汇编语言上机实验

1. 实验目的

考查学生对汇编语言的综合运用能力，其中包括汇编语言基本指令、子函数和宏的运用、十六进制和十进制的转换等功能。

1. 实验要求

一个学生的信息包括姓名、班级、学号（班内序号）、成绩。

试编写程序实现以下功能：

1. 能录入学生成绩（十进制形式）；提示：输入形如：Tom 301 16 95回车，每名学生用一行；选择一个特殊符表示输入结束
2. 能按要求（如班内序号或成绩）进行排序显示；
3. 能统计平均成绩；
4. 能统计不及格、60~70,、70~80、80~90、90~100各分数段的人数

提示：要有必要的提示信息。应该以上往下规划程序结构，划分各个子程序的功能和调用关系。

1. 算法思想

程序使用了16\_10\_print宏，以完成将16进制数转换成10进制数并输出的功能。该宏中分出了四种情况分别讨论，以实现目标功能。

程序中的input\_number是完成学生数目的输入功能的宏。其算法思想为，将num的十位和个位分别用两个寄存器存储（假设为A、B），依次输入数字，每次输入时，将前一次输入的数乘10并加在当前输入数上，以完成学生数量存储。

程序中的input\_info是完成学生信息输入功能的宏。程序中利用dup嵌套实现学生信息info的存储，通过lea si,info和对info的偏移，来完成信息的逐个处理，同时利用存储学生人数的算法，完成对信息的存储。

程序中的menu是实现菜单功能的宏。其中paixu、score\_find、stu\_find联合完成对学生成绩排序并输出的功能，具体算法为：对101-0每个数进行检查，看每个分数上是否有对应的学生，若有，则输出。Cal、sum、divide联合完成对学生平均分的计算，具体算法为：先计算所有学生的总分数，再取出学生总人数，完成平均分计算。score\_section\_、select、section0\_59、section60\_69、section70\_79、section80\_89、section90\_100、print\_select联合完成对分数段的统计。Select完成对分数的比较判断、以跳转到后续代码，对相应分数段人数+1。SectionA\_B完成对A-B分数段的人数+1。

1. 算法流程图

主程序开始

主程序开始

Input\_num输入学生人数

Input\_info:输入各个学生信息

Menu:列出功能菜单 用户给出选择

3

2

4

1

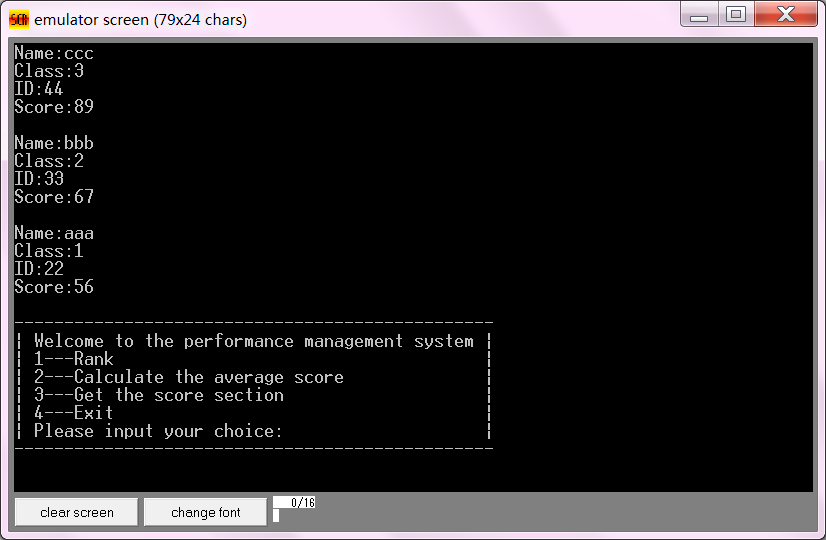
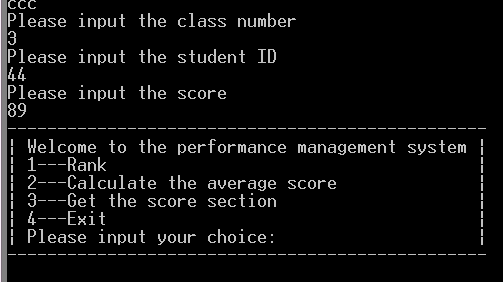
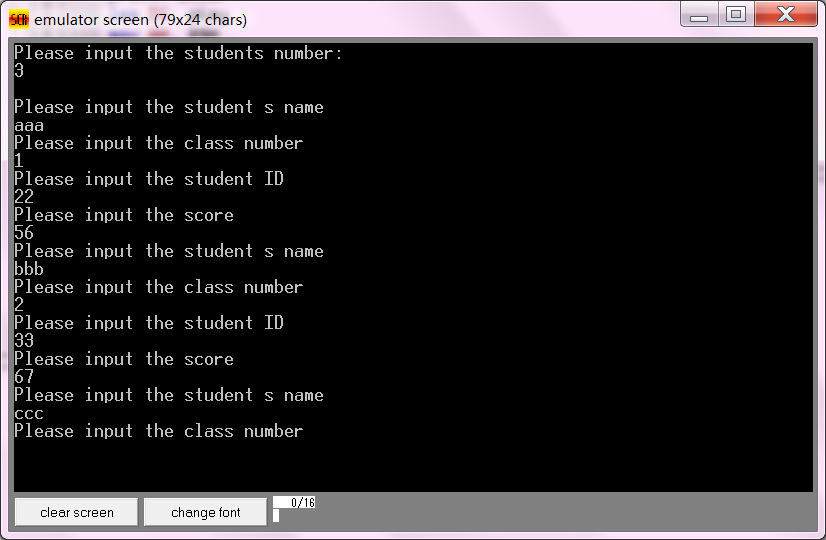
Score\_section获取各分数段人数

