1. 请简述 Flynn 分类法将计算机系统结构分成哪四类。

答：1.单指令流、单数据流 SISD

2.单指令流、多数据流 SIMD

3.多指令流、单数据流 MISD

4.多指令流、多数据流 MIMD

2. 请简述程序局部性原理。

答：局部性原理指程序执行时所访问的存储器地址分布不是随机的，而是相对的聚集。它分为时间局部性和空间局部性。

1. 程序的时间局部性：程序即将用到的信息很可能就是目前正在使用的信息；
2. 程序的空间局部性：程序即将用到的信息很可能与目前正在使用的信息在空间上相邻或者临近；

3. 请简述 Amdahl 定律。

答：加快某部件执行速度所能获得的系统性能加速比，受限于该部件的执行时间占系统中总执行时间的百分比。

$$加速比=\frac{系统性能_{改进后}}{系统性能_{改进前}}=\frac{总执行时间_{改进前}}{总执行时间_{改进后}}$$

4. 影响 CPU 时间的因素包括什么？（提示：从 CPU 公式入手，考虑 3 个参数的影响因素）

答：时钟周期时间、CPI（每条指令执行的平均时钟周期数）、IC（指令条数）

5. 请简述冯诺依曼体系结构的特点。

答：1.计算机应该包括五个基本组成部分：运算器、控制器、存储器、输入输出设备；

2.计算机内部应采用二进制来表示指令和数据；

3.采用存储程序方式；

6. 请简要说明提高计算机系统并行性的 3 种技术途径，并分别从单机和多机系统的角度举例。

答：1.时间重叠：引入时间因素，让多个处理过程在时间上相互错开，轮流重叠地使用同一套硬件设备的各个部分，以加快硬件周转而赢得速度；

举例：单处理机（部件功能专用化）：指令流水线

多处理机（处理机专用化）：专用外围处理机、专用处理机、异构型多处理机系统

2.资源重复：引入空间因素，以数量取胜。通过重复设置硬件资源，大幅度地提高计算机系统的性能。

举例：单处理机（重复设置功能部件）：阵列处理机

多处理机（重复设置处理机）：容错系统、同构型多处理机系统

3.资源共享：一种软件方法，它使多个任务按一定的顺序轮流使用桶一套硬件设备。

举例：单处理机：分时系统

多处理机：分布式系统

7. 请分析影响平均访存时间的因素有哪些？针对每个因素，各举出 1 种改进方法。

答：1.平均访存时间=命中时间+不命中率 x 不命中开销；

2.（1）减少命中时间：使用容量小并且结构简单的 cache、虚拟 cache、cache 访问流水化、踪迹 cache

（2）降低不命中率：增加块大小、增加 cache 容量、提高相联度、伪相联 cache、硬件预取、编译器预取、编译器优化、牺牲 cache

（3）减少不命中开销：采用两级 cache、读不命中优先于写、写缓冲合并、请求字处理技术、非阻塞 cache

8. 请简述 Cache 的 3C 失效。

答：1.强制性不命中：当第一次访问一个块时，该块不在 cache 中，需从下一级存储器中调入 cache;

2.容量不命中：如果程序执行时所需的块不能全部调入 cache 中，则当某些块被替换后，若又重新被访问，就会发生不命中;

3.冲突不命中：在组相联或直接映像 cache 中，若太多的块映像到同一组（块）中，则会出现该组中某个快被别的块替换（即使别的组或块有空闲位置），然后又被重新访问的情况。

9. 请针对 3C 失效的每种失效给出一种降低失效率的方法，并分别分析该种方法的缺点。

答：1.强制性不命中：方法：增加块的大小

缺点：会增加不命中开销

2.容量不命中：方法：增加 cache 容量

缺点：增加成本，可能增加命中时间

3.冲突不命中：方法：提高相联度

缺点：增加命中时间

10.请简述伪相联的基本思想。

答：在逻辑上把直接映像 cache 的空间上下平分为两个区。对于任何一次访问，伪相联 cache 先按直接映像 cache 的方式去处理。若命中，则其访问过程与直接映像 cache 的情况一样。若不命中，则到另一区相应位置查找，找到则发生伪命中；否则访问下一级存储器。

