操作系统第二次作业

20373068 周宇光

1. 假设有一个简单的计算机硬件系统，CPU 是同学们自己设计的，有内存和硬盘（可以没有MMU），请设计一个尽量简单的启动过程。（要求：列出必要的硬件支持、启动软件的基本功能和启动过程，只要将控制权交给操作系统镜像就算完成启动）

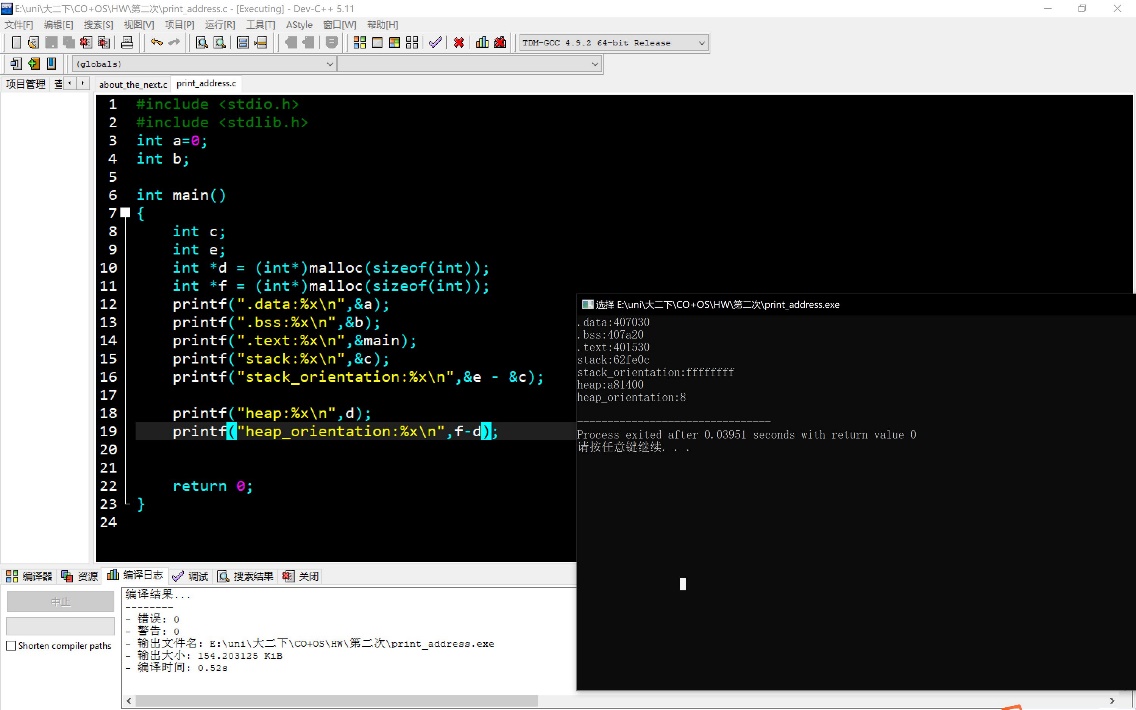
启动过程：首先，对系统加电。加电后，从硬盘的MBR中读取设备相关的启动信息、磁盘分区信息等。检查完MBR后，设计的Booter开始运行，由MBR信息初始化内存分布、各种CP0寄存器的值以及CPU参数。然后，由loader将操作系统内核的镜像映射到内存，并根据已有的内存分布设置好堆栈的位置。最后，由Loader跳转至操作系统内核的C程序入口。

硬件支持：包含CP0寄存器的MIPS流水线CPU、内存、有MBR的硬盘

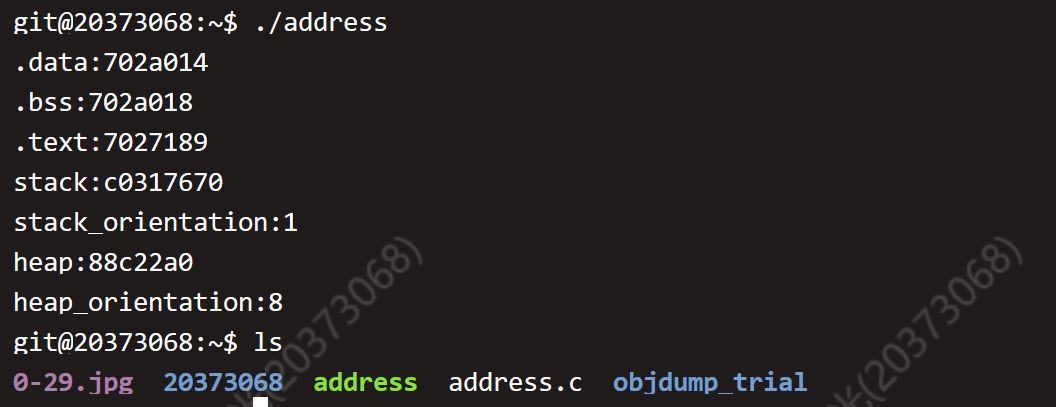
启动软件的基本功能：Booter负责stage1的各种初始化工作；Loader负责映像操作系统内核，跳转至stage2的操作系统

