

微机原理实验三

PB13206106

罗勇冠

一.实验目的

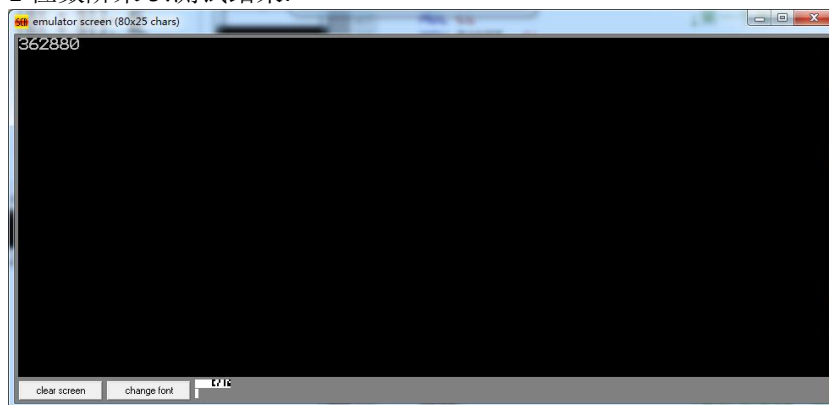
通过本次实验熟练掌握汇编语言的字符读取和字符输出操作,并学会用汇编语言计算阶乘并保存大整数数字。

二.实验内容

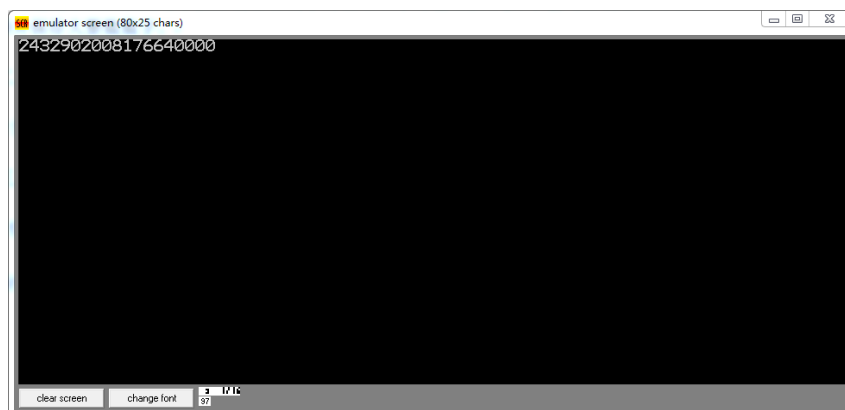
计算 $N!$ ($0 < N < 20$)。

三.实验结果

1 位数阶乘 9!测试结果:



两位数阶乘 20!测试结果:



四.实验分析

实验整体思路是,利用一个大数组存储阶乘结果的各位数,通过 `CL` 从 1 递增到阶乘数,遍历乘以大数组存储的数,并从低位开始,满十进位,直到最后计算出结果,再从高位向低

位逐位输出结果。

首先在数据段定义中，定义数据如下，

```
BUB DB 20 DUP (0)           ;大小为 20 的数组，存放长整型结果
RANGE DB 0                   ;阶乘数
PLUS DW 0                    ;存放进位
```

然后读取阶乘数 **N**，若 **N** 为一位数，将 ASCII 码转化为数字存放入 **RANGE** 变量中，再按控制字符结束输入，若 **N** 为两位数，将第一位输入数字乘 10 加上第二位输入数字得到该两位数，存入 **RANGE** 变量中，再按任何字符结束输入。读取字符代码如下：

```
MOV AH, 1                    ;读取字符
INT 21H
SUB AL, 30H                  ;转换 ASCII 为数字
MOV BL, AL
MOV RANGE, BL                ;将数字放入 RANGE 中
MOV AH, 1                    ;继续读字符
INT 21H
CMP AL, 30H                  ;第二次输入字符和 0 的 ASCII 码比较
JB RMUL                      ;若为其他控制字符，跳转
SUB AL, 30H                  ;否则，转换成数字
MOV DL, AL
MOV AL, BL
MOV CL, 10
MUL CL                       ;第一个数字乘 10，转换成十位
MOV RANGE, AL
ADD RANGE, DL                ;十位数和个位数相加，得到两位阶乘数
MOV AH, 1                    ;读取最后一个控制字符，不作处理
INT 21H
```

