

华南农业大学综合性、设计性实验报告

实验项目名称： 汇编语言综合设计

实验项目性质： 综合性、设计性实验

所属课程名称： 汇编语言程序设计

开 设 时 间： 2023-2024 学年第二学期

指 导 教 师： 陈霓

所属课程		汇编语言程序设计		开 课 学 期		2024 年春季	
实验项目性质		综合性、设计性		报告提交日期		2024. 6. 20	
学 号		姓 名		分 工		组长评分	
SCAU		Link		程序设计和报告撰写		100	
实验题目		图示 HANOI 的移动过程					
评分标准	1.选题难度，包括完成目标的难度，20%； 2.完成质量，主要考虑其设计方法和实现技术，50%； 3.实验报告撰写质量，30%； 4.雷同程序（包括从网上下载的雷同程序），按 0 分计。						
成绩单	学 号	姓名	个人成绩		总成绩	教师签名	
	SCAU	Link					

一、问题描述

1.问题背景

有一种被称为汉诺塔(Hanoi)的游戏。该游戏是在一块铜板装置上,有三根杆(编号 A、B、C),在 A 杆自下而上、由大到小按顺序放置 n 个盘子(如图 1)。游戏的目标:把 A 杆上的盘子全部移到 C 杆上,并仍保持原有顺序叠好。

操作规则:每次只能移动一个盘子,并且在移动过程中三根杆上都始终保持大盘在下,小盘在上,操作过程中盘子可以置于 A、B、C 任一杆上。



图 1 汉诺塔问题

2.系统功能描述

本实验的目标是使用汇编语言编写一个程序,该程序能够模拟并可视化汉诺塔问题的解决过程。程序将提供以下功能:

用户界面: 提供一个简单的用户界面,展示汉诺塔的三根柱子和不同大小的盘子。

盘子表示: 每个盘子的大小用不同长度的线型表示,以便区分。

移动过程: 程序能够按照汉诺塔问题的规则,逐步展示盘子从一个柱子移动到另一个柱子的过程。

递归可视化: 通过递归调用的可视化,展示算法的递归结构和调用栈的变化。

用户交互: 允许用户输入盘子的数量,程序根据输入的数量自动计算并展示解决方案。

动画效果: 为移动过程添加动画效果,提高可视化的直观性和趣味性。

二、数据结构

在本实验中,数据结构的设计主要用于存储汉诺塔问题的状态信息,包括盘子的分布、用户输入、递归调用的栈等。

数据定义

1. 全局变量定义:

len_left dw 0	;哪里开始打印
len_right dw 0	;哪里结束打印
COLUMN_A dw 0,100 dup(?)	;表示当前柱子上有哪几个盘
COLUMN_B dw 0,100 dup(?)	;表示当前柱子上有哪几个盘
COLUMN_C dw 0,100 dup(?)	;表示当前柱子上有哪几个盘

2. 提示信息和用户输入:

tip db 'input n:',0ah,0dh,'\$'	;输入提示
n db 0	;表示 n 个盘子
tempn db 0	;临时存放 N 值

3. 提示信息和用户输入:

nowA dw 1	;当前 A 对应哪个柱子
nowB dw 2	;当前 B 对应哪个柱子
nowC dw 3	;当前 C 对应哪个柱子

4. 递归调用的辅助变量:

mover dw 0	;偏移位置
flag dw 0	

5. 递归调用的栈:

STACKS SEGMENT	;定义了一个 80 字的栈空间
----------------	-----------------

子程序定义

hanoi proc near

实现汉诺塔递归算法的核心子程序。根据盘子数量，递归地将盘子从起始柱子移动到目标柱子，中间使用辅助柱子。

move PROC FAR

负责在屏幕上模拟移动盘子的过程。它更新盘子在柱子上的位置，并调用 Print 子程序来刷新屏幕显示。

Print PROC FAR

负责在屏幕上绘制汉诺塔的柱子和盘子。它使用 len_left 和 len_right 确定每根柱子上盘子的打印位置，并调用 Draw_dot 子程序来绘制点。

Store1 proc near

在递归调用中保存当前柱子的状态，即将 B 柱子的编号存储到 C 柱子的变量中，并递减盘子数量 n。

REC1 proc near

恢复之前由 Store1 保存的状态，即将 C 柱子的编号恢复到 B 柱子的变量中，并递增盘子数量 n。

Store2 proc near

保存当前柱子的状态，即将 A 柱子的编号存储到 B 柱子的变量中，并递减盘子数量 n。

REC2 proc near

恢复之前由 Store2 保存的状态，即将 B 柱子的编号恢复到 A 柱子的变量中，并递增盘子数量 n。

Store3 proc near

将当前盘子数量 n 存储到临时变量 tempn 中，并将 n 设置为 1，用于递归调用的基础情况。

REC3 proc near

从临时变量 tempn 恢复盘子数量 n，用于递归调用的恢复状态。

init proc near

初始化汉诺塔问题的初始状态，将所有盘子放在 A 柱子上，并按大小顺序排列。

Draw_dot

一个底层的绘制过程，用于在屏幕上绘制单个点。它根据传入的颜色和位置信息，在视频缓冲区中绘制点。

Clear_page

清除屏幕上的缓冲区数据，用于在打印新的汉诺塔状态前清空屏幕。

move_page

将缓冲区的数据移动到显示区域，用于更新屏幕上的显示内容。

三、算法描述

汉诺塔问题是一个经典的递归问题，其基本思想是将 n 个盘子从一根柱子（称为源柱）移动到另一根柱子（目标柱），在移动过程中需要遵循以下规则：

1. 只能移动一个盘子。
2. 移动的盘子必须是在柱子顶部的盘子。
3. 将一个较大的盘子放在较小的盘子上面。

算法实现

在本实验中, 我使用汇编语言实现了汉诺塔问题的递归解决方案。以下是算法的具体描述:

1.初始化 (init 过程)

程序开始时, 所有盘子按大小顺序放置在 A 柱上。初始化过程将用户输入的盘子数量存储在 `n` 中, 并将所有盘子的初始状态设置在 `COLUMN_A` 中。

2.递归算法 (hanoi 过程)

`hanoi` 过程是递归算法的核心。首先检查是否只有一个盘子需要移动 (基本情况)。如果是, 则直接调用 `move` 过程将盘子从源柱移动到目标柱, 并刷新屏幕显示。

如果有多个盘子需要移动, 算法将问题分解为三个步骤:

- 1.将上面的 `n-1` 个盘子从源柱移动到辅助柱 (使用递归调用 `hanoi`)。
- 2.将最大的盘子 (第 `n` 个盘子) 从源柱移动到目标柱, 并刷新屏幕显示。
- 3.再次递归调用 `hanoi`, 将辅助柱上的 `n-1` 个盘子移动到目标柱。

3.状态保存与恢复 (Store1, REC1, Store2, REC2, Store3, REC3 过程)

在递归调用过程中, 为了正确地移动盘子, 需要临时改变柱子的编号和盘子数量。这些过程负责在递归调用前后保存和恢复这些状态。

4.移动过程 (move 过程)

该过程负责更新盘子在柱子上的位置。它通过计算偏移量 `mover` 来确定新盘子的位置, 并更新 `COLUMN_A`, `COLUMN_B`, `COLUMN_C` 中的数据。

5.屏幕刷新 (Print 过程)

每次盘子移动后, 都需要调用 `Print` 过程来刷新屏幕显示。该过程根据 `COLUMN_A`, `COLUMN_B`, `COLUMN_C` 中的数据绘制柱子和盘子。

6.绘制点 (Draw_dot 过程)

这是一个底层过程, 用于在屏幕上绘制单个点。`Print` 过程使用它来绘制柱子和盘子。

7.屏幕清除 (Clear_page 过程)

在每次刷新屏幕之前, `Clear_page` 过程被调用来清除屏幕上的旧内容, 确保新的绘制不会被旧内容干扰。

8.屏幕更新 (move_page 过程)

最后, `move_page` 过程将缓冲区的内容更新到显示区域, 使得屏幕上的显示与当前的汉诺塔状态同步。

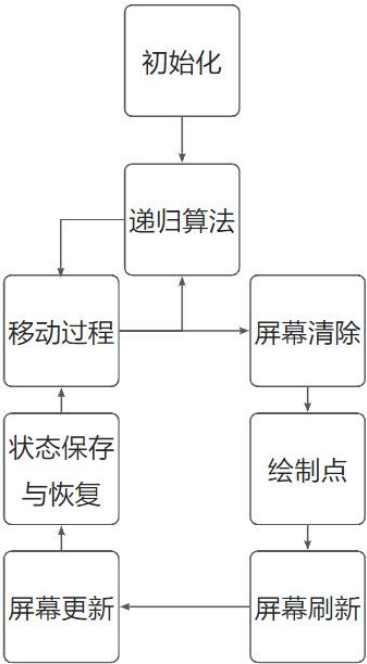


图 2 Hanoi 算法流程图

Algorithm 1 汉诺塔算法

```
1: procedure HANOI( $n, source, auxiliary, target$ )
2:   if  $n == 1$  then
3:     将盘子 1 从  $source$  移动到  $target$ 
4:     PRINTRESULT
5:   return
6:   end if
7:   HANOITOWER( $n - 1, source, target, auxiliary$ )           ▷ 递归步骤 1
8:   MOVEPLATE( $source, target$ )                             ▷ 移动最大的盘子
9:   PRINTRESULT
10:  HANOITOWER( $n - 1, auxiliary, source, target$ )           ▷ 递归步骤 2
11: end procedure
12: procedure MOVEPLATE( $fromRod, toRod$ )
13:  计算盘子在  $fromRod$  上的位置
14:  计算  $toRod$  上新盘子的位置
15:  更新盘子位置, 将盘子从  $fromRod$  移动到  $toRod$ 
16:  PRINTRESULT
17: end procedure
18: procedure PRINTRESULT
19:  清除屏幕
20:  根据当前盘子的位置状态绘制所有柱子和盘子
21:  刷新屏幕显示
22: end procedure
23: procedure INITIALIZE( $n$ )
24:  设置盘子数量为  $n$ 
25:  初始化所有盘子在  $source$  柱子上, 按大小顺序排列
26: end procedure
27: procedure MAIN
28:  输入盘子数量  $n$ 
29:  INITIALIZE( $n$ )
30:  HANOITOWER( $n, "A", "B", "C"$ )
31: end procedure
```

