中南大學

CENTRAL SOUTH UNIVERSITY

计算机原理与汇编语言

课程设计

题目名称：吃豆子程序

姓名：苗子阳

学号：**8208200907**

专业：计算机科学与技术

班级：计算机科学与技术 **2005** 班

指导教师：贺建飚

编写日期：**2022.04.23**

目录

一、问题描述

1. 题目要求

在屏幕上显示多行“豆子”（用“.”表示），用一个“嘴巴”（用字符“C”表示），程序运行时，单击空格，“嘴巴”开始从左到右逐行还是“吃豆子”，一直到“豆子”被吃完停止或者单击空格暂停。

2. 题目分析

围绕要求，本程序应该完成以下几点**核心内容**：

- **显示**“嘴巴”与“豆子”
- **暂停与继续运行**功能
- **循环**吃“豆子”

而为了更好地进行**用户交互**，还辅助设计以下几点程序功能：

- **自定义**“豆子”数量
- **选择**开始或者结束
- 除了暂停之外，加入**直接结束**游戏功能
- 友好**美观**的 dos 界面

二、系统设计

1. 主要功能流程图

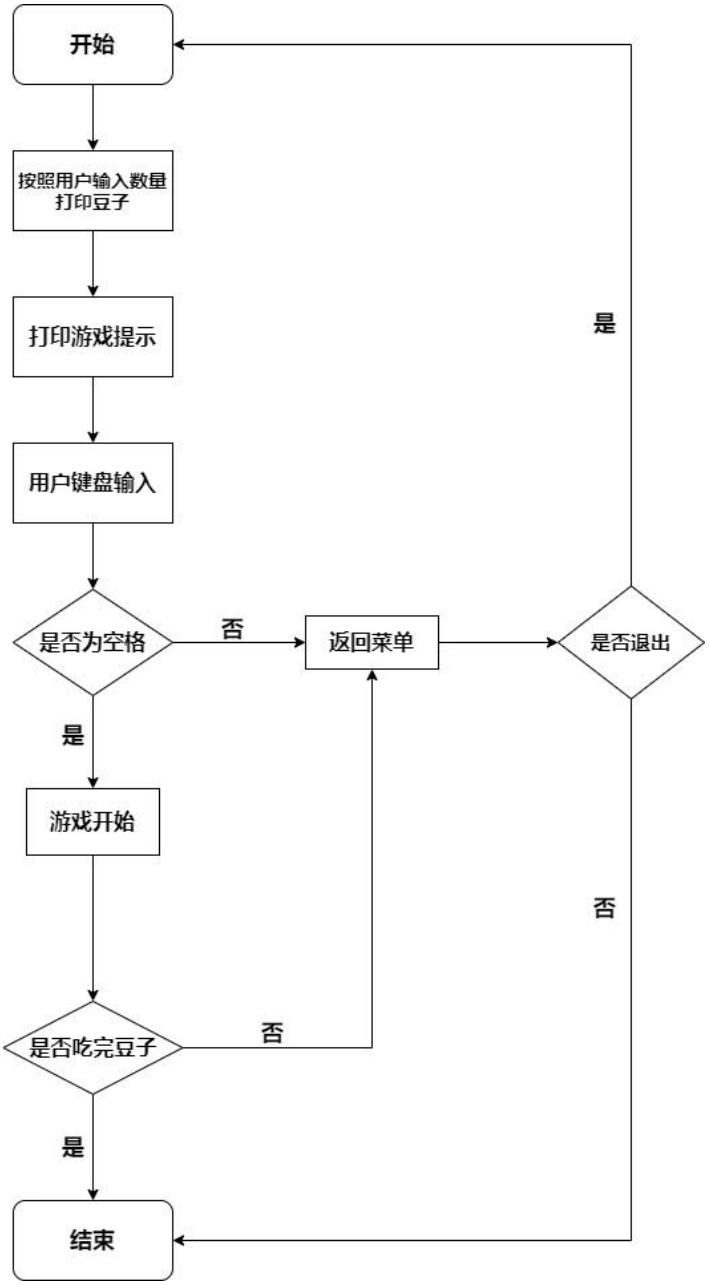


图 2-1 主要功能流程图

1.1 实现思路

本程序的主要功能模块，主要通过调用 BIOS 中断服务中 INT 10H 的界面功能

来实现各类页面跳转与按键设计。通过调用 BIOS 中断服务中 INT 16H 来监测用户键盘输入，进行游戏进程控制。具体模块的功能设计将继续进行阐述。

2. 游戏模块设计

2.1 游戏模块流程图

