首先看一个小例子，我在查找这个问题的时候发现的：

<http://bbs.chinaunix.net/thread-1719587-1-1.html>

关于bss段，请教个问题，它是不是并不占用可执行程序的硬盘空间？  
hello.c文件内容如下

1. #include <stdio.h>
2. int main()
3. {
4. return 0;
5. }

复制代码

编译后  
[root@localhost ctest]# ll hello  
-rwxr-xr-x 1 root root **4613** 06-10 16:06 hello  
[root@localhost ctest]# size hello  
   text    data     bss     dec     hex filename  
    803     248       8    1059     423 hello  
  
可见这时hello的大小为 4613字节，用size去看hello，其bss段为8字节  
  
  
然后修改代码，填上一行 int bss[1000]; (该数组在运行时应占4000字节 ）

1. #include <stdio.h>
2. int bss[1000];//这行是增加的，它应位于bss段
3. int main()
4. {
5. return 0;
6. }

复制代码

[root@localhost ctest]# ll hello  
-rwxr-xr-x 1 root root 4633 06-10 16:07 hello  
[root@localhost ctest]# size hello  
   text    data     bss     dec     hex filename  
    803     248    4032    5083    13db hello  
  
发现hello的大小变为4633,比4613增加了20字节，而不是增加了4000字节，这应该就说明了bss段的数据并不占用可执行文件的空间吧？  
但奇怪的是 第二次对hello执行size命令时， 其bss段大小变为4032,而第一次时是8   
  
第一次  
[root@localhost ctest]# size hello  
   text    data     bss     dec     hex filename  
    803     248       8    1059     423 hello  
  
第二次   
  
[root@localhost ctest]# size hello  
   text    data     bss     dec     hex filename  
    803     248    4032    5083    13db hello  
  
实际上hello仅增加了20字节 ，是不是size命令得到的bss段的大小是指程序运行后，映射到虚拟内存空间后，bss段实际所占空间的大小？而不是说在硬盘上的大小？ 这样理解对么？