1. 编写一段程序，分别输出属于该程序代码段、数据段、堆和栈的地址。（提示：不要求一定要输出各个段的首地址，不要编一个ELF 解析程序，只要输出不同类型变量的地址就可以。）

首先，我在本地windows系统下，编写了一个输出地址的程序，其代码和输出如下：



可以看到，各个代码段的分布。为了对比，我在linux的跳板机上也进行了这份代码的编译运行。结果如下：



发现了一点我不大能理解的，就是为什么linux和windows的栈增长方向不一样？我怀疑可能是两个变量c和e编译时顺序的问题。

1. 动态内存分配需要对内存分区进行管理，一般使用位图和空闲链表两种方法。128MB的内存以n字节为单元分配，对于链表，假设内存中数据段和空闲区交替排列，长度均为64KB。并假设链表中的每个节点需要记录32位的内存地址信息、16位长度信息和16位下一节点域信息。这两种方法分别需要多少字节的存储空间？那种方法更好？

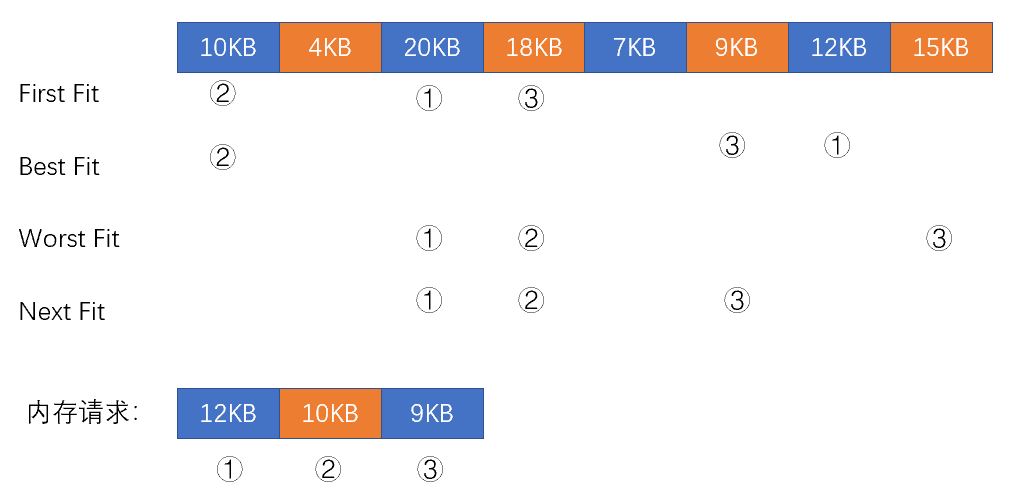
如果以位图管理，则需要一位来记录一个B的内存状况，对于128MB的内存空间，需要16MB的bit-map。

如果以链表的方式来管理，则对于每一段都需要建立一个链表项，总计有2K段，每个单元的大小为64位=8B，因此需要2KB的内存空间。

显然，链表管理法的效果更好，因为内存空间以64KB为单位，足够连续。如果位图方法一位记录更多信息，比如一位记录一个物理页（4KB）大小的使用情况，那么位图法的效率会高很多。

1. 在一个内存系统中，按内存地址排列的空闲区大小是: 10KB、4KB、20KB、18KB、7KB、9KB、12KB和15KB。对于连续的内存请求：12KB、10KB、9KB。使用FirstFit、BestFit、WorstFit和NextFit将找出哪些空闲区？

匹配情况如下图所示



1. 解释逻辑地址、物理地址、地址映射，并举例说明。

逻辑地址是CPU使用的地址，在32位MIPS处理器中，逻辑地址的范围是0x00000000-0x7fffffff；

物理地址是某个具体的物理设备中，存储单元的地址。

地址映射是指逻辑地址和物理地址之间的映射关系。比如在kseg0区，逻辑地址0x80000000-0x9fffffff被映射到了物理地址0x00000000-0x1fffffff中。