**Chapter 1**

**1.**

1. manage hardware resources
2. control program running,
3. improve manmachine interface and provide support for application

**2.**

当CPU处于内核模式时，才能执行某些指令，只有当程序在内核模式下执行时，才能访问硬件设备。只有当CPU处于内核模式时，才能控制何时启用或禁用中断。CPU在用户模式下可以强制保护关键资源。

**Chapter 2**

**3.**

provide the interface between a running program and the operating system.Exact type and amount of information vary

according to OS and call.

**4.**

1. Program execution。该操作系统将代表用户进行调度。用户无法处理此服务，因为用户需要访问该硬件。

2. I/O operations。用户可以很容易地访问I/O流。用户不需要知道机器中数据的物理访问情况，如果没有用户提供的界面，用户就不能这样做。

3. File-system manipulation。用户不需要担心访问和更新文件系统表。由于这种复杂性，这种访问最好由操作系统来处理，用户很少自己处理。

4. Communications。在内存映射的情况下，这对操作系统处理对内存区域的访问和控制非常有益的。用户无法访问这样的系统来共享地图。

5. Error detection。如果在较低的级别上出现错误，会通知用户以便他们可以采取行动。例如，如果堆上没有内存，用户不能这样做，因为这对用户来说工作太多了。

**5.**

一种是shared memory模式，一种是message passing模式。

message passing的优缺点：消息可以通过公共的邮箱等内容在进程之间直接或间接交换。它可以用于交换较少的数据，并且更容易实现计算机间通信，并且易于维护、保护。但是，它的速度比共享内存模型慢。

Shared memory的优缺点：它使得通信的速度得到最快发展，并且具有方便性，但在进程之间的保护和同步方面存在一些问题。

**Chapter 3**

**6.**

当进行上下文切换时，内核会将旧进程状态保存在其PCB中，然后加载经调度而要执行的新进程的上下文。上下文包括CPU寄存器的值、进程状态和内存管理信息等。

**7.**

**C**

**8.**

长期调度选择哪一个进程应该被引入到准备好的队列，特点是：少执行（例如，一次几分钟），控制多重编程的程度，选择一个良好的I/O-bound流程的流程组合和cpu-bound进程。

短期调度程序，选择下一个应该执行哪个进程并进行分配中央处理器，特点是：执行相当频繁（例如，每100 ms一次）、必须是非常快。

中期调度：交换输出：从内存中移除进程以减少多道程序数，交换输入：将交换输出进程重新引入到内存中。

区别：

长期调度：job scheduling, select job from external storage to memory and create a process

短期调度：process scheduling, select the ready process to run on the processor

中期调度：solves the problem of insufficient memory, using secondary storage to alleviate (controls the degree of multiprogramming)

