1. 冯诺伊曼体系: 将程序指令和数据一起存储的计算机设计概念结构

早期计算机仅含固定用途的程序

改变程序得更改结构、重新设计电路

把程序存储, 设计通用电路

核心: 存储程序指令, 设计通用电路

为实现体系: 必须有存储器(存储程序)，控制器(控制程序的上下文切换等)，运算器(完成运算操作)， 输入设备，输出设备

扩展: 对应JVM虚拟机, 输入设备: ClassLoader .class文件 输出设备: 字节码 控制器: 程序计数器 运算器: 虚拟机栈 存储器: 堆

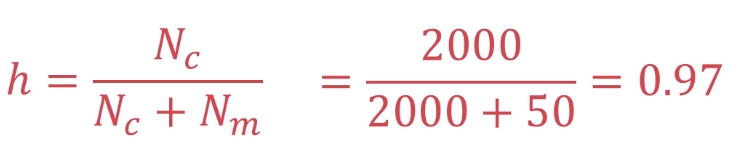
冯诺伊曼计算机体系的瓶颈: CPU速度与IO设备速度的不一致

1. 计算机存储器: 局部性原则, CPU访问存储器时, 无论是存取指令还是存取数据，所访问的存储单元都趋向于聚集在一个较小的连续区域中(实现缓存置换的基础).

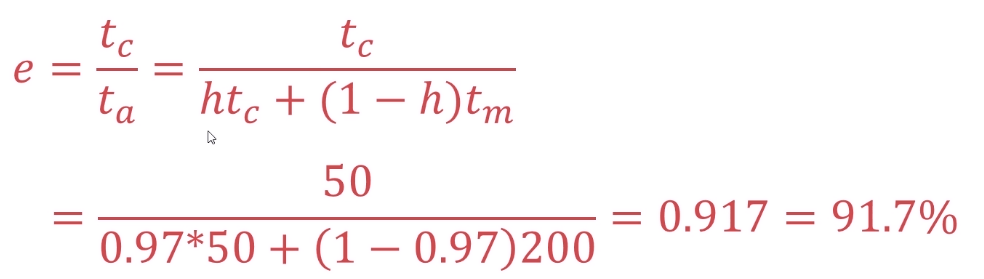
直接将内存中的一段置换至CPU缓存即可

例: 假设CPU在执行某段程序时, 共访问了Cache命中2000次，访问主存50次，已知Cache的存取时间为50ns,主存的存取时间为200ns, 求Cache主存系统的命中率、访问效率和平均访问时间。

命中率:



访问效率: -> Cache的存取时间/实际存取时间



平均访问时间:

 = 0.97 \* 50 + (1 – 0.97) \* 200 = 54.5ns

高速缓存替换算法:

随机算法:

先进先出算法:

最不经常使用算法:

最近最少使用算法: