# 第二章 中央处理器

1. **2.1 CPU组成**
2. **2.2 数据表示**
   1. 数据表示：**由硬件直接辨识的数据类型**（这节唯一要掌握）

堆栈，队列，浮点数

* 1. 数据表示的目的是 确定哪些数据类型用硬件/软件实现。

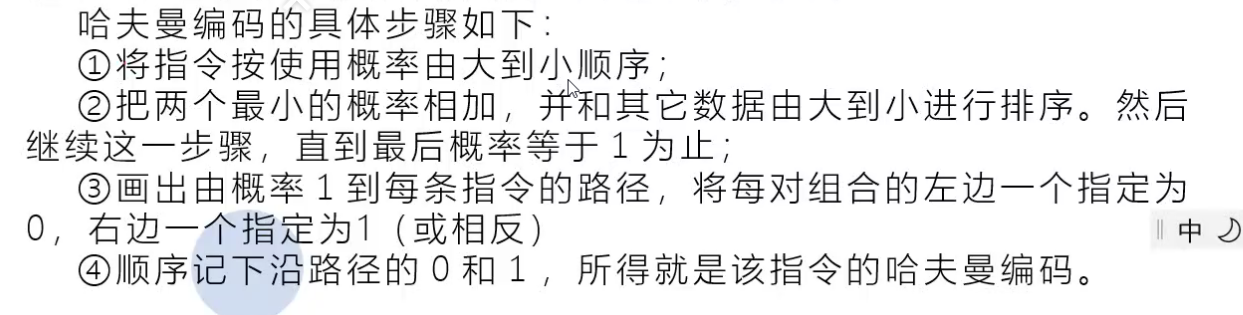
1. **2.3 指令优化**

指令 = 操作数（数据表示） + 操作码

* 1. **2.3.1 指令格式优化**

优化操作码，地址码优化（不讲）

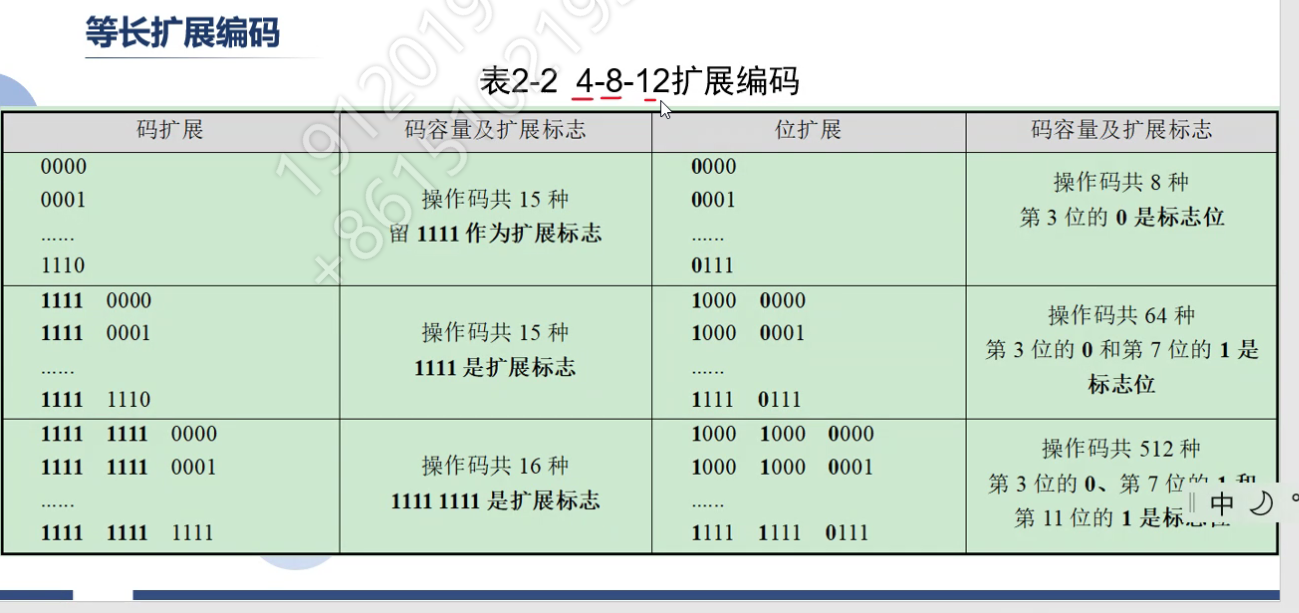
* + 1. 压缩方式
       1. **定长编码**
       2. **哈夫曼编码**
          1. 哈夫曼压缩：出现概率最大的事件用最少的位（或最短的时间）来表示（或处理）。
          2. 过程：