11.请例举一种编译器优化的方法，并举例说明。

答：1.数组合并

```
/* 修改前 */
int x[100];
int y[100];
```

若x[i]和y[i]经常一起访问

```
/* 修改后 */
struct merge{
    int x;
    int y;
};
struct merge
merged_array[100];
```

2.内外循环交换

举例：

```
/* 修改前 */（存储器按行存储，按列操作）
    for （ j = 0 ; j < 100 ; j = j+1 ）
        for （ i = 0 ; i < 5000 ; i = i+1 ）
            x [ i ][ j ] = 2 * x [ i ][ j ];
```

a00, a01, a02 ..... ao99

a10, a11, a12 ..... a199

a990, a991,a992.....a9999

获取a00,a10,.....a990每次都失效

```
/* 修改后 */
        for （ i = 0 ; i < 5000 ; i = i+1 ）
            for （ j = 0 ; j < 100 ; j = j+1 ）
                x [ i ][ j ] = 2 * x [ i ][ j ];
```

修改后程序顺序依次访问地访问同一个cache块中的各个元素，然后在访问下一块的各个元素

3.循环融合

```
/* 修改前 */
for（i = 0; i < 100; i = i+1 ）
    for（j = 0; j < 100; j = j+1 ）
        x[i][j] = a[i][j] + b[i][j];
for（i = 0; i < 100; i = i+1 ）
   for（j = 0; j < 100; j = j+1 ）
       y[i][j] = a[i][j] - b[i][j];
```

```
/* 修改后 */
for（i = 0; i < 100; i = i+1 ）
   for（j = 0; j < 100; j = j+1 ）{
       x[i][j] = a[i][j] + b[i][j];
       y[i][j] = a[i][j] - b[i][j];
}
```

4.分块：通过提高局部性减少不命中。把对数组的整行或整列的访问改为按块进行，尽量集中访问，减少替换，提高访问的局部性。

```
/* 修改前 */

      for （ i = 0; i<N; i = i+1 ）
         for （ j = 0; j < N; j = j+1 ）
         {
            r = 0;
            for （ k = 0; k < N; k = k+1）
            {
                r = r + y[ i ][ k ] * z[ k ][ j ];
            }
            x[ i ][ j ] = r;
         }
```

最坏的情况：所有的访问都不命中
不命中次数：$2N^3+N^2$

```
/* 修改后 */
 for （ jj = 0; jj < N; jj = jj+B ）
    for （ kk = 0; kk < N; kk = kk+B ）
       for （ i = 0; i < N; i =i+1 ）
          for （ j = jj; j < min（jj+B-1, N）; j = j+1 ）
          {
              r = 0;
              for （ k = kk; k < min（kk+B-1, N）; k = k+1）
              {
                  r = r + y[ i ][ k ] * z[ k ][ j ];
              }
              x[ i ][ j ] = x[ i ][ j ] + r;
          }
```

（不命中次数：$2N^3$ /B ＋N^2）

12.请说明流水线中有哪三种相关？分别会引起哪种流水线中的冲突（冒险）？

答：1.数据相关，会引起数据冲突中的写后读(RAW)冲突

2.名相关，会引起数据冲突中的读后写(WAR)和写后写(WAW)冲突

3.控制相关，会引起控制冲突

13.请说明什么是静态调度？什么是动态调度？动态调度的优点是什么？

答：1.静态调度：依靠编译器对代码进行调度，也就是在代码被执行之前进行调度，以减少相关和冲突；

2.动态调度：在程序的执行过程中，依靠专门硬件对代码进行调度，减少数据相关导致的停顿；

3.动态调度的优点：

（1）能够处理一些在编译时情况不明的相关，并简化了编译器；

（2）能够使本来是面向某一流水线优化编译的代码在其他的流水线（动态调度）上也能高效地执行；

4.动态调度的缺点：以硬件复杂性的显著增加为代价；

14.请简述 Tomasulo 算法的基本思想。

答：1.记录和检测指令相关，操作数一旦就绪就立即执行，把发生写后读(RAW)冲突的可能性减少到最小（通过 CDB）；

2.通过寄存器换名来消除读后写(WAR)冲突和写后写(WAW)冲突；

15.请说明什么是动态分支预测？有何优点？

答：1.动态分支预测：采取硬件技术。在程序执行时，根据每一条转移指令过去的转移历史记录预测下一次转移的方向。提前预测分支方向，减少或消除控制相关导致的流水线停顿；

