进入用户程序后，DS和ES仍然指向段header，SS仍然指向加载器的栈空间。

图片包含 图形用户界面

AI 生成的内容可能不正确。

根据用户程序入口点，进入到start，注意这里先初始化SS和SP，然后再初始化DS，注意ES仍然指向段header

电脑屏幕截图

AI 生成的内容可能不正确。

145行显示完第一段信息之后，将code2的地址和begin的地址分别作为CS和IP入栈，retf时直接进入到code2的begin。

文本

AI 生成的内容可能不正确。

进入到code2的begin后，用同样的方法继续跳转到code1的continue执行。之后DS切换到data2，bx初始化为msg1，调用put\_string，调用完成后使用jmp $一直循环。

put\_string，首先入栈相关寄存器，先从0x3d5数据端口号的0xe和0xf索引值读出光标位置的16位数。光标在屏幕上的位置保存在显卡内部的两个光标寄存器中，每个寄存器是8位，组合形成16位的数值。标准VGA文本模式是25行，每行80个字符。

put\_string有两个分支：1、如果不是回车符，进入put\_0a；2、如果是回车符，进入set\_cursor。

文本

AI 生成的内容可能不正确。

光标指示字符位置，一个字符在显存中对应两个字节。可以将光标位置乘2，得到字符在显存中的偏移地址。之后再将偏移地址回退光标位置，前进一个字符。

put\_0a有两个分支：1、如果不是换行符，进入put\_other，之后继续进入roll\_screen；2、如果是换行符，进入roll\_screen。

文本

AI 生成的内容可能不正确。

光标位置如果超过了25\*80-1，需要进行滚屏，设定源区域为显存内偏移地址为0xa0(十进制160，也就是屏幕第二行第一列)开始，用DS:SI，设定目标区域为显存内偏移地址为0，屏幕第一行第一列开始，用ES:DI，CX设定传送字数，字数为24\*80，执行rep movsw完成传送(cld将df清零正向传送)。同时清除屏幕最后一行，0x720表示黑底白字的空白字符。

roll\_screen有两个分支：1、如果光标未超出屏幕，进入set\_cursor；2、如果超出屏幕，进入cls后再进入set\_cursor。

set\_cursor通过将bx值写回端口以设置光标位置。

图示

AI 生成的内容可能不正确。