

同济大学

汇编语言程序设计

（上机大作业报告）

学 号 : 1953610

姓 名 : 孙久昌

专业/年级: 2019 级 信息安全

所在院系 : 电信学院计算机系

任课教师 : 张光长

完成日期 : 2020 年 7 月 27 日

一. 设计题目

0、成绩统计 ★★★★★

从键盘输入一个班级（每班人数 10~60 人）某课程考试成绩，要求分别统计优（ ≥ 90 分）、良（80~89 分）、中（70~79 分）、及格（60~69 分）、不及格（ ≤ 59 分）的人数和百分比（小数点保留 1 位，四舍五入），以及人数、最高分、最低分、平均分（小数点保留 1 位，四舍五入）。提示：如有缺席考试的学生其成绩不统计在内。

根据题目要求，本次作业基于 8086 汇编语言编写了一个简单的学生成绩统计工具，名为 cghelper。

二. 功能描述

1. 使用方法

启动该工具后，程序提示用户在输入记录、列出全部记录、显示统计信息、清空记录的选项中给出一个选择（图 2.1），并执行相应的操作。执行完毕后，程序回到主菜单并提示用户继续选择操作。

```
Enter today's date (m-d-y): 7-24-2020
The IBM Personal Computer DOS
Version 1.00 (C)Copyright IBM Corp 1981
A>B:
B>cghelper
# Class Grade Helper (c) 2020 stream.
Ready: Choose an operation.

1, Input record(s).  2, List all.
3, Show analysis.   4, Clear.
0, Quit.

=>
```

图 2.1 程序开始执行

如果用户选择输入记录，程序将询问要输入的记录总数，并提示分别键入每组记录的所需信息（图 2.2）。第一次输入完毕后，用户可以选择该操作以追加记录。最多可以保存 90 条记录。

每条记录包括三个部分，即姓名、学号、成绩。其中，姓名一项为不超过 10 个字符的文本内容，可以为空。学号一项只允许由数字组成，不超过七位数。成绩一项要求为[0, 100]区间内的正整数。

```

3, Show analysis.  4, Clear.
0, Quit.
=> 1
How many records in total? 57
Enter info required for student #1.

Name :
ID : 1950062
Score : 99

* Record added.
Enter info required for student #2.

Name :
ID : 1954004
Score : 97

* Record added.
Enter info required for student #3.

Name :
ID : 1952721
Score :

```

图 2.2 输入记录

```

* Totally 57 record(s).
Ready: Choose an operation.
1, Input record(s).  2, List all.
3, Show analysis.  4, Clear.
0, Quit.
=> 3
* Totally 57 score records calculated.
* Students in all segments counts as follows:
Excellent: 17 total, 29.8%.
Good: 13 total, 22.8%.
Medium: 11 total, 19.2%.
Pass: 3 total, 5.2%.
Fail: 7 total, 12.2%.
Absent: 6 total, 10.5%.
* The highest score is 99, lowest 12, and the average 77.0.
Enter to continue....

```

图 2.3 统计信息

选择列出记录，程序会把全部记录信息作为一个列表进行显示（图 2.4）。当输入的记录项较多时，列表可以分屏显示（图 2.5）。需要输入一组新的数据时，选择清空记录即可（图 2.6）。

如果用户选择显示统计信息，程序将按题目要求给出所输入记录的统计信息，包括总人数、各段人数与占比、最高分、最低分与平均分（图 2.3）。

如果程序在执行操作中遇到特殊情形（如在无记录时试图进行统计等），程序会给出提示，这时用户按照字面意思操作即可（图 2.7）。

Cghelper 并不通过要求某个确定的步骤来限制用户的操作，用户可以根据需要使用该程序的功能。

```

=> 2
ID      Name      Score
-----
1952277      0
1951318      0
1953488      0
1956210      0
1956219      0
1956223      0
1953277      12
1951970      20
1953475      22
1953798      30
1952634      37
1952542      43
1950885      51
1952125      63
1954105      66
1950056      69
1952339      71
1951101      71
1953190      73
Enter to continue...

```

图 2.4 列出记录

```

1956031      88
1951097      89
1956216      89
1950980      90
1951399      90
1951336      91
1951315      91
1951126      91
1952122      92
1952221      92
1952212      93
1952735      93
1952622      94
1950064      95
1951045      95
1952636      96
1951245      96
1952721      97
1954004      97
1950062      99
* Totally 57 record(s).
Enter to continue....

```

图 2.5 记录显示完毕

```

Ready: Choose an operation.
1, Input record(s).  2, List all.
3, Show analysis.  4, Clear.
0, Quit.
=> 4
* All records are cleared.
Ready: Choose an operation.
1, Input record(s).  2, List all.
3, Show analysis.  4, Clear.
0, Quit.
=> 2
ID      Name      Score
-----
* Totally 0 record(s).
Enter to continue....