* + - * 1. 例子：



* + - 1. **扩展哈夫曼编码**
         1. 等长扩展编码（类似地址码扩展）



位扩展：每次保留一个二进制位不使用，用于进一步扩展

码扩展：每次保留一个编码不使用，用于进一步扩展

* + - * 1. 例子



* + 1. 评价标准
       1. 哈夫曼信息源熵



* + - 1. 信息冗余量 = 1 - 平均最优码长 / 实际码长

**信息冗余量 越高， 平均码长效率越低**

* 1. **2.3.2 指令系统分析**
     1. CPU的两种架构
        1. RISC：精简指令集
        2. CISC：复杂指令集

1. 2.4 指令集

知道概念即可

* 1. 2.4.1 RISC 指令集
  2. 2.4.2 CISC 指令集

1. **2.5 时钟频率（略过）**
2. **2.6 并行**

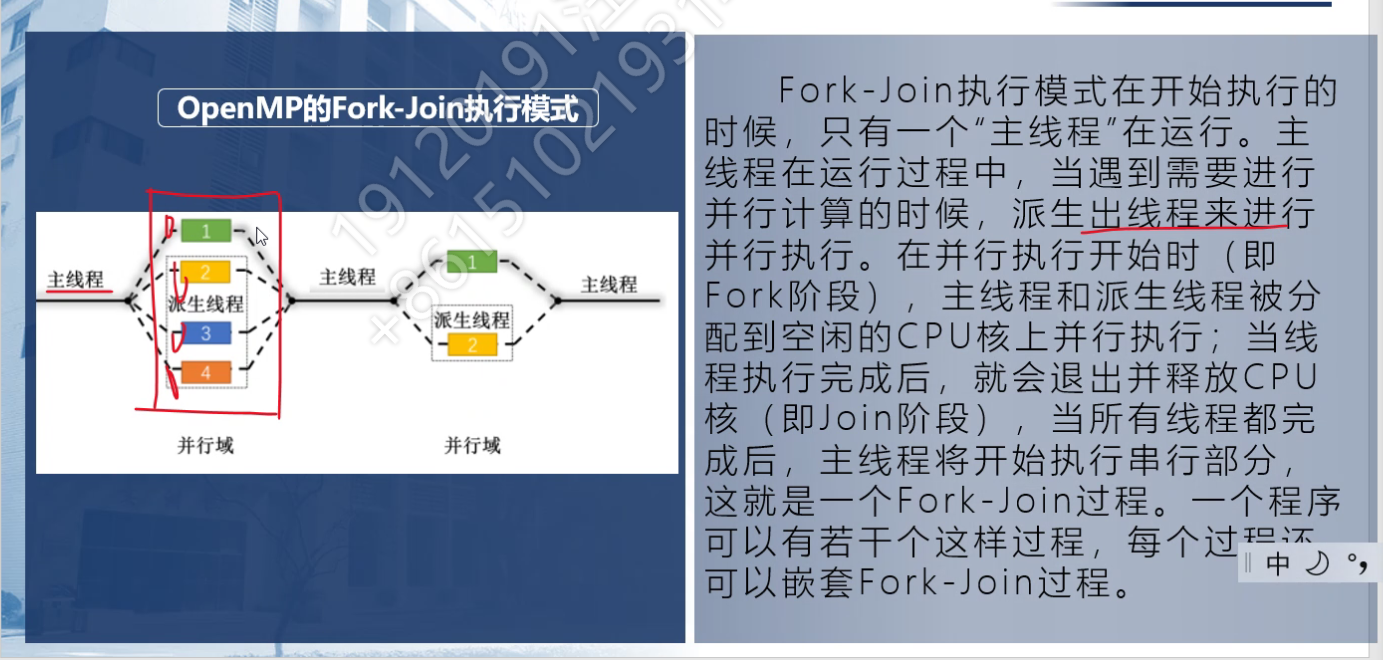
**了解并行的概念，系统结构中有哪些并行**

* 1. **2.6.1 指令级并行**
     1. 宏观上，同时执行多条指令；微观上，不同指令在同一时刻使用CPU的不同部分。
     2. 时间并行；空间并行；时间和空间并行相结合
  2. **2.6.2 线程级并行**
     1. **1. 超线程技术**

