（1）总体程序从deal子程序块开始；

（2）deal子程序块功能为从data段中将数据取出来，按行保存到es:[di]所指向的table表中，子程序块流程图思想和实验七基本一致；

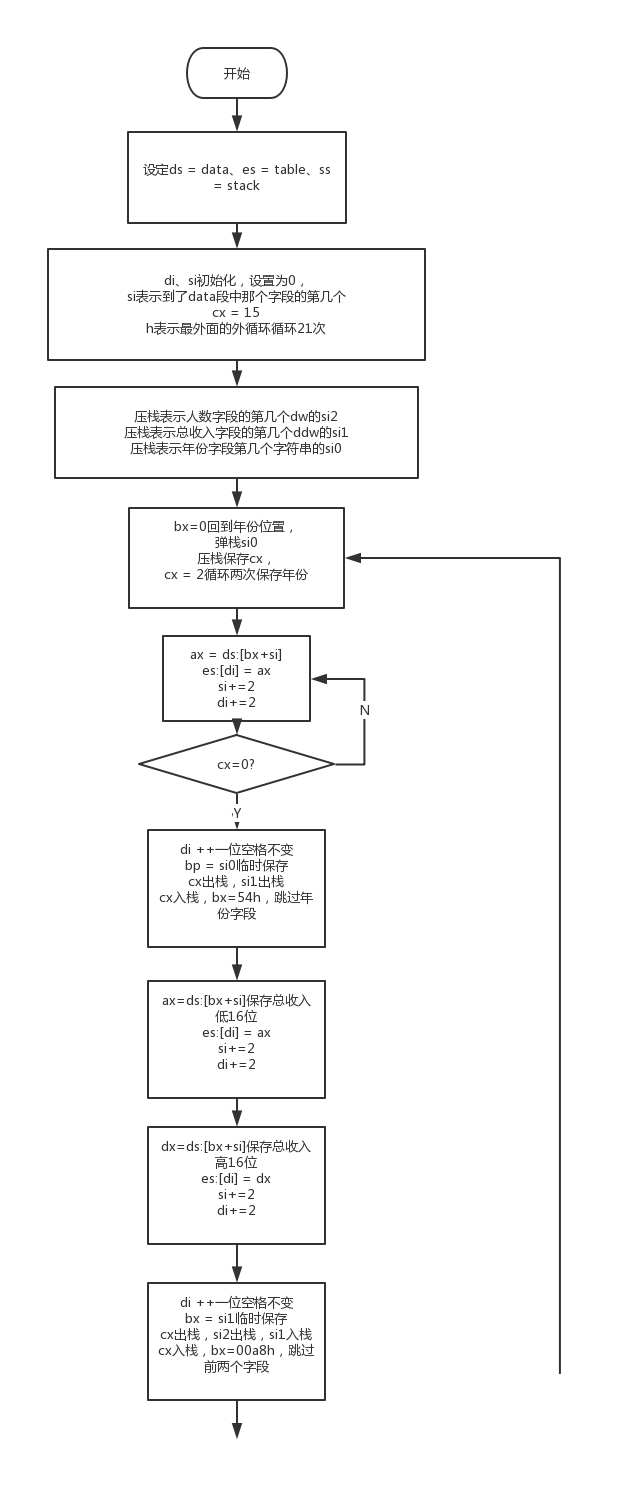


图5-1 流程图

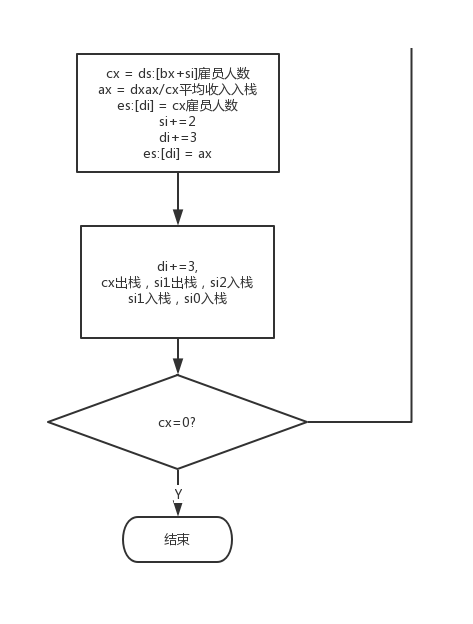


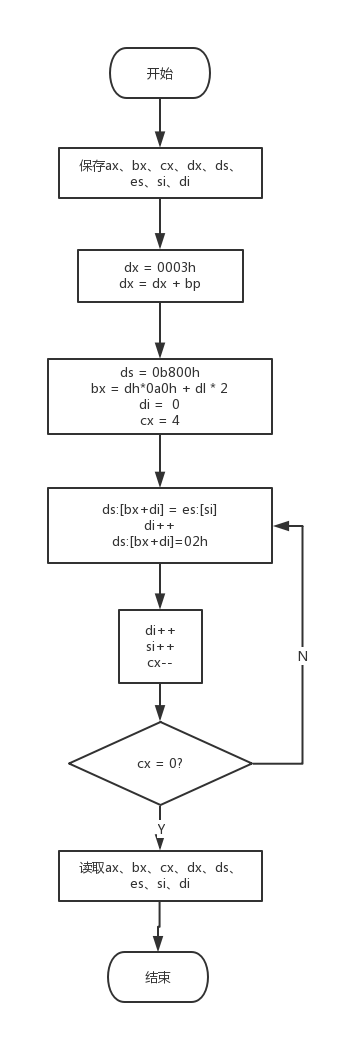
图5-2 流程图（续图5-1）