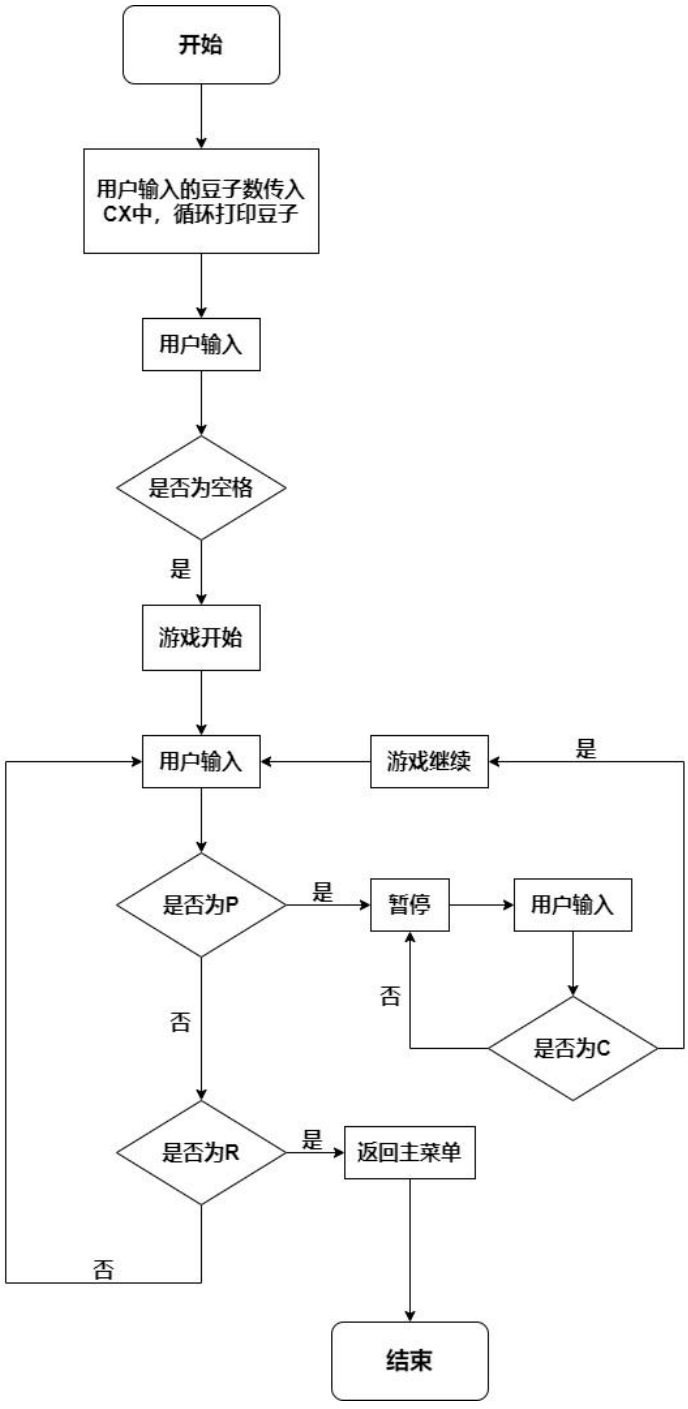


图 2-2 游戏模块流程图

2.2 实现思路

先对于用户输入的豆子数设置循环界限，在文本模式下（80*25）的游戏界面打印出以‘.’表示的豆子。并且在豆子下方打印提示信息：

“Press Space to Start or Press R to return.”

“Press P to pause. Press C to continue.”

用户键入空格，以‘C’代表的嘴巴开始自动吃豆子，代表游戏开始。在任意时刻，用户都可以通过键入‘r’来返回主菜单。用户可以通过键入‘p’来暂停游戏，通过键入‘c’来继续游戏。为了实现上述功能，需要调用 BIOS 终端服务 INT 16H 中对键盘缓冲区的特定功能。**值得注意的是，INT 16H 中 AH=00H 的功能不能使用，因为该功能是循环等待直到用户键入，会造成游戏等到用户键入一下才能动一下。要使用 AH=01H 的功能才能正确运行，该功能不需要循环等待，是动态处理的。**

3. 主菜单模块设计

3.1 主菜单模块流程图

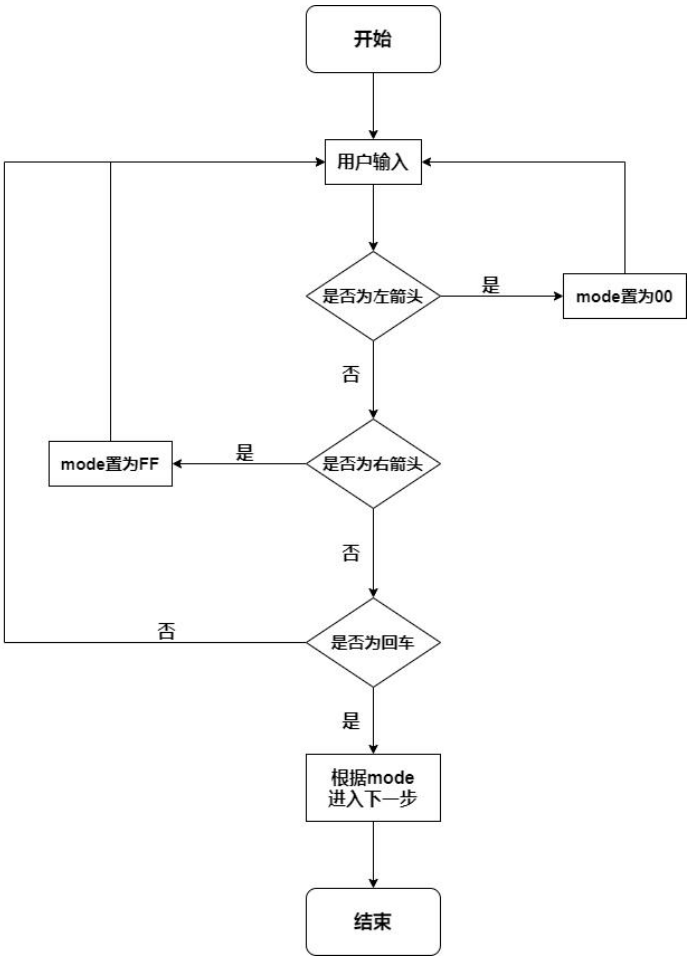


图 2-3 主菜单模块流程图

3.2 实现思路

为了实现主菜单中开始游戏与退出的功能，**建立变量 mode**，存储用户选择的功能。设定 mode 初始化为 00，代表选择 start，start 选项闪烁；用户点击按键切换模式，mode 取反变为 FF，代表选择 quit，quit 选项闪烁。最后以回车键判定用户做出最终选择。

4. 界面文字打字机效果

4.1 打字机效果流程图

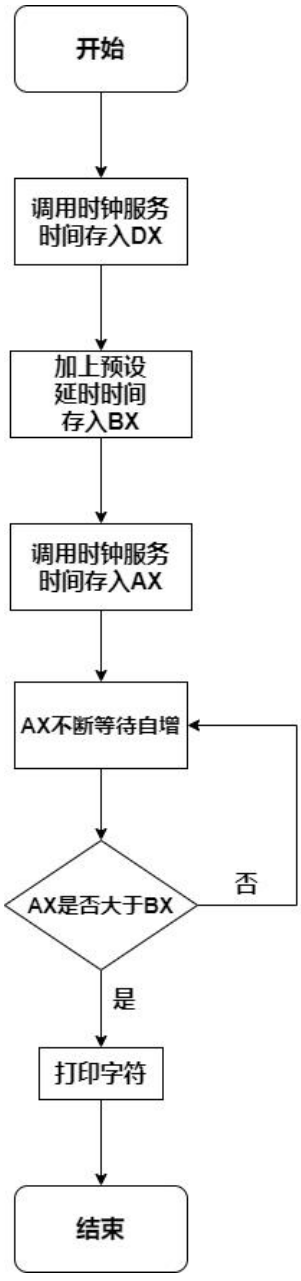


图 2-4 打字机效果流程图

三、源代码清单

1. 数据定义段

定义延时时间 delaytime（服务打字机效果）、模式选择 modeflag（服务菜单模式选择）、一些字符串的预定义（用于提示用户）、一个用于存放用户输入的 NUM 数组（服务打印用户输入的豆子数）

```
DATA SEGMENT

    DELAYTIME DB 01H

    MODEFLAG DB 0
    MODE1 DB 'START'
    MODE2 DB 'QUIT'
    TIPS3 DB 'MODE SELECT'
    STRING  DB  13,10,'Press Space to Start or Press R to
return',13,10,'$'
    STRING1  DB  13,10,'Press ENTER to continue',13,10,'$'
    INPUT_MSG DB 13,10,'PLEASE INPUT THE NUMBER OF PEAS:','$'
    PAUSE_MSG DB 13,10,'Press P to pause.  Press C to
continue.','$'
    NUM DW 40 DUP(?)

DATA ENDS
```

2. 处理用户输入数字子程序

将用户输入的数字串计算转换成对应数字，存入 NUM[0]中。

```
input proc near
    mov bx, 0
abc:
    mov ah, 1
    int 21h
    cmp al, 0dh
    jz exitt
    and ax, 000fh
    xchg ax, bx
    mov cx,10
    mul cx
```

```
    add bx, ax
    jmp abc
exitt:
    ret
input endp
```

3. 欢迎界面程序段

通过 INT 10H 中断建立界面，再通过打字机效果打印欢迎语句。同时提示用户输入游戏中的豆子数量，输入完成后，点击回车进行下一步菜单页面。

```
START:
    MOV AX,DATA
    MOV DS,AX
;重开一个界面
    MOV ES, AX
    MOV BX, 0
    MOV AH, 0FH
    INT 10H
    PUSH AX
    PUSH BX

    MOV AL, 03H
    MOV AH, 0
    INT 10H
    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'W'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'E'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'L'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
```

```
    MOV AL, 'C'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'O'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'M'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'E'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, ' '
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'T'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'O'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, ' '
    INT 10H
```

```
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'T'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'H'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'E'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, ' '
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'G'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'A'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'M'
    INT 10H
    CALL DELAY
```

```
    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'E'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, ' '
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'E'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'A'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'T'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, ' '
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'P'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
```

```
    MOV AL, 'E'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'A'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, '!'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, '-'
    INT 10H
    CALL DELAY
    MOV AH, 0EH
    MOV BH, 0
    MOV AL, '-'
    INT 10H
    CALL DELAY
    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'B'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'y'
    INT 10H
    CALL DELAY
    MOV AH, 0EH
    MOV BH, 0
    MOV AL, ' '
    INT 10H
    CALL DELAY

    MOV AH, 0EH
```

```
    MOV BH, 0
    MOV AL, 'm'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'i'
    INT 10H
    CALL DELAY
    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'a'
    INT 10H
    CALL DELAY
    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'o'
    INT 10H
    CALL DELAY
    LEA dx,offset INPUT_MSG
    MOV ah,09h
    int 21h
    call input
    mov NUM[0],bx
    MOV  AX,DATA
    MOV  DS,AX
    LEA  DX,STRING1
    MOV  AH,9
    INT  21H

    MOV ah,07H
    INT 21H
    cmp al,' '
    JNZ SELECT
```

4. 选择菜单界面主逻辑

同样，在新的页面中显示标题，调用**模式选择子程序**，根据子程序运行结果进行下一步的运行选择：若 mode 为 0，说明用户选择 start，开始游戏；否则退出。

```
SELECT:
    MOV AX, DATA
```

```
    MOV DS, AX
    MOV ES, AX
    MOV BX, 0
    MOV AH, 0FH
    INT 10H
    PUSH AX
    PUSH BX
;设置选择界面尺寸、模式
    MOV AL, 03H
    MOV AH, 0
    INT 10H
    MOV BP, OFFSET TIPS3;指向字符串
    MOV CX, 11D
    MOV DH, 0D
    MOV DL, 33;显示屏中央
    MOV AL, 01
    MOV BL, 0EH
    MOV AH, 13H
    INT 10H
    CALL MODESELECT
    CMP MODEFLAG, 0
    JE GOGAME1
    JMP EXIT
```

4.1 模式选择子程序

这里的模式选择，是用户在点击回车确认前的行为。即通过**判断用户是否点击了左右键**来移动选择，根据用户点击左右键的情况，在用户选择的选项根据 INT 10H 进行上色与换背景色，营造出一种“选中”的感觉。

