

# 计算机组成原理实验报告

## 13349051 劳嘉辉 计科一班

### 实验内容

使用 C++ 或者 C 语言生成一个程序。这个程序可以将 LC2K 汇编语言转换成机器码。

### 实验目的

更好地了解 LC2K 汇编语言以及其如何编码成机器码的过程。

### 代码显示

```
int  
readAndParse(FILE *inFilePtr, char *label, char *opcode, char *arg0,  
char *arg1, char *arg2)
```

在这个函数里面，先使用 `fgets` 读取一行指令，然后用 `strchr` 函数来判断是否超出了大小。接下来的步骤尤为关键。这里使用了 `sscanf(ptr, "%[^\\t\\n ]"` 判断是否有 `label`。这种做法能成功的原因有两个：1) 规范的输入格式 2) `[^\\t\\n]` 表示的是空格和 `tab` 都被忽略。如果有 `Label` 的话就会读入 `label` 这一个 `char*` 里，如果没有的话就会变空。

```
sscanf(ptr,  
"%*[\\t\\n ]%[^\\t\\n ]%*[\\t\\n ]%[^\\t\\n ]%*[\\t\\n ]%[^\\t\\n ]%*[\\t\\n ]%[^\\t\\n ]",  
opcode, arg0, arg1, arg2);
```

依次读入控制码，源 1 寄存器码，源 2 寄存器码和目标寄存器码。

R 型指令 (add, nand)

```
if (!strcmp(opcode, "add")) {  
    /* do whatever you need to do for opcode "add" */  
    fprintf(tempFilePtr, "%s", "0000000000");  
    fprintf(tempFilePtr, "%s", registernum(arg0));  
    fprintf(tempFilePtr, "%s", registernum(arg1));  
    fprintf(tempFilePtr, "%s", "00000000000000");  
    fprintf(tempFilePtr, "%s\\n", registernum(arg2));  
}
```

因为 R 型指令中的寄存器的地址都已经记录下来，所以只需要按照特定的格式输出即可。

## I 型指令

对于 lw,sw 来说

```
else if (!strcmp(opcode, "lw")){
    /* do whatever you need to do for opcode "lw" */
    fprintf(tempFilePtr, "%s", "0000000010");
    fprintf(tempFilePtr, "%s", registernum(arg0));
    fprintf(tempFilePtr, "%s", registernum(arg1));
    if (isNumber(arg2)){
        int num;
        sscanf(arg2, "%d", &num);
        if (num <= 32767 || num >= -32768){
            char* result = dectobin(num, 16);
            fprintf(tempFilePtr, "%s\n", result);
        }
    }
    else
    {
        printf("%s", "OffsetFields that don't fit in 16 bits");
        exit(1);
    }
}
else
{
    rewind(inFilePtr);
    char temp[MAXLINELENGTH];
    strcpy(temp, arg2);
    bool check = false;
    int labelnum = 0;
    int commandnum = 0;
    while (!feof(inFilePtr))
    {
        if (!readAndParse(inFilePtr, label, opcode, arg0, arg1,
arg2)) {
            /* reached end of file */
        }
        if (!strcmp(temp, label) && !check)
        {
            labelnum = commandnum;
            check = true;
        }
        else if (!strcmp(temp, label) && check)
        {
            printf("%s", "Duplicate labels are detected!");
            exit(1);
        }
    }
}
```

```

        commandnum++;
    }
    if (!check)
    {
        printf("%s", "Undefined labels are detected!");
        exit(1);
    }

    rewind(inFilePtr);
    for (int i = 0; i < programcontrol; i++)
        if (!readAndParse(inFilePtr, label, opcode, arg0, arg1, arg2))
        {
            /* reached end of file */
        }

    char* result = dectobin(labelnum, 16);
    fprintf(tempFilePtr, "%s\n", result);
}
}