执行后ds指向data段，es指向table表最开始位置，令si = 0，bp=0300h（bp记录行数，传给dh），cx = 15h，之后进入（3）；

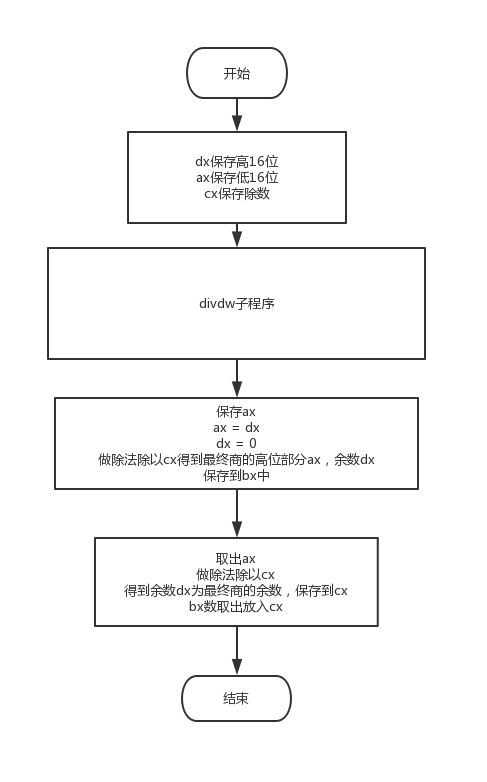
（3）进入string子程序块；

（4）string子程序块功能为每次显示table一行中的4个年份字符。以es:[si]指向数据，es指向table表每一行，si指向每一行的字符，dx =bp+0003h（从bp行第三列开始显示，之后每次循环bp增加），ds:[bx+di]作为显存寻址，cx=4，进入（5）；