值得一提的是，这里的逻辑比较简单，仅仅是**将 start 作为初始状态先上色**闪烁，一旦用户点击了左右键，就一定会切换到 exit，直接先将 mode 模式选择变量取反。是很简单的逻辑，**点击一次就切换一次 mode 的状态**，直到用户点击回车进入下一步。

```
MODESELECT PROC
    MODESELECTLOP1:
        CMP MODEFLAG, 0FFH;判断选择的模式
        JE MODESELECTLOP2
        MOV AH, 06H
        MOV AL, 0
        MOV BH, 07H
        MOV CH, 12D
        MOV CL, 0
        MOV DH, 13D
```

```
    MOV DL, 79
    INT 10H

    MOV AH, 13H
    MOV BL, 9EH
    MOV BH, 0
    MOV BP, OFFSET MODE1
    MOV AL, 1
    MOV CX, 5
    MOV DH, 3D
    MOV DL, 32D
    INT 10H

    MOV AH, 13H
    MOV BL, 08H
    MOV BH, 0
    MOV BP, OFFSET MODE2
    MOV AL, 1
    MOV CX, 4
    MOV DH, 3D
    MOV DL, 41D
    INT 10H
    JMP MODESELECTNEXT

MODESELECTLOP2:
    MOV AH, 06H
    MOV AL, 0
    MOV BH, 07H
    MOV CH, 18D
    MOV CL, 0
    MOV DH, 19D
    MOV DL, 79
    INT 10H

    MOV AH, 13H
    MOV BL, 08H
    MOV BH, 0
    MOV BP, OFFSET MODE1
    MOV AL, 1
    MOV CX, 5
    MOV DH, 3D
    MOV DL, 32D
    INT 10H
```

```
        MOV AH, 13H
        MOV BL, 9EH
        MOV BH, 0
        MOV BP, OFFSET MODE2
        MOV AL, 1
        MOV CX, 4
        MOV DH, 3D
        MOV DL, 41D
        INT 10H

    MODESELECTNEXT:
    ;读取键盘缓冲区
        MOV AH, 00
        INT 16H
        CMP AH,4BH;左键
        JE MODESELECTRESET
        CMP AH,4DH;右键
        JE MODESELECTRESET
        CMP AH,1CH
        JNE MODESELECTNEXT
    RET;回车返回
    MODESELECTRESET:
        NOT MODEFLAG;置位
        JMP MODESELECTLOP1
MODESELECT ENDP
```

5. 游戏运行主程序

新建页面，为打印豆子与用户提示信息做准备。该部分程序段是在豆子开始被吃之前，对游戏页面进行初始化。

```
GAME:
    MOV ah,02h
    MOV bh,0
    MOV dl,0
    MOV dh,10
    INT 10h
    MOV AH, 0FH
    INT 10H
;   PUSH AX
;   PUSH BX
;设置选择界面尺寸、模式
    MOV AL, 03H
    MOV AH, 0
```

```
    INT 10H
    MOV cx,NUM[0]
setpoint:;循环打印豆子

    MOV ah,02H
    MOV dl,'.'
    INT 21H
    LOOP setpoint
    MOV  AX,DATA
    MOV  DS,AX
    LEA  DX,STRING
    MOV  AH,9
    INT  21H
    MOV  AX,DATA
    MOV  DS,AX
    LEA  DX,PAUSE_MSG
    MOV  AH,9
    INT  21H
    MOV ah,07H
    INT 21H
    cmp al,' '
    JNZ SELECT

    MOV ah,02H
    MOV bh,0
    XOR dx,dx
    INT 10H
    MOV ax,0B800H ;显示取缓存起始位置，打印到屏幕需要从这里开始。
    MOV ds,ax
    XOR bx,bx
```

5.1 游戏运行控制程序段

将 NUM[0]中存放的用户输入数递给 CX，**利用指针指向当前位置**，将**当前位置赋为 C**，即吃豆人；将**走过的位置赋为‘ ’**，即空格，代表已经吃掉了豆子。中间通过对键盘键入的判断，来选择进行暂停还是返回菜单还是不予处理继续游戏。

值得一提的是，这里将**指针的下一位是否为豆子**作为游戏是否结束的标准。

```
re:
    MOV cx,NUM[0]
eat_start:
    MOV si,0FFFFH
    MOV di,004FFH
```

```
    ;代表在 25×80 的文本显示方式下，屏幕可有 2000 个字符位置，（si 为源
变址寄存器，si 与 di 有自动增量和自动减量功能，用于变址很方便）
    nextone:
    sub si,1
    JNZ nextone

    MOV byte ptr [bx],' ';走过的位置置为空格
    MOV byte ptr [bx+2],'C';当前位置置为 C
    xor ax,ax
    mov ah,01H
    int 16h
    cmp al,'p'
    JE PAUSE
    cmp al,'r'
    JE SELECT
cont:
    MOV AH, 0ch ; 清除键盘缓冲区
    INT 21h
    CMP byte ptr [bx+4],'.'
    JNZ exit
    ADD bx,2
    LOOP eat_start
```

5.2 暂停程序段

用户输入 p 之后，进入暂停模式。不断循环等待用户直到输入 c，才继续进行游戏。当然，用户依然可以输入 r 来返回主菜单。

```
PAUSE:
    MOV AH, 0
    INT 16H
    CMP Al,'c';右键
    JE cont
    cmp al,'r'
    JE SELECT
    JMP PAUSE
```

6. 退出界面程序段

与欢迎界面相同，调用打字机效果，与用户道别。

```
exit:
    MOV AX, DATA
```

```
    MOV DS, AX
    MOV ES, AX
    MOV BX, 0
    MOV AH, 0FH
    INT 10H

    PUSH AX
    PUSH BX
    MOV AL, 03H
    MOV AH, 0
    INT 10H


    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'G'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, '0'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, '0'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'D'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, ' '
    INT 10H
    CALL DELAY

    MOV AH, 0EH
```

```
    MOV BH, 0
    MOV AL, 'B'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'Y'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'E'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, '!'
    INT 10H
    CALL DELAY

    MOV ax,4C00H
    INT 21H
```