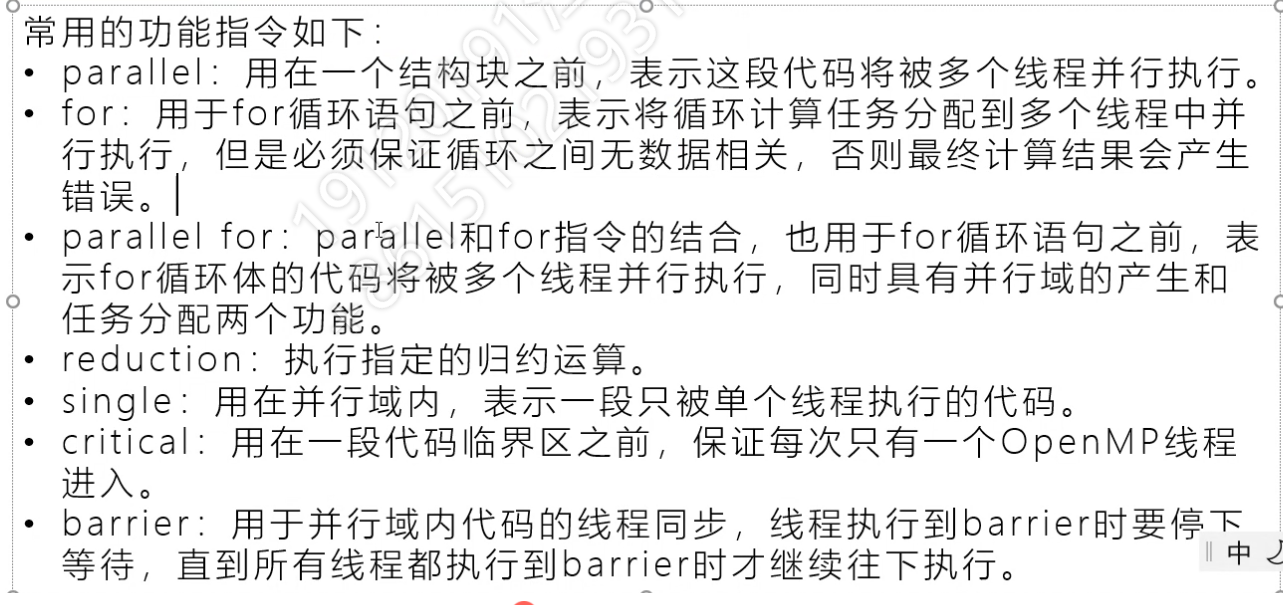
多核没有产生时，出现的并行技术

* + - 1. 在同一时间里，应用程序可以使用芯片的不同部分。
  1. **2.6.3 数据并行和SIMD指令集合**
     1. 把需要处理的数据划分成若干部分，分配到不同的处理部件上，并在每个处理部件上运行相同的处理程序对分派的数据进行处理。

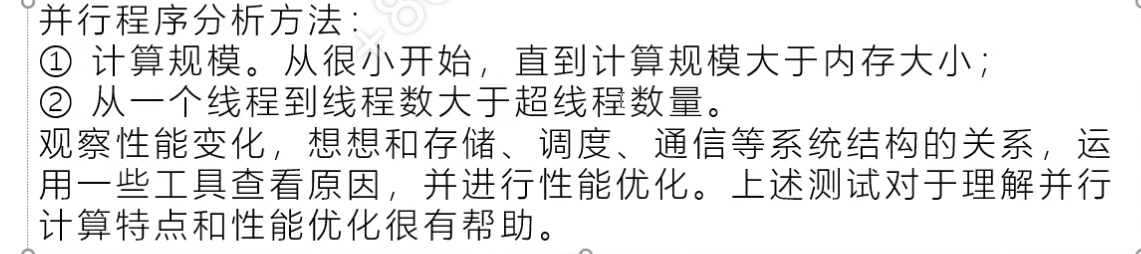
1. **2.7 多核技术**
   1. **现代处理器结构**
      1. 早期：多CPU技术
      2. 多核技术
         1. 多个CPU内核集成到单个集成电路芯片中。
         2. 每个内核具有：独立的L1缓冲和L2缓存，共享末级（L3）缓存和内存控制器
2. 2.8 CPU内部互联（了解）
3. 2.9 CPU外部互联（了解）
4. **2.10 OpenMP多线程并行编程**
   1. 资源分配的最小单位：进程
   2. CPU调度的最小单位：线程
   3. OpenMP的编程模型以线程为基础，通过编译制导语句
   4. Fork-Join 指令模式