（5）将table中的一个字符读取出来并且显示，cx = cx - 1，判断cx是否为0，否继续（4），是（6）

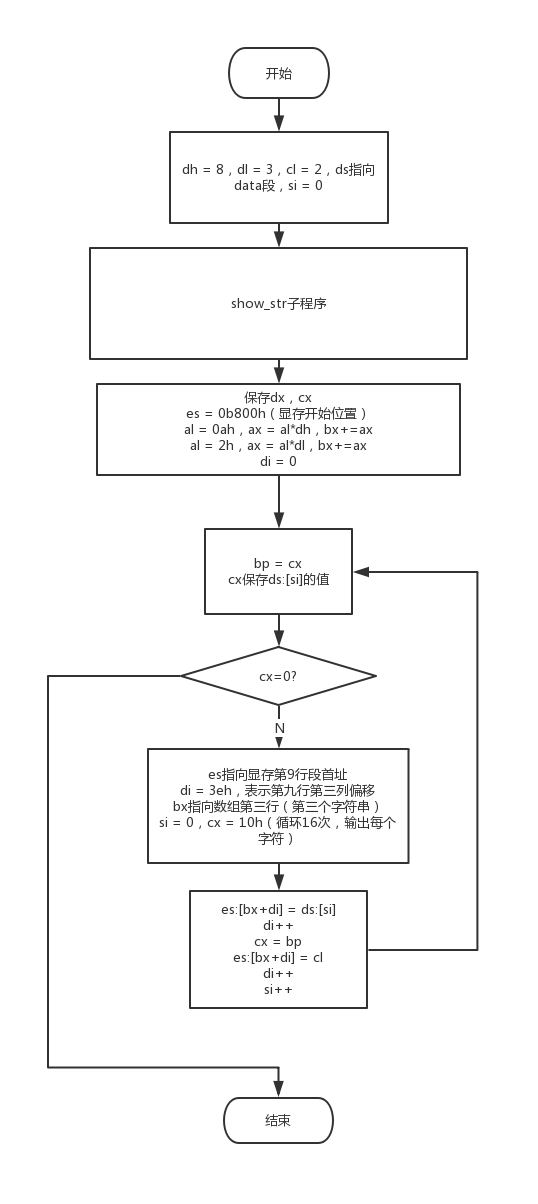


1. 进入ddprint子程序段；
2. ddprint子程序块功能为将dd类型的数据，即table表中的总收入字段，先从table表中读出，并且调用Divdw子程序做能消除溢出影响的双字除法，再讲除法得到的数临时保存在data中，再调用show\_str子程序来将这个数字显示出来。令ds指向data段，di指向0，以es:[si]的形式读取table表中总收入字段的双字，将低字保存在ax中，将高字保存在dx中，进入（8）；
3. 判断当前数字是否为0，是则跳至（11）。否令cx = 0ah，调用divdw子程序
4. Divdw子程序功能是输入参数被除数高位存入dx，被除数低位存入ax，除数存入cx，输出dx是商的高位部分，ax是商的低位部分，cx为余数。 子程序块流程图思想和实验十子程序二基本一致；

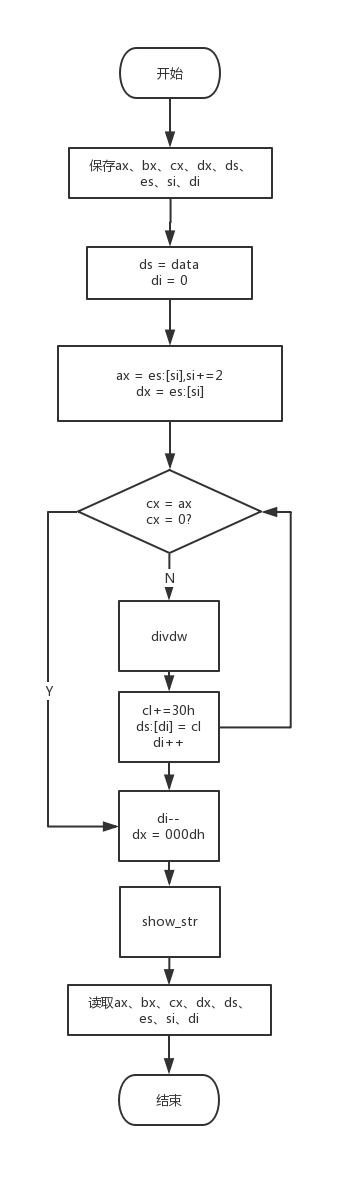
****

1. 将余数的低位部分cl加上30h可以用ascii码显示出来，将cl放入ds:[di]中临时保存，跳转至（8）；
2. di指向已经换为10进制字符的双字的最后一个字符，dx = 000dh（从bp行第13列开始显示），调用show\_str子程序，调用完跳到（15）；
3. Show\_str子程序功能为将ds:[di]指向的字符字节倒着显示在屏幕上，先将dx += bp计算出要显示的行列位置，es:[bx+si]显存寻址；
4. 每次判断di是否已经小于0（即是否已经把ds:[di]中临时保存的字符全部输出），是则跳至（14），否则es:[bx+si] = ds:[di]将除法结束的字符全部显示出来。