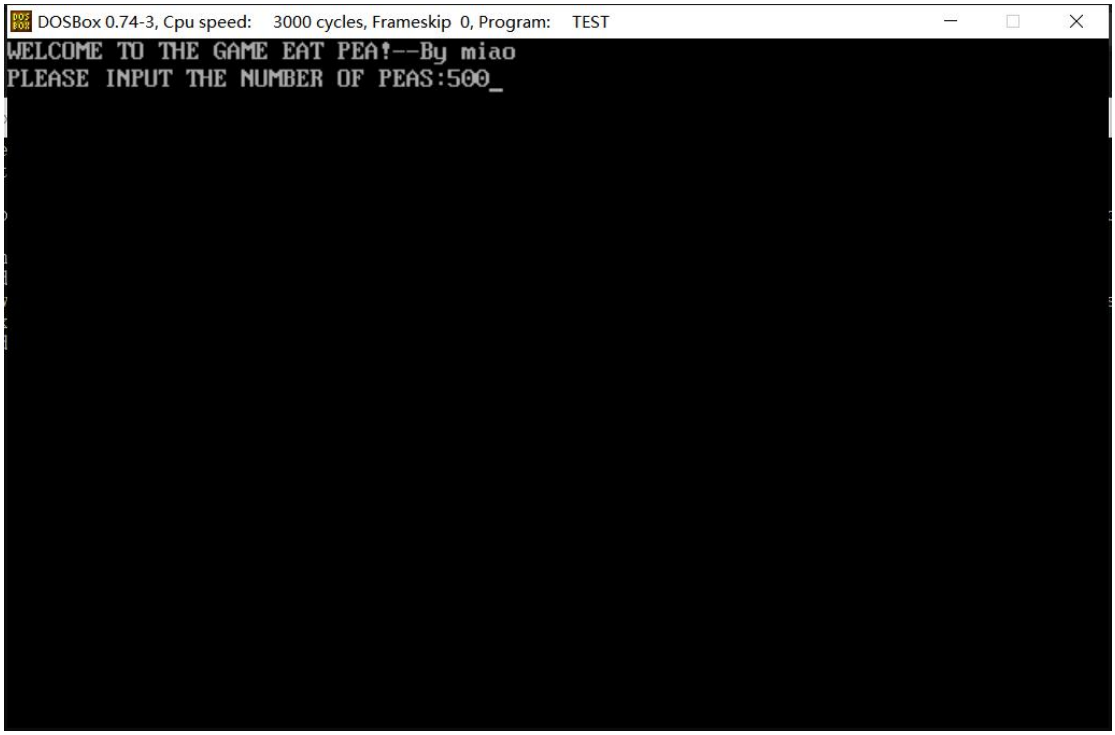
7. 打字机效果子程序

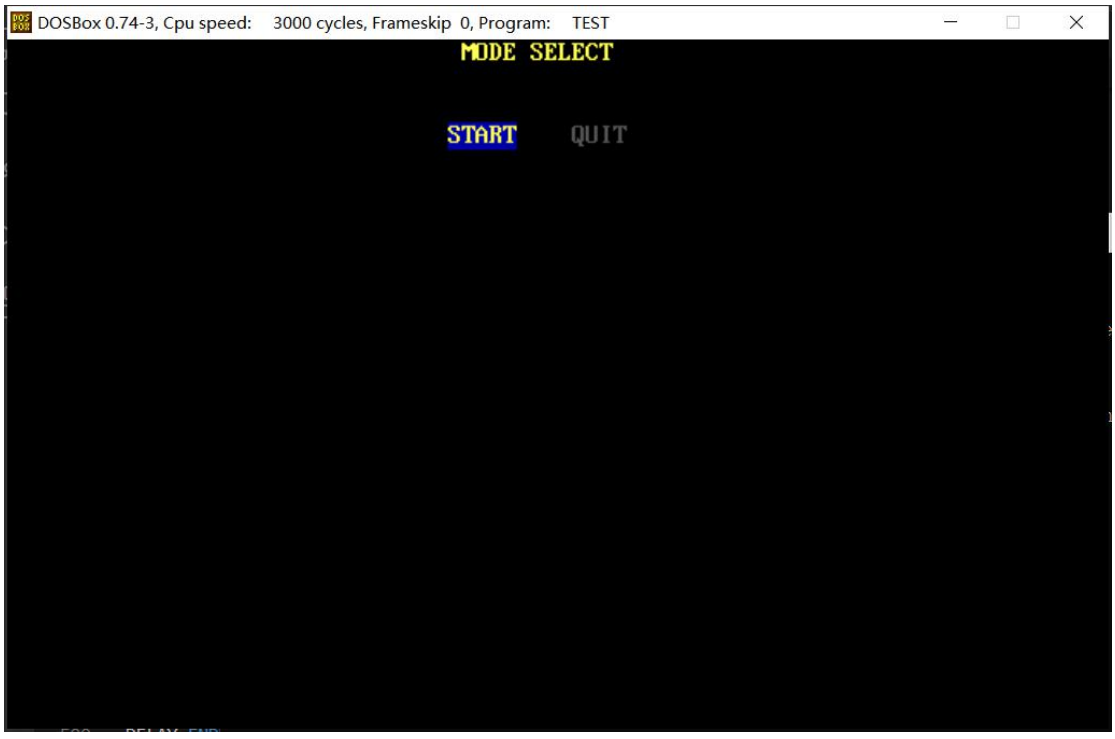
先通过一次时钟服务记录当前时间，加上预设的延时长度 delaytime，得到我希望延时到某一时刻。接着通过一次循环，**等待当前时间到达延时时刻，再打印字符**，形成打字机效果。