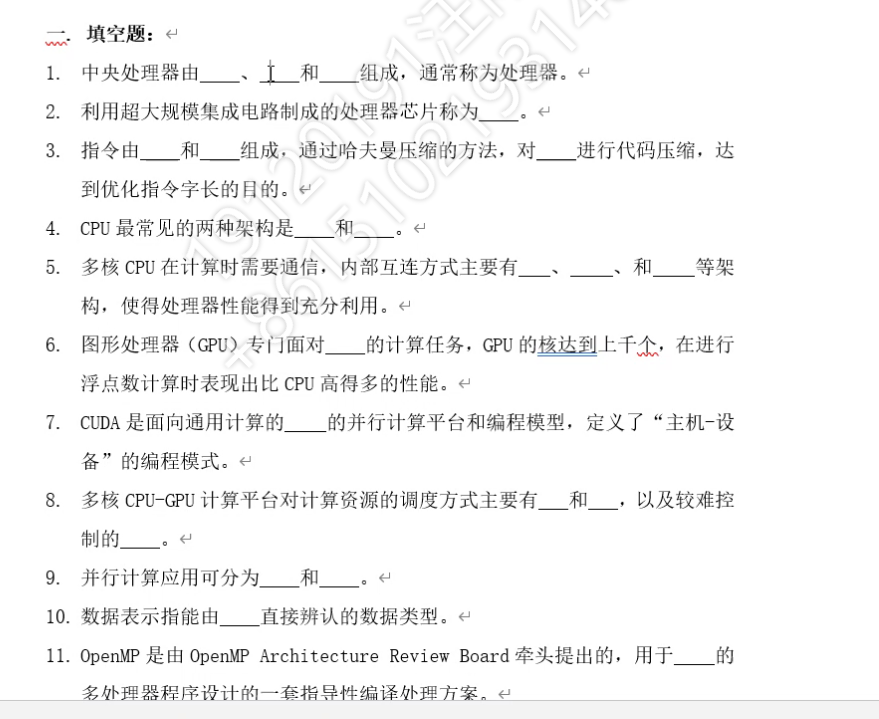
* 1. **2.10.1 编译制导语句**
     1. # pragma omp <directive>...
     2. 常用功能



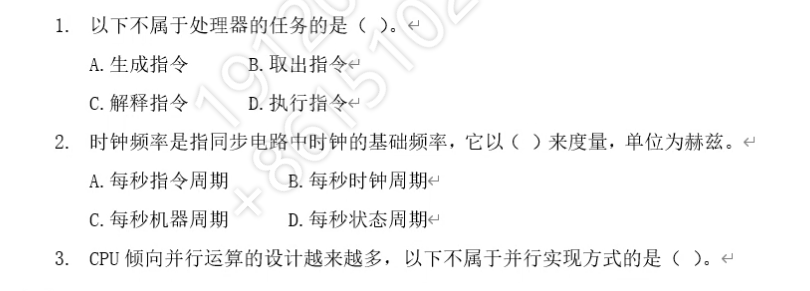
* 1. **2.10.2 API 函数**、
     1. omp\_get\_thread\_num: 返回线程号。
     2. omp\_set\_num\_threads：设置后续并行域的线程个数。
     3. omp\_get\_num\_threads: 返回当前并行域中的线程数。
     4. omp\_get\_num\_procs: 返回系统中处理器的个数。
     5. 并行程序分析方法：