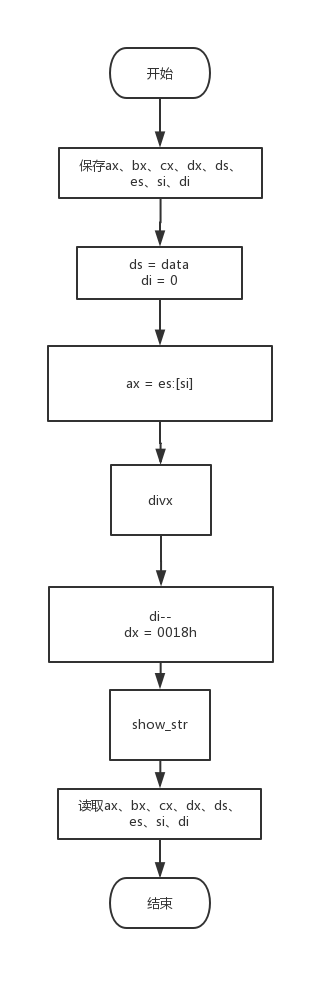
子程序流程图思想和实验A十子程序一基本一致



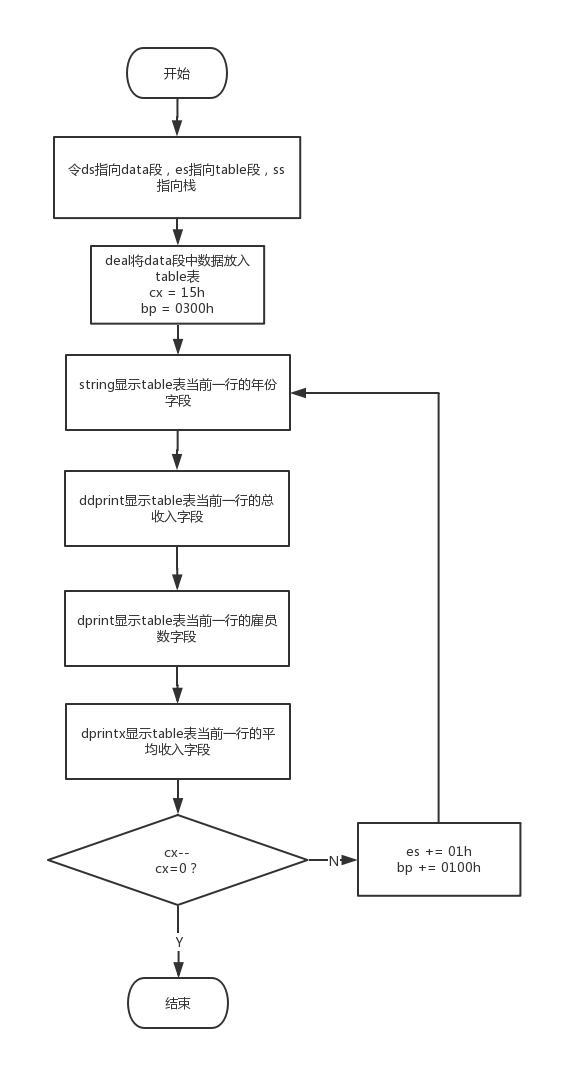
1. 返回调用位置。



1. 进入dprint子程序块；
2. dprint子程序块的功能是显示单字雇员人数字段。思想与ddprint相类似，先从table表中读出，并且做除法，再将除法得到的数临时保存在data中，再调用show\_str子程序来将这个数字显示出来。令ds指向data段，di指向0，以es:[si]的形式读取table表中雇员数的单字，保存在ax中，令dx = 0，进入（17）；
3. 进入divx单字除法并保存的子程序，返回后跳转至（19）
4. Divx单字除法并保存的子程序主要就是做除法和将每个余数变成ascii字符可显示的数保存到ds:[di]中，主要过程与双字除法并保存的思想基本一致，不做赘述。
5. di指向换为10进制的单字的最后一个字符，dx = 0018h（从bp行第24列开始显示），调用show\_str子程序，调用完跳到（20）；



