

OS 第一章作业

191220008 陈南瞳

问答题

1、What are the two main functions of an operating system?

操作系统既是“管理员”，又是“服务员”。

①对内作为“管理员”，做好计算机系统软硬件资源的管理、控制与调度，在裸机基础上形成虚拟机供应用程序使用，并对程序执行进行控制和协调，提高系统效率和资源利用率。

②对外作为“服务员”，是用户与硬件的接口和人机界面，在管理好资源的基础上，为用户提供友善运行环境和各种公共服务。

2、On early computers, every byte of data read or written was handled by the CPU (i.e., there was no DMA). What implications does this have for multiprogramming?

多道程序设计是在计算机内存中同时存放几道相互独立的程序，使它们在管理程序控制之下，相互穿插的运行。进行多程序编程的主要原因是在等待I/O完成的同时给CPU做一些事情。在早期计算机中，CPU与其他硬件设备串行工作，由于I/O速度慢，CPU处理速度快，导致CPU长时间处于等待状态，而多道程序设计则可以让CPU利用等待时间去做其他事情，提高效率和吞吐率。

3、What is the difference between kernel and user mode? Explain how having two distinct modes aids in designing an operating system.

处于用户态执行时，进程所能访问的内存空间、对象和指令受到限制，其所处于占有的处理机是可被抢占的；而处于核心态执行中的进程，则能访问所有的内存空间、对象和指令，且所占有的处理机是不允许被抢占的。

在操作系统中设置两种模式是为了在用户态下运行用户程序时，无法调用一些特权指令，访问也受到限制，起到了良好的保护和封装。

4、What is a trap instruction? Explain its use in operating systems.

陷阱指令可以让CPU的执行模式从用户态陷入内核态。这种机制可以让用户调用一些内核的函数。

5、What type of multiplexing (time, space, or both) can be used for sharing the following resources: CPU, memory, disk, network card, printer, keyboard, and display?

时间：处理器，网卡，打印机，键盘

空间：内存，磁盘

两者兼有：显示

6、To a programmer, a system call looks like any other call to a library procedure. Is it important that a programmer know which library procedures result in system calls? Under what circumstances and why?

在通常情况下，对于程序的逻辑本身而言，系统调用和库函数调用并无差别。但在涉及程序性能时，由于系统调用涉及上下文的切换，会带来额外的时间开销，所以通常会减低程序性能。因此在考虑程序性能时，程序员需要知道哪些库函数会进行系统调用，以便于控制时间开销。

7、Explain how separation of policy and mechanism aids in building microkernel-based operating systems.

机制和策略的分离使得在内核中仅实现少量的基本原语，它们不依赖于任何特定的策略，使得软件适应性好且易于开发。然后将它们用于在用户级别实施更复杂的机制和策略，及时存在改变策略的需要，系统本身也可以保持不变。

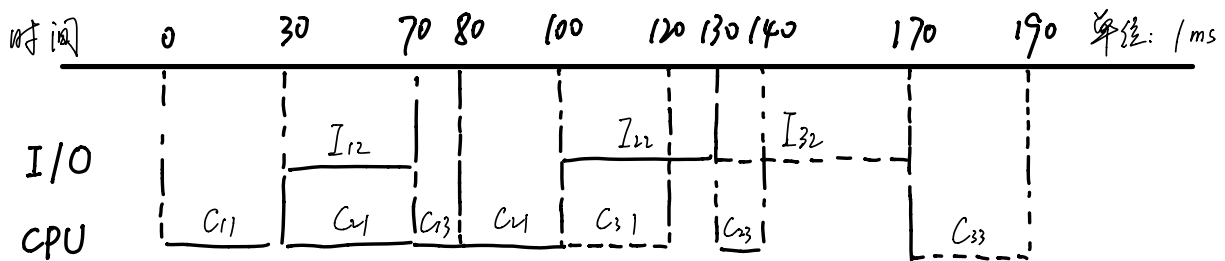
应用题

3. 设有三道程序,按照 A、B、C 的优先次序运行,其内部计算和 I/O 操作时间如下图所示。

A	B	C
$C_{11} = 30 \text{ ms}$	$C_{21} = 60 \text{ ms}$	$C_{31} = 20 \text{ ms}$
$I_{12} = 40 \text{ ms}$	$I_{22} = 30 \text{ ms}$	$I_{32} = 40 \text{ ms}$
$C_{13} = 10 \text{ ms}$	$C_{23} = 10 \text{ ms}$	$C_{33} = 20 \text{ ms}$

试画出多道运行的时间关系图(忽略调度执行时间)。完成三道程序共花费多少时间?比单道运行节省多少时间?若处理器调度程序每次进行程序转换费时 1 ms,试画出各程序状态转换的时间关系图。

① 多道运行,忽略调度(抢占式):

