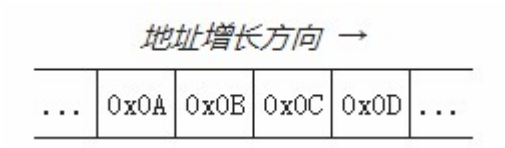
C基础：内存布局

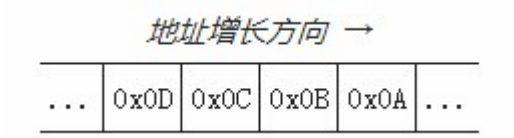
大端与小端

大端和小端是表示字节顺序的方式，又称端序或尾序(**Endianness**)

- 大端(**big-endian**)
  - 高字节存储在低地址上，低字节存储在高地址上
  - 以8bit为单位的数据`0x0A0B0C0D`为例，



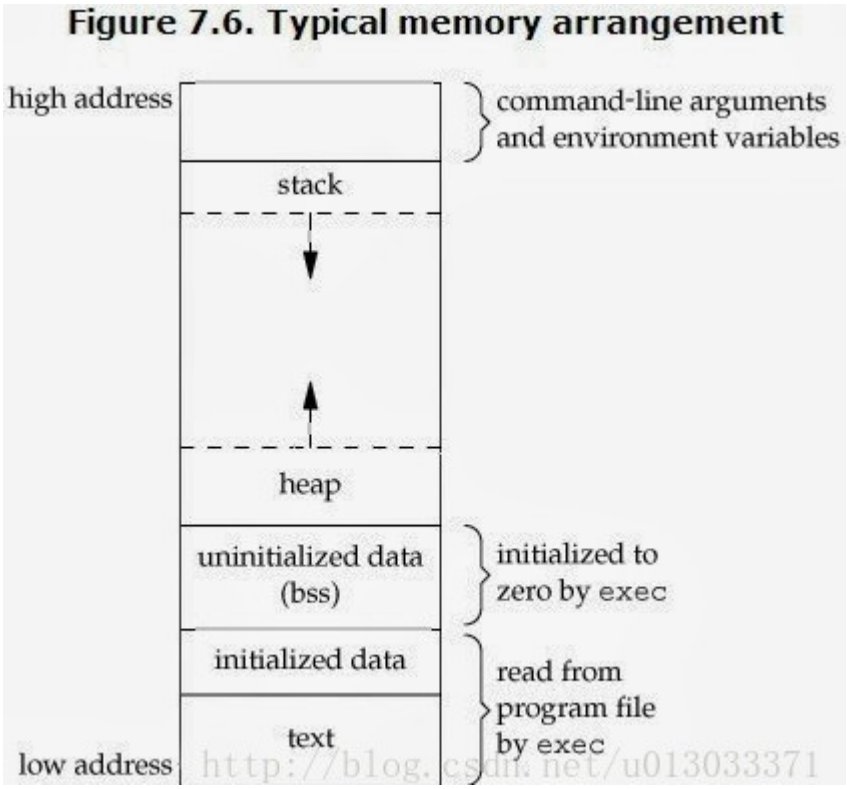
- 小端(**little-endian**)
  - 高字节存储在高地址上，低字节存储在低地址上
  - 以8bit为单位的数据`0x0A0B0C0D`为例，