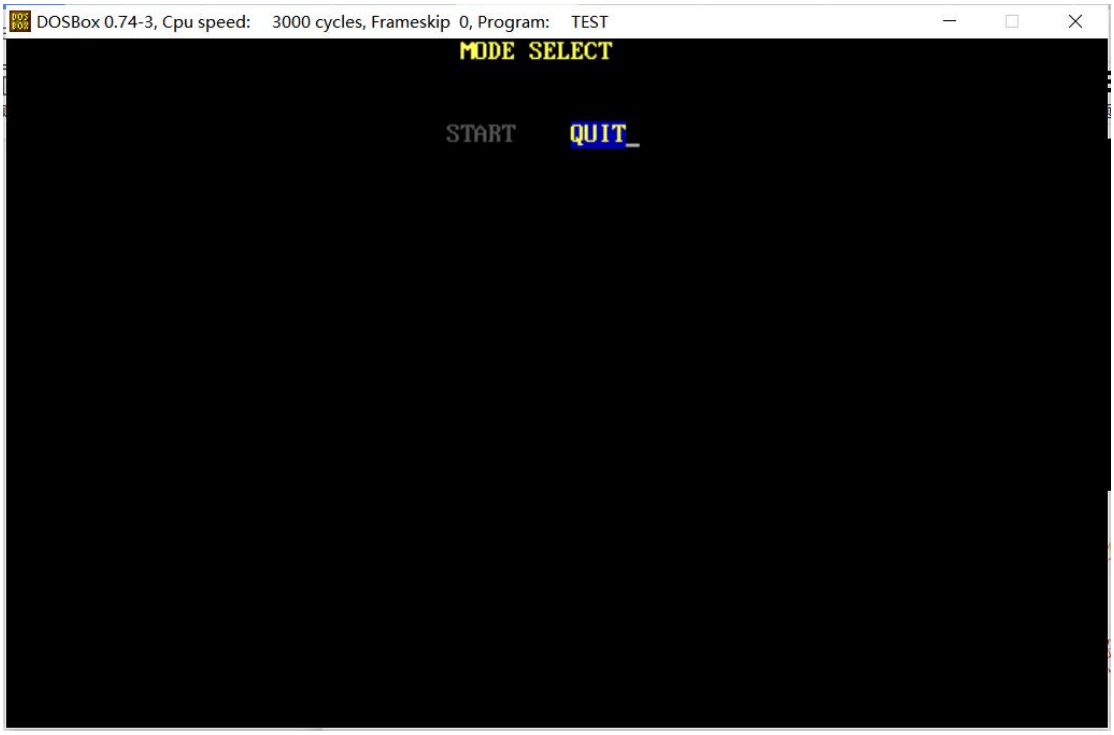
```
DELAY PROC
    MOV AH, 0
    INT 1AH  ;时钟服务
    MOV BX, DX
    MOV AX, 0
    MOV AL, DELAYTIME  ;延时长度
    ADD BX, AX
DELAYLOP:
    MOV AH, 0
    INT 1AH
    CMP DX, BX;循环延时长度直到与 BX 相同
    JE DELAYNEXT
```

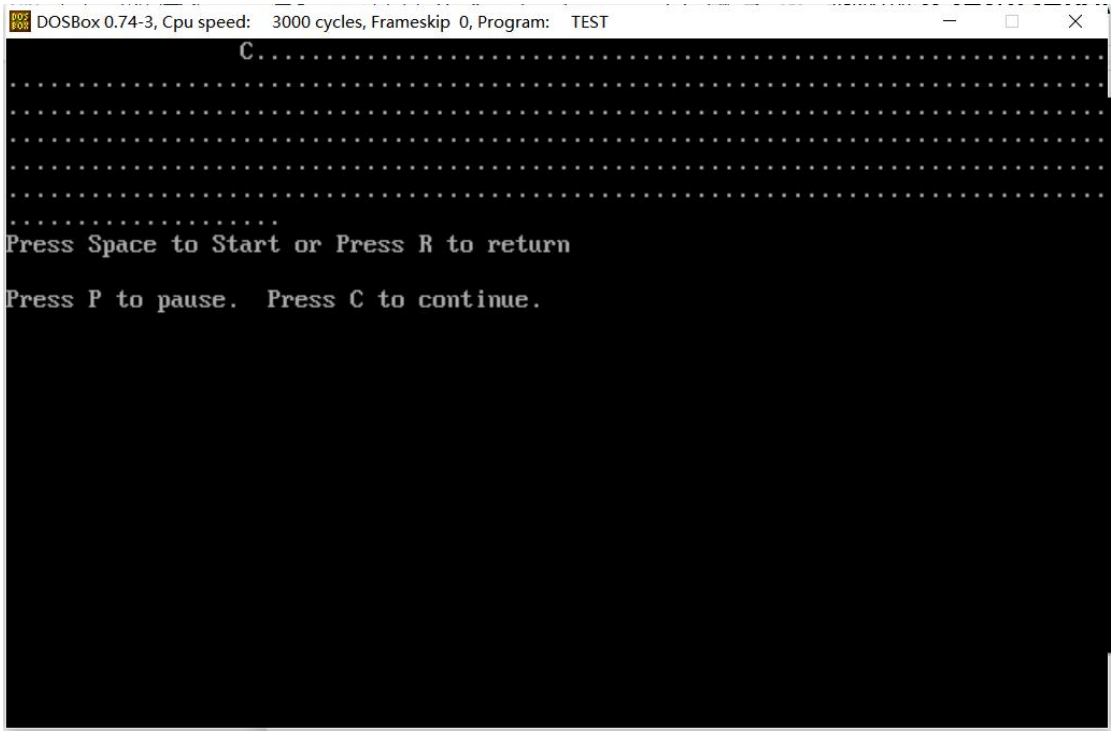
```
    JMP DELAYLOP
DELAYNEXT:
    RET
DELAY ENDP
```