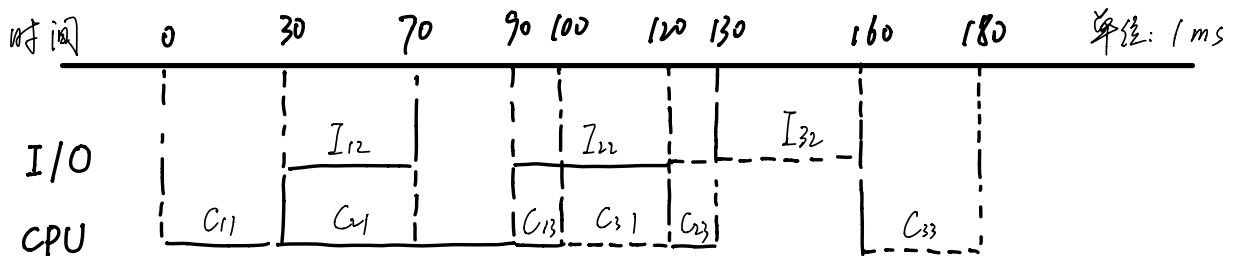


单道运行: $30 + 40 + 10 + 60 + 30 + 10 + 20 + 40 + 20 = 260 \text{ ms}$

抢占式: 190 ms

节省: $260 - 190 = 70 \text{ ms}$

② 多道运行,忽略调度(非抢占式):

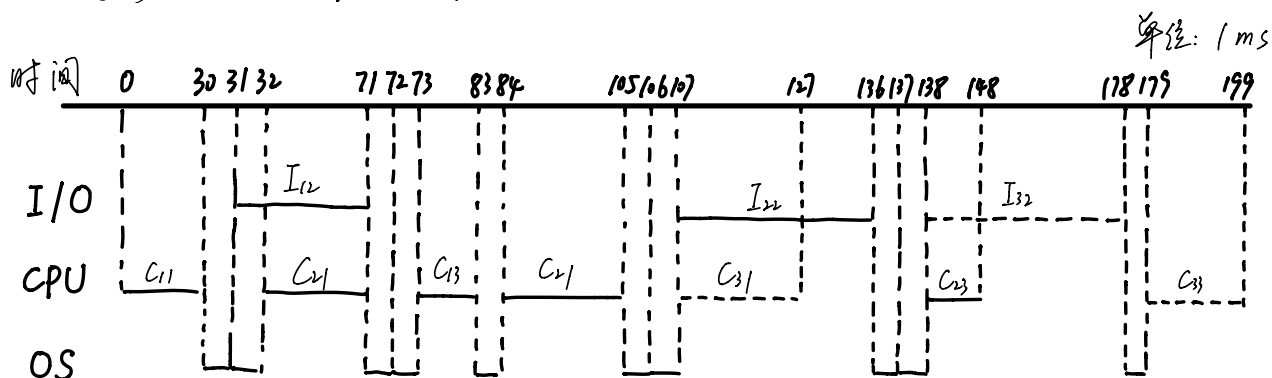


单道运行: $30 + 40 + 10 + 60 + 30 + 10 + 20 + 40 + 20 = 260 \text{ ms}$

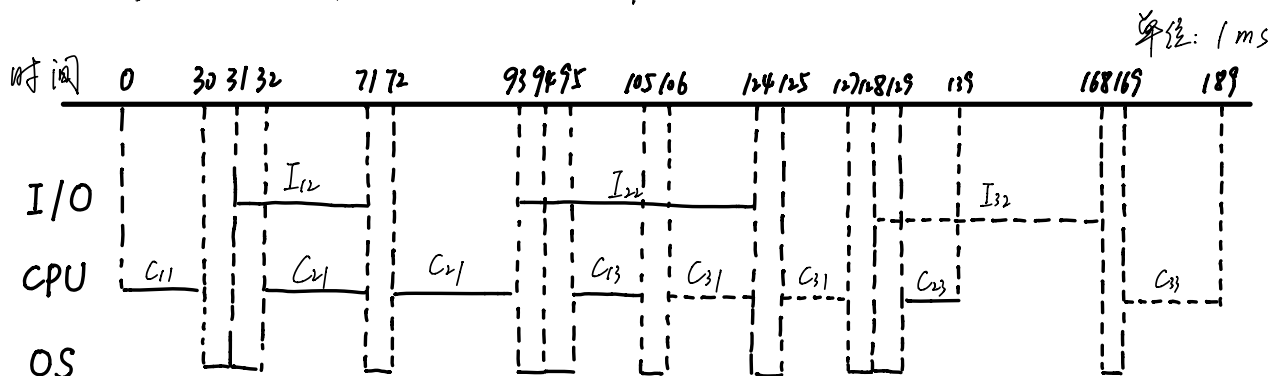
非抢占式: 180 ms

节省: $260 - 180 = 80 \text{ ms}$

③ 多道运行, 考虑调度 1ms (抢占式):



