

# 工资计算实验报告

2251656 于思源

## 一、将工资存储到 Table 中

### 1. 基本思路

在 SALINIT.ASM 中定义数据，将公司的总收入和员工人数储存在 .data 段中，随后使用 SALPRO.ASM 读取数据，并计算每年的平均收入，再将年份、总收入、员工人数和人均收入格式化存储到 TABLE 段中。

### 2. 核心代码

读取数据：

；复制年份

mov ax, [si]

mov [di], ax

add si, 2

add di, 4

；复制总收入（32 位分段存储：高 16 位在 BX，低 16 位在 AX）

mov ax, word ptr [si] ; 取总收入的低 16 位

mov bx, word ptr [si + 2] ; 取总收入的高 16 位

mov [di], ax ; 将低 16 位写入表格

mov [di + 2], bx ; 将高 16 位写入表格

add si, 4

add di, 4

；复制雇员人数

mov ax, word ptr [si]

mov [di], ax

add si, 2

add di, 2

；计算人均收入 = 总收入 / 雇员人数

mov cx, [si - 2] ; 雇员人数在 CX 中

；清空 DX 用于存储商的高 16 位部分（因为我们进行的是 32 位除法）

xor dx, dx ; 将 DX 清零，形成 32 位的被除数 BX:AX

；执行 32 位除法 (DX:AX) / CX，结果在 AX 中

div cx ; 32 位除法，结果低 16 位在 AX，中间结果保存在 DX

；将人均收入写入表格

mov [di], ax ; 存储除法结果

add di,

## 二、打印 TABLE 中的数据

### 1. 基本思路

在第一题的基础上，实现了将 TABLE 中的数据逐行输出，系那是为指定格式的表格形式。

### 2. 核心代码

打印数据：

；将 DX 中的数字转换为字符串并显示

mov cx, 5 ; 假设最大 5 位数字

mov bx, 10 ; 除数

lea si, buffer ; 指向缓冲区

convert\_loop:

xor dx, dx ; 清除 DX

div bx ; AX / 10, 余数在 DX, 商在 AX

add dl, '0' ; 转换余数为 ASCII

dec si ; 指向前一个字符位置

mov [si], dl ; 存储字符

loop convert\_loop

；显示转换后的字符串

mov ah, 9

lea dx, [si]

int 21h

## 三、实验总结

在第一题中，通过 ProcessData 过程完成了数据的格式化和存储。在此过程中，遇到的主要问题是无法直接存储 32 位数据，需要分段存储高低 16 位数据，并利用 16 位寄存器进行计算。最终成功将符合格式的数据存储在 TABLE 段，为第二题的打印输出打下了基础。

在第二题中，成功实现了数据的输出功能。PrintTable 过程逐行读取 TABLE 中的数据，并格式化输出。遇到的主要挑战是将整数数据转换为字符串并在屏幕上输出。通过 PrintWord 和 NewLine 实现了逐行的表格显示，最终完成了实验目标。

整个实验帮助我们熟悉了数据格式化存储和数据输出的方法，同时也进一步理解了如何在 DOS 环境下使用汇编语言进行简单的表格数据处理和输出。