

建立 OPTAB 前置作業：

```
opfile = fopen("opcode.txt","r");
if(opfile==NULL){
    printf("Open opcode.txt fail.\n");
    exit(0);
}
for(int i=0;i<MAX;i++){
    optable[i]=(operation*)malloc(sizeof(operation));
}
for(int i=0;i<MAX;i++){
    strcpy(optable[i]->op_name,"");
    strcpy(optable[i]->op_value,"");
}
char ch;
char line[100];

int o=0; //o:total optable num
char str[20]={0};
int i,j,k;
```

建立 OPTAB:

```
while(fgets(line, sizeof(line), opfile) != NULL){ //先把OPTAB建立起來
    i=0;
    while(line[i]!=' '){
        str[i]=line[i];
        i++;
    }
    strcpy(optable[o]->op_name,str);
    memset(str,0,sizeof(str));
    i++;
    k=0;
    while(line[i]!='\n'){
        str[k]=line[i];
        i++; k++;
    }
    strcpy(optable[o]->op_value,str);
    memset(str,0,sizeof(str));
    o++;
}
```

建立 srctable && SYMTAB 前置作業：

```

srcfile = fopen("source.txt","r");
if(srcfile==NULL){
    printf("Open source.txt fail.\n");
    exit(0);
}
for(i=0;i<MAX;i++){
    srctable[i]=(source*)malloc(sizeof(source));
}
for(i=0;i<MAX;i++){
    strcpy(srctable[i]->label,"");
    strcpy(srctable[i]->op,"");
    strcpy(srctable[i]->mem,"");
    strcpy(srctable[i]->code,"");
    strcpy(srctable[i]->locsix,"");
    srctable[i]->loc=0;
    srctable[i]->nextloc=0;
}
for(i=0;i<MAX;i++){
    symtable[i]=(sym*)malloc(sizeof(sym));
}
for(i=0;i<MAX;i++){
    strcpy(symtable[i]->label,"");
    strcpy(symtable[i]->address,"");
}

int s=0;        //s:total source code num
int sy=0;       //sy:total stmtable

```

將 src. txt 的內容先存起來：

```

while(fgets(line, sizeof(line), srcfile) != NULL){ //先將sourc.txt資料儲存起來
    i=0;

    while((line[i]>='a'&&line[i]<='z')||((line[i]>='A'&&line[i]<='Z'))){
        str[i]=line[i];
        i++;
    }
    printf("1:%s ",str);
    strcpy(srctable[s]->label,str);
    memset(str,0,sizeof(str));
    i++; k=0;
    while(!isalpha(line[i])){
        i++;
    }

    while((line[i]>='a'&&line[i]<='z')||((line[i]>='A'&&line[i]<='Z'))){
        str[k]=line[i];
        i++; k++;
    }
    printf("2:%s!!!\n",str);
    strcpy(srctable[s]->op,str);
    memset(str,0,sizeof(str));
    i++; k=0;
    while(line[i]==' '||line[i]=='\n'){
        i++;
    }

    while(line[i]!='\n'){
        str[k]=line[i];
        i++; k++;
    }
    printf("3:%s ",str);
    strcpy(srctable[s]->mem,str);
    memset(str,0,sizeof(str));
    printf("\n");
    s++;
}

```

開始 pass1~~找到 START 並且開始

```

//////////////////////////////////////////pass1-loc

long int loc;
char num_temp[10]={0};
char program_name[10]={0};
int startpassone=0;
while(strcmp(srctable[startpassone]->op,"START")){ //從START開始
    startpassone++;
}

```

將 START 這行的初始化資料記下來

```

strcpy(program_name,srctable[startpassone]->label); //記住program名稱、開始位置
strcpy(num_temp,srctable[startpassone]->mem);
int l = strlen(num_temp);

for(i=0;i<l;i++){
    int num=sixteentoten(num_temp[i]);
    loc += num*pow(16,l-i-1); //將位置轉16進位(方便算)
}

```

開始掃 src. txt:

End == true 就停下

一開始先把位置轉成 16 進位並存起來

```

for(i=startpassone+1;i<s && !end;i++){ //掃到end結束
    srctable[i]->loc=loc;
    char sadd[6]={0};
    int t;
    int loc_t=loc;
    for(int a=0;a<6;a++){ //每到新一行就把自己位置記下
        t=loc_t%16;
        sadd[5-a]=tentosixteen(t);
        loc_t=loc_t/16;
    }
    sadd[6]='\0'; //!!!!!!本次遇到最大問題:最好每個字串後方都加'\0'!!!!!!!!!!!!!!
    strcpy(srctable[i]->locsix,sadd);
    memset(sadd,0,sizeof(sadd));
}

```

這次遇到的最大問題~~~字串尾巴會莫名其妙多一堆奇怪的東西，要加一個'\0'才能預防= =

再來建立 SYMTAB:

```

if(strcmp("",srctable[i]->label)){ //動態建立SYMTAB
    printf("%s: %ld: %s\n",srctable[i]->label,loc,add);
    for(j=0;j<sy;j++){ //check error->重複定義了label位置
        if(!strcmp(symtable[j]->label,srctable[i]->label)){
            printf("Duplicate Symbol Error!\n");
            exit(0);
        }
    }
    strcpy(symtable[sy]->label,srctable[i]->label);
    strcpy(symtable[sy]->address,srctable[i]->locsix);
    sy++;
}

```

Label 處有東西就將其與其位置存至 SYMTAB

但若 SYMTAB 已經有了，代表它被重新定義，這是不被允許的，報錯並且 exit

如果 opcode 是 OPTAB 內的，loc+=3

```

for(j=0;j<o&&!end;j++){
    if(!strcmp(optable[j]->op_name,srctable[i]->op)){
        opexist=true;
        break;
    }
}

if(opexist){
    loc+=3;
}

```

WORD: +3

BYTE: C: loc+3    B: loc+長度/2

```

if(!strcmp("WORD",srctable[i]->op)){ // 非OPTAB內資料處理
    loc+=3;
}else if(!strcmp("BYTE",srctable[i]->op)){
    char tem=srctable[i]->mem[0];
    switch(tem){
        case 'C':
            loc+=3;
            break;
        case 'X':
            int b_len=strlen(srctable[i]->mem);
            b_len-=3;
            loc+=b_len/2;
            break;
    }
}

```

RESW RESB:掃進來的是 10 進位字串，數字 X (W:3 / B:1)

```
else if(!strcmp("RESW",srctable[i]->op)){
    int ll=strlen(srctable[i]->mem);
    int tt=0;
    for(int a=0;a<ll;a++){
        k=sixteentoten(srctable[i]->mem[a]);
        tt+=k*pow(10,ll-a-1);
    }
    loc+=tt*3;
else if(!strcmp("RESB",srctable[i]->op)){
    int ll=strlen(srctable[i]->mem);
    int tt=0;
    for(int a=0;a<ll;a++){
        k=sixteentoten(srctable[i]->mem[a]);
        tt+=k*pow(10,ll-a-1);
    }
    loc+=tt;
```

掃到 END，end=true

```
}else if(!strcmp("END",srctable[i]->op)){
    end=true;
```

Op 哪裡都找不到->報錯退出

```
}else{ //都不是->有未被定義->報錯
    printf("%s\n",srctable[i]->op);
    printf("Undefined Symbol Errors!\n"); //set error flag
    exit(0);
}
```

Pass1 結束，得到 loc 以及 SYMTAB

```
//////////pass1-finish
////show loc&symtable
printf("Finish pass1.\n");
printf("Location:\n");
for(i=0;i<s;i++){
    printf("%s\n",srctable[i]->locsix);
}
printf("\nSYMTAB:\n");
for(i=0;i<sy;i++){
    printf("%s: %s\n",symtable[i]->label,symtable[i]->address);
}
```

Pass2 開始:

開始撰寫 object file

```

startline=startpassone+1;
fputs("H",objectfile);
fputs(program_name,objectfile);
fputs(" ",objectfile);
fputs(srctable[startline]->locsix,objectfile);|
long int programlength = loc-firstloc;           //得到總長度並寫入第一行
char addpl[6];
int tpl;
for(int a=0;a<6;a++){
    tpl=programlength%16;
    addpl[5-a]=tentosixteen(tpl);
    programlength=programlength/16;
}
addpl[6]='\0';
fputs(addpl,objectfile);
fputs("\n",objectfile);

```

設置幾個暫存器

```

char textc[MAX]={0};           //暫存object file資料
int textl=0;                   //暫存目前存放了多少byte
char startlocsix[MAX]={0};     //存這次從哪個位置開始存放
bool space=true;              //避免RESW RESB導致開始位置是空的狀況

```

最簡單狀況~

```

while(strcmp(srctable[++i]->op,"END")){           //op=END結束
    opexist=false;
    if(space){
        strcpy(startlocsix,srctable[i]->locsix);
        space=false;
    }
    for(j=0;j<o;j++){                               //計算objectcode
        if(!strcmp(optable[j]->op_name,srctable[i]->op)){
            opexist=true;
            ob_code[0]=optable[j]->op_value[0]; ob_code[1]=optable[j]->op_value[1]; //OPTAB內找opvalue
            symexist=false;
            for(k=0;k<sy;k++){
                if(!strcmp(symtable[k]->label,srctable[i]->mem)){           //SYM內找mem位置
                    ob_code[2]=symtable[k]->address[2];
                    ob_code[3]=symtable[k]->address[3];
                    ob_code[4]=symtable[k]->address[4];
                    ob_code[5]=symtable[k]->address[5];
                    symexist=true;
                    break;
                }
            }
        }
    }
}

```

,X 情況

```

int ml = strlen(srctable[i]->mem);
if(srctable[i]->mem[ml-1]=='X'&&srctable[i]->mem[ml-2]==','){           //??,X 狀況
    char mm[ml-2];
    for(int a=0;a<ml-2;a++){
        mm[a]=srctable[i]->mem[a];
    }
    for(k=0;k<sy;k++){
        if(!strcmp(symtable[k]->label,mm)){
            char x=symtable[k]->address[2];
            int xx=(int)x-48;
            xx+=8;
            char xxx=tentosixteen(xx);
            ob_code[2]=xxx;
            ob_code[3]=symtable[k]->address[3];
            ob_code[4]=symtable[k]->address[4];
            ob_code[5]=symtable[k]->address[5];
            symexist=true;
            break;
        }
    }
}
}

```

要針對 obcode[2]做處理，因為 X 是這位最高位元，所以要+  $2^3$



都不是||空的 -> 填 0

```
if(!symexist){
    ob_code[2]='0';
    ob_code[3]='0';
    ob_code[4]='0';
    ob_code[5]='0';
}
```

存起來

```
ob_code[6]='\0';
strcpy(srctable[i]->code,ob_code);
```

撰寫 object file 情況，後面多次重覆就不再提起

```
if(textl+3<=30){ //以最多30bytes一行為基準
    strcat(textc,ob_code);
    textl+=3;
}else{
    char plq[2]; //滿了就填到object file內並且清空暫存
    int ttlq;
    for(int a=0;a<2;a++){
        ttlq=textl%16;
        plq[1-a]=tentosixteen(ttlq);
        textl=textl/16;
    }
    plq[2]='\0';
    fputs("T",objectfile);
    fputs(startlocsix,objectfile);
    memset(startlocsix,0,sizeof(startlocsix));
    strcpy(startlocsix,srctable[i]->locsix);
    fputs(plq,objectfile);
    fputs(textc,objectfile);
    fputs("\n",objectfile);
    memset(textc,0,sizeof(textc));
    textl=3;
    strcat(textc,ob_code);
}
memset(ob_code,0,sizeof(ob_code));
break;
```

## WORD 情況

```
int wbl;  
if(!opexist){  
    if(!strcmp("WORD",srctable[i]->op)){  
        wbl=strlen(srctable[i]->mem);  
        int loc_t=0;  
        for(int b=0;b<wbl;b++){  
            loc_t+=(((int)(srctable[i]->mem[b])-48)*pow(10,wbl-b-1));  
        }  
        char addd[6]={0};  
        int t;  
        for(int a=0;a<6;a++){  
            t=loc_t%16;  
            addd[5-a]=tentosixteen(t);  
            loc_t=loc_t/16;  
        }  
        addd[6]='\0';  
        strcpy(srctable[i]->code,addd);  
        memset(add,0,sizeof(add));  
    }  
}
```

讀入的 mem 為 10 進位字串，先轉成數字，再轉成 16 進位字串，其字串就是 obcode

## BYTE:C 狀況

```
else if(!strcmp("BYTE",srctable[i]->op)){  
    char tem=srctable[i]->mem[0];  
    int cx;  
    switch(tem){  
        case 'C':  
            wbl=strlen(srctable[i]->mem);  
            cx=wbl-3;  
            for(int a=0;a<cx;a++){  
                int cinascii = (int)srctable[i]->mem[a+2];  
                char asciiinsixteen[2]={0};  
                int t;  
                for(int aa=0;aa<2;aa++){  
                    t=cinascii%16;  
                    asciiinsixteen[1-aa]=tentosixteen(t);  
                    cinascii=cinascii/16;  
                }  
                asciiinsixteen[2]='\0';  
                srctable[i]->code[a*2]= asciiinsixteen[0]; srctable[i]->code[a*2+1]= asciiinsixteen[1];  
                memset(asciiinsixteen,0,sizeof(asciiinsixteen));  
            }  
            /////  
        }  
    }
```

其內容(字串)即為 obcode，其 obcode 是各個字元的 ASCII code 再轉成 16 進位

## BYTE:X 狀況

```
case 'X':  
    wbl=strlen(srctable[i]->mem);  
    cx=wbl-3;  
    char xadd[2]={0};  
    for(int a=0;a<cx;a++){  
        xadd[a]=srctable[i]->mem[a+2];  
    }  
    xadd[2]='\0';  
    strcpy(srctable[i]->code,xadd);  
    memset(xadd,0,sizeof(xadd));  
}
```



其內容直接為 obcode

RESW/RESB 狀況

```
}else if(!strcmp("RESW",srctable[i]->op)||!strcmp("RESB",srctable[i]->op)){
    space=true;
    if(textl!=0){
        char pl[2];
        int ttl;
        for(int a=0;a<2;a++){
            ttl=textl%16;
            pl[1-a]=tentosixteen(ttl);
            textl=textl/16;
        }

        pl[2]='\0';
        fputs("T",objectfile);
        fputs(startlocsix,objectfile);
        memset(startlocsix,0,sizeof(startlocsix));
        fputs(pl,objectfile);
        fputs(textc,objectfile);
        fputs("\n",objectfile);
        memset(textc,0,sizeof(textc));
        textl=0;
    }
}
```

遇到就直接把目前暫存器內的丟到 file 內並清空，space 設 true(開始位置直到非 RESW/RESB 都要一直變)

END~~~~把目前存著的最後丟到 file 內，就結束啦!

```
char pl[2];
int ttl;
for(int a=0;a<2;a++){
    ttl=textl%16;
    pl[1-a]=tentosixteen(ttl);
    textl=textl/16;
}

pl[2]='\0';
fputs("T",objectfile);
fputs(startlocsix,objectfile);
memset(startlocsix,0,sizeof(startlocsix));
space=true;
fputs(pl,objectfile);
fputs(textc,objectfile);
fputs("\n",objectfile);
//////////
fputs("E",objectfile);
fputs(srctable[startline]->locsix,objectfile);

printf("\nFinish pass2.\n");
printf("src program with loc && obcode:\n");
printf("\nloc  label  op  mem obcode\n");
for(i=0;i<s;i++){
    printf("%s  %s  %s  %s\n",srctable[i]->locsix,srctable[i]->label,srctable[i]->op,srctable[i]->mem,srctable[i]->code);
}

fclose(objectfile);
```