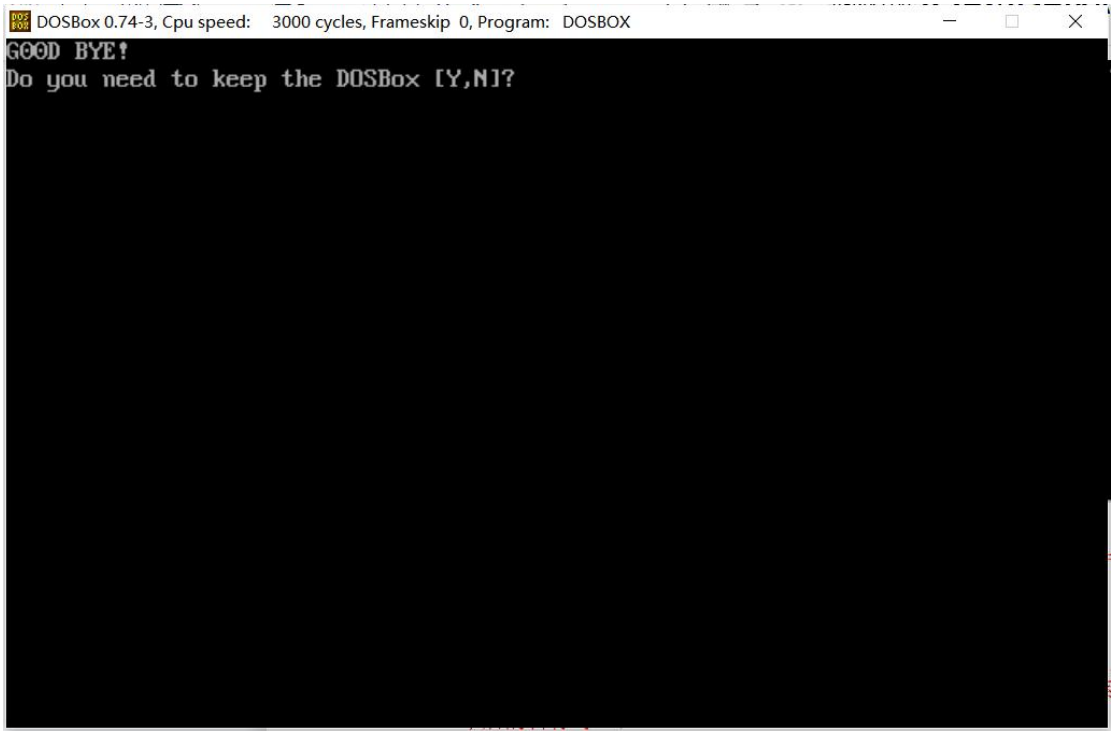
四、运行结果测试与分析


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TEST
WELCOME TO THE GAME EAT PEA!--By miao
PLEASE INPUT THE NUMBER OF PEAS:500_


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TEST
MODE SELECT
START
QUIT


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TEST
MODE SELECT
START QUIT


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: TEST
C
Press Space to Start or Press R to return
Press P to pause. Press C to continue.



分析：总体来说，程序运行结果符合预期。不过仍有一些情况需要注意：用户输入数字的时候，**没有加入输入非数字字符的判断**。倘若出现这种情况，虽然**不影响后续游戏**，因为豆子数量不会超过 80*25 个，但仍然是**考虑欠佳**的部分。

五、结论与心得

本次汇编程序设计的过程当中，我遇到了许许多多的问题。比如利用 INT 16H 中断 AH=00 的功能时，**会发生按一下键盘才动一下的现象**，仔细查阅 BIOS 中断服务表发现，是由于**循环等待**而发生这样的 bug。修改之后，就可正常运行了。在处理完这个问题之后，又发现新的问题。假如用户输入 p 进行暂停前，输入了其它的字符，**导致缓冲区不为空，中断服务不被触发怎么办？** 查阅资料后，我发现 INT 21H 的中断功能 AH=00 的时候，会进行**缓冲区的清除**，写入这行代码后，果然错误迎刃而解。同时，也出现了比如用户输入了数字，但是总是**没办法生成对应数量的豆子**的问题。在一番辛苦地调试之中，终于发现是 **CX 的内容在过程中被更新**，经过处理修改后便可个性化设置豆子数目。

这样的问题还有很多。在调试的过程中，我慢慢学会了如何去**查询中断表**，**找寻符合自己需求的服务功能**。当吃豆人按照我的控制进行吃豆子的时候，我获得了慢慢的成就感。汇编语言与高级语言不同，需要对问题理解**更透彻、对每个细节一清二楚**才行。这也锻炼了我编写代码的严谨性思维，受益匪浅。

