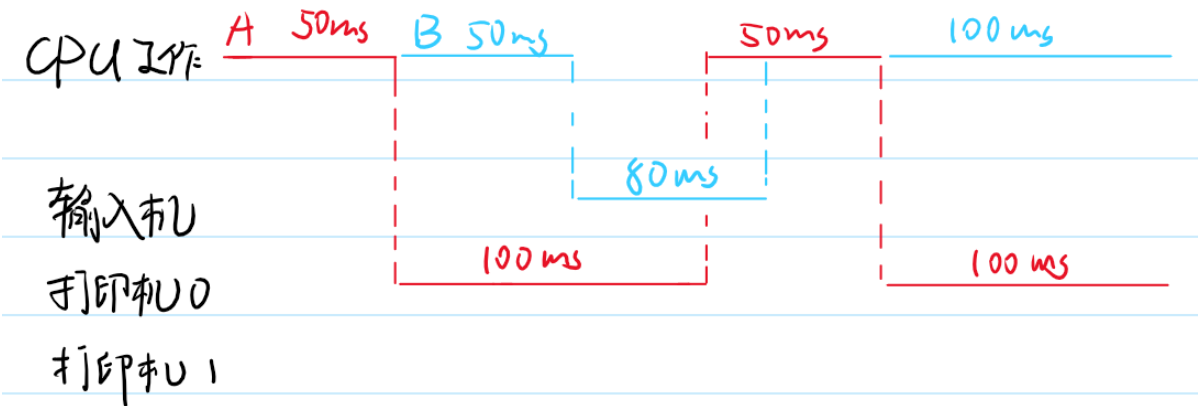


第一次作业

1.

(1) 如图所示：

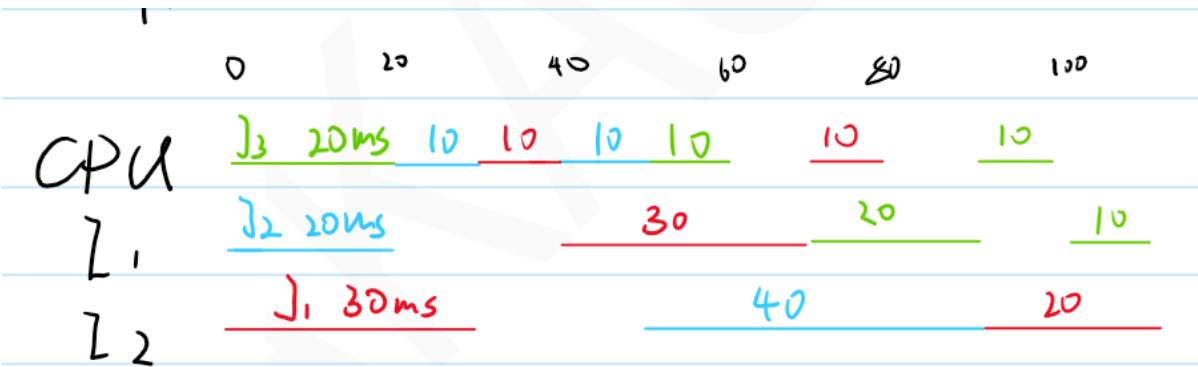


(2) 在如图所示的执行方案中，CPU有空闲等待，在第100ms到第150ms空等。因为此时程序A和程序B均没有完成IO操作，CPU没有可以执行的任务。

(3) 在如图所示的执行方案中，程序A和程序B均存在等待现象。程序A等待打印完毕；程序B等待输入完毕，也等待程序A放弃CPU。此外，如果程序需要打印机或者输入机，但没有可用的打印机或输入机资源时，也需要等待。

2.

程序运行过程如图所示：



各个作业的周转时间：

J1:110ms

J2:90ms

J3:110ms

CPU利用率：80/110=72.8%

I1利用率：80/110=72.8%

I2利用率：90/110=81.8%

3.

(1) 硬件形式多种多样，无论如何进行抽象，总需要面对其间差异的部分。例如，完全可移植的操作系统一定需要面对不同的指令集架构，此时操作系统需要针对不同的指令集进行编译和优化，这可能需要对操作系统内核的底层代码进行修改。此外，不同的外设和硬件组件也需要特定的驱动程序才能在操作系统上正确运行，这也会影响操作系统的可移植性。

(2) 一是硬件抽象层：提供一组标准的硬件抽象接口，以便操作系统可以在不同的硬件架构上运行。例如内存管理、中断处理和设备访问等，并且可以在不同的硬件上进行实现和优化。操作系统的内核可以使用这些接口来访问底层硬件资源，而无需关心底层硬件的具体实现。二是API：提供一组标准的系统调用接口，以便应用程序可以在不同的操作系统上运行。这些接口可以提供一些通用的系统功能，例如文件访问、网络通信和进程管理等，并且可以在不同的操作系统上进行实现和优化。应用程序可以使用这些接口来访问操作系统的功能，而无需关心操作系统的实现。综上，对下，操作系统通过硬件抽象层适应不同的硬件；对上，不同的操作系统通过API为应用程序提供统一的服务。