④ 多道运行, 考虑调度 1ms (非抢占式):



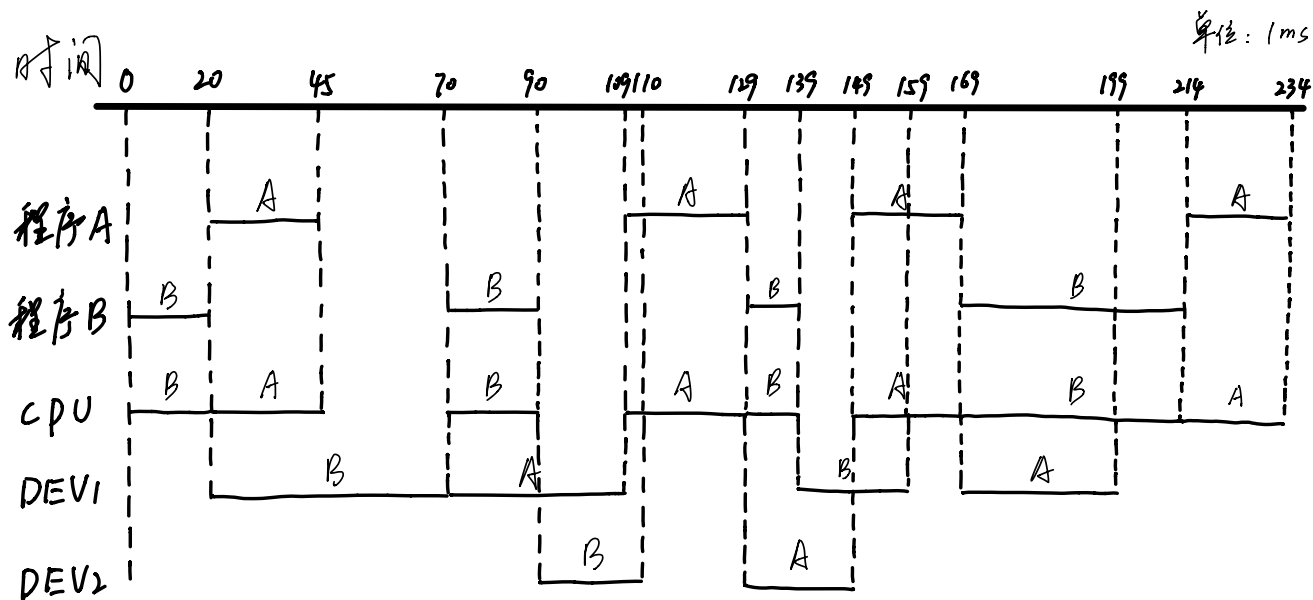
9. 在单机系统中, 有同时到达的两个程序 A、B, 若每个程序单独运行, 则使用 CPU、DEV₁ (设备 1)、DEV₂ (设备 2) 的顺序和时间如下表所示。

程序	运行情况/ms						
	CPU	DEV ₁	CPU	DEV ₂	CPU	DEV ₁	CPU
程序 A	25	39	20	20	20	30	20
	CPU	DEV ₁	CPU	DEV ₂	CPU	DEV ₁	CPU
程序 B	20	50	20	20	10	20	45
	CPU	DEV ₁	CPU	DEV ₂	CPU	DEV ₁	CPU

给定下列条件:

- (1) DEV₁ 和 DEV₂ 是不同的 I/O 设备, 它们能够同时工作。
 - (2) 程序 B 的优先级高于程序 A。但是当程序 A 占用 CPU 时, 即使程序 B 需要使用 CPU, 也不能打断程序 A 的执行而应等待。
 - (3) 当使用 CPU 之后控制转向 I/O 设备, 或者使用 I/O 设备之后控制转向 CPU, 由控制程序执行中断处理, 但这段处理时间可以忽略不计。试解答下列问题:
- (1) 哪个程序先结束?
 - (2) 程序全部执行结束需要多长时间?
 - (3) 程序全部执行完毕时, CPU 的利用率是多少?
 - (4) 程序 A 等待 CPU 的累计时间是多少?
 - (5) 程序 B 等待 CPU 的累计时间是多少?

画出运行图：



(1) A在234ms结束, B在214ms结束

因此B先结束

(2) 程序全部执行完需要234ms

$$(3) \text{CPU利用率} = \frac{20 + 25 + 20 + 20 + 10 + 20 + 45 + 20}{234} \times 100\%$$

$$= 76.92\%$$

(4) A等待CPU: $20 + 15 = 35\text{ms}$

(5) B等待CPU: $19 + 10 = 29\text{ms}$