附录：

完整 asm 文件代码：

```
DATA SEGMENT

    DELAYTIME DB 01H

    MODEFLAG DB 0
    MODE1 DB 'START'
    MODE2 DB 'QUIT'
    TIPS3 DB 'MODE SELECT'
    STRING  DB  13,10,'Press Space to Start or Press R to return',13,10,'$'
    STRING1  DB  13,10,'Press ENTER to continue',13,10,'$'
    INPUT_MSG DB 13,10,'PLEASE INPUT THE NUMBER OF PEAS:','$'
    PAUSE_MSG DB 13,10,'Press P to pause.  Press C to continue.','$'
    NUM DW 40 DUP(?)

DATA ENDS
CODES SEGMENT
    ASSUME CS:CODES,DS:DATA
input proc near
    mov bx, 0
abc:
    mov ah, 1
    int 21h
    cmp al, 0dh
    jz exitt
    and ax, 000fh
    xchg ax, bx
    mov cx,10
    mul cx
    add bx, ax
    jmp abc
exitt:
    ret
input endp
START:
```

```
    MOV AX,DATA
    MOV DS,AX
;重开一个界面
    MOV ES, AX
    MOV BX, 0
    MOV AH, 0FH
    INT 10H
    PUSH AX
    PUSH BX
    MOV AL, 03H
    MOV AH, 0
    INT 10H
    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'W'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'E'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'L'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'C'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'O'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
```

```
MOV AL, 'M'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'E'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, ' '
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'T'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'O'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, ' '
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'T'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'H'
INT 10H
```

```
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'E'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, ' '
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'G'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'A'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'M'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'E'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, ' '
INT 10H
CALL DELAY
```

```
MOV AH, 0EH
MOV BH, 0
MOV AL, 'E'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'A'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'T'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, ' '
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'P'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'E'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'A'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
```

```
MOV AL, '!'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, '-'
INT 10H
CALL DELAY
MOV AH, 0EH
MOV BH, 0
MOV AL, '-'
INT 10H
CALL DELAY
MOV AH, 0EH
MOV BH, 0
MOV AL, 'B'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'y'
INT 10H
CALL DELAY
MOV AH, 0EH
MOV BH, 0
MOV AL, ' '
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'm'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'i'
INT 10H
CALL DELAY
MOV AH, 0EH
MOV BH, 0
```

```
    MOV AL, 'a'
    INT 10H
    CALL DELAY
    MOV AH, 0EH
    MOV BH, 0
    MOV AL, 'o'
    INT 10H
    CALL DELAY
    LEA dx,offset INPUT_MSG
    MOV ah,09h
    int 21h
    call input
    mov NUM[0],bx
    MOV  AX,DATA
    MOV  DS,AX
    LEA  DX,STRING1
    MOV  AH,9
    INT  21H

    MOV ah,07H
    INT 21H
    cmp al,' '
    JNZ SELECT
SELECT:
    MOV AX, DATA
    MOV DS, AX
    MOV ES, AX
    MOV BX, 0
    MOV AH, 0FH
    INT 10H
    PUSH AX
    PUSH BX
;设置选择界面尺寸、模式
    MOV AL, 03H
    MOV AH, 0
    INT 10H
    MOV BP, OFFSET TIPS3;指向字符串
    MOV CX, 11D
    MOV DH, 0D
    MOV DL, 33;显示屏中央
    MOV AL, 01
    MOV BL, 0EH
    MOV AH, 13H
    INT 10H
```

```
    CALL MODESELECT
    CMP MODEFLAG, 0
    JE GAME
    JMP EXIT



GAME:
    MOV ah,02h
    MOV bh,0
    MOV dl,0
    MOV dh,10
    INT 10h
    MOV AH, 0FH
    INT 10H
;   PUSH AX
;   PUSH BX
;设置选择界面尺寸、模式
    MOV AL, 03H
    MOV AH, 0
    INT 10H
    MOV cx,NUM[0]
setpoint:;循环打印豆子
    MOV ah,02H
    MOV dl,'.'
    INT 21H
    LOOP setpoint
    MOV  AX,DATA
    MOV  DS,AX
    LEA  DX,STRING
    MOV  AH,9
    INT  21H
    MOV  AX,DATA
    MOV  DS,AX
    LEA  DX,PAUSE_MSG
    MOV  AH,9
    INT  21H
    MOV ah,07H
    INT 21H
    cmp al,' '
    JNZ SELECT

    MOV ah,02H
    MOV bh,0
    XOR dx,dx
```

```
    INT 10H
    MOV ax,0B800H ;显示取缓存起始位置，打印到屏幕需要从这里开始。
    MOV ds,ax
    XOR bx,bx
re:
    MOV cx,NUM[0]
eat_start:
    MOV si,0FFFFH
    MOV di,004FFH
    ;代表在 25×80 的文本显示方式下，屏幕可有 2000 个字符位置，（si 为源变址寄存器，si 与 di 有自动增量和自动减量功能，用于变址很方便）
    nextone:
    sub si,1
    JNZ nextone
    MOV byte ptr [bx],' ';走过的位置置为空格
    MOV byte ptr [bx+2],'C';当前位置置为 C
    xor ax,ax
    mov ah,01H
    int 16h
    cmp al,'p'
    JE PAUSE
    cmp al,'r'
    JE SELECT
cont:
    MOV AH, 0ch ; 清除键盘缓冲区
    INT 21h
    CMP byte ptr [bx+4],'.'
    JNZ exit
    ADD bx,2
    LOOP eat_start
exit:
    MOV AX, DATA
    MOV DS, AX
    MOV ES, AX
    MOV BX, 0
    MOV AH, 0FH
    INT 10H

    PUSH AX
    PUSH BX
    MOV AL, 03H
    MOV AH, 0
    INT 10H
```

```
MOV AH, 0EH
MOV BH, 0
MOV AL, 'G'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, '0'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, '0'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'D'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, ' '
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'B'
INT 10H
CALL DELAY

MOV AH, 0EH
MOV BH, 0
MOV AL, 'Y'
INT 10H
CALL DELAY

MOV AH, 0EH
```

```
    MOV BH, 0
    MOV AL, 'E'
    INT 10H
    CALL DELAY

    MOV AH, 0EH
    MOV BH, 0
    MOV AL, '!'
    INT 10H
    CALL DELAY
    MOV ax,4C00H
    INT 21H

PAUSE:
    MOV AH, 0
    INT 16H
    CMP Al,'c';右键
    JE cont
    cmp al,'r'
    JE SELECT
    JMP PAUSE

DELAY PROC
    MOV AH, 0
    INT 1AH  ;时钟服务
    MOV BX, DX
    MOV AX, 0
    MOV AL, DELAYTIME  ;延时长度
    ADD BX, AX
DELAYLOP:
    MOV AH, 0
    INT 1AH
    CMP DX, BX;循环延时长度直到与 BX 相同
    JE DELAYNEXT
    JMP DELAYLOP
DELAYNEXT:
    RET
DELAY ENDP

MODESELECT PROC
    MODESELECTLOP1:
        CMP MODEFLAG, 0FFH;判断选择的模式
        JE MODESELECTLOP2
        MOV AH, 06H
```

```
    MOV AL, 0
    MOV BH, 07H
    MOV CH, 12D
    MOV CL, 0
    MOV DH, 13D
    MOV DL, 79
    INT 10H

    MOV AH, 13H
    MOV BL, 9EH
    MOV BH, 0
    MOV BP, OFFSET MODE1
    MOV AL, 1
    MOV CX, 5
    MOV DH, 3D
    MOV DL, 32D
    INT 10H

    MOV AH, 13H
    MOV BL, 08H
    MOV BH, 0
    MOV BP, OFFSET MODE2
    MOV AL, 1
    MOV CX, 4
    MOV DH, 3D
    MOV DL, 41D
    INT 10H
    JMP MODESELECTNEXT

MODESELECTLOP2:
    MOV AH, 06H
    MOV AL, 0
    MOV BH, 07H
    MOV CH, 18D
    MOV CL, 0
    MOV DH, 19D
    MOV DL, 79
    INT 10H

    MOV AH, 13H
    MOV BL, 08H
    MOV BH, 0
    MOV BP, OFFSET MODE1
    MOV AL, 1
```

```
        MOV CX, 5
        MOV DH, 3D
        MOV DL, 32D
        INT 10H

        MOV AH, 13H
        MOV BL, 9EH
        MOV BH, 0
        MOV BP, OFFSET MODE2
        MOV AL, 1
        MOV CX, 4
        MOV DH, 3D
        MOV DL, 41D
        INT 10H
    MODESELECTNEXT:
    ;读取键盘缓冲区
        MOV AH, 00
        INT 16H
        CMP AH,4BH;左键
        JE MODESELECTRESET
        CMP AH,4DH;右键
        JE MODESELECTRESET
        CMP AH,1CH
        JNE MODESELECTNEXT
    RET;回车返回
    MODESELECTRESET:
        NOT MODEFLAG;置位
        JMP MODESELECTLOP1
MODESELECT ENDP
    JMP exit
CODES ENDS
    END START
```