Endr:退出系统

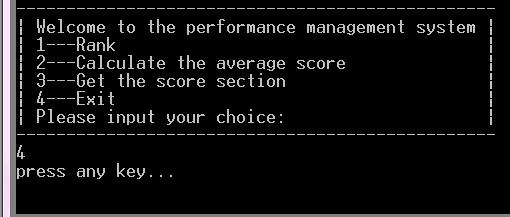
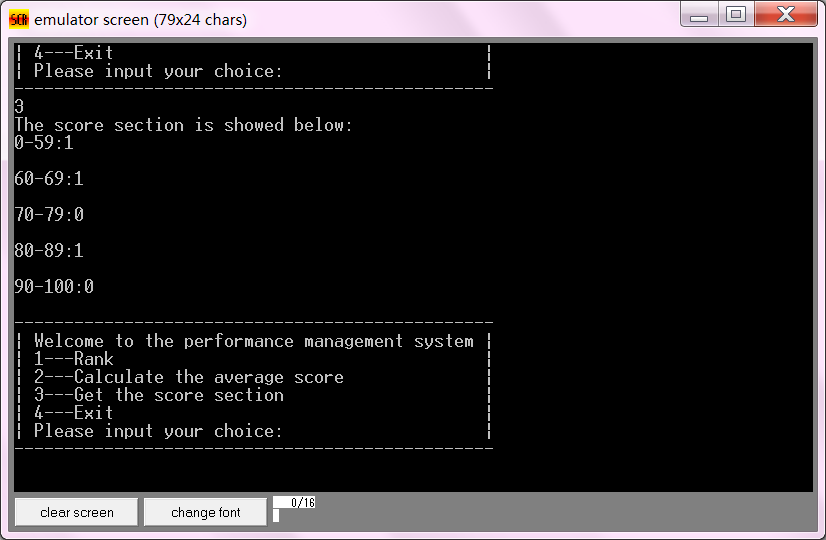
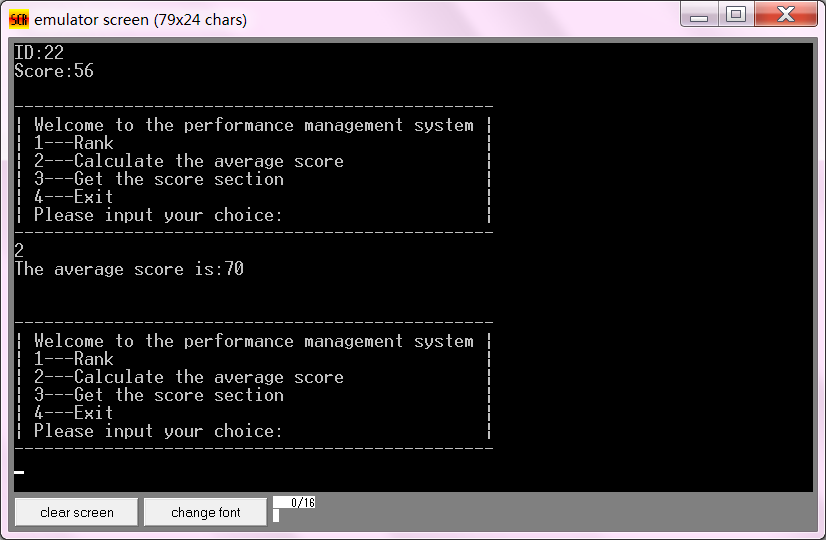
Cal:计算平均分

