

Lab6a-LC-3-Assembler-Report

郑浩博 3210105321

Algorithm explanation

Firstly, we should know that the **core of Lab6a** is deal with the "Label". Why? Because when we deal with the instruction such as `ADD NOT BR ...`, we can just read it, judge it and translate it into binary code which we write before in our c code.

But when it comes to "Label", we should know its row location.

Thus I think I could create a "*.txt" file (Couldn't create file in PTA, thus I use char array instead.) and when here comes "Label" offset in an instruction, I put it directly in file (array), while when here comes the instruction I translate it into binary code and put code in file (array). What's more, when meeting the Label before instruction, I would put it into a Label array, and put its row location into a Label location array:

```
void DoLabelpre(char* nowLabel);
void DoLabelafter(int n, char c);
char Label[MAXLABELNUMBER][MAXLABEL]; //存储label
int LabelRow[MAXLABELNUMBER]; //存储label对应的行数
int NowRow = 0;
char file[100000];
...
void DoLabelpre(char* nowLabel){
    static int labelnum=0;
    strcpy(Label[labelnum], nowLabel);
    LabelRow[labelnum]=NowRow;
    labelnum++;
}
...
void DoLabelafter(int n, char c){
    char flag;
    int i=1;
    char labeltodo[MAXLABEL];
    labeltodo[0]=c;
    labeltodo[1]='\0';
    UNUSE=scanf("%c",&flag);
    while(flag!='\n' && flag!=' '){
        labeltodo[i]=flag;
        labeltodo[i+1]='\0';
        i++;
        UNUSE=scanf("%c",&flag);
    }
    if(n==9){
        strcat(file, "9");
        strcat(file, labeltodo);
        strcat(file, "\n");
    }else{
        strcat(file, " ");
        strcat(file, labeltodo);
    }
}
```

```

        strcat(file, "\n");
    }
}

```

Using this method, I could deal with Label. When meeting the `.END` instruction, we end the reading and begin to deal with the file (array) . if the char we read from file (array) is `'0' or '1' or '\n'` , we print it directly. if the char we read from file (array) is a "Label", I would compare it with the Label in Label array one by one, and print the row offset between the location of Label and the row location now :

```

int main(){
    scanfInstruction();
    int i=0;
    char c;
    int nowrow=0;
    char nowlabel[MAXLABEL];
    while(file[i]!='\0'){//处理file数组, print出最终结果
        c=file[i];
        i++;
        if(c=='\n'){
            nowrow++;
        }
        if(c==' '){// 自己设计的label 的预处理标志
            for(int j=0;j<MAXLABEL;j++){
                nowlabel[j]=file[i+j];
                if(file[i+j]=='\n'){
                    nowlabel[j]='\0';
                    i+=j;
                    break;
                }
            }
        }
        for(int i=0;i<NowRow;i++){
            if(mycmp(Label[i],nowlabel)==0){
                Printf11(LabelRow[i]-nowrow-1);
                break;
            }
        }
    }else if(c=='9'){ // 自己设计的label 的预处理标志
        for(int j=0;j<MAXLABEL;j++){
            nowlabel[j]=file[i+j];
            if(file[i+j]=='\n'){
                nowlabel[j]='\0';
                i+=j;
                break;
            }
        }
    }
    for(int i=0;i<NowRow;i++){
        if(mycmp(Label[i],nowlabel)==0){
            Printf9(LabelRow[i]-nowrow-1);
            break;
        }
    }
}else { //如果是0 1 \n 直接输出
    printf("%c",c);
}

```

```

    }
}
return 0;
}

```

Now we complete most of our **LC-3 Assembler**, we just need deal with the simple instruction:

```

//字数限制，仅展示部分定义
void DoAnd(){
    char flag;
    int n;
    strcat(file,"0101");
    UNUSE=scanf(" ");
    ScanfReg();
    UNUSE=scanf(", ");
    ScanfReg();
    UNUSE=scanf(", ");
    UNUSE=scanf("%c",&flag);
    if(flag=='R' || flag=='r'){
        strcat(file,"000");
        UNUSE=scanf("%d",&n);
        PrintReg(n);
        strcat(file,"\n");
    }else if(flag=='#'){//flag == #
        strcat(file,"1");
        Scanfimm5();
    }else if(flag=='x'){
        strcat(file,"1");
        Scanfimmx(5);
    }else if(flag=='0'){
        strcat(file,"1");
        imm0(5);
    }
    NowRow ++;
}

void DoAdd();
void DoNot();
void DoLd(){
    char c;
    strcat(file,"0010");
    UNUSE=scanf(" ");
    ScanfReg();
    UNUSE=scanf(", ");
    UNUSE=scanf("%c",&c);
    if(c=='#'){
        Scanfimm(9);
    }else if(c=='0'){
        imm0(9);
    }else DoLabelafter(9,c);
    NowRow ++;
}

void DoLdr();
void DoLdi();
void DoLea();
void DoSt();