2.优点：根据程序的执行过程动态地改变转移的预测方向，有更好的准确度和适应性

16.请简述分支历史表 BHT 的基本思想。

答：用 BHT 来记录分支指令最近一次或几次的执行情况（成功或不成功），并据此进行预测。分为只有 1 个预测位的分支预测缓冲和两位二进制位作为预测位的分支预测缓冲；

17.请简述分支目标缓冲器 BTB 的基本思想。

答：将分支成功的分支指令的地址和它的分支目标地址都放到一个缓冲区中保存起来，缓冲区以分支指令的地址作为标识，这个缓冲区就是 BTB。目的是将分支的开销降为 0；

18.请简述基于硬件的前瞻算法基本思想。

答：对分支指令的结果进行猜测，并假设这个猜测总是对的，然后按这个猜测结果继续取出、流出和执行后续的指令。执行指令的结果不是写回到寄存器或存储器，而是放到一个称为 ROB 的缓冲器中。等到相应的指令得到“确认”之后，才将结果写入寄存器或存储器。目的是在猜测错误时能够恢复现场。

19.请简述向量体系结构和 GPU 体系结构的差异。

答：1.向量体系结构：

（1）“窄而深”；

（2）ALU 宽度窄，指令流水线深；

（3）单次指令流水后能处理更多数据，掩盖不必要的流水线时间；

2.GPU 体系结构：

（1）“宽而浅”；

（2）ALU 宽度宽，指令流水线浅；

（3）流水线本身比较简单，直接对更多的数据进行并行计算，同一时刻处理更多数据

20.请简述 GPU 和 CPU 在设计理念上的差异性。

答：1.CPU 面向延迟设计：

(1)强大的 ALU，降低运算延迟；

(2)大量的 cache，把长延迟内存访问转换为短延迟 cache 访问；

(3)复杂的控制逻辑，分支预测降低分支延迟，数据转发降低数据延迟；

2.GPU 面向吞吐量设计：

(1)小型 cache：提高内存吞吐量；

(2)简单控制：无分支预测，无数据转发；

(3)节能型 ALU：多而长延迟，但具有高吞吐量的大量流水线；

需要大量线程才能进行容错。

21. 请简述 GPU 各个层次组件间的相似性。

答：在任务的划分上存在着自相似性。每一个第 i 级的任务，都可以划分为一组 i-1 级的任务，并且同一级别的任务之间是并行的。

22. 请简述 GPGPU 虚拟化的思想。

答：在硬件上把多个 SM 分隔成多个切片，在软件上给每个用户分配一个 SM 切片，使每个用户都能满负荷使用 SM，提高 GPU 的 SM 利用率。

23. 请简述向量长度寄存器和向量屏蔽寄存器的作用。

答：1.向量长度寄存器的作用：将软件层程序中实际向量长度 N 与硬件层向量寄存器中的元素数目 64 相适配；

2.向量屏蔽寄存器的作用：当向量长度小于 64 时，或者条件语句控制下对向量某些元素进行单独运算时使用。即使掩码中有大量的 0，使用 VM 的向量指令速度依然远远快于标量计算模式；