```

图 2.6 清空记录

```

1, Input record(s).  2, List all.
3, Show analysis.  4, Clear.
0, Quit.
=> 3
* Currently no records. Enter some and try again.
Ready: Choose an operation.
1, Input record(s).  2, List all.
3, Show analysis.  4, Clear.
0, Quit.
=> 1
* Please enter a number below. Try again.
* Can't save so many records. Try again.
How many records in total? 1
Enter info required for student #1.
Name :
ID : 1950062
* Please enter a number below in [0, 100]. Try again.
Score :

```

图 2.7 错误提示

2. 运行环境

本程序可以在 1981 年生产的原装 IBM PC 5150¹上执行，完整并正确地完
成全部功能操作；上文的样例图片都是其 PC-DOS 1.0 操作系统的视频信号采集
图像。因此，本程序适用于有史以来的全部 *PC 兼容运行环境*。下面是一些本程
序在其他平台上的运行截图。

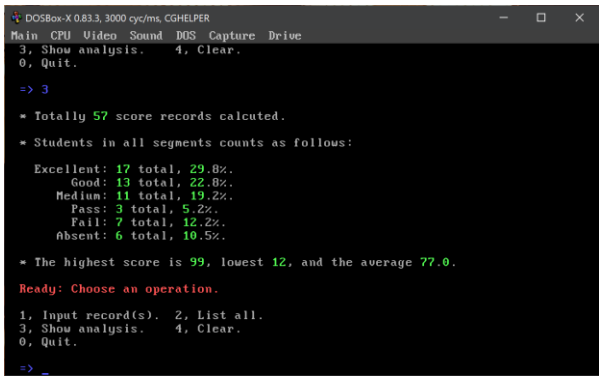


图 2.8 DOSBox-X²截图

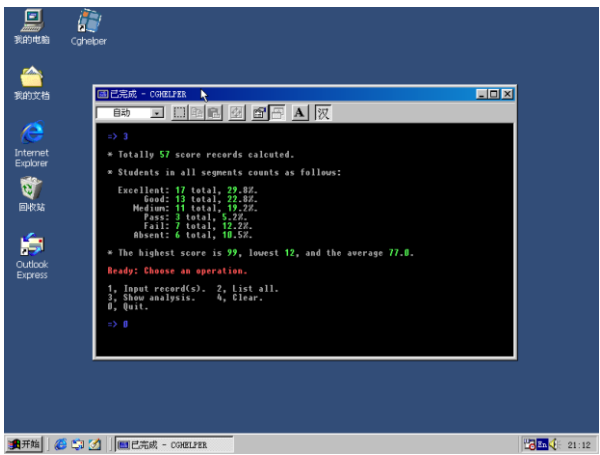


图 2.9 Windows 98 截图

CPU: 奔腾 OverDrive MMX 200³

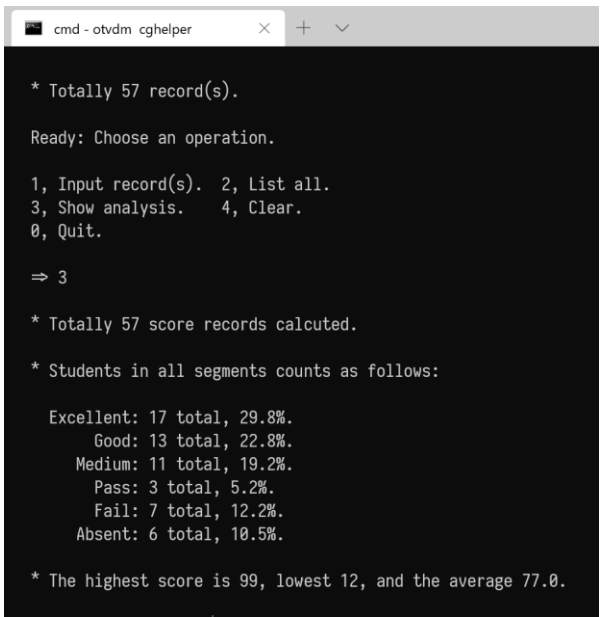


图 2.10 Windows 终端⁴ 截图

借助兼容层 otvdm⁵，运行在 amd64 架
构的 UEFI 计算机上。

三. 设计实现与分析

1. 数据存储

本程序的主要功能是组织和分析所提供的数据。程序的设计目标应该是基于数据，提供一个与用户相联系的中间界面。所以，程序的所有操作都离不开数据。

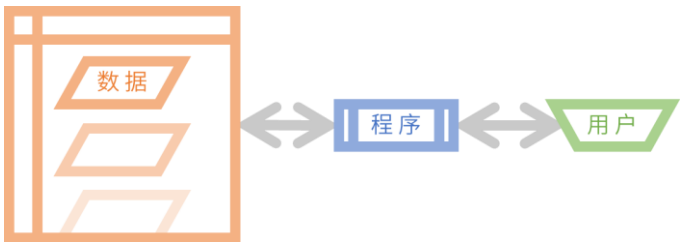


