于程序B开始运行。程序A的运行轨迹为：计算50ms，打印100ms，计算50ms，打印100ms；程序B的运行轨迹为：计算50ms，输入80ms，计算100ms。试说明：(a) 两道程序运行时，CPU有无空闲等待？若有，请给出其等待时间段，并说明原因。(b) 程序A和B有无等待CPU的情况？若有，请给出等待时间段，并说明原因。(3) 若采用不可剥夺调度策略，不考虑调度和切换时间，则完成两个作业需要的时间最少是多少。请说明原因。

答：以下是根据题目画的图



1. CPU在100-150ms处于空闲等待的状态，原因是当程序A在打印时，CPU不执行任何计算任务，此时程序B也处于输入阶段，CPU也无法执行其他任务，因此出现了空闲等待。
2. 程序B出现等待CPU的情况，在180-200ms，原因是程序A正在使用CPU进行计算。
3. 两个作业需要的时间最少是530ms，这是因为每个程序都需要等待前一个程序执行完成后才能运行，因此两个作业所需要的时间是各别完成的时间叠加。

2. 微内核与外核的主要区别是什么？它们分别有哪些优势和劣势？微内核的性能一定差吗？瓶颈在哪里？

微内核与外核的主要区别是内核和“用户”空间的架构。微内核以客户和服务器为标准，将一些核心的功能以独立的服务器运行在“用户“空间，用户通过访问内核再访问服务器。而外核则是将整个操作系统作为一个应用程序运行再”用户“空间。

微内核的优势在于易于扩展、易于移植，相对于外核更可靠和安全，它的劣势是在于性能相对较低，因为核心功能模块化后，需要进行更多的进程间的通信，而进程间的开销较大。外核的劣势在于可靠性和安全性相对地低，而它的优势是OS无抽象，能够在理论上提供最优性能、应用对计算有更精确的实时等控制。

微内核的性能不一定差，关键在于具体实现和优化。

微内核的性能瓶颈主要在于进程间通信的开销和模块间的接口设计。