```

```

void DoStr();
void DoSti();
void DoTrap(){
    strcat(file,"11110000");
    int n=0;
    char c='0';
    UNUSE=scanf(" x");
    do{
        if(c>='a'&&c<='f'){
            n=n*16+c-87;
        }else if(c>='A'&&c<='F'){
            n=n*16+c-55;
        }else if(c>='0'&&c<='9'){
            n=n*16+c-48;
        }
        UNUSE=scanf("%c",&c);

    }while(c != '\n' && c!= ' ');
    char num[9]; //转换为二进制输出
    num[8]='\0';
    for(int i=7;i>=0;i--){
        num[i]=n%2+48;
        n /= 2;
    }
    strcat(file,num);
    strcat(file,"\n");
    NowRow ++;
}
void DoHalt();
void DoOrig();
void DoBr(int i);
void DoJmp();
void DoRet();
void DoJsr();
void DoJsrr();
void DoRti();
void DoFill();
void DoBlkw(){
    char c;
    int n;
    UNUSE=scanf(" ");
    UNUSE=scanf("%c",&c);
    if(c=='#'){
        UNUSE=scanf("%d",&n);
    }else if(c=='x'){
        c='0';
    }
    do{
        if(c>='a'&&c<='f'){
            n=n*16+c-87;
        }else if(c>='A'&&c<='F'){
            n=n*16+c-55;
        }else if(c>='0'&&c<='9'){
            n=n*16+c-48;
        }
        UNUSE=scanf("%c",&c);
    }
}

```

```

        }while(c != '\n' && c!= ' ');
    }else if(c=='0')n=0;
    for(int i=0;i<n;i++){
        strcat(file,"0111011101110111");
        strcat(file,"\n");
    }
    NowRow +=n;
}
void DoStringz();

```

And we would meet the conversion from decimal or hexadecimal to binary:

```

void Scanfimm5();
void Scanfimm6();
void scanfimm(int w){//读入立即数（十进制） 并输出5位(或6位)二进制
    int n;
    UNUSE=scanf("%d",&n);
    char num[w+1];
    num[w]='\0';
    if(n>=0){
        for(int i=w-1;i>=0;i--){
            num[i]=n%2+48;
            n /= 2;
        }
    }else {
        n=-n;
        for(int i=w-1;i>=0;i--){
            num[i]=n%2;
            num[i]=1-num[i];
            num[i]+=48;
            n /= 2;
        }
        for(int i=w-1;i>=0;i--){
            num[i]++;
            if(num[i]=='2'){
                num[i]='0';
            }else break;
        }
    }
    strcat(file,num);
    strcat(file,"\n");
}

void imm0(int w);
void scanfimmx(int w){//读入立即数（十六进制） 并输出5位(或6位)二进制
    int n=0;
    char c='0';
    int flag=0;
    UNUSE=scanf(" x");//读入十六进制数
    do{
        if(c=='-'){
            flag=1;
            UNUSE=scanf("%c",&c);
            continue;
        }
    }while(c != '\n' && c!= ' ');
    for(int i=0;i<n;i++){
        strcat(file,"0111011101110111");
        strcat(file,"\n");
    }
    NowRow +=n;
}
void DoStringz();

```

```

    }
    if(c>='a'&&c<='f'){
        n=n*16+c-87;
    }else if(c>='A'&&c<='F'){
        n=n*16+c-55;
    }else if(c>='0'&&c<='9'){
        n=n*16+c-48;
    }
    UNUSE=scanf("%c",&c);
}while(c != '\n' && c!= ' ');

char num[w+1];
num[w]='\0';
if(flag==1)n=-n;
if(n>=0){
    for(int i=w-1;i>=0;i--){
        num[i]=n%2+48;
        n /= 2;
    }
}else {
    n=-n;
    for(int i=w-1;i>=0;i--){
        num[i]=n%2;
        num[i]=1-num[i];
        num[i]+=48;
        n /= 2;
    }
    for(int i=w-1;i>=0;i--){
        num[i]++;
        if(num[i]=='2'){
            num[i]='0';
        }else break;
    }
}
strcat(file,num);
strcat(file,"\n");
}
void Printoff9(int off);
void Printoff11(int off);

```

Then we complete our **LC-3 Assembler** .