图 3.1 对程序作用的预估

Cghelper 以记录的形式保存学生成绩，并为记录定义了结构体数据类型 `Stu struc`。程序将所有记录都通过内存来存储。程序运行时，内存中会有固定的 1.2KB 空间被占用，用于存储最多 60 条的学生成绩记录。

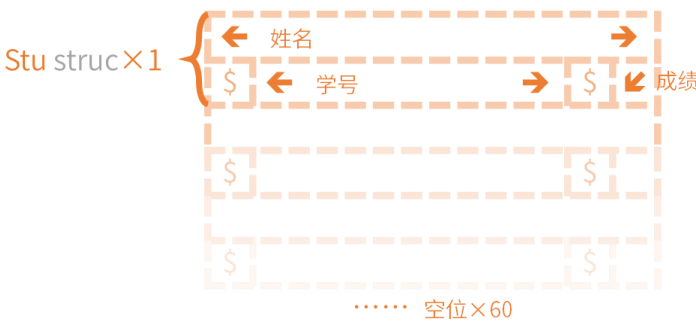


图 3.2 记录存储区

单独的一条记录长 20 个字节。前 10 个字节用于保存姓名，第 12-18 个字节用于保存学号。两项之后的第 11 个字节位置和第 19 个字节位置分别紧跟一个符号 `$`。这是因为程序用到了 DOS 功能调用来输出记录的内容，它们可以用作字符串的截断标记，方便输出。姓名左对齐地存储在其空间内，右侧用空格填充；学号右对齐地存储在其空间内，左侧用空格填充。成绩则直接以二进制数的形式存储在第 20 个字节的位置。

本程序使用实践中的常规方法来使用记录存储区，也就是通过“画布方法”将记录保存在存储空间中。具体地，程序维护一个记录计数器，它的初始值为 0，表示尚未存储任何记录。添加记录时，记录计数器的值相应增加；如果再次开始添加记录，程序根据计数器的值从新的空位置继续。列出记录时，程序根据记录计数器列出相应条数的记录。这样，如果要

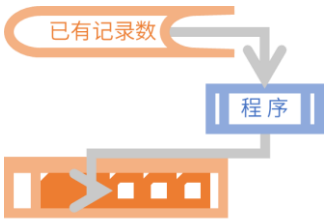


图 3.3 记录存储区的访问模式

清空记录，只需要将记录计数器置零即可，这样后续的数据就将不能被访问到，并会被之后添加的新记录覆盖。

2. 任务模块划分

具体来说，程序应该处理的工作可以分为三个方面，即数据维护、数据处理、用户交互。以上工作需要被分配到不同的子功能中进行处理。考虑到交互的友好性和实现的便利性，程序基于交互菜单的设计模式设计了主框架，以此将不同的子功能集成起来。

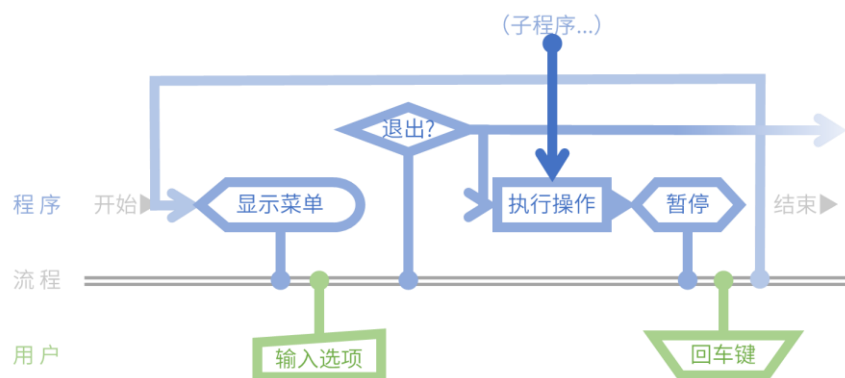


图 3.4 程序主框架的设计

Cghelper 的功能模块划分可以抽象为三个调用层级。最顶层是程序主框架；从主框架出发，可以通过模块的调用关系到达第二层，即实现功能选项的子程序调用；而在第二层子程序中，又包括了用来完成功能步骤的第三层子程序调用。同时，还有一些对系统中断调用的封装，它们对应于特定的实用功能，以独立子程序的形式供程序其他部分调用。