1. 进入dprintx子程序块；
2. dprintx子程序块功能是显示平均收入的，思想与dprint完全一致，唯一有区别的是令dx = 0022h（即显示在第bp行第34列）；
3. Cx减一，判断是否为0，为0则跳至（23）。否则令es += 01h（显示时是以es:[si]作为table的寻址方式，es指向每一行），令bp+=0100h，即显示行加一。跳转至（3）
4. 程序结束



源程序：

assume cs:codesg

data segment

db '1975','1976','1977','1978','1979','1980','1981','1982','1983'

db '1984','1985','1986','1987','1988','1989','1990','1991','1992'

db '1993','1994','1995'

dd 16,22,382,1356,2390,8000,16000,24486,50065,97479,140417,197514

dd 345980,590827,803530,1183000,1843000,2759000,3753000,4649000,5937000

dw 3,7,9,13,28,38,130,220,476,778,1001,1442,2258,2793,4037,5635,8226

dw 11542,14430,15257,17800

asis db 8 dup (0)

data ends

table segment

db 21 dup ('year summ ne ?? ')

table ends

stacksg segment

dw 8 dup (0)

stacksg ends

codesg segment

start: mov ax,data

mov ds,ax

mov ax,table

mov es,ax

mov ax,stacksg

mov ss,ax

mov sp,10h

mov di,0

mov cx,15h

mov si,0

deal: push si;压栈si2

push si;压栈si1

push si;压栈si0

s: mov bx,0

pop si;出栈si0

push cx;压栈cx

mov cx,2h

s0: mov ax,ds:[bx+si];循环2次移动年份

mov es:[di],ax

add si,2h

add di,2h

loop s0

inc di

mov bp,si;暂时保存si0

pop cx;cx出栈

pop si;si1出栈

push cx;cx入栈

mov bx,54h

mov ax,ds:[bx+si];移动收入的第一个字，保存到ax循环2次移动总收入

mov es:[di],ax

add si,2h;向前移动2个字节

add di,2h

mov dx,ds:[bx+si];移动收入的第二个字，保存到dx循环2次移动总收入

mov es:[di],dx

add si,2h;向前移动2个字节

add di,2h

inc di;移动一个空格

mov bx,si;暂时保存si1

pop cx;cx出栈

pop si;si2出栈

push bx;si1入栈

push cx;cx入栈

mov bx,00a8h

mov cx,ds:[bx+si]

div cx;计算人均收入

push ax;人均收入入栈

mov ax,ds:[bx+si];移动一个字节，移动雇员人数

mov es:[di],ax

add si,2

add di,2

inc di

pop ax;人均收入出栈

mov es:[di],ax

add di,2

inc di

pop cx;cx出栈

pop dx;si1出栈

push si;si2入栈

push dx;si1入栈

push bp;si0入栈

loop s

x: mov cx,15h

mov bp,0300h

k: mov si,0

string: push ax

push bx

push cx

push dx

push ds

push es

push si

push di

s1: mov dh,0h

mov dl,3h

add dx,bp

mov ax,0b800h

mov ds,ax;显存位置

mov al,0A0h

mul dh;计算出行的位置

mov bx,ax

mov al,2h

mul dl;计算出列的位置

add bx,ax

mov di,0h;显存寄存器寻址

mov cx,04h

s12: mov ah,es:[si]

