

（深圳）

实验报告

开课学期： 2024春季

课程名称：计算机组成原理（实验）

实验名称： 从C语言到机器码

实验性质： 设计型

实验学时： 4 地点： T2506

学生班级： 5 班

学生学号： 220110515

学生姓名： 金正达

作业成绩：

实验与创新实践教育中心制

2024年5月

|  |
| --- |
| 1、指令解析 |
| 要求：查看**可执行程序**的反汇编文件，在算术运算、移位运算、访存、分支跳转四种指令中，每种分别选出一条机器指令，并参照示例对它们进行解析。 |
| **示例：**  指令1：   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 机器码 | 16进制 | 2进制 | | | | | c3418513 | 1100 0011 0100 0001 1000 0101 0001 0011 | | | | | opcode | 001 0011 | funct3 | 000 | funct7 | - | | rd | 0101 0（x10/a0） | rs1 | 0001 1（x3/gp） | rs2 | - | | imm | 1100 0011 0100（-972的补码） | | | | | | 指令功能 | (a0) ← (gp) - 972 | | | | |   指令1：1018c: 00e787b3 add a5,a5,a4   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 机器码 | 16进制 | 2进制 | | | | | 00e787b3 | 0000000 01110 01111 000 01111 0110011 | | | | | opcode | 0110011 | funct3 | 000 | funct7 | 0000000 | | rd | 01111 | rs1 | 01111 | rs2 | 01110 | | imm | - | | | | | | 指令功能 | (a5) ← (a5) + (a4) | | | | |   指令2：102c4: 40295913 srai s2,s2,0x2   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 机器码 | 16进制 | 2进制 | | | | | 40295913 | 010000000010 10010 101 10010 0010011 | | | | | opcode | 0010011 | funct3 | 101 | funct7 | 0100000 | | rd | 10010 | rs1 | 10010 | rs2 | - | | imm | 00010（2） | | | | | | 指令功能 | (rd) ← (rd) >> 2 | | | | |   指令3：10150: fca42e23 sw a0,-36(s0)   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 机器码 | 16进制 | 2进制 | | | | | fca42e23 | 1111110 01010 01000 010 11100 0100011 | | | | | opcode | 0100011 | funct3 | 010 | funct7 | - | | rd | - | rs1 | 01000 | rs2 | 01010 | | imm | 111111011100(-36的补码) | | | | | | 指令功能 | Memory((rs1) - 36) ← (rs2) | | | | |   指令4：101b4: 00f71863 bne a4,a5,101c4 <find\_substr+0x80>   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 机器码 | 16进制 | 2进制 | | | | | 00f71863 | 0000000 01111 01110 001 10000 1100011 | | | | | opcode | 1100011 | funct3 | 001 | funct7 | - | | rd | - | rs1 | 01110 | rs2 | 01111 | | imm | 000000001000 | | | | | | 指令功能 | If ((rs1) != (rs2) ) pc ← pc + 16 | | | | | |

|  |
| --- |
| 2、汇编程序实验结果 |
| 要求：自行构造5个测试用例，对自行编写的字符串匹配汇编程序进行测试，并记录运行结果：  【用例1】母串：用任意字符将学号填充至16位；子串：学号后4位。  【用例2】母串包含子串的前几个字符，但不构成匹配。  【用例3】母串不包含子串，但将母串首尾衔接后包含子串。  【用例4】母串包含多个子串。  【用例5】母串包含子串的逆序串。 |
| **示例：**  【用例1】母串：3/j\_22011**0abc**$#% 子串：0abc  运行结果： |
| 【用例1】母串：fh2244o220110515cj()#) 子串：0515  运行结果： |
| 【用例2】母串：hgsdfabcdhefdj 子串：abcdef  运行结果： |
| 【用例3】母串：abcdefg 子串：ga  运行结果： |
| 【用例4】母串：absgabdhhabjk 子串：ab  运行结果： |
| 【用例5】母串：adhbag 子串：ab  运行结果： |

|  |
| --- |
| 3、思考与讨论 |
| （1）用自己的语言描述子程序的工作流程。  1.调用子程序：  主程序或其他子程序通过使用调用指令来调用子程序。调用指令会将子程序的入口地址压入堆栈，并跳转到子程序的起始地址执行。  2.保存上下文：  子程序开始时，通常会保存当前的寄存器状态，以便在子程序结束时恢复。该步骤通过将寄存器的值压入堆栈来完成。  3.执行子程序代码：  4.恢复上下文：  子程序在结束前，会将先前保存的寄存器状态恢复。该步骤通过从堆栈中弹出寄存器值来完成恢复。  5.返回到调用者：  子程序结束时，使用返回指令将控制权返回给调用它的代码。返回指令会从堆栈中弹出返回地址，并跳转到该地址继续执行。  （2）进入子程序时为何要保护现场，以及子程序返回前为何要恢复现场？  保护现场：  保存主程序的运行环境，防止子程序执行期间对其产生影响。  恢复现场：  将主程序的运行环境恢复到调用子程序之前的状态，以确保主程序能够正确继续执行。  （3）试对比分析编译器生成的汇编程序与自行编写的汇编程序各有什么异同。  异：   1. 编译器生成的代码结构较为复杂，包含了更多的跳转标签，且使用了更多的局部变量存储和加载操作。自行编写的代码结构相对简单，逻辑更为直观； 2. 编译器生成的代码在进入函数时保存了更多的上下文，在返回时进行恢复。这使得函数调用更为通用，但增加了开销。自行编写的代码通过宏定义简化了栈操作，只在必要时保存和恢复寄存器，更为高效； 3. 编译器生成的代码为了通用性和稳定性，包含了一些冗余操作和更多的寄存器保存与恢复。自行编写的代码减少了不必要的存储。   同：   1. 都实现了查找子串的功能，通过嵌套循环来匹配主字符串和子字符串中的字符。 2. 都使用了一些相同的寄存器，如a0, a1, a2等作为函数参数传递和返回值。 |