图 3 Hanoi 算法伪代码

四、效果与测试情况

软件环境

任何支持汇编语言开发和运行的环境。使用相应的汇编器和链接器将汇编代码编译成可执行程序。个人在 DOSBox 虚拟环境下运行测试。

1.基本测试: 输入较小的盘子数量 (如 $n=1$ 或 $n=2$ 或 $n=3$), 验证程序能够正确完成汉诺塔的移动, 并观察可视化效果。

2.中等规模测试: 输入中等数量的盘子 (如 $n=5$ 或 $n=7$), 测试程序的递归深度和性能。

3.大规模测试: 输入较大的盘子数量 (如 $n=10$ 或更高), 测试程序的稳定性和长时间运行的性能。

测试结果

1.基本测试: 程序成功完成了 1、2、3 个盘子的汉诺塔移动, 可视化效果清晰, 盘子的移动顺序和位置准确无误。

2.中等规模测试: 对于 5 个和 7 个盘子的情况, 程序同样能够正确完成移动任务, 但移动次数增多, 递归深度更深, 可视化更新速度略有下降。

3.大规模测试: 在 13 个盘子的情况下, 程序运行正确, 但由于递归次数大幅增加, 导致可视化更新速度变慢。

程序实际效果

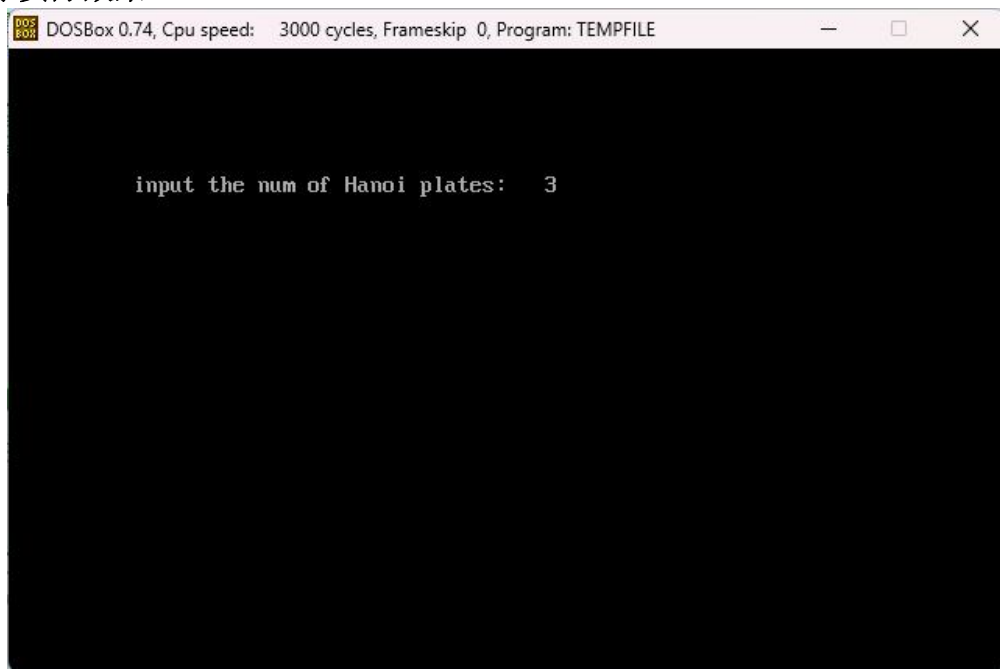


图 4 程序初始页面



图 5 Hanoi 移动结果(小规模 n)

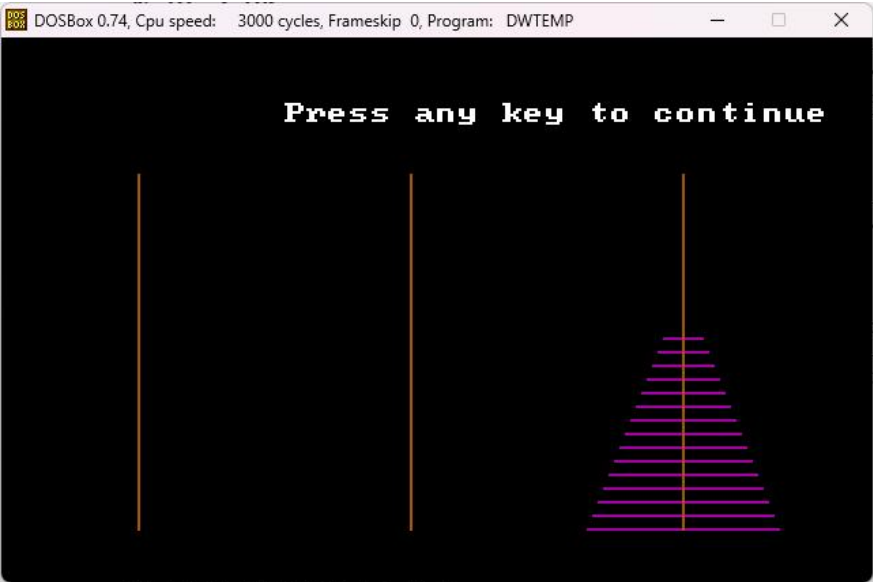


图 6 Hanoi 移动结果(大规模 n)

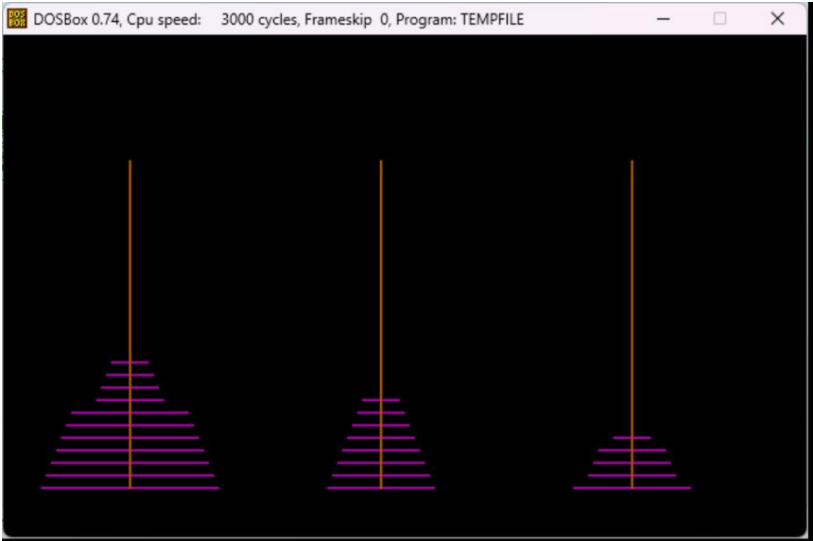


图 7 Hanoi 移动过程

五、分析与讨论

不足之处

在处理大量盘子时，递归调用的深度增加，导致程序运行速度变慢，影响用户体验。并且当盘子数量较多时，部分细节可能无法清晰显示。盘子无法全部展示出来。

此外，程序的可视化界面也可以做得更加美观。

优化思路

考虑使用迭代算法代替递归，减少函数调用的开销，提高程序性能。

程序可以增加比如暂停、恢复、步进等控制功能，让用户能够更好地控制和观察汉诺塔的移动过程。

实验过程体会

这次汇编语言综合性实验对我来说是一项兴奋且具有挑战性的任务，利用汇编语言实现模拟并可视化 Hanoi 的移动过程。

在高级语言中，我们可以方便地使用现成的库和函数来处理算法流程、屏幕可视化，而在汇编语言中，我们必须直接操作计算机的底层。这包括了理解寄存器、内存和指令集等概念。通过手动编写指令，我对计算机如何执行每一条指令以及如何管理数据有了更深入的了解。

最重要的是，我在这个过程中培养了解决问题的能力和耐心。汇编语言编程是一项复杂而繁琐的任务，常常需要仔细检查和调试代码。遇到错误时，我学会了分析问题的根源，并逐步调试代码以找到解决方案。这个过程需要耐心和恒心，但当我成功地解决问题时，获得的成就感是无与伦比的。

通过本次实验，加深了对汇编语言的理解，不仅学习了汇编语言的编程技巧，还加深了对递归算法和算法可视化的理解，理解了递归的基本原理和应用场景。同时，也认识到了在实际编程中可能遇到的挑战，如性能优化、用户体验改进等，在解决实验中遇到的问题时，提高了分析问题和解决问题的能力，为未来的学习和研究打下了坚实的基础。

六、参考资料

- [1]沈美明,温冬婵. IBM-PC 汇编语言程序设计. 北京: 清华大学出版社, 2001
- [2]叶晓霞,彭小红.基于 DEBUG 查找汇编语言源程序中逻辑错误的方法研究.信息记录材料,2019,20(10):152-155
- [3] EnTaroAdunZ. 8086 汇编 - 图示 HANOI 的移动过程 [OL](2017-07-31)[2021-6-20].
<https://blog.csdn.net/EnTaroAdunZ/article/details/75252335>

七、致谢

本次实验设计能够顺序完成,离不开老师的指引教导。这次综合实验设计,得益于老师平时的教导,老师对汇编语言的精深理解和教学热情极大地激发了我对计算机底层与硬件的兴趣。在老师的教导下,我得以掌握关键的编程概念和技巧,这些都在我的汉诺塔程序设计中发挥了重要作用,使我在做实验之前便掌握了大部分有关汇编语言的知识,能够将精力更多花在汉诺塔问题的算法设计和模拟过程可视化的方面。

我将继续努力,不负期望,将所学知识应用于未来的学习和研究之中。

八、实验报告原创声明

本人郑重声明:所呈交的实验报告是本人综合所学知识独立所取得的工作成果。除了参考所标明的参考资料外,本报告不包含任何其他个人或集体已经发表或撰写的成果,保证不会是抄袭、拼凑的报告。对本实验报告的完成做出贡献的个人/集体,均已在文中以明确的方式表明。

本人完全意识到本声明的意义。

组长: Link