**深 圳 大 学 实 验 报 告**

**课程名称：­ 计算机系统**

**实验项目名称： 试验四 LC-3汇编语言试验 简单游戏设计**

**学院： 计算机与软件学院**

**专业： 软件工程**

**指导教师： 蔡晔**

**报告人： 郭昌华 学号：2022190025 班级： 02B**

**实验时间： 2024年5月 23日 星期四**

**实验报告提交时间： 2024年 6月 7日 星期四**

**教务部制**

|  |
| --- |
| **实验目的：**   1. 分析和理解试验指定的需解决问题。 2. 利用LC-3的汇编代码设计实现相关程序。 3. 通过LC-3仿真器调试和运行相关程序并得到正确的结果。 |
| **实验内容与实验要求:**   * **游戏设计与规则：**实现一个简易版的四子棋游戏，该游戏为双人轮流下棋的桌面游戏，棋盘为6x6的网格。玩家需连成一行（水平、垂直或对角线）四颗棋子以获胜。游戏不允许悔棋，且已有棋子的位置不可再落子。若棋盘填满而无人获胜，则游戏以平局结束。 * **交互流程：**程序需通过LC-3计算机系统的键盘和输出窗口实现玩家交互。开始时展示空棋盘，并以ASCII码表示棋盘状态（"-"代表空位，"O"代表玩家一，"X"代表玩家二）。玩家一先手，随后双方交替操作。玩家通过输入列号（1-6）选择落子位置，系统自动处理行号，从上至下填充。输入验证需严格，确保在合法范围内，否则提示错误并要求重新输入。 * **程序实现细节：**内存中用36个单元存储棋盘状态，分别用0、1、-1表示空位、玩家一和玩家二的棋子。程序需使用LC-3汇编语言编写，包括处理输入输出的TRAP调用（如GETC和OUT），并采用子程序设计以提高模块化和可维护性。所有输入输出必须使用ASCII字符。程序的起始地址为.ORIG x3000。 * **判定胜负与结束条件：**每次落子后，程序需检查是否有玩家获胜，通过遍历棋盘判断是否存在四子连线。若某方获胜，输出“Player X Wins.”（X为获胜玩家编号），平局则输出“Tie Game.”，并终止程序运行（Halt）。现场考核时，需准备讲解所设计的子程序及其功能，以及判定赢家的具体算法逻辑。 |
| **实验步骤：**   1. **问题分析及对应代码：**   **分析：**  要实现题目要求的功能,我经过不断的思考,不断拆解问题为一个又一个子程序,最终实现了这个较为复杂的功能.在主函数部分,我设置了一个循环,**依次执行:打印棋盘,获取输入,判断输入合法性,放置棋子,棋盘状态检测,交换控制权**这几个子程序,直至有玩家获得胜利或棋盘被置满,输出胜利信息或平局提示.同时,若用户的输入不合法,输出响应提示,并重新获取输入.  本题的关键点在我看来可以分成两部分.首先是**棋子的放置部分**,由于默认从第六行开始尝试放置,我将放置棋子分为六次尝试,依次从第六行向上尝试,放置成功则退出,为成功则输出提示.而每一次放置实质上都是等价的,于是可以再编写一个放置棋子到某一行的子程序,大大提升了代码可读性,使代码的逻辑功能更加清晰.  第二个关键点是对于**棋盘状态的检测**,即胜利和平局的检测.我首先检测平局,根据题目的游戏规则,实际上平局出现时一定是第一行放满了棋子,所以只需要判断最高行即可.而对于胜利情况的检测,为了提高检测的效率,我采取了**”当前点拓展”**的方式.因为我们知道,胜利条件的出现必然附带一次落子,且该落子显然在胜利的四子之一中,于是我们只需要以当前落子处为起点,**分四个方向不断尝试寻找相连的同类棋子,统计个数**即可,任意一个方向检测到超过四个棋子相连,游戏胜利.具体为了实现这个思路,我编写了CheckWin子程序,在其中调用如下:  CheckHorizontalWin()  CheckVerticalWin()  CheckDiagnalWin()  CheckAntiDiagnalWin()  这四个方向的胜利检测子程序,最终实现了题目要求的功能.  **代码：**  **由于实现该完整程序我使用了600行左右的汇编代码,此处只展示和解释一部分关键汇编代码.**    图 1 游戏主循环    图 2 提示信息数据区部分数据    图 3 输出提示子程序    图 4 交换控制权    图 5 获取用户列输入    图 6 检测输入是否合法    图 7 交换控制权    图 8 打印棋盘    图 9 辅助棋盘打印,根据不同数字打印棋子字符    图 10 逐行放置棋子    图 11 辅助放置棋子,只针对一行    图 12 胜利检测    图 13 打印最终结果    图 14 平局检测    图 15 获取指定行列的数值    图 16 水平方向胜利检测(其余方向类似)    图 17 加载程序到模拟器中    图 18 错误输入提示    图 19 棋盘的循环打印    图 20 水平方向胜利检测示例    图 21垂直方向胜利检测示例    图 22主对角线方向胜利检测示例    图 23副对角线方向胜利检测示例 |
| **实验结论：**  本次实验极大增强了我对LC3汇编语言的深入理解与实践能力，尤其在子程序应用方面收获颇丰。我认识到，将复杂任务拆解为小块——即各个子程序，是解决复杂问题的有效策略。这样做不仅避免了繁琐的循环结构，还显著提升了代码的可读性和逻辑清晰性。举例来说，我将放置棋子的操作独立成专门的子程序，简化了代码结构。同样，在设计胜利判定机制时，我依据横向、纵向及两个对角线方向，创建了四个独立的子程序，有效降低了实现这一多维度检测逻辑的难度。  此外，实验加深了我对LC3汇编子程序调用机制的掌握，尤其是在寄存器管理方面。由于LC3仅配备8个寄存器，处理多状态或调用子程序时，必须精心安排寄存器的使用，包括临时保存与恢复寄存器内容至内存的技巧。这个过程对于保持子程序调用的正确性和效率至关重要。尤其值得注意的是寄存器R7的特殊性，它会在执行子程序跳转（JSR）指令时自动调整，因此在编程时必须给予特别关注，确保其正确性和程序的稳定性。 |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。