1. 源程序实际上就是一个由值\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_组成的\_\_\_\_\_\_\_\_序列。\_\_\_\_\_\_\_\_个\_\_\_\_\_\_\_\_被组织成一组，称为\_\_\_\_\_\_\_\_。每个字节表示程序中的某些\_\_\_\_\_\_\_\_。
2. ASCII 字符集使用一个唯一的\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_的整数值表示每个字符。
3. 只由\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_构成的文件称为文本文件，\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_称为二进制文件。
4. 如果 shell 命令行的第一个单词不是一个\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_，那么 shell 就会假设这是一个\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
5. Bus，即\_\_\_\_\_\_\_\_，是\_\_\_\_\_\_\_\_整个系统的一组\_\_\_\_\_\_\_\_，它负责在\_\_\_\_\_\_\_\_之间传递\_\_\_\_\_\_\_\_。总线传递\_\_\_\_\_\_\_\_的字节块，也就是\_\_\_\_\_\_\_\_。字中的\_\_\_\_\_\_\_\_，各个系统各不相同，要么是\_\_\_\_\_\_\_\_个字节，\_\_\_\_\_\_\_\_位，要么是\_\_\_\_\_\_\_\_个字节，\_\_\_\_\_\_\_\_位。
6. 每个 I/O 设备都通过一个\_\_\_\_\_\_\_\_或\_\_\_\_\_\_\_\_与\_\_\_\_\_\_\_\_相连。
7. 控制器与适配器之间的功能都是传递信息，区别主要在于它们的\_\_\_\_\_\_\_\_。控制器是\_\_\_\_\_\_\_\_或\_\_\_\_\_\_\_\_上的芯片组。而适配器则是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
8. 主存是一个\_\_\_\_\_\_\_\_存储设备。在处理器执行程序时，用来存放\_\_\_\_\_\_\_\_和\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。物理上，主存是由\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_组成的。逻辑上说，存储器是一个\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_，每个\_\_\_\_\_\_\_\_都有其\_\_\_\_\_\_\_\_，这些\_\_\_\_\_\_\_\_是从\_\_\_\_\_\_\_\_开始的。组成程序的每条机器指令都由\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_组成。
9. 中央处理器（CPU），简称处理器，是\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。处理器的核心是一个\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_的\_\_\_\_\_\_\_\_，称为\_\_\_\_\_\_\_\_，在任何时刻，它都指向\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
10. 0，1，位 / 8，位，字节 / 文本字符
11. 单字节大小的整数值
12. ASCII 字符，所有其他文件都
13. 内置的 shell 命令，可执行文件的名字
14. 总线，贯穿，电子管道，各个部件，信息字节，定长，字（word），字节数，4，32，8，64
15. 控制器，适配器，I/O 总线
16. 封装方式，设备本身，主板（主印制电路板），一块插在主板插槽上的卡
17. 临时，程序，程序处理的数据，一组动态随机存取存储器（DRAM）芯片，线性的字节数组，字节，唯一的地址（数组索引），地址，零，不同数量的字节
18. 解释（或执行）存储在主存中指令的引擎，大小为一个字，存储设备（或寄存器），程序计数器（PC），主存中的某条机器语言指令（即含有该条指令的地址）
19. 寄存器文件（register file）是一个小的\_\_\_\_\_\_\_\_，由一些\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_组成。
20. ALU 的中文名是\_\_\_\_\_\_\_\_。
21. 运行 hello world 程序：

