系統程式期末筆記

第一到四周

楊淞元

110710502 | 資工二

内容

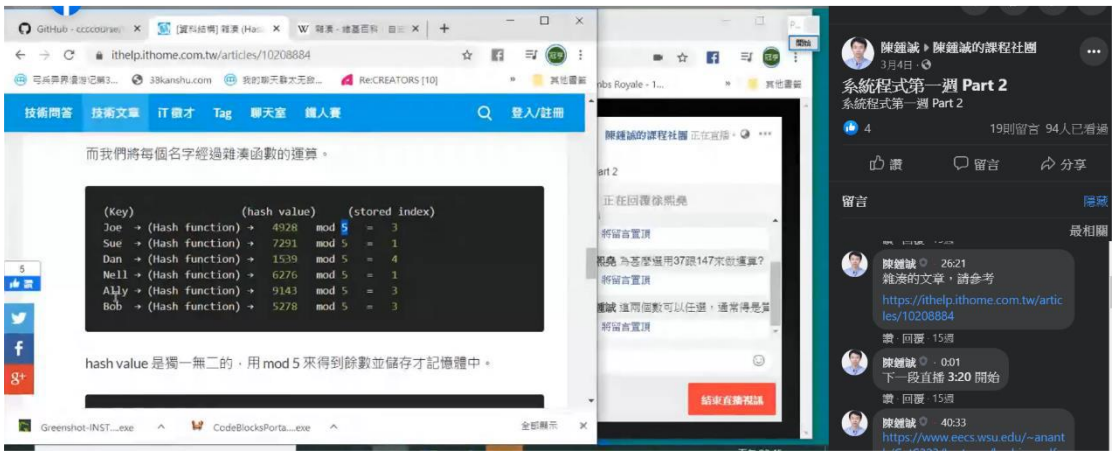
第一周

%u 無號數　　%d 有號數

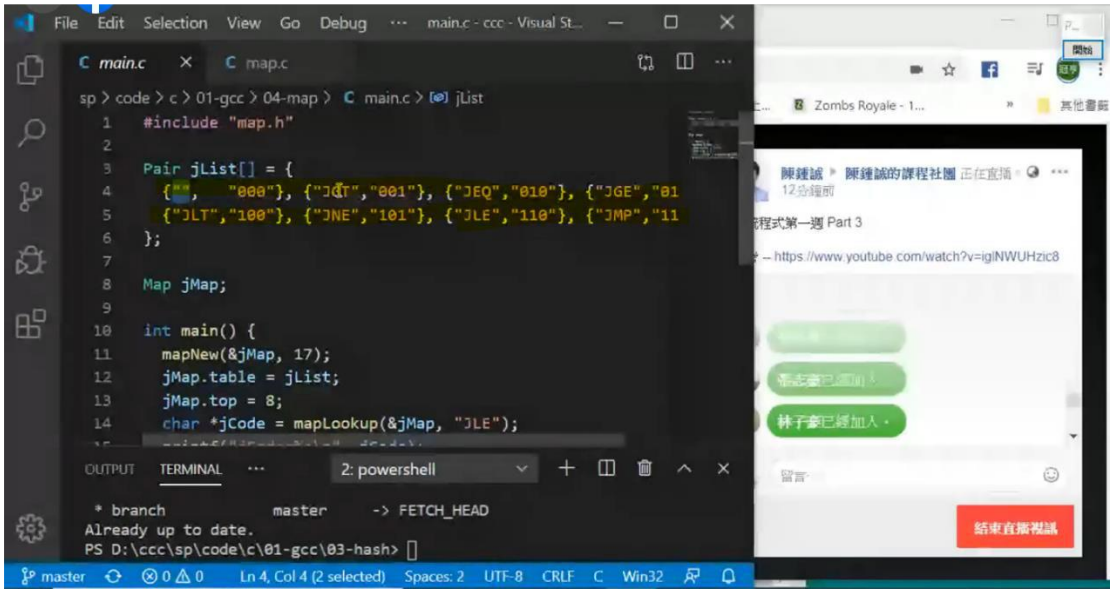
hash 雜湊:把一個字串用固定方式轉成同樣數字

雜湊的文章，請參考

https://ithelp.ithome.com.tw/articles/10208884



程式碼:



argv 參數變數　　argc 參數個數

parse

nextTemp( )產生下一個臨時變數的代號

ch( )取得下一個字元

next( )取得字元，同時進到下一格

isNext( )判斷下一個字是不是”+”或”-”

puts=printf 輸出　　*----->0 次以上

git pull origin master　ASSIGN 指定

lex 詞彙解析

map 查詢 循序搜尋　mapDump 輸出

struct 結構(裡面可有很多欄位)

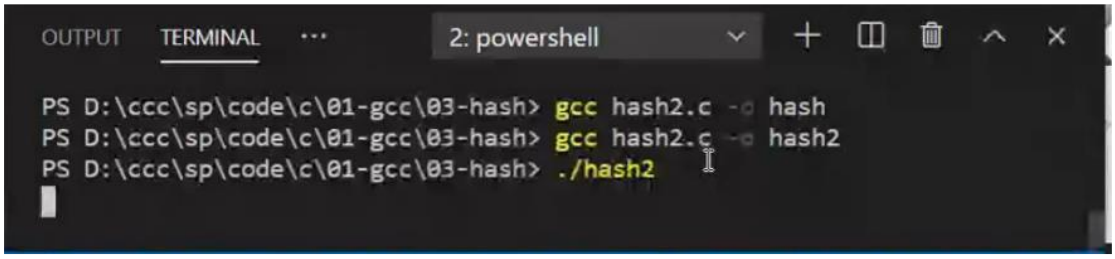
hashMap 尋找雜湊函數的值

mapAdd 新增一筆資料

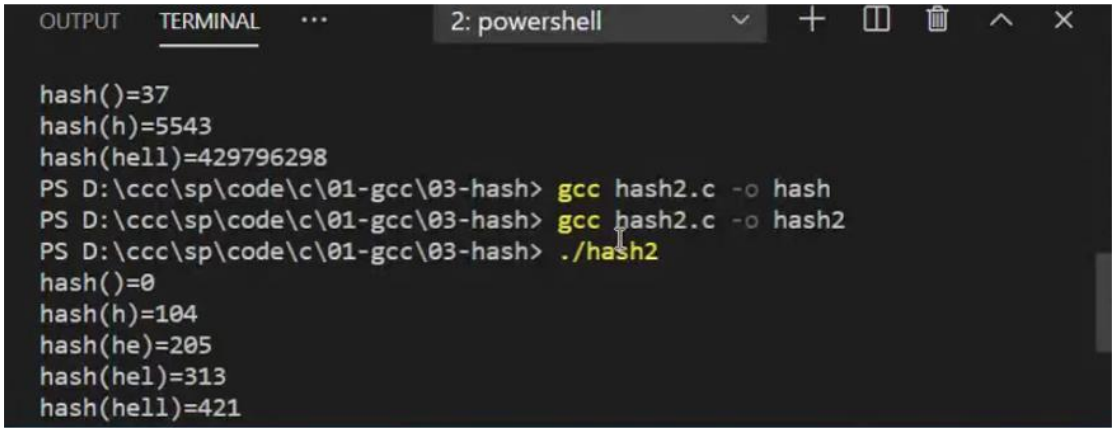
mapFind 找到資料

hash2.c

指令



結果



箭頭是用來《存取指標指向的結構內之欄位》

，例如

typedef struct _Map {

Pair *table;

int size;

int top;

} Map;

Map* mapNew(Map *map, int size) {

map-

>table = N

ULL;

map-

>size = size;

map-

>top = 0;

return map;

}

其中

map

是個結構指標，箭頭就可以用來存取其中的欄位

size, table, top

補充:

struct

是

structure

的短寫，也就是結構

struct

裡面可以有很多欄位，每個欄位都有名字和型態。

```
typedef struct _Pair {
char *key;
void *value;
} Pair;
typedef struct _Map {
Pair *table;
int size;
int top;
} Map;
```

補充:

```
Pair jList[] = {
{"", "000"}, {"JGT","001"}, {"JEQ","010"}, {"JGE","011"},
{"JLT","100"}, {"JNE","101"}, {"JLE","110"}, {"JMP","111"}
};
Map jMap;
```

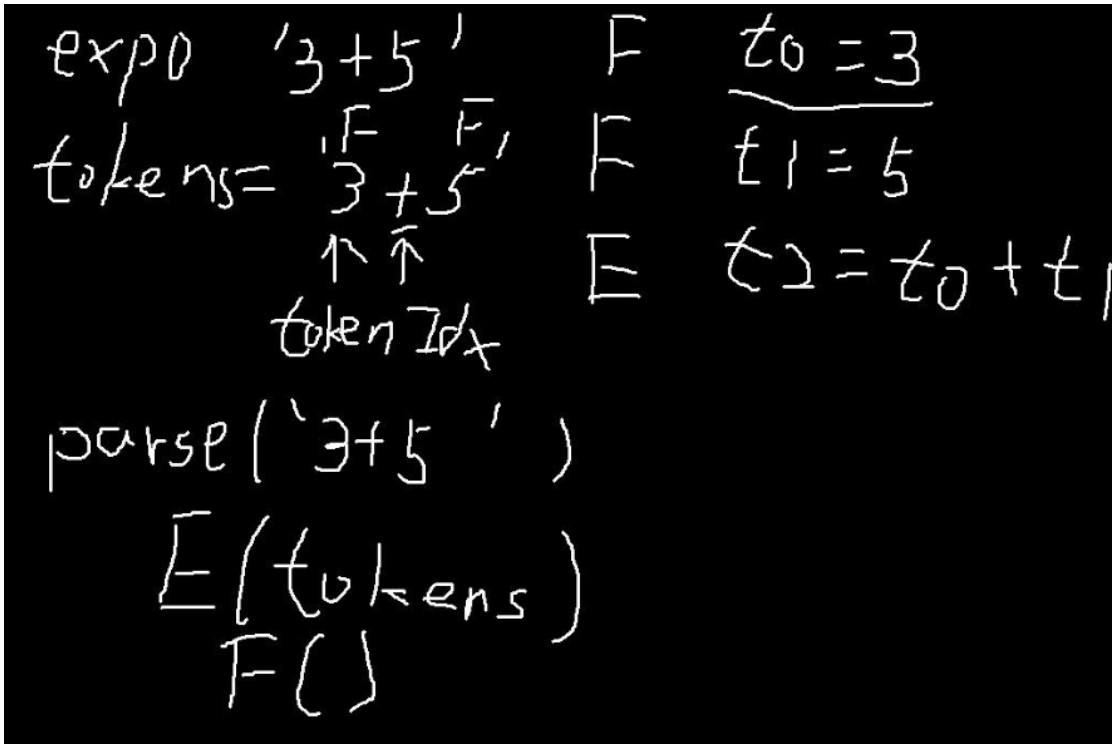
補充:

http://misavo.com/blog/%E9%99%B3%E9%8D%BE%E8%AA%A0/%E6%9B%B8%E7%B1%8D/C%E8%AA%9E%E8%A8%80/%E9%AB%98%E7%AD%89/structure

第二周

生成語法

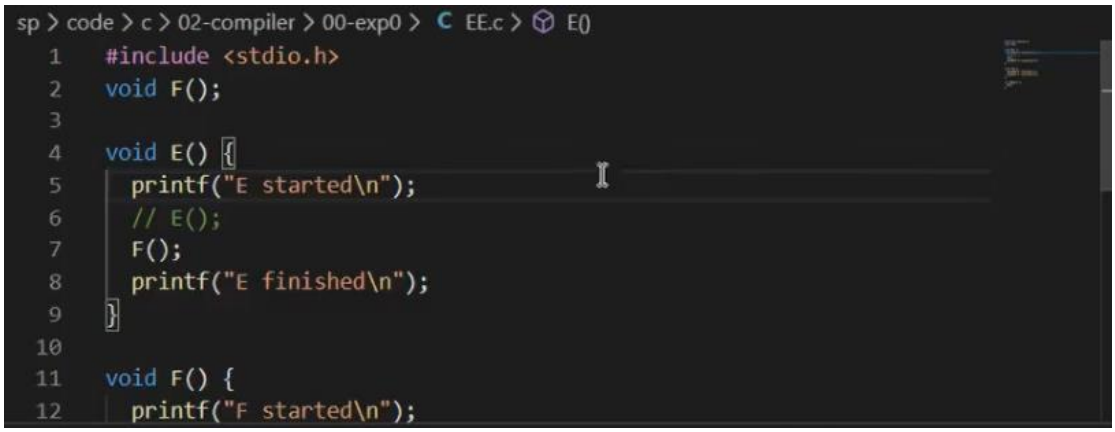
中間碼’3+5’ t0=3, t1=5, t2=t0+t1

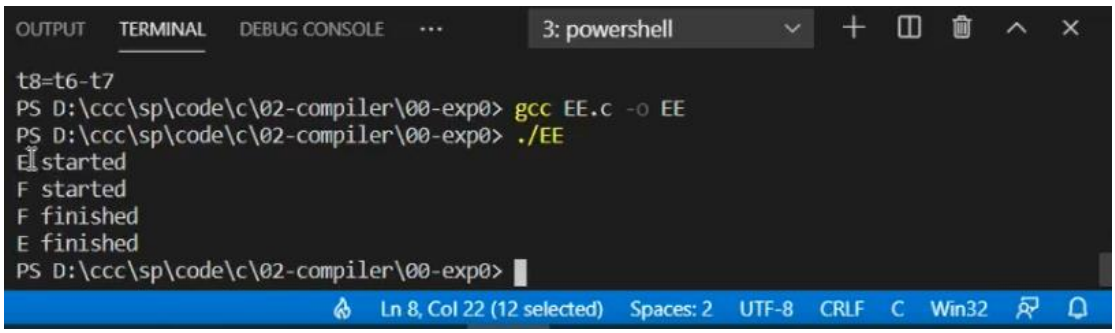

expo '3+5'
tokens= '3 + 5'
tokenIdx
parse('3+5')
E(tokens)
F()
F
t0 = 3
F
t1 = 5
E
t2 = t0 + t

EE.c:

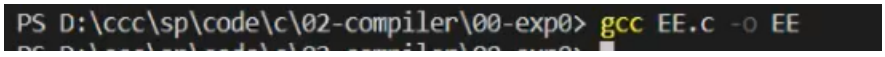

sp > code > c > 02-compiler > 00-exp0 > EE.c > F()
#include <stdio.h>
void F();
void E() {
printf("E started\n");
// E();
F();
printf("E finished\n");
}
void F() {
printf("F (char [12])"F finished\n"
printf("F finished\n");
}


sp > code > c > 02-compiler > 00-exp0 > EE.c > main()
8 printf("E finished\n");
9 }
10
11 void F() {
12 printf("F started\n");
13 printf("F finished\n");
14 }
15
16 int main() {
17 E();
18 }


sp > code > c > 02-compiler > 00-exp0 > EE.c > E()
1 #include <stdio.h>
2 void F();
3
4 void E() {
5 printf("E started\n");
6 // E();
7 F();
8 printf("E finished\n");
9 }
10
11 void F() {
12