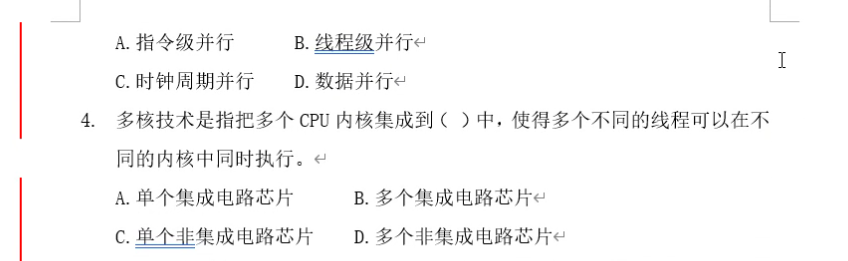


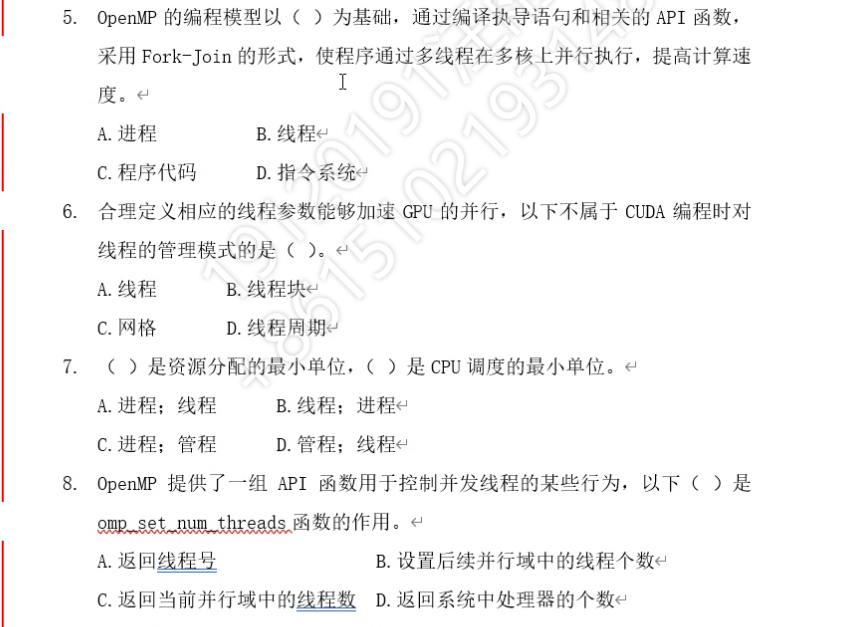
1. 2.11 GPU（概念）
2. 2.12 CUDA（概念）
3. **练习**
   1. **一、填空**

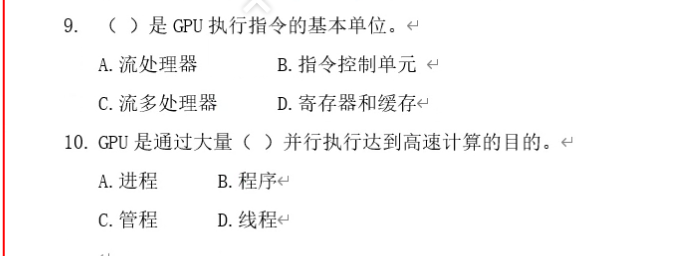


* + 1. 1. 运算器、控制器、寄存器
    2. 3. 操作码；地址码；指令码
    3. 4. RISC；CISC
    4. 10. 硬件
    5. 11. 共享内存
  1. **二、选择**