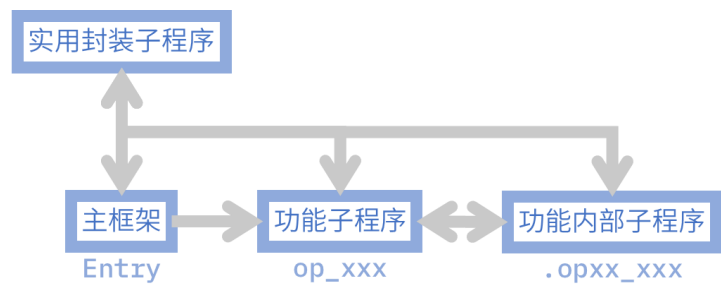


图 3.5 程序的功能模块划分

对于 cghelper 实现的输入、列表、统计、清空四个功能选项，分别有 `op_input`、`op_list`、`op_analyze`、`op_clear` 四个功能子程序调用相对应。主程序通过地址表调用它们。

a 输入记录

输入记录是一个批量的处理操作。为了方便一次输入多组记录，cghelper 输入记录子程序的主体是一个循环结构，调用内部子程序 `.opin_one` 分别单独处理每组记录的输入过程。

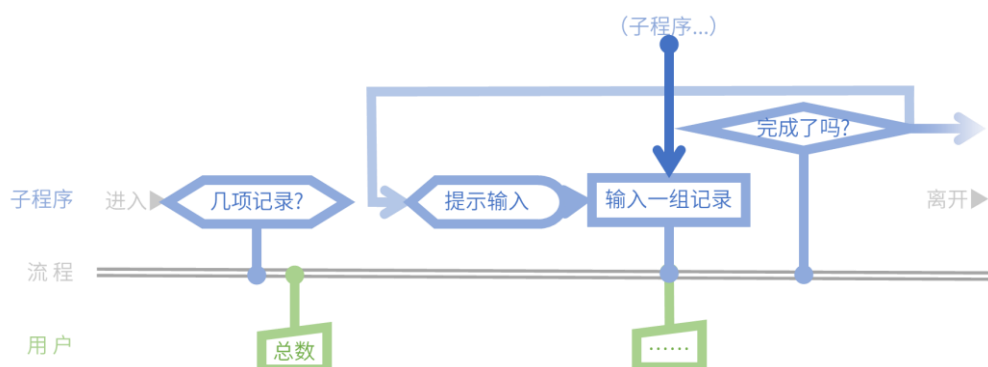


图 3.6 `op_input` 的流程

要输入一组记录，就需要分别向用户获取一组记录中各项的数据。Cghelper 的输入操作均有封装子程序 `input` 处理，调用它就可以得到格式正确的用户输入，因此 `.opin_one` 的工作就是获得用户输入后，将数据保存到记录存储中。其中，保存姓名学号两项可以通过串传送完成，没有数据的区域用空格填满。保存成绩需要一个文本到数字转换过程，这部分由封装子程序 `parsenum` 处理。