mov ds:[bx+di],ah;保存字符

inc di

mov al,02h

mov ds:[bx+di],al;保存字符属性

inc di

inc si;向后读取

loop s12

back3: pop di

pop si

pop es

pop ds

pop dx

pop cx

pop bx

pop ax

mov si,05h

ddprint:push ax

push bx

push cx

push dx

push ds

push es

push si

push di

mov ax,data

mov ds,ax

mov di,0

mov ax,es:[si]

add si,2h

mov dx,es:[si]

s11: mov cx,ax;判断x是不是等于0

jcxz back12;,等于0返回

mov cx,0ah

call divdw;每次除以10

add cl,30h;ascii码变换

mov ds:[di],cl

inc di

jmp s11

divdw: push bx

push di

push ax

mov ax,dx

mov dx,0;求解H/N

div cx

mov bx,ax;储存int(H/N)

pop ax

div cx

mov cx,dx

mov dx,bx

pop di

pop bx

ret

back12: dec di

mov dh,0h

mov dl,0dh

call show\_str

jmp back33

show\_str:add dx,bp

mov ax,0b800h

mov es,ax;显存位置

mov al,0A0h

mul dh;计算出行的位置

mov bx,ax

mov al,2h

mul dl;计算出列的位置

add bx,ax

mov si,0h;显存寄存器寻址

s22: mov cx,di

inc cx

jcxz ok1

mov ah,ds:[di]

mov es:[bx+si],ah;保存字符

inc si

mov al,02h

mov es:[bx+si],al;保存字符属性

inc si

dec di;向后读取

jmp s22

ok1: ret

back33: pop di

pop si

pop es

pop ds

pop dx

pop cx

pop bx

pop ax

mov si,0ah

dprint: push ax

push bx

push cx

push dx

push ds

push es

push si

push di

mov ax,data

mov ds,ax

mov di,0;ds:[di]保存字符串

mov ax,es:[si]

mov dx,0h

call divx

jmp back31

divx: nop

s21: mov dx,0h

mov cx,ax;判断x是不是等于0

jcxz ok2;等于0返回

mov cx,0ah

div cx;每次除以10

add dl,30h;ascii码变换

mov ds:[di],dl

inc di

jmp s21

ok2: ret

back31: dec di

mov dh,0h

mov dl,18h

call show\_str

jmp back32

back32: pop di

pop si

pop es

pop ds

pop dx

pop cx

pop bx

pop ax

mov si,0dh

dprintx:push ax

push bx

push cx

push dx

push ds

push es

push si

push di

mov ax,data

mov ds,ax

mov di,0;ds:[di]保存字符串

mov ax,es:[si]

mov dx,0h

call divx

back41: dec di

mov dh,0h

mov dl,22h

call show\_str

jmp back35

back35: pop di

pop si

pop es

pop ds

pop dx

pop cx

pop bx

pop ax

dec cx

jcxz e

push ax

push cx

mov ax,es

mov cx,01h

add ax,cx

mov es,ax

pop cx

pop ax

mov ax,0100h

add bp,ax

jmp far ptr k

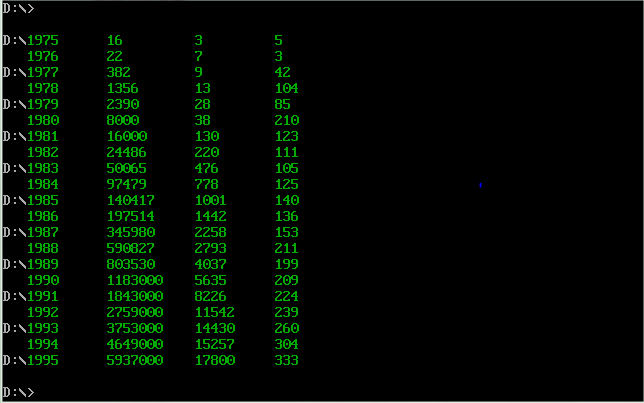
e: mov ax,4c00h

int 21h

codesg ends

end start

程序运行结果：



功能描述：

程序运行后可以在屏幕上显示出来4列数据，第一列为data数据段中的年份，第二列为data数据段与年份对应的总收入，第三列为data数据段与年份对应的雇员数，第四列为data数据段与年份对应的平均收入。