现代PC机，以x86架构为基础的机器上，大多数都是采用了小端模式

C的内存模型和运行时内存布局

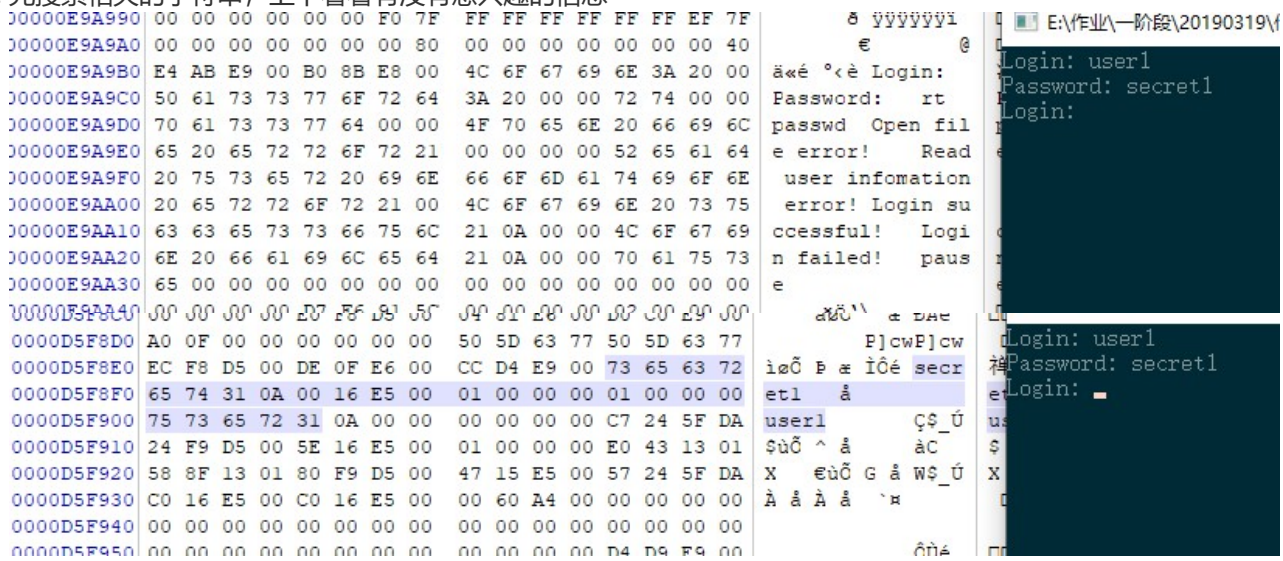
- BSS段(**bss segment**)
  - 通常是指用来存放程序中**未初始化**的全局变量的一块内存区域，属于静态内存分配
- 数据段(**data segment**)
  - 通常是指用来存放程序中**已初始化**的全局变量 的一块内存区域，属于静态内存分配
- 代码段(**code segment/text segment**)
  - 通常代码段和只读数据段合成为文本段(**text segment**)， 包含实际要执行的代码（机器指令）和常量
- 堆(**heap**)
  - 堆是用于存放进程运行中被动态分配的内存段，它的大小并不固定，可动态扩张或缩减
- 栈(**stack**)
  - 栈内存只在程序运行时出现，在函数内部使用的变量、函数的参数以及返回值将使用栈空间，栈空间由编译器自动分配和释放。其操作方式类似于数据结构中的栈



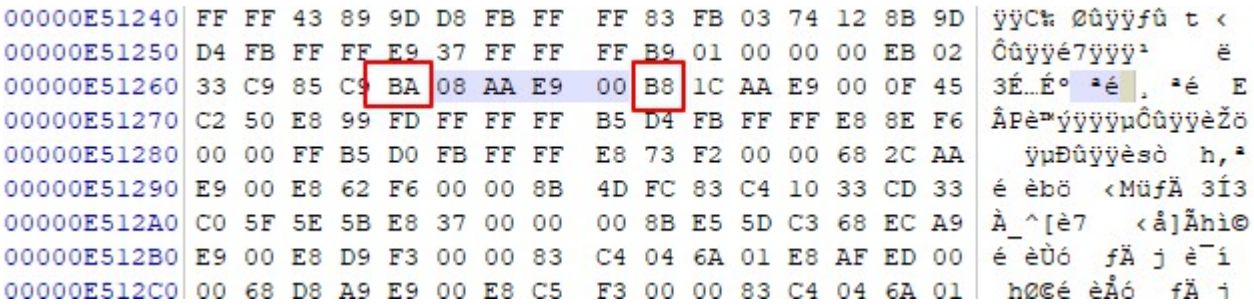
用WinHex破解作业

以我的作业为例

1. 先搜索相关的字符串，上下看看有没有感兴趣的信息



2. 没有直接看到登录账户，那么可以考虑下修改跳转等来达到目的。先看看哪个地方引用了登录成功的信息



3. 看到引用字符串地址前有**BA**和**B8**，这是`mov edx，xxx`和`mov eax，xxx`的格式，在往后看看。传送了两个地址后面有个**50**表示`push eax`的操作，那么考虑改下这个**push**操作，因为最后打印成功的结果肯定有函数调用，那么这个**push**操作很有可能就是这次的调用。我们改成`push edx`