```

如果 `offsetfield` 的值大于 32767 或者小于 -32768 的话，程序会通过 `exit(1)` 退出。如果 `offsetfield` 是一个数值，那么就会通过一个十进制转换二进制的函数将这个十进制数转换成二进制输出。如果是负数的话会以 16 位补码形式输出。如果 `offsetfield` 是一个 label，那么就会通过一个函数找出这个 Label 的所在的地址值，如果这个地址不存在或者重复了，就会通过 `exit(1)` 退出。如果成功找到了 label 所在的地址值，那么就将这个十进制值转换成二进制值输出。

对于 `beq` 来说，不同的一点在于处理 label 的时候，如果成功找到了 Label 所在的地址值，那么就会通过 `pc+1+offset=labelnum` 这个公式找出 `offset` 即偏移量是多少，再进行转换。

## J 型指令

```

else if (!strcmp(opcode, "jalr")) {
    /* do whatever you need to do for opcode "jalr" */
    fprintf(tempFilePtr, "%s", "0000000101");
    fprintf(tempFilePtr, "%s", registernum(arg0));
    fprintf(tempFilePtr, "%s", registernum(arg1));
    fprintf(tempFilePtr, "%s\n", "0000000000000000");
}
}

```

O 型指令

```
else if (!strcmp(opcode, "halt")) {
    /* do whatever you need to do for opcode "halt" */
    fprintf(tempFilePtr, "%s\n", "000000011000000000000000000000");
}
else if (!strcmp(opcode, "noop")) {
    /* do whatever you need to do for opcode "noop" */
    fprintf(tempFilePtr, "%s\n", "000000011100000000000000000000");
}
```

因为都有特定的格式，只需按照格式输出即可。

.fill

```
else if (!strcmp(opcode, ".fill")) {
    /* do whatever you need to do for direction ".fill" */
    if (isNumber(arg0))
    {
        int num;
        sscanf(arg0, "%d", &num);
        char* result = dectobin(num, 32);
        fprintf(tempFilePtr, "%s\n", result);
    }
    else
    {
        rewind(inFilePtr);
        char temp[MAXLINELENGTH];
        strcpy(temp, arg0);
        bool check = false;
        int labelnum = 0;
        int commandnum = 0;
        while (!feof(inFilePtr))
        {
            if (!readAndParse(inFilePtr, label, opcode, arg0, arg1,
arg2)) {
                /* reached end of file */
            }
            if (!strcmp(temp, label) && !check)
            {
                labelnum = commandnum;
                check = true;
            }
            else if (!strcmp(temp, label) && check)
            {
                printf("%s", "Duplicate labels are detected!");
                exit(1);
            }
        }
    }
}
```

```

        commandnum++;
    }
    if (!check)
    {
        printf("%s", "Undefined labels are detected!");
        exit(1);
    }

    rewind(inFilePtr);
    for (int i = 0; i < programcontrol; i++)
        if (!readAndParse(inFilePtr, label, opcode, arg0, arg1, arg2))
        {
            /* reached end of file */
        }

        char* result = dectobin(labelnum, 32);
        fprintf(tempFilePtr, "%s\n", result);
    }
}
else{
    printf("%s", "Unrecognized opcodes are detected!");
    printf("The unrecognized opcode is %s", opcode);
    exit(1);
}

    if (!readAndParse(inFilePtr, label, opcode, arg0, arg1, arg2)) {
        /* reached end of file */
    }
}
}

```

其实跟 lw 差不多，这里不作解释。

最后我还设置了两个函数 `bintohehex` 和 `bintodec`。这两个函数允许用户选择以十六进制或者十进制的形式看见机器码。

## 实验心得

这次实验收获很多。首先是对 c 语言又有了更近一步的了解。这个程序我全过程都是使用了 c 语言。刚开始感觉有点吃力，因为在输入方面例如 `sscanf` 和 `scanf` 这两个函数之前都没怎么接触，对输入的格式都不是特别地了解。其次是文件的操作跟 c++ 也有很大的区别。不过通过这次实验使我对 c 语言又有了更深的认识。当然了，这次实验也让我比较清楚地了解到 LC2K 转换成机器码的过程。这对以后的计算机组成原理的学习有十分大的好处。