* 初始时，shell 程序等待我们输入一个命令。当我们在键盘上输入字符串“./hello”后，shell 程序将字节逐一读入\_\_\_\_\_\_\_\_，再把它们存放到\_\_\_\_\_\_\_\_中。
* 当我们在键盘上敲回车键时，shell 程序就知道我们已经结束了命令输入，shell 执行一系列指令来加载可执行的 hello 文件，这些指令将 hello 目标文件中的代码和数据（字符串“hello, world\n”就是该程序的数据）从\_\_\_\_\_\_\_\_复制到\_\_\_\_\_\_\_\_。利用\_\_\_\_\_\_\_\_技术，数据可以不通过\_\_\_\_\_\_\_\_而直接从\_\_\_\_\_\_\_\_到达\_\_\_\_\_\_\_\_。
* 一旦目标文件 hello 中的代码和数据被加载到\_\_\_\_\_\_\_\_，\_\_\_\_\_\_\_\_就开始执行 hello 程序中的 main 程序中的机器语言指令。这些指令将字符串“hello, world\n”从\_\_\_\_\_\_\_\_复制到\_\_\_\_\_\_\_\_，再从\_\_\_\_\_\_\_\_复制到\_\_\_\_\_\_\_\_，最终显示在屏幕上。
* 可见，系统花费了大量的时间把信息从一个地方挪到另外一个地方。hello 程序的机器指令最初是存放在\_\_\_\_\_\_\_\_上，当程序加载时，它们被复制到\_\_\_\_\_\_\_\_；当处理器运行程序时，指令又从\_\_\_\_\_\_\_\_复制到\_\_\_\_\_\_\_\_。数据串“hello, world\n”开始时在\_\_\_\_\_\_\_\_上，然后被复制到\_\_\_\_\_\_\_\_，最后从\_\_\_\_\_\_\_\_复制到\_\_\_\_\_\_\_\_。从程序员的角度来说，这些复制就是\_\_\_\_\_\_\_\_，键盘了程序“真正的”工作。因此，系统设计者的一个主要目标就是使这些复制操作尽可能快地完成。

1. 根据机械原理，较大从存储设备比较小的存储设备运行得\_\_\_\_\_\_\_\_。
2. 从磁盘驱动器上读取一个字的时间开销要比从主存中读取的开销大\_\_\_\_\_\_\_\_倍。
3. 高速缓存至关重要。一个典型的寄存器文件只存储\_\_\_\_\_\_\_\_字节的信息，而主存里可以存放\_\_\_\_\_\_\_\_字节。然而，从寄存器文件中读数据比从主存中读数据几乎要快\_\_\_\_\_\_\_\_倍。针对这种处理器与主存之间的差异，系统设计者采用了\_\_\_\_\_\_\_\_的存储设备，简称\_\_\_\_\_\_\_\_，作为暂时的集结区域，存放\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
4. 存储设备，单个字长的寄存器
5. 算术/逻辑单元

* 寄存器，主存
* 磁盘，主存，直接存储器存取（DMA），处理器，磁盘，主存
* 主存，处理器，主存，寄存器文件，寄存器文件，显示设备
* 磁盘，主存，主存，处理器，磁盘，主存，主存，显示设备，开销

1. 慢
2. 1000 万
3. 几百，几十亿，100，更小更快，高速缓存存储器（cache memory），高速缓存，处理器近期可能会需要的信息
4. 位于处理器芯片上的 L1 高速缓存的容量可以达到\_\_\_\_\_\_\_\_，访问速度几乎和访问\_\_\_\_\_\_\_\_一样快。一个容量为\_\_\_\_\_\_\_\_到\_\_\_\_\_\_\_\_的更大的 L2 高速缓存通过一条特殊的总线连接到处理器，进程访问 L2 高速缓存的时间要比访问 L1 告诉缓存的时间长 \_\_\_\_\_\_\_\_ 倍，但这仍然比访问\_\_\_\_\_\_\_\_的时间快\_\_\_\_\_\_\_\_倍。L1 和 L2 高速缓存是用一种叫做\_\_\_\_\_\_\_\_的硬件技术实现的。比较新的处理能力强的系统甚至有三级高速缓存。系统可以获得一个很大的存储器，同时访问速度也很快，原因是利用了高速缓存的\_\_\_\_\_\_\_\_原理，即\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。通过让高速缓存里存放可能经常访问的数据，大部分的内存操作都能在快速的高速缓存中完成。本书得出的一个重要结论之一就是，\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_的程序员，能\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
5. 在 Eclipse 中，如果知道一个类属于哪个 jar 包？使用使用\_\_\_\_\_\_\_\_快捷键。
6. 数万字节，寄存器文件，数十万，数百万字节，5，5~10，静态随机访问存储器（SRAM），局部性，程序具有访问局部区域里的数据和代码的趋势，意识到高速缓存存储器存在，利用高速缓存将程序的性能提高一个数量级
7. Ctrl + Shift + T