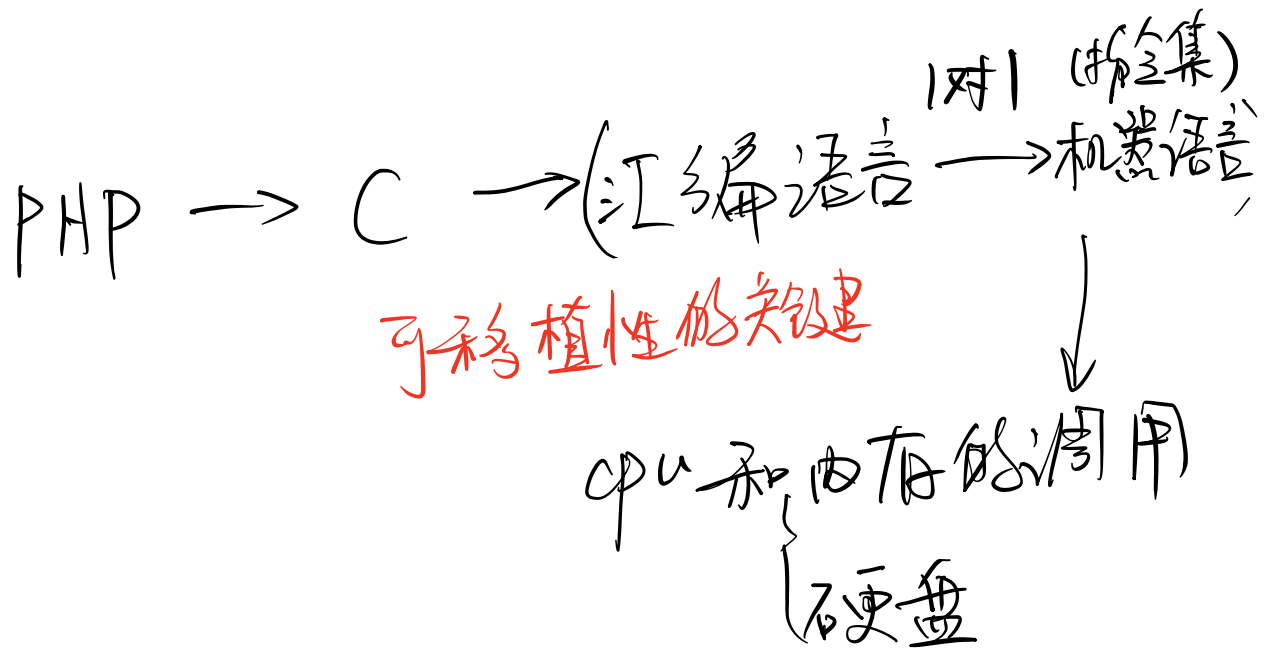


程序执行过程:



CPU指令集即 CPU 能执行的指令的集合,

指令是对 CPU 中器件的开、闭操作, 对

应 0、1 状态; 不同 CPU 结构, 要运行

同一个计算逻辑，所进行的升、闭也不同，也即指令集不同，所以同样功能的程序在不同 CPU 架构中对应的机器码  $\rightarrow$  汇编代码也不同，也不能直接移植程序。

但用 C 语言写的程序，既可以在 ARM 的 CPU 运行，又可以在 intel 的 CPU 运行，是因为用不同编译器转换，同样的 C 语言被翻译成了适合于不

同 opu 的机器码，也即编译器使得高级语言和汇编语言解耦了。

而汇编语言与机器语言<sup>(指令)</sup>一一对应，汇编语言的意义只是方便人读机器代码，相当于指定的助记符。

高级语言写的程序源代码有可移植性，但最终的可执行文件并不具有可移植性，即便在同样 opu

值 0x7c1c1c1c

之间, 因为操作系统不同。

可执行文件并不是机器码, 不是直接对cpu的一条条指令, 而是通过操作系统提供的库等进行链接来实现对应功能。不同操作系统可执行程序格式不同, 提供的基础库也不同, 所以即便cpu相同, 要么不识别格式, 要么调用不了要用好的

函数，也不行，所以，`isSecure`  
机密而可执行文件不机密，反编译也是不被允许的也很难的（如同散列值求原值，却不可能）

java 可执行程序的可移植性

在于虚拟机中间层对不同系统有不同对待，其可执行程序对虚拟机而言就是源代码的角色。