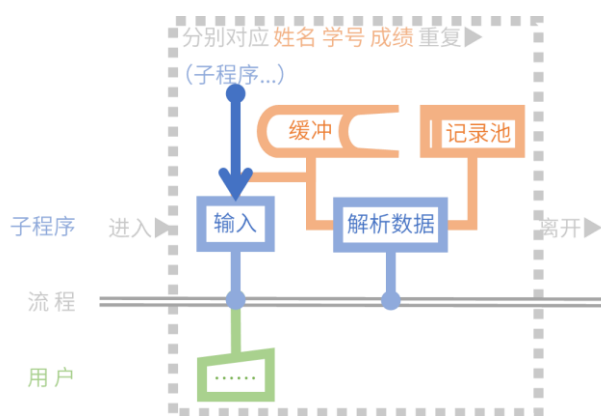


图 3.7 `.opin_one` 的流程

b 列出记录

记录的存储方式决定了其显示起来并不复杂。显示流程如图 3.8 所示。

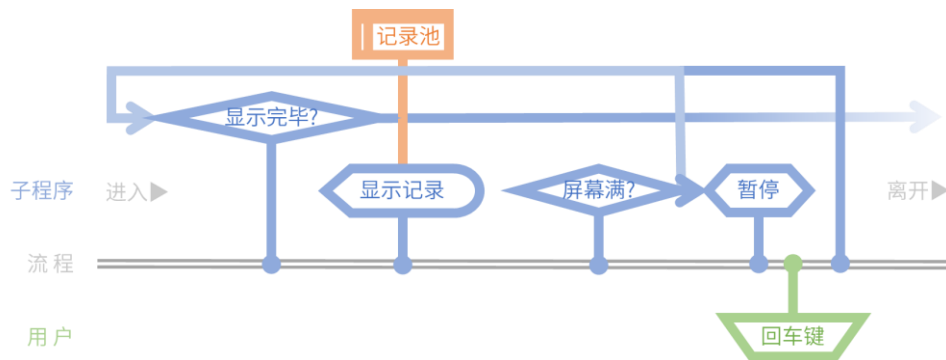


图 3.8 op_list 的流程

c 显示统计信息

结果保存在变量中，先计算再显示就可以了。如果没有数据，这时是没有统计信息的，提示错误并返回即可。

缺考成绩计作 0 分，不参与最低分和平均分计算。

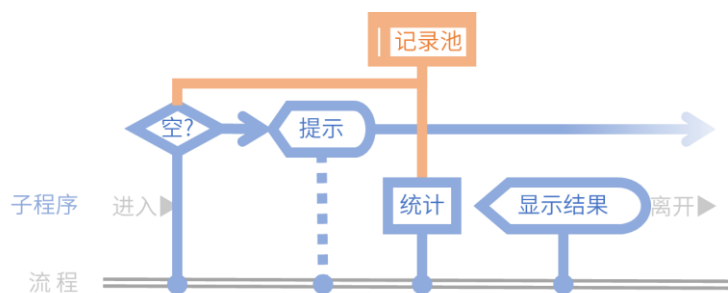


图 3.9 op_analyze 的流程

有小数点后位数要求的数值都作为定点数处理。为了得到四舍五入后保留一位小数的商，先将除数的 $\frac{1}{2}$ 加到扩大10倍的被除数上，作整数除法后再除以10，即可在余数中得到小数部分的数位。

如图 3.10，蓝色线条为以上描述的算法 $q \approx \frac{1}{10} \left\lfloor \frac{10a + \frac{b}{2}}{b} \right\rfloor$ 的结果，蓝圈为除数为整数时结果的值；绿色线条为精确相除后四舍五入的结果，紫色十字为两者之差。这一算法在除数为正整数的情形下可以保证得到正确的结果。

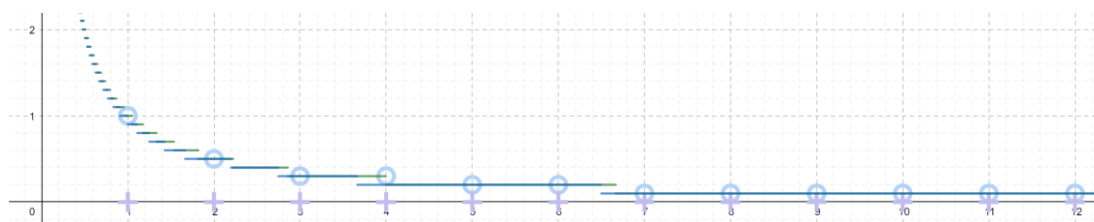


图 3.10 舍入算法的正确性分析

在程序的数据范围限制内，计算中使用的类型不会导致溢出。

d 清空记录

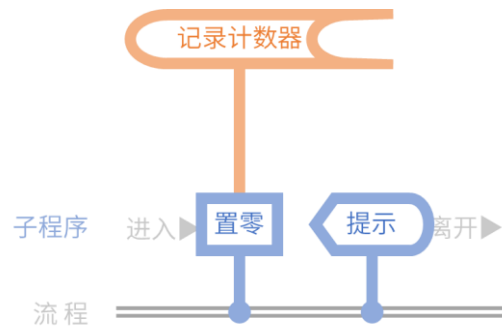


图 3.11 op_clear 的流程

e 实用封装

设计封装子程序的考虑因素是：实现好一个单一的功能，并可以复用到各种不同情形中。

以 `input` 为例，输入的过程可以实现为一个循环。先进行 DOS `0Ah` 号功能调用等待用户输入完毕，然后检查输入正确性：正确则结束，错误则重试。对于不同类型的输入，检查输入正确性的过程也不同。因此 `input` 接受一个子程序地址作为参数以调用。它可以指定为和 `input` 一起定义的 `inchk_num`（检查输入是否为数字）、`inchk_good`（允许任意输入），或者也可以是如 `.inchk_score`（检查输入是否为有效成绩）等。

