

## 4109056001 施昶宇-期末 project

Pass 1

### 1. Intermediate file

Loc	Source statement		
1000	COPY	START	1000
1000	FIRST	STL	RETADR
1003	CLOOP	JSUB	RDREC
1006		LDA	LENGTH
1009		COMP	ZERO
100C		JEQ	ENDFIL
100F		JSUB	WRREC
1012		J	CLOOP
1015	ENDFIL	LDA	EOF
1018		STA	BUFFER
101B		LDA	THREE
101E		STA	LENGTH
1021		JSUB	WRREC
1024		LDL	RETADR
1027		RSUB	
102A	EOF	BYTE	C'EOF'
102D	THREE	WORD	3
1030	ZERO	WORD	0
1033	RETADR	RESW	1
1036	LENGTH	RESW	1
1039	BUFFER	RESB	4096

2039	RDREC	LDX	ZERO
203C		LDA	ZERO
203F	RLOOP	TD	INPUT
2042		JEQ	RLOOP
2045		RD	INPUT
2048		COMP	ZERO
204B		JEQ	EXIT
204E		STCH	BUFFER,X
2051		TIX	MAXLEN
2054		JLT	RLOOP
2057	EXIT	STX	LENGTH
205A		RSUB	
205D	INPUT	BYTE	X'F1'
205E	MAXLEN	WORD	4096
2061	WRREC	LDX	ZERO
2064	WLOOP	TD	OUTPUT
2067		JEQ	WLOOP
206A		LDCH	BUFFER,X
206D		WD	OUTPUT
2070		TIX	LENGTH
2073		JLT	WLOOP
2076		RSUB	
2079	OUTPUT	BYTE	X'05'
		END	FIRST

## 2. Symbol table

Label Name	Address
FIRST	1000
CLOOP	1003
ENDFIL	1015
EOF	102A
THREE	102D
ZERO	1030
RETADR	1033
LENGTH	1036
BUFFER	1039
RDREC	2039
RLOOP	203F
EXIT	2057
INPUT	205D
MAXLEN	205E
WRREC	2061
WLOOP	2064
OUTPUT	2079

## Pass 2

### 1. source program

Loc	Source statement			Object code
1000	COPY	START	1000	
1000	FIRST	STL	RETADR	141033
1003	CLOOP	JSUB	RDREC	482039
1006		LDA	LENGTH	001036
1009		COMP	ZERO	281030
100C		JEQ	ENDFIL	301015
100F		JSUB	WRREC	482061
1012		J	CLOOP	3C1003
1015	ENDFIL	LDA	EOF	00102A
1018		STA	BUFFER	0C1039
101B		LDA	THREE	00102D
101E		STA	LENGTH	0C1036
1021		JSUB	WRREC	482061
1024		LDL	RETADR	081033
1027		RSUB		4C0000
102A	EOF	BYTE	C'EOF'	454F46
102D	THREE	WORD	3	000003
1030	ZERO	WORD	0	000000
1033	RETADR	RESW	1	
1036	LENGTH	RESW	1	
1039	BUFFER	RESB	4096	

2039	RDREC	LDX	ZERO	041030
203C		LDA	ZERO	001030
203F	RLOOP	TD	INPUT	E0205D
2042		JEQ	RLOOP	30203F
2045		RD	INPUT	D8205D
2048		COMP	ZERO	281030
204B		JEQ	EXIT	302057
204E		STCH	BUFFER,X	549039
2051		TIX	MAXLEN	2C205E
2054		JLT	RLOOP	38203F
2057	EXIT	STX	LENGTH	101036
205A		RSUB		4C0000
205D	INPUT	BYTE	X'F1'	F1
205E	MAXLEN	WORD	4096	001000
2061	WRREC	LDX	ZERO	041030
2064	WLOOP	TD	OUTPUT	E02079
2067		JEQ	WLOOP	302064
206A		LDCH	BUFFER,X	509039
206D		WD	OUTPUT	DC2079
2070		TIX	LENGTH	2C1036
2073		JLT	WLOOP	382064
2076		RSUB		4C0000
2079	OUTPUT	BYTE	X'05'	05
		END	FIRST	

## 2. final object program

---

```

HCOPY 00100000107A
T0010001E1410334820390010362810303010154820613C100300102A0C103900102D
T00101E150C10364820610810334C0000454F46000003000000
T0020391E041030001030E0205D30203FD8205D2810303020575490392C205E38203F
T0020571C1010364C0000F1001000041030E02079302064509039DC20792C1036
T002073073820644C000005
E001000

```

## 流程說明：

使用兩個串列連結分別存取 opcode 和 source

```
/**
 *
 */
typedef struct opcode *opnode;
typedef struct opcode{
    char name[10];
    char number[2];
    opnode link;
}opcode;
opnode opcode_root=NULL;
/**
 *
 */
typedef struct source *source_node;
typedef struct source{
    char location[5];
    char labelname[10];
    char opcode[10];
    char name[10];
    char object_code[6];
    source_node link;
}source;
source_node source_root=NULL;
/**
 *
 */
```

先讀取 opcoed. txt 存入 struct opcode

```
while(fscanf(fp1,"%s%s",&name,&number)!=EOF)
    opcode_creat(name,number);
```

```

void opcode_creat(char name[],char number[]){
    opnode new_one=(opnode)malloc(sizeof(opcode));
    strcpy(new_one->name,name);
    strcpy(new_one->number,number);
    new_one->link=NULL;
    if(!opcode_root){
        opcode_root=new_one;
        return;
    }
    opnode current=opcode_root;
    while(current->link!=NULL)
        current=current->link;
    current->link=new_one;
}

```

接著讀取 source

```

while(fscanf(fp2,"%[^\\n] ",&arr)!=EOF){//cut source
    cut=0,count=0,d=0;
    reset(name);
    reset(op);
    reset(c);
    for(i=0;i<strlen(arr);i++){
        if(arr[i]==' '){
            reset(arr1);
            strncpy(arr1,arr+cut,i-cut);
            if(check(arr1)){
                strcpy(op,arr1);
                d=1;//opcode
            }
            else
                strcpy(name,arr1);
            cut=i+1;
            count++;
        }
        if(count==2||d==1){
            strncpy(arr1,arr+cut,strlen(arr)-i);
            strcpy(c,arr1);
            break;
        }
    }
    if(count==0)
        strncpy(op,arr,strlen(arr));
    source_creat(name,op,c);
}

```

一次讀取一行，並將字串切割成三個部分，中間以 tab 做分割，將切的第一個字串和 opcode 相比，如果一樣代表 labelname(source 第一行) 是空的。

```
int check(char name[]){
    opnode current=opcode_root;
    while(current){//if name==opcode return 1
        if(!strcmp(current->name,name)||!strcmp(name,"START")
            ||!strcmp(name,"RSUB")||!strcmp(name,"BYTE")||!strcmp(name,"WORD")
            ||!strcmp(name,"RESB")||!strcmp(name,"RESW")||!strcmp(name,"END"))
            return 1;
        current=current->link;
    }
    return 0;
}
/*****/
```

存入 struct source

```
void source_creat(char labelname[],char opcode[],char name[]){
    source_node new_one=(source_node)malloc(sizeof(source));
    strcpy(new_one->location,"0000");
    strcpy(new_one->object_code,"000000");
    strcpy(new_one->labelname,labelname);
    strcpy(new_one->opcode,opcode);
    strcpy(new_one->name,name);
    new_one->link=NULL;
    if(!source_root){
        source_root=new_one;
        return;
    }
    source_node current=source_root;
    while(current->link!=NULL)
        current=current->link;
    current->link=new_one;
}
/*****/
```



## Pass 1:

1. 尋找 “START” ，並將一開始的 location 設為 “START” 的起始位置。

```
while(current->link!=NULL){
    if(!strcmp(current->opcode,"START")){//opcode==START
        strcpy(current->location,current->name);
        strcpy(current->link->location,current->name);
        strcpy(num,current->name);
        current=current->link;
        break;
    }
    current=current->link;
}
```

2. 如果 opcode 是 “RESB” , location 加上當前列的 operand(轉 16 進位)。

```
if(!strcmp(current->opcode,"RESB")){//opcode==RESB
    dectohex(hextodec(num)+3,hex);
    strcpy(current->location,hex);
    dectohex(atoi(current->name),hex);
    dectohex((hextodec(current->location)+hextodec(hex)-3),hex);
    strcpy(num,hex);//1039
}
```

3. 如果 opcode 是 “RESW” , location 加上當前列的 operand\*3(轉 16 進位) , 因為 WORD 3 個 BYTE 。

```
else if(!strcmp(current->opcode,"RESW")){//opcode==RESW
    dectohex(hextodec(num)+3,hex);
    strcpy(current->location,hex);
    dectohex(atoi(current->name),hex);
    dectohex((hextodec(current->location)+atoi(current->name)*3-3),hex);
    strcpy(num,hex);
}
```

4. 如果 opcode 是 “BYTE” , LOCCTR 要加的數量, 取決於  
OPERAND 的長度和型態。

如果是十六進位數字(X), 則兩個數字一個 byte

如果是字元(C), 則一個字元一個 byte。

```
else if(!strcmp(current->opcode,"BYTE")){//opcode==BYTE
    dectohex(hextodec(num)+3,hex);
    strcpy(current->location,hex);
    if(current->name[0]=='C'){
        dectohex((hextodec(current->location)+strlen(current->name)-6),hex);
    }
    else{
        dectohex((hextodec(current->location)+(strlen(current->name)/2)-4),hex);
    }
    strcpy(num,hex);
}
```

5. 其它 location 直接+3。

```
else{
    dectohex(hextodec(num)+3,hex);
    strcpy(current->location,hex);
    strcpy(num,current->location);
}
```

6. 直到讀到 “END” 為止。
7. 將檔案寫入 `pass.txt`。

```
void show_source(){
    source_node current=source_root;
    pass1=fopen("pass1.txt","w");
    while(current!=NULL){
        fprintf(pass1,"%s ",current->location);
        fprintf(pass1,"%s %s %s\n",current->labelname,current->opcode,current->name);
        current=current->link;
    }
}
/*****/
```

8. 當 `labelname(source 第一行)` 長度不為 0 時，將 `labelname(source 第一行)` 和 `locatoin` 寫入 `symbol.txt`，直到結尾。

```
void symbol_table(){
    symbol=fopen("symbol.txt","w");
    fprintf(symbol,"%s %s\n","Label Name","Address");
    source_node current=source_root;
    while(current){
        if(!strcmp(current->opcode,"START")){
            current=current->link;
            break;
        }
        current=current->link;
    }
    while(current){
        if(strlen(current->labelname)!=0)
            fprintf(symbol,"%s %s\n",current->labelname,current->location);
        current=current->link;
    }
}
/*****/
```

9. `pass1` 結束。

## Pass 2:

1. 先將 object code 的前兩位數值填入相對應的 opcode，沒有的設” 0”。

```
source_node current=source_root;
while(current){
    if(!strcmp(current->opcode,"START")){//find STRAT
        current=current->link;
        break;
    }
    current=current->link;
}
while(current){
    opnode ptr=opcode_root;
    while(ptr){
        if(!strcmp(ptr->name,current->opcode)){
            current->object_code[0]=ptr->number[0];
            current->object_code[1]=ptr->number[1];
            break;
        }
        ptr=ptr->link;
    }
    current=current->link;
}
```

2. 當 name(source 的第三行)的長度為 0 時，不用改變，直接下一列。

```
if(strlen(current->name)==0){
    current=current->link;
    continue;
}
```

3. 當 opcode 是 “RESW” 或者是 “RESB” ，清空 object code，什麼都不用做。

```
else if(!strcmp(current->opcode,"RESW")||!strcmp(current->opcode,"RESB")){//
    reset(current->object_code);
}
```

4. 當 opcode 是 “ ,X” 時，將當前的 location 轉成二進位，並將最前面的原設為 1，最後再轉回 16 進位，儲存到 object code 裡。

```
else if(current->name[strlen(current->name)-1]=='X'&&current->name[strlen(current->name)-2]==','){
    char hex[4],arr1[10];
    int j=5;
    strncpy(arr1,current->name,strlen(current->name)-2);
    while(cur){
        if(!strcmp(arr1,cur->labelname))
            break;
        cur=cur->link;
    }
    int n=hextodec(cur->location)+4096*8;//opcode + x + address(15bit)
    while(n>65536){
        n-=32768;
    }
    dectohex(n,hex);
    current->object_code[j--]=hex[3];
    current->object_code[j--]=hex[2];
    current->object_code[j--]=hex[1];
    current->object_code[j--]=hex[0];
}
```

5. 當 opcode 是 “BYTE” 且 name (source 的第三行) 的第 0 個字元為 ’ X’ 時，清空當前 object code，直接將 ’ ’ 裡的字元由最左邊填入。

```
else if((!strcmp(current->opcode,"BYTE"))&&current->name[0]=='X'){
    reset(current->object_code);
    int i,j=0;
    for(i=2;i<=strlen(current->name)-2;i++){
        current->object_code[j++]=current->name[i];
    }
}
```

6. 當 opcode 是 “BYTE” 且 name (source 的第三行) 的第 0 個字元為 ' C ' 時，清空當前 object code，直接將 ' ' 裡的字元轉成 ASCII 碼(10 進位)由最左邊填入。

```
else if(!strcmp(current->opcode,"BYTE"))&&current->name[0]=='C'){
    reset(current->object_code);
    int i,j=0,num;
    char hex[4];
    for(i=2;i<=strlen(current->name)-2;i++){
        dectohex(current->name[i],hex);
        current->object_code[j++]=hex[2];
        current->object_code[j++]=hex[3];
    }
}
```

7. 當 opcode 是 “WORD” 時，將 name (source 的第三行) 轉成 16 進位，存入 object code。

```
else if(!strcmp(current->opcode,"WORD")){
    int j=5;
    char hex[4];
    dectohex(atoi(current->name),hex);
    current->object_code[j--]=hex[3];
    current->object_code[j--]=hex[2];
    current->object_code[j--]=hex[1];
    current->object_code[j--]=hex[0];
}
```



8. 其它的尋找到對應的 labelname(source 的第一行)，直接存入即可。

```
else{
    while(cur){
        if(!strcmp(cur->labelname,current->name)){
            current->object_code[2]=cur->location[0];
            current->object_code[3]=cur->location[1];
            current->object_code[4]=cur->location[2];
            current->object_code[5]=cur->location[3];
            break;
        }
        cur=cur->link;
    }
}
```

9. 重複 2-8 直到結尾。
10. 將檔案寫入 `pass2.txt`。

```
void pass2_show(){
    pass2=fopen("pass2.txt","w");
    source_node current=source_root;
    while(current!=NULL){
        fprintf(pass2,"%s  ",current->location);
        fprintf(pass2,"%s      %s  %-20s  %s\n",current->labelname,current->opcode
,current->name,current->object_code);
        current=current->link;
    }
}
```

11. 寫入” H” 、程式名稱、開始位置寫入 `record.txt` 。

```
while(current){
    if(!strcmp(current->opcode,"START")){
        for(i=0;i<strlen(current->labelname);i++){
            name[i]=current->labelname[i];
        }
        for(i=strlen(current->name)-1;i>=0;i--){
            length[j--]=current->name[i];
        }
        fprintf(record,"%c%s00%s", 'H', name, length);
        strcpy(name, "000000");
        for(i=3;i>=0;i--){
            name[i+2]=length[i];
        }
        break;
    }
    current=current->link;
}
```

12. 計算總長度，寫入 `record.txt` 。

```
while(strcmp(current->link->opcode,"END")){
    current=current->link;
    hexdec(current->location)-hexdec(length)+strlen(current->object_code)/2;//count total length
    dectohex(hexdec(current->location)-hexdec(length)+strlen(current->object_code)/2,length);
    fprintf(record,"00%s\n",length);
    current=source_root;
    while(current){
        if(!strcmp(current->opcode,"START")){
            current=current->link;
            break;
        }
        current=current->link;
    }
}
```

13. 計算一行可放的長度，一行最多 10 個，遇到 “RESW” 直接換行，寫入 `record.txt`，直到 “END” 結束。

```
for(i=0;i<10;i++){//count number
    if(!strcmp(ptr->opcode,"RESW")||!strcmp(ptr->link->opcode,"END")){
        if(!strcmp(ptr->link->opcode,"END")){
            j++;
            break;
        }
    }
    object_len+=strlen(ptr->object_code);//count object_sum
    j=i;
    ptr=ptr->link;
}
```



14. 寫入 end record

```
fprintf(record, "\n%c%s", 'E', name);
```

15. 結束 pass2

備註：我寫的兩個函式

1. 將十六進位的字串轉成十進位的數字。

```
int hextodec(char num[]){
    int index=0,i,count=1,temp;
    for(i=strlen(num)-1;i>=0;i--){
        switch(num[i]){
            case 'A': temp=10; break;
            case 'B': temp=11; break;
            case 'C': temp=12; break;
            case 'D': temp=13; break;
            case 'E': temp=14; break;
            case 'F': temp=15; break;
            default :temp=num[i]-'0'; break;
        }
        index+=temp*count;
        count*=16;
    }
    return index;
}
```

2. 將十進位的字串轉成十六進位的數字。

```
void dectohex(int num, char hex[]){
    int a=num, i=3;
    strcpy(hex, "0000");
    while(num){
        a=num%16;
        switch(a){
            case 0: hex[i]='0'; break;
            case 1: hex[i]='1'; break;
            case 2: hex[i]='2'; break;
            case 3: hex[i]='3'; break;
            case 4: hex[i]='4'; break;
            case 5: hex[i]='5'; break;
            case 6: hex[i]='6'; break;
            case 7: hex[i]='7'; break;
            case 8: hex[i]='8'; break;
            case 9: hex[i]='9'; break;
            case 10: hex[i]='A'; break;
            case 11: hex[i]='B'; break;
            case 12: hex[i]='C'; break;
            case 13: hex[i]='D'; break;
            case 14: hex[i]='E'; break;
            case 15: hex[i]='F'; break;
            default: break;
        }
        i--;
        num/=16;
    }
}
```

加分題：

這支(4109056001 施昶宇-期末加分題 project)程式可以處理 XIC/XE pass1 的部分。

Loc	Source statement		
0000	COPY	START	0
0000	FIRST	STL	RETADR
0003		LDB	#LENGTH
		BASE	LENGTH
0006	CLOOP	+JSUB	RDREC
000A		LDA	LENGTH
000D		COMP	#0
0010		JEQ	ENDFIL
0013		+JSUB	WRREC
0017		J	CLOOP
001A	ENDFIL	LDA	EOF
001D		STA	BUFFER
0020		LDA	#3
0023		STA	LENGTH
0026		+JSUB	WRREC
002A		J	@RETADR
002D	EOF	BYTE	C'EOF'
0030	RETADR	RESW	1
0033	LENGTH	RESW	1
0036	BUFFER	RESB	4096
1036	RDREC	CLEAR	X
1038		CLEAR	A
103A		CLEAR	S
103C		+LDT	#4096
1040	RLOOP	TD	INPUT
1043		JEQ	RLOOP

1046		RD	INPUT
1049		COMPR	A,S
104B		JEQ	EXIT
104E		STCH	BUFFER,X
1051		TIXR	T
1053		JLT	RLOOP
1056	EXIT	STX	LENGTH
1059		RSUB	
105C	INPUT	BYTE	X'F1'
105D	WRREC	CLEAR	X
105F		LDT	LENGTH
1062	WLOOP	TD	OUTPUT
1065		JEQ	WLOOP
1068		LDCH	BUFFER,X
106B		WD	OUTPUT
106E		TIXR	T
1070		JLT	WLOOP
1073		RSUB	
1076	OUTPUT	BYTE	X'05'
		END	FIRST