今天的任务

继续看代码，了解具体实现

怎么分离编译器和解释器为两个独立的程序

遇到的问题

1. fsys是什么，在哪里定义的，怎么使用
2. long型全局变量kk是什么，为什么在main函数中被初始化为al：

kk = al;

1. fprintf()没有执行成功
2. 输出的中间代码从序号101开始
3. interpret怎么读取文件内容
4. 输出的中间代码从序号2开始，前两条指令没有输入

解决的问题

1. 在main函数调用了block(declbegsys | statbegsys | period)，传到block函数的参数被重命名为fsys。即fsys初始时= declbegsys | statbegsys | period。在接下来使用中fsys不断被迭代，是合法的字符的集合
2. kk用来对齐当前标示符，al是define的一个常量，是标识符的最大长度。

main函数中初始化为al。

用途：防止上一个标识符存在a中的内容影响到当前标识符，比如上一个标识符为“qwerty”，现在的标识符为“abcd”，如果不清后几位则a中会保存"abcdty"，这显然是错误的

1. getch
2. getsym
3. main
4. 并不是fprintf的问题，而是在前面判断语句中没有成功创建outfile

错误：

|  |
| --- |
| if(outfile=fopen(outfilename,"w")==NULL){  printf("can't output the code");  exit(1);  } |

正确：

|  |
| --- |
| if(**(**outfile=fopen(outfilename,"w")**)**==NULL){  printf("can't output the code");  exit(1);  } |

一定要加括号，否则并不会执行而是纯粹判断。

1. 因为直接在listcode里创建文件并且关闭了所以每次调用block都是重新创建了文件并写入。所以要创建一个全局变量outfile。

|  |
| --- |
| if(**(**outfile=fopen(outfilename,"w")**)**==NULL){  printf("can't output the code");  exit(1);  } |

1. while没有遇到文件结束符

fscanf()得到的值赋给code[].f,l,a.

1. 改变listcode的位置，在全部执行完后再调用listcode从0开始输出

待解决的问题

1. block()
2. 中间代码的前两个指令在哪里输出的？应该是gen()里，第一条和第二条都是jmp

关键代码

|  |
| --- |
| void initialize(){  int no;  char tmp[80];  while(!feof(infile)){  fscanf(infile,"%s",tmp);//序号  no=atoi(tmp);    fscanf(infile,"%s",tmp);//code.f  //code[0].f=tmp;  if(!strcmp(tmp,"opr")){  code[no].f=opr;  //printf("f:%d\n",code[no].f);  }  else if(!strcmp(tmp,"lit")){  code[no].f=lit;  //printf("d:%d\n",code[no].f);  }  else if(!strcmp(tmp,"lod")){  code[no].f=lod;  }  else if(!strcmp(tmp,"sto")){  code[no].f=sto;  }  else if(!strcmp(tmp,"cal")){  code[no].f=cal;  }  else if(!strcmp(tmp,"int")){  //printf("jjj");  code[no].f=Int;  }  else if(!strcmp(tmp,"jmp")){  code[no].f=jmp;  }  else if(!strcmp(tmp,"jpc")){  code[no].f=jpc;  }  else{  break;  printf("error!\n");  }  fscanf(infile,"%s",tmp);//code.l  code[no].l=atoi(tmp);  //printf("code.l:%d\n",code[no].l);  //strcpy(code[0].f,tmp);  fscanf(infile,"%s",tmp);//code.a  code[no].a=atoi(tmp);  //printf("code.a:%d\n",code[no].a);  //printf("%s\n",tmp);  //memset(tmp,0,80);  }  //printf("code.a:%d\n",code[3].a);  } |

其他想法

编译器：把源程序翻译成中间语言程序 gen分析出中间语言

解释器：执行中间代码

编译器和解释器的接口是二进制的中间代码文件。编译器将中间代码文件导出，解释器接收中间代码文件。

思路：listcode将中间代码写入一个txt中，interpret读取这个文件并且执行其中的代码

明天的任务