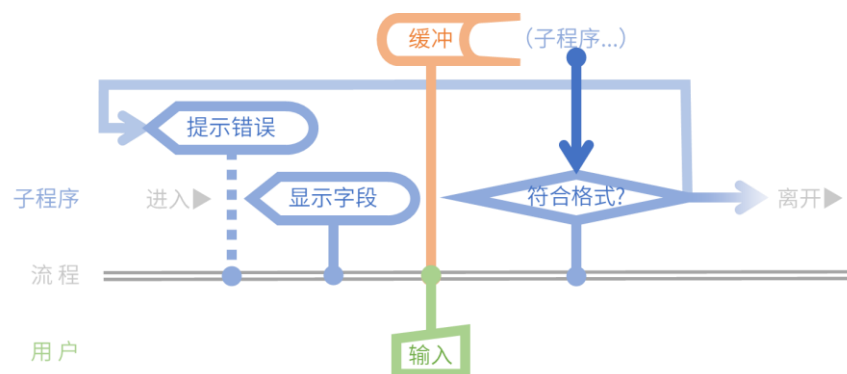


图 3.12 input 的流程

3. 子程序定义

Cghelper 的子程序均不保留调用前的寄存器值。

`getop` 根据用户的选择，在 `0 - 4` 之间确定接下来要执行的选项。
没有参数。

返回: `BX` - 选项值×2（方便寻址）。

`op_input` 执行相应的功能操作。

`op_list` 参数/返回: `recpool` , `reccount`

`op_analyze`

`op_clear`

`input` 获取用户的输入。

参数: `DS:DX` - 要显示的提示语。

`CS:BX` - 用于检查输入的子程序地址。

`infield[0]` - 最大输入长度。

返回: `infield[1]` - 实际输入长度。

`infield+2` - 输入内容。

`inchk_bad` 离开子程序，告诉 `input` 输入是错误的。

返回: `CF = 1`

`inchk_good` 离开子程序，告诉 `input` 输入是正确的。

返回: `CF = 0`

`inchk_num` 检查输入是否为数字。

返回: `CF` - 标志位会根据判断结果被设置。

`parsenum` 将字符串解析为数字。

参数: `infield+2` - 文字。

返回: `AL` - 数。

`pause` 提示用户按回车键继续。

没有参数。没有返回值。

`print` 输出字符串。

参数: `DS:DX` - 字符串，以 '\$' 结束。

putc 更改后续输出文字的颜色，并输出字符串。

参数：BL - CGA 文本模式定义的文字颜色。

DS:DX - 字符串，以 '\$' 结束。

putbanner 输出程序名称。

没有参数。没有返回值。

putmenu 输出功能菜单。

没有参数。没有返回值。

putdecimal 输出数字。十进制。

如果彩色显示可用，数字会显示为绿色（下同）。

参数：AL - 数。

没有返回值。

putdecimal_right 输出数字，右对齐到指定宽度。十进制。

参数：AL - 数。

CX - 对齐宽度。

没有返回值。

四. 构建测试说明

1. 程序的生成

Cghelper 的源代码仅包含在一个单独的源文件中，其生成过程没有其他特殊要求。如果一款编译器具有生成 MZ 格式文件的能力，并且接受 MASM 5 语法的汇编源码作为输入，就可以用它来将本程序的源代码编译为可用的 DOS 可执行文件。连接过程不需要引入符号表等外部内容。

2. 关于调试

要调试该程序，首先需要清楚地了解程序各部分的组织结构，让调试器跳过不相关的部分，并逐步进入执行期间需要观察的部分。通常，按重要性排序，需要特别注意的部分包括：寄存器值的变化、语句的执行顺序以及内存中的数据更新。由于本程序的规模很小，因此调试相对容易。

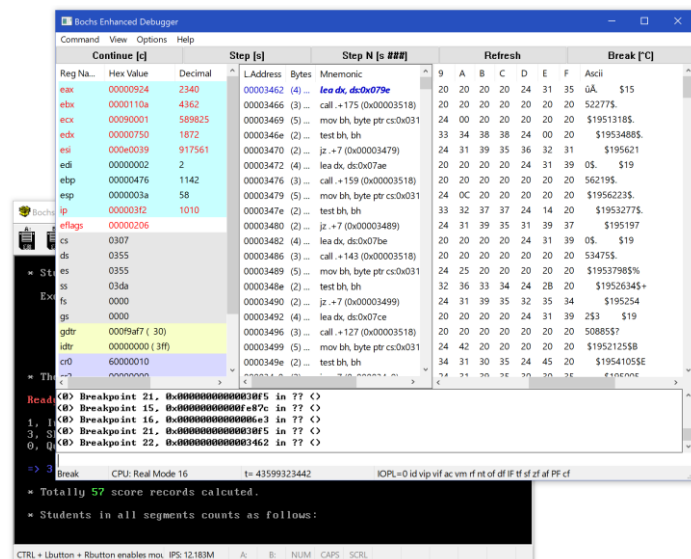


图 4.1 使用 bochs 虚拟机进行调试

3. 可用性测试

本程序经过了充分的测试。测试数据考虑了对数量范围（10~60）和得分范围（0~100）的全面覆盖，并模拟了实际使用场景。可以验证程序的各功能均正确实现，执行状态稳定，足以满足实用要求。