**Chapter 4**

**9.**

1. **C**

**10.**

有可能，假设有A、B两个进程，都有N个任务，那么当A完成第一个任务之后，又转到B完成第一个任务，然后又转回来，一直保持这种模式，就是并发而不是并行。

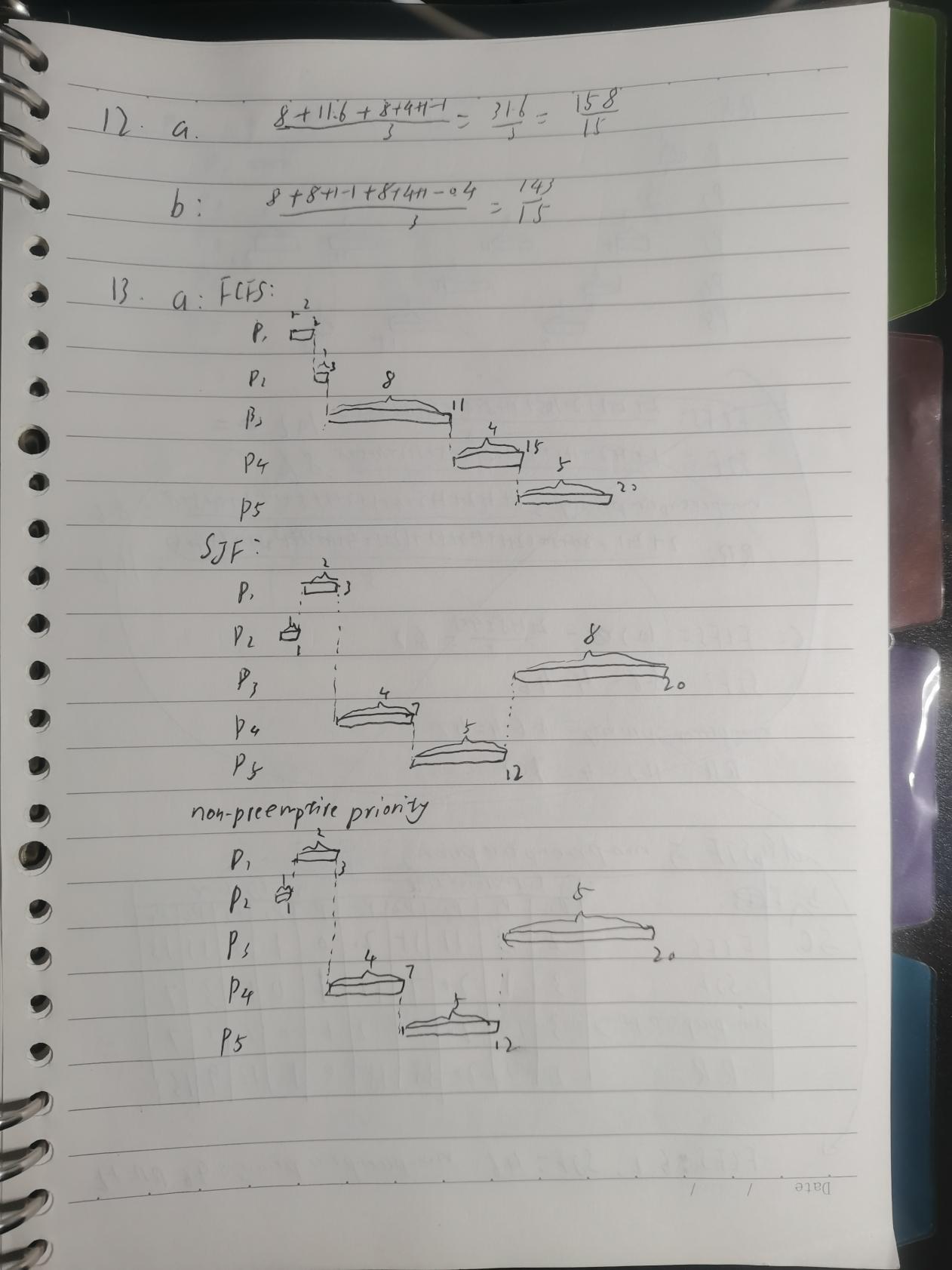
**Chapter 5**

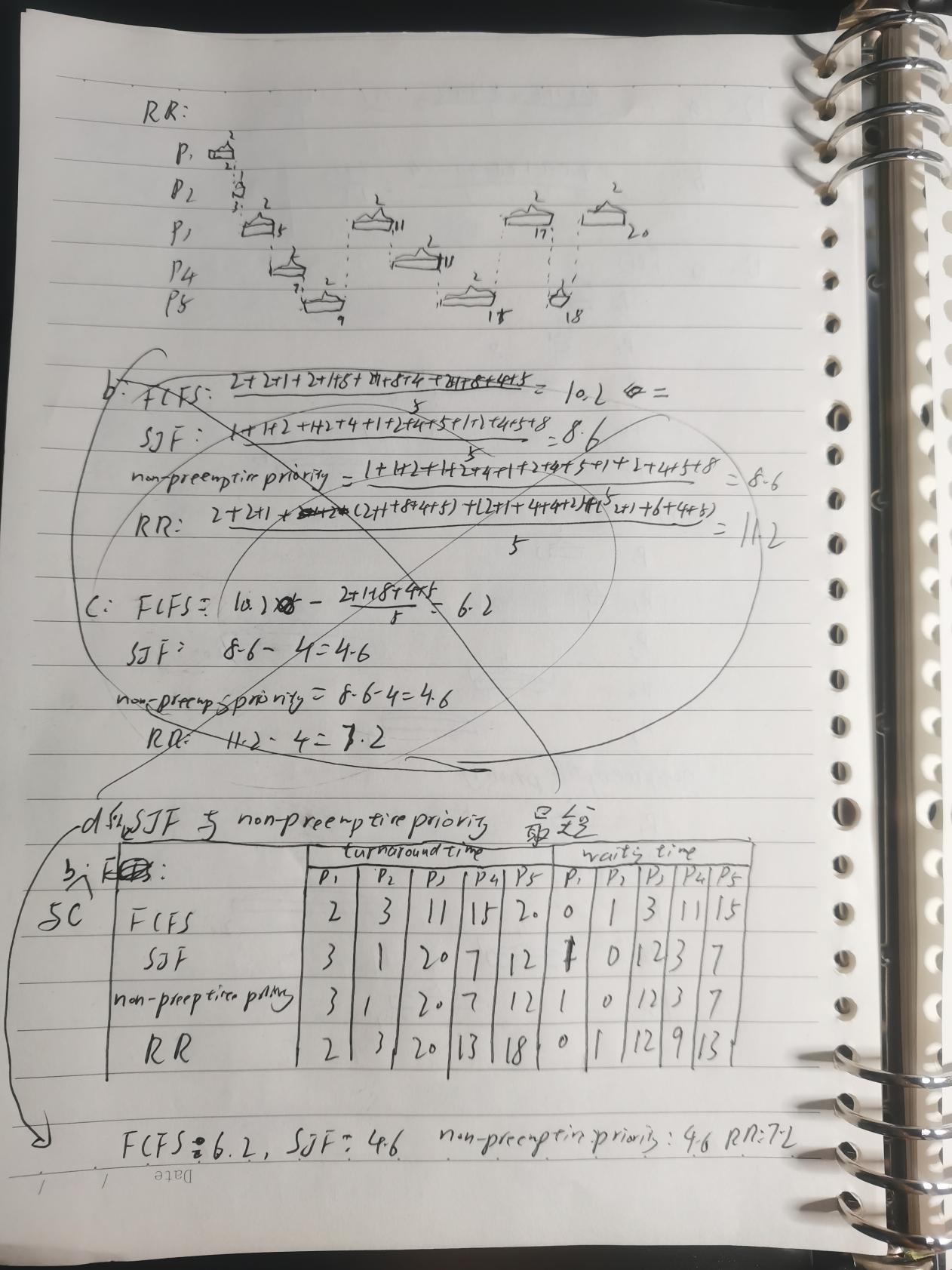
**11.**

对于长期调度程序，知道进程是IO约束的还是CPU约束的之后，便可以按照一定比例组合被调度的进程，使得CPU和IO设备都会一直处于负载状态，从而提高资源的使用率。

对于短期调度（CPU调度），IO约束的进程CPU区间都比较短，可以根据最短作业优先准则，可以优先调度IO约束进程，加快效率。

1. **至17**

****

****