接下来是阶乘处理部分，先将大数组的 0 位存入 1，循环变量 **CL** 初始值亦为 1，相乘后若结果满十则向高位传递，进位存储在 **PLUS** 变量中，大数组下一位与 **CL** 相乘时加上 **PLUS** 变量，继续乘至大数组的最高位，进入第二遍外层循环，**CL** 自增 1。当 **CL** 值等于 **RANGE** 时，执行最后一遍最外层循环，然后循环结束，大数组中为最后结果，**BUB[0]** 为个位数，以此类推至大数组最高位。详细循环代码如下：

```
RMUL:                        ;阶乘标号入口
MOV BUB[0], 1                ;数组第一个数为 1
MOV CL, 1                    ;CL 为 1
MULTI:
MOV SI, 0                    ;SI 为 0
CER:
MOV AL, BUB[SI]
MUL CL                       ;初始时 AL=1*1
ADD AX, PLUS                 ;加上上一位乘阶的进位
CMP AL, 10
JAE GETP                     ;AL 大于等于 10 则跳转
MOV BUB[SI], AL
MOV PLUS, 0
```

```

JMP NOSFT
GETP:                                ;若 AL 大于 10，则除以 10，余数放回原数组，商作为
进位存进 PLUS 中
MOV BL, 10
DIV BL
MOV BUB[SI], AH
XOR AH, AH
MOV PLUS, AX
NOSFT:                                ;AL 不大于 10 的跳转入口
ADD SI, 1                            ;SI=SI+1
CMP SI, 20
JB CER                                ;SI 小于 20 则跳回 CER 标号
ADD CL, 1                            ;CL=CL+1
CMP CL, RANGE                        ;若 CL 不大于阶乘数，则继续执行
JBE MULTI

```

最后是结果的输出，从大数组的最高位，即第 20 位开始判断，若不为 0，则输出该位，并标记 BL 为 1，表明接下来各位数全部都要输出。若高位为 0，则数组下标递减，处理下一位，标记 BL 为 0，表明还未处理到结果的最高位。循环遍历到大数组的第 0 位后输出结束。输出部分详细代码如下：

```

MOV BL, 0
MOV SI, 20
PRINT:                                ;输出结果
CMP BUB[SI-1], 0
JNE PRINTB                            ;若该位不是 0，输出该数字
CMP BL, 0                             ;若该位是 0，处理下一位
JE NEXT
PRINTB:
MOV BL, 1
MOV DL, BUB[SI-1]
ADD DL, 30H                           ;转换成 ASCII 码输出该位
MOV AH, 2
INT 21H
NEXT:                                ;处理下一位
SUB SI, 1
CMP SI, 1
JAE PRINT

```

五.意见建议

该实验思路较为简单，需要注意的是输出时若最高位为 0，则通过标记寄存器 BL=0 表明还未到达结果的最高位，若递减至某一位不为 0，表明到达结果的最高位，此后所有位全部输出，设置标记寄存器 BL=1 以输出正确结果。

六.源代码

DATA SEGMENT

BUB DB 20 DUP (0) ;大小为 20 的数组，存放长整型结果

RANGE DB 0 ;阶乘数

PLUS DW 0 ;存放进位

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA

START:

MOV AX, DATA

MOV DS, AX

MOV AH, 1 ;读取字符

INT 21H

SUB AL, 30H ;转换 ASCII 为数字

MOV BL, AL

MOV RANGE, BL ;将数字放入 RANGE 中

MOV AH, 1 ;继续读字符

INT 21H

CMP AL, 30H ;第二次输入字符和 0 的 ASCII 码比较

JB RMUL ;若为其他控制字符，跳转

SUB AL, 30H ;否则，转换成数字

MOV DL, AL

MOV AL, BL

MOV CL, 10

MUL CL ;第一个数字乘 10，转换成十位

MOV RANGE, AL

ADD RANGE, DL ;十位数和个位数相加，得到两位阶乘数

MOV AH, 1 ;读取最后一个控制字符，不作处理

INT 21H

RMUL: ;阶乘标号入口

MOV BUB[0], 1 ;数组第一个数为 1

MOV CL, 1 ;CL 为 1

MULTI:

MOV SI, 0 ;SI 为 0

CER:

MOV AL, BUB[SI]

MUL CL ;初始时 AL=1*1

ADD AX, PLUS ;加上上一位乘阶的进位

CMP AL, 10

JAE GETP	;AL 大于等于 10 则跳转
MOV BUB[SI], AL	
MOV PLUS, 0	
JMP NOSFT	
GETP:	;若 AL 大于 10，则除以 10，余数放回原数组，商作为
进位存进 PLUS 中	
MOV BL, 10	
DIV BL	
MOV BUB[SI], AH	
XOR AH, AH	
MOV PLUS, AX	
NOSFT:	;AL 不大于 10 的跳转入口
ADD SI, 1	;SI=SI+1
CMP SI, 20	
JB CER	;SI 小于 20 则跳回 CER 标号
ADD CL, 1	;CL=CL+1
CMP CL, RANGE	;若 CL 不大于阶乘数，则继续执行
JBE MULTI	
MOV BL, 0	
MOV SI, 20	
PRINT:	;输出结果
CMP BUB[SI-1], 0	
JNE PRINTB	;若该位不是 0，输出该数字
CMP BL, 0	;若该位是 0，处理下一位
JE NEXT	
PRINTB:	
MOV BL, 1	
MOV DL, BUB[SI-1]	
ADD DL, 30H	;转换成 ASCII 码输出该位
MOV AH, 2	
INT 21H	
NEXT:	;处理下一位
SUB SI, 1	
CMP SI, 1	
JAE PRINT	
MOV AH, 4CH	;结束返回
INT 21H	
CODE ENDS	
END START	