以下按题目要求列出四组代表性的测试数据集，以及对应的正确结果和程序实际给出的结果。

85, 75, 48, 61, 44, 100, 65, 82, 97, 37, 75, 85, 48, 79, 77, 94, 84, 48, 33, 0, 64, 84, 99, 70, 95, 68, 71, 52, 89, 95, 95, 44, 42, 87, 66, 57, 96, 73, 97, 69, 92, 45, 68, 96, 84, 68, 46, 85, 88, 55, 53, 52, 96, 94, 95, 92, 57, 成绩: 35, 76, 89

总数: 60

优: 15人(25%)

良: 11人(18.3%)

中: 8人(13.3%)

及格: 8人(13.3%)

不及格: 17人(28.3%)

缺考: 1人(1.7%)

最高: 100

最低: 33

平均: 72.8

* Totally 60 score records calculated.

* Students in all segments counts as follows:

Excellent: 15 total, 25.0%.

Good: 11 total, 18.3%.

Medium: 8 total, 13.3%.

Pass: 8 total, 13.3%.

Fail: 17 total, 28.3%.

Absent: 1 total, 1.7%.

* The highest score is 100, lowest 33, and the average 72.8.

89, 48, 79, 65, 59, 42, 100, 99, 73, 39, 69, 96, 46, 58, 73, 94, 95, 88, 47,
成绩: 41, 86, 52, 87, 79, 52, 45, 77, 24, 44, 35, 0, 59, 95, 0, 59

总数: 35

优: 6人(17.1%)

良: 4人(11.4%)

中: 5人(14.3%)

及格: 2人(5.7%)

不及格: 16人(45.7%)

缺考: 2人(5.7%)

最高: 100

最低: 24

平均: 66.5

```
.....
* Totally 35 score records calculated.
* Students in all segments counts as follows:
Excellent: 6 total, 17.1%.
Good: 4 total, 11.4%.
Medium: 5 total, 14.3%.
Pass: 2 total, 5.7%.
Fail: 16 total, 45.7%.
Absent: 2 total, 5.7%.
* The highest score is 100, lowest 24, and the average 66.5.
.....
```

81, 84, 82, 91, 10, 84, 20, 91, 62, 85, 46, 93, 73, 82, 63, 71, 30, 65, 61,
成绩: 51, 58, 36, 82, 10, 87, 45, 66, 54, 72, 62, 87, 62, 47, 68

总数: 34

优: 3人(8.8%)

良: 9人(26.5%)

中: 3人(8.8%)

及格: 8人(23.5%)

不及格: 11人(32.4%)

缺考: 0人(0%)

最高: 93

最低: 10

平均: 63.6

```
.....
* Totally 34 score records calculated.
* Students in all segments counts as follows:
Excellent: 3 total, 8.8%.
Good: 9 total, 26.5%.
Medium: 3 total, 8.8%.
Pass: 8 total, 23.5%.
Fail: 11 total, 32.4%.
* The highest score is 93, lowest 10, and the average 63.6.
.....
```

成绩: 63, 26, 55, 50, 35, 96, 86, 58, 57, 59

总数: 10

优: 1人(10%)

良: 1人(10%)

中: 0人(0%)

及格: 1人(10%)

不及格: 7人(70%)

缺考: 0人(0%)

最高: 96

最低: 26

平均: 58.5

```
.....
* Totally 10 score records calculated.
* Students in all segments counts as follows:
Excellent: 1 total, 10.0%.
Good: 1 total, 10.0%.
Pass: 1 total, 10.0%.
Fail: 7 total, 70.0%.
* The highest score is 96, lowest 26, and the average 58.5.
.....
```

五. 讨论和小结

1. 课程学习结语

汇编语言程序设计是一门既有知识性又有技能性的课程。通过线上授课的形式，这两周时间的教学就像一场充实的夏令营，参与的同学们是伴随着充分的练习、交流与探索、实验度过的。现在这门课程已经结课了，潜藏在计算机背后的精巧的逻辑与秩序世界，穿过了一层层复杂的工程架构和漫长的历史变迁，在同学们面前展开了一扇全新的门，将抽象的数理思维和具象的现实相连接。

本期课程的课堂效果较好，达到了增强知识，消除疑惑的效果。线上学习环境包括在线课堂与交流群；上机实验条件包括程序运行环境、调试环境等；学习资料有参考电子书、线上教程等。