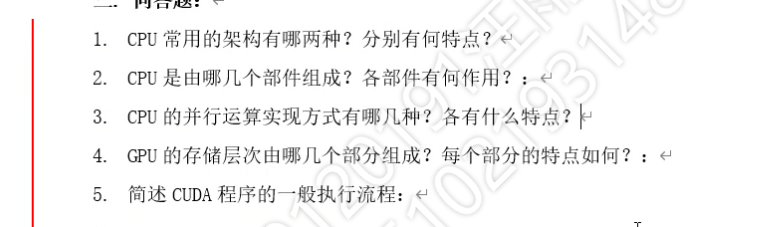








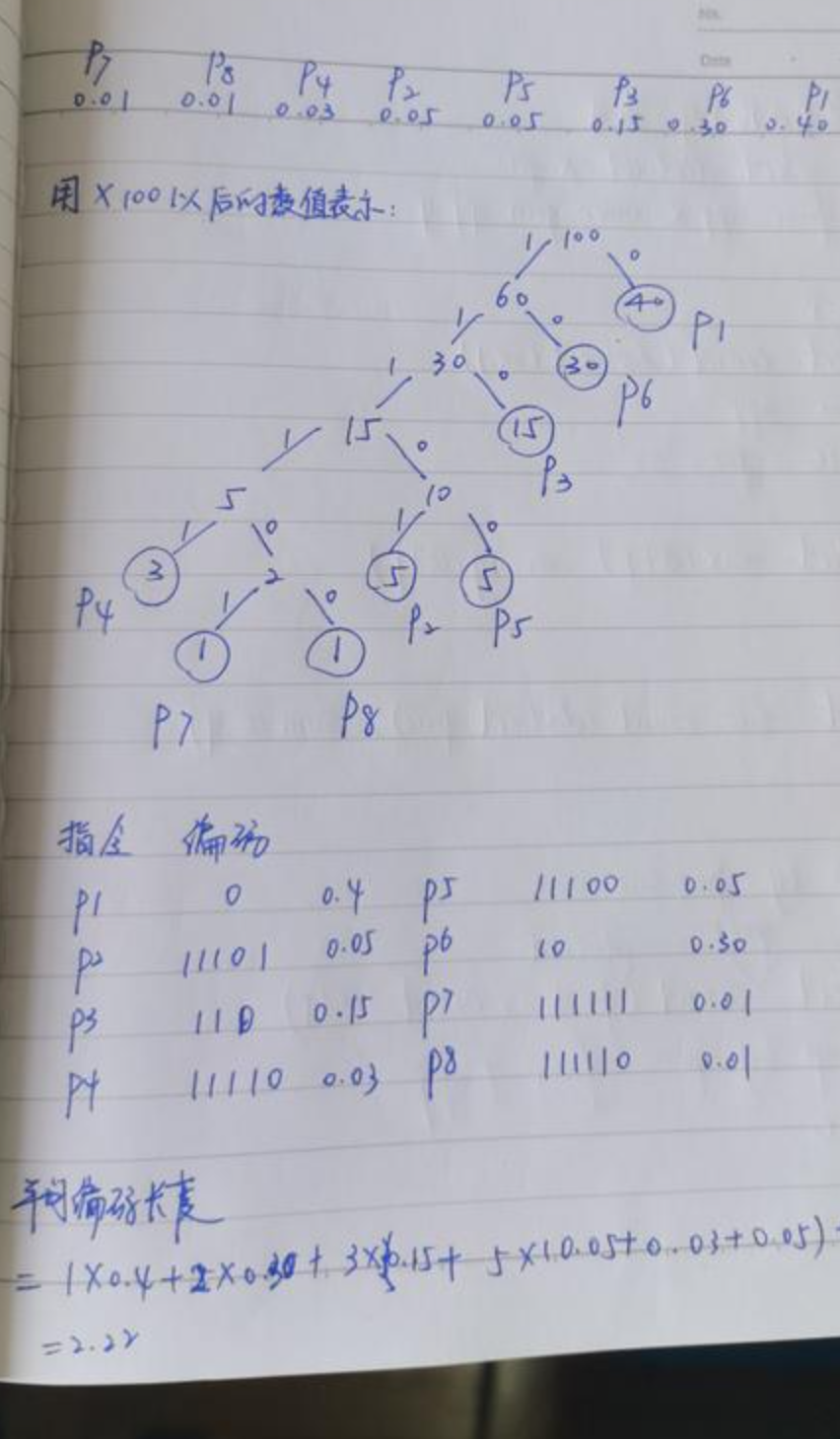
* + 1. 1. A
    2. 2. B
    3. 3. C
    4. 4. A
    5. 5. B
    6. 6.
    7. 7. A
    8. 8. B
  1. **三、问答题**



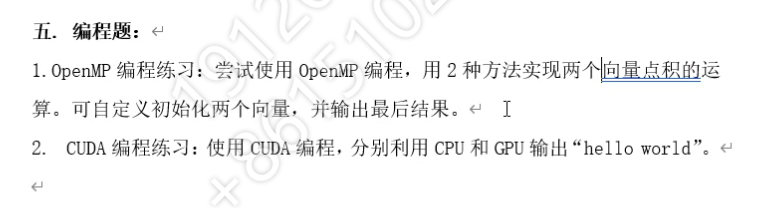
* 1. **四、计算题**



* + 1. 答案



* 1. **五、 编程题**



可以考虑1