Paixu:排序并输出

1. 程序运行实例



（上图是排序界面，由于屏幕滚到导致部分未能显示）



1. 源代码

data segment

stunum db 0H

temp db 0H

avescore db 0H

info db 63h dup(1ch dup(0h))

enter db 0Dh,0AH,'$'

section db 05h dup(0h)

pkey db "press any key...$"

menu1 db 0Dh,0AH,'------------------------------------------------',0Dh,0AH,'| Welcome to the performance management system |',0Dh,0AH,'| 1---Rank |',0Dh,0AH,'| 2---Calculate the average score |',0Dh,0AH,'| 3---Get the score section |',0Dh,0AH,'| 4---Exit |',0Dh,0AH,'| Please input your choice: |',0Dh,0AH,'------------------------------------------------',0Dh,0AH,'$'

num\_input db 'Please input the students number:',0Dh,0AH,'$'

name\_input db 'Please input the student’s name',0Dh,0AH,'$'

class\_input db 'Please input the class number',0Dh,0AH,'$'

id\_input db 'Please input the student ID',0Dh,0AH,'$'

score\_input db 'Please input the score',0Dh,0AH,'$'

wrong\_input db 'Input error,please input the right order:$'

name\_output db 0dh,0ah,'Name:$'

class\_output db 0dh,0ah,'Class:$'

id\_output db 0dh,0ah,'ID:$'

score\_output db 0dh,0ah,'Score:$'

ave\_output db 0dh,0ah,'The average score is :$'

score\_section db 0dh,0ah,'The score section is showed below:$'

score1 db 0ah,0dh,'0-59:$'

score2 db 0ah,0dh,'60-69:$'

score3 db 0ah,0dh,'70-79:$'

score4 db 0ah,0dh,'80-89:$'

score5 db 0ah,0dh,'90-100:$'

data ends

stack segment

dw 128 dup(0)

ends

16\_10\_print macro a ;把十六进制转换成十进制并输出

local one,two,two1,three

push ax

push bx

push cx

push dx

one:

xor dx,dx ;初始化寄存器dx

mov al,a

mov bl,100

and ax,00ffh

div bl

mov bh,ah

mov dl,al

add dl,30h

mov dh,dl

cmp dl,30h

jz two

mov ah,02h

int 21h

two:

mov bl,10

mov ax,0

mov al,bh

div bl

mov bh,ah

mov dl,al

add dl,30h

cmp dl,30h ;判断十位是否为0

jnz two1 ;十位不等于0则输出

cmp dh,30h ;十位等于0则看百位

jz three ;百位也等于零则直接输出个位

two1:

mov ah,02h

int 21h

three:

mov dl,bh

add dl,30h

mov ah,02h

int 21h

pop dx

pop cx

pop bx

pop ax

endm

printf macro str ;输出str内容

push AX

push DX

lea DX,str ;输出菜单提示指令

mov AH,09H

int 21H

pop DX

pop AX

endm

scanf\_AL macro ;输入一个字符

mov AH,01h

int 21H

endm

input\_number macro ;输入学生个数，以回车结尾

local input\_num,end\_num

printf num\_input

lea si,stunum

mov DL,0H

input\_num: ;输入学生个数，以回车结尾

scanf\_AL

cmp AL,0DH

je end\_num

sub AL,30H ;将asc码转换成十进制存储

mov DH,AL

mov AL,DL

mov DL,0AH

mul DL ;将上一位数乘10，加上下一位数，完成存储

add AL,DH

mov DL,AL

jmp input\_num

end\_num:

mov [si],DL ;此时内部的数据为16进制的

add si,1

printf enter

endm

input\_info macro ;输入每个学生的个人信息

local info\_,name\_,next,class,id,score,next1

lea si,info

info\_:

printf enter

printf name\_input

mov cx,13h

name\_:

scanf\_AL

cmp al,0dh

jz next

mov [si],al

add si,1 ;si往后偏移一位

loop name\_

next:

mov [si],24h ;24h是$的编码

add cx,1 ;把输入的名字整合成20位长，即将si偏移地址整体向后移动20位（包括输入的名字）

add si,cx

printf enter

printf class\_input

mov cx,02h

class:

scanf\_AL

mov [si],al ;用同样的方式输入class

add si,1

loop class

mov [si],24h

add si,1

printf enter

printf id\_input

mov cx,02h

id:

scanf\_AL ;用同样的方式输入id

mov [si],al

add si,1

loop id

mov [si],24h

add si,1

printf enter

printf score\_input

mov cx,03h

xor bx,bx

xor ax,ax

mov dl,0H

score:

scanf\_AL ;用同样的方式输入score

cmp al,0dh

jz next1

sub al,30h

mov bl,al

mov al,dl

mov bh,0ah

mul bh

add al,bl

mov dl,al

loop score

next1:

mov [si],dl

add si,1

mov [si],24h

add si,1

dec dh

jnz info\_

endm

menu macro ;完成菜单功能操作的宏

menu:

printf menu1

scanf\_AL

cmp al,31h ;根据输入的数字，完成功能跳转

jz paixu

cmp al,32h

jz cal

cmp al,33h

jz score\_section\_

cmp al,34h

jz endr

paixu:

local score\_find,stu\_find,cal,sum,divide,score\_section\_,select,section0\_59,section60\_69,section70\_79,section80\_89,section90\_100,print\_select

lea di,stunum

xor ax,ax

xor cx,cx

mov ch,65h ;65h=101

score\_find: ;从100-0开始对每个分数寻找是否有对应的学生

sub ch,1

js menu

lea si,info

sub si,1ch ;1ch=28

mov cl,[di]

add cl,1

stu\_find: ;寻找对应的分数

dec cl

jz score\_find

add si,1ch

cmp [si+26],ch ;ch此时的值为64h (100)

jnz stu\_find

mov dx,si

printf name\_output

mov ah,09h

int 21h

add dx,14h ;偏移20位，找到class位置

printf class\_output

int 21h

add dx,03h ;偏移3位，找到id位置

printf id\_output

int 21h

printf score\_output

16\_10\_print [si+26]

printf enter

jmp stu\_find

cal:

xor cx,cx

xor ax,ax

xor dx,dx

lea di,stunum ;根据学生人数，雪顶循环次数

mov cl,[di]

add cl,1

lea si,info

sub si,1ch

sum:

add si,1ch ;计算所有同学总分数

dec cl

jz divide

add al,[si+26]

jmp sum

divide: ;完成除法，计算平均值

div [di]

printf ave\_output

16\_10\_print al

printf enter

printf enter

jmp menu

score\_section\_: ;计算各分数段人数

printf score\_section

xor cx,cx

xor ax,ax

xor dx,dx

lea si,info

sub si,1ch

lea di,stunum

mov cl,[di]

add cl,1

lea di,section

select:

dec cl

jz print\_select

add si,1ch

cmp [si+26],3ch ;是否小于60

js section0\_59

cmp [si+26],46h ;是否小于70

js section60\_69

cmp [si+26],50h ;是否小于80

js section70\_79

cmp [si+26],5ah ;是否小于90

js section80\_89

cmp [si+26],65h ;是否小于101

js section90\_100

section0\_59:

add [di],1 ;将0-59分数段人数+1

jmp select

section60\_69:

add [di+1],1 ;将60-69分数段人数+1

jmp select

section70\_79:

add [di+2],1 ;将70-79分数段人数+1

jmp select

section80\_89:

add [di+3],1 ;将80-89分数段人数+1

jmp select

section90\_100:

add [di+4],1 ;将90-100分数段人数+1

jmp select

print\_select:

printf score1

16\_10\_print [di] ;输出完之后，将寄存器清零

mov [di],0

printf enter

printf score2

16\_10\_print [di+1] ;输出完之后，将寄存器清零

mov [di+1],0

printf enter

printf score3

16\_10\_print [di+2] ;输出完之后，将寄存器清零

mov [di+2],0

printf enter

printf score4

16\_10\_print [di+3] ;输出完之后，将寄存器清零

mov [di+3],0

printf enter

printf score5

16\_10\_print [di+4] ;输出完之后，将寄存器清零

mov [di+4],0

printf enter

jmp menu

endm

;代码段

code segment

assume cs:code,DS:data,SS:stack

start:

mov AX, data

mov DS, AX

input\_number ;输入学生个数的宏

input\_info ;输入每个学生的个人信息

menu ;完成菜单功能的宏

endr:

printf enter

lea DX, pkey

mov AH, 09H

int 21H ; output string at DS:DX

; wait for any key....

mov AH, 01H

int 21H

printf enter

mov AX, 4c00H ; exit to operating system.

int 21H

ends

end start ; set entry point and stop the assembler.