对课程知识点的掌握进展顺利。作为对机械演算过程的描述，符号机器码与高级语言相比并不特殊，因此许多概念是在两者之间通用的；而且所有的高级语言设计模式都无一例外地对应到明确的汇编实现。但是，如果将汇编语言投入实际使用，则需要通过大量的细致和专注来克服单条语句功能的有限、程序规模带来的复杂性、寄存器数目的局限和其他限制。因此，本课程的思维价值高于应用价值。实际的汇编语言程序通常伴随着许多优化技巧，这部分知识是需要实践中进一步提高掌握的。

通过课程提升的能力：大概有码字速度、实现构想的能力、针对需求探索最佳实践的能力等。撰写实验报告时没有遇到阻碍。

期待本课程未来可以：更多地关注学生的知识背景，知识点的解释接近学生的思维习惯；鼓励学生亲自动手操作；减少形式要求，多关注代码本身。

2. 关于作业设计

以下是对题目要求中问题的回答。

自学的内容和时间

在课堂上讲授的内容足以完成本作业的设计，在完成过程中没有太多的自学内容。

参考书目或网络资料链接

参考资料是 PC 中断表⁶。还有一张 PC-DOS 软盘映像⁷也许可算参考的网络资源。

参考书目是 *IBM PC Assembly Language and Programming (5th Edition)*, Peter Abel 著。

上机的地点、时数

上机地点：家。用时数：3 天，大概 16 个小时。其中编码用时 1 天，测试 1 天，完成报告 1 天。

独立完成哪些部分

全部。

合作完成哪些部分

没有。

是否有独创的地方

本次作业完成的是一个经典项目，相关技术实现都很成熟。把程序中的数字全部显示成绿色也许可以算一个独创设计。

Cghelper 的完成是一段汇编语言的上手过程，也是一次理解汇编语言的独特经历。完成这个项目之后，再去读 *xchg rax,rax* 这本谜语集⁸，其中的乐趣也能理解不少了，这也说明对汇编语言确实有了新的理解。汇编语言编程作为一种基础技能，相信它可以在算法优化和底层设施开发等领域的后续学习中发挥重要作用。

Cghelper 的实现值得进一步的改进，比如

- 1) 把运行前并没有确定内容的记录池也放进数据段，导致程序在硬盘上要多占用约 1.17KiB 的无意义空间。解决方法是通过 `.data?` 定义一个 BSS 数据段，运行时再准备内存空间。这样就要对程序中 ES 寄存器的初始化逻辑加以修改，并且要调整访问数据时的寻址方式。
- 2) 把大量重复的 "\$" 符号放在记录保存区域中占用空间，而实际每个字段的长度都是确定的，并不是必须以 DOS 字符串的形式输出。可以通过自定义一个子程序用来读取固定数目的字符输出，代替本程序的实现。
- 3) 功能过于简单，不能动态删除、直接修改某条记录，不方便从外部加载数据或将输入的数据外部保存。

基于本程序的设计架构，以上内容的实现都是可行的。但是如果实现了这些内容，程序的复杂性将明显提升，超出课程上机大作业的要求。因此不将这些需求加以实现——这样的取舍是基于理论和实践之间关系的考量。

3. 源码清单

本程序的源码 `K1953610.asm` 已随作业报告一起提供。同时随附的还有可打印格式的源码列表文件，保存为 [cghelper-listing.pdf](#)，可供备用。

4. 签名

孙久昌 2020-7-26

-
- ¹ [The IBM PC's debut - At the beginning](https://www.ibm.com/ibm/history/exhibits/pc25/pc25_intro.html), https://www.ibm.com/ibm/history/exhibits/pc25/pc25_intro.html
 - ² [The Dosbox-X project](https://dosbox-x.com/), <https://dosbox-x.com/>
 - ³ [采用 MMX™ 技术的英特尔® 奔腾® 处理器，200 MHz，66 MHz 前端总线](#), <https://ark.intel.com/content/www/cn/zh/ark/products/49966/intel-pentium-processor-with-mmx-technology-200-mhz-66-mhz-fsb.html>
 - ⁴ [Windows 终端概述 | Microsoft Docs](#), <https://docs.microsoft.com/zh-cn/windows/terminal/>
 - ⁵ [Otvdm/winevdm - Columbia University](#), <http://www.columbia.edu/~em36/otvdm.html>
 - ⁶ [int table - stanislavs.org](#), https://stanislavs.org/helppc/int_table.html
 - ⁷ [WinWorld: PC-DOS 1.x](#), <https://winworldpc.com/product/pc-dos/1x>
 - ⁸ [xorpd | xchg rax,rax](#), https://www.xorpd.net/pages/xchg_rax/snip_00.html