24. 请简述指令编队的思想。

答：由一组不包含结构冒险的向量指令组成，一个编队中的所有向量指令在硬件条件允许时可以并行执行。

25. 请简述链接技术的思想。

答：1.当两条指令出现“写后读”相关时，若它们不存在功能部件冲突和向量寄存器冲突，就有可能把它们所用的功能部件头尾相接，形成一个链接流水线，进行流水线处理。

2.把流水线定向的思想引入到向量执行过程，对两条流水线进行联合控制

26. 请简述分段开采技术的思想。

答：当向量的长度大于向量寄存器的长度时，必须把向量分成长度固定的段，然后循环分段处理，每一次循环只处理一个向量段。

27. 请在 PVP、SMP、MPP、DSM 和 COW 中任选一种，简要描述其特点。（MIMD 并行机结构模型）

答：PVP（并行向量处理机），SMP（对称多处理机），MPP（大规模并行处理机），DSM（分布式共享存储多处理机），COW（工作站集群）

	并行向量处理机	对称多处理机	大规模并行处理机	分布式共享存储多处理机	工作站集群
属性	PVP	SMP	MPP	DSM	COW
结构类型	MIMD	MIMD	MIMD	MIMD	MIMD
处理器类型	专用定制	商用	商用	商用	商用
互连网络	定制交叉开关	总线、交叉开关	定制网络	定制网络	商用网络（以太 ATM）
通信机制	共享变量	共享变量	消息传递	共享变量	消息传递
地址空间	单地址空间	单地址空间	多地址空间	单地址空间	多地址空间
系统存储器	集中共享	集中共享	分布非共享	分布共享	分布非共享
访存模型	UMA	UMA	NORMA	NUMA	NORMA
代表机器	Cray C-90，Cray T-90，银河1号	IBM R50，SGI Power Challenge，曙光1号	Intel Paragon，IBMSP2，曙光 1000/2000	Stanford DASH，Cray T 3D	Berkeley NOW，Alpha Farm

28. 什么是多处理机 Cache 一致性问题?

答：1.允许共享数据进入 Cache，导致多个处理器的 Cache 中都有同一存储块的副本;

2.某个处理器对其 Cache 中的数据进行修改后，使得其 Cache 中的数据与其他的 Cache 中的数据不一致;

29. 请简述监听式协议的原理。

答：1.每个 Cache 除了包含物理存储器中块的数据拷贝之外，也保存着各个块的共享状态信息;

2.Cache 通常连在共享存储器的总线上，当某个 Cache 需要访问存储器时，它会把请求放到总线上广播出去，其他各个 Cache 控制器通过监听总线来判断它们是否有总线上请求的数据块。如果有，就进行相应的操作。

30. 请简述目录式协议的原理。

答：目录是一种数据结构，记录了 cache 块的一些状态：未缓冲、共享态、专有态。Cache 块的位向量，用于记录该 cache 块的共享情况，即位向量的每一位代表一个处理器，若其为 1，则表明在该位置的 CPU 有该 cache 块的副本。当处理器对某一块进行写操作时，需哟啊根据位向量，通知对应的 CPU 进行作废或更新操作;

31. 请比较说明写作废协议和写更新协议性能上的差别。

答：1.在对同一个数据进行多次写操作而中间无读操作的情况下，写更新需进行多次写广播操作，而写作废只需一次作废操作;

2.在对同一 Cache 块的多个字进行写操作的情况下，写更新对于每一个写操作都要进行一次广播，而写作废仅在对该块的第一次写时进行作废操作即可;

3.考虑从一个处理器 A 进行写操作后到另一个处理器 B 能读到该写入数据之间的延迟时间。

32. 请解释，在目录式协议中，什么是本地节点、宿主节点、远程节点和共享集合?

答：1.本地节点：发出访问请求的节点;

2.宿主节点：包含所访问的存储单元及其目录项的节点;

3.远程节点：可以和宿主节点是同一个节点，也可以不是同一个节点;

4.共享集合：位向量记录拥有其副本的处理器的集合。

33. 请简述目录式协议中，目录的三种结构。

答：1.全映像目录：每一个目录项都包含一个 N 位的位向量，其每一位对应于一个处理机;

2.有限映像目录：提高其可扩展性和减少目录所占用的空间；

3.链式目录：用一个目录指针链表来表示共享集合。当一个数据块的副本数增加（或减少）时，其指针链表就跟着变长（或变短）。