**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 中断实验**

**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6月 20日 星期四**

**实验报告提交时间： 2024年 6月 20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:**   * **实验内容**： * 理解中断的概念。 * 编写用户程序，该程序持续间隔输出“ICS”字符，并通过计数代码控制输出速度。 * 实现键盘中断处理程序，响应用户输入，将字符打印10次，不使用TRAP指令。 * 手动设置操作系统使能代码，包括栈空间初始化、中断向量表设置和键盘中断使能。 * **实验要求**： * 用户程序需嵌入DELAY子程序控制输出速度，确保屏幕输出可读。 * 中断处理程序需暂存和恢复寄存器状态，响应键盘输入，且不使用TRAP指令。 * 完成操作系统使能代码的编写，模拟操作系统的行为，包括栈空间的初始化、中断向量表的设置和键盘中断的使能。 * 确保实验在Windows环境下进行，因为Linux版本不支持中断。 * 参考教材相关章节和附录获取更多信息和指导。 * 实验中不能使用TRAP指令显示字符，而应通过读写DSR实现。 |
| **实验步骤：**   1. **问题分析及对应代码：**   **分析：**   1. **初始化栈指针**：我首先将栈指针初始化到 x3000，这是因为在没有操作系统的情况下，我需要手动设置一个栈空间来保存程序的状态。     图 1**初始化栈指针**：   1. **设置中断向量表**：我通过加载键盘中断处理程序的入口地址到中断向量表的指定位置，确保当键盘中断发生时，程序能够跳转到正确的处理代码。     图 2**设置中断向量表**   1. **使能键盘中断**：我通过设置 INTERRUPT\_MASK 并更新 KBSR 寄存器来使能键盘中断，这样LC-3就能够响应键盘输入产生的中断信号。     图 3**使能键盘中断**   1. **编写用户程序主体**：我创建了一个循环 PRINT\_LOOP，用于周期性地打印“ICS”字符。我还调用了 DELAY 子程序来控制打印速度，以便于观察输出。     图 4**编写用户程序主体**   1. **实现 DELAY 子程序**：我编写了一个 DELAY 子程序，通过递减计数器 COUNT 来实现延时效果。     图 5 **DELAY 子程序**   1. **编写**：我实现了一个中断处理程序，它首先保存了所有使用到的寄存器状态，然后打印提示文本。     图 6**中断处理程序**   1. **处理用户输入**：在中断处理程序中，我使用了 GET\_INPUT 子程序来读取用户输入的字符，并将其存储在 INPUT\_STRING 中。     图 7**处理用户输入**   1. **显示输入字符**：我通过 OUTPUT\_IT 循环将用户输入的字符打印了指定的次数，这是通过 OUTPUT\_TIMES 定义的。     图 8**显示输入字符**   1. **恢复寄存器状态和返回**：在中断处理程序的最后，我恢复了所有保存的寄存器状态，并使用 RTI 指令返回到中断前的程序状态。     图 9**恢复寄存器状态和返回**   1. **数据定义**：我定义了所有需要的常量、地址和字符串，包括中断向量地址、键盘状态寄存器地址、输入提示和输入字符串的缓冲区。     图 10**数据定义**  结果截图:    图 11中断前的输出    图 12中断状态,等待输入至回车    图 13输入完毕,正常打印    图 14打印完毕后回到了用户程序正常运行 |
| **实验结论：**  本次实验显著提升了我对LC-3汇编语言中断机制的理解和实现能力，尤其是在处理内存映射寄存器方面。通过这次实验，我深刻认识到中断在操作系统中处理异步事件的重要性，以及在汇编层面实现中断处理的复杂性。  首先，我学习了如何初始化栈指针，这是模拟操作系统行为的关键步骤。在LC-3上，由于没有操作系统的自动栈管理，我必须手动设置栈空间，以确保中断处理程序能够正确地保存和恢复现场。  其次，我深入理解了中断向量表的设置。通过将键盘中断处理程序的入口地址写入中断向量表，我确保了在键盘中断发生时，程序能够被正确地引导至中断服务例程。这一步骤对于中断的快速响应至关重要。  在中断处理程序中，我特别关注了四个内存映射的寄存器：KBSR（键盘状态寄存器）、KBDR（键盘数据寄存器）、DSR（显示状态寄存器）和DDR（数据方向寄存器）。这些寄存器的管理是实现中断处理的核心。   * KBSR 用于控制和检测键盘中断的状态，我通过设置IE位（中断使能位）来使能键盘中断。 * KBDR 存储了用户按下的键的数据，我在中断服务例程中读取这个寄存器来获取用户输入。 * DSR 用于检测显示是否准备好接收新数据，我在输出字符前检查这个寄存器，以确保不会在显示更新时覆盖现有内容。 * DDR 用于设置数据方向，我通过修改这个寄存器来控制数据的输入和输出。   我学会了如何在中断处理程序中保存和恢复这些寄存器的状态，以确保中断处理不会干扰主程序的执行。此外，我也意识到了在中断服务例程中不能使用 TRAP 指令的限制，这要求我必须使用内存映射寄存器来实现字符的显示和其他I/O操作。  通过这次实验，我不仅掌握了中断处理的基本概念和实现方法，还提高了我的汇编语言编程技能，特别是在寄存器管理和内存映射寄存器的使用上。这些经验对于我未来在更复杂的系统编程和操作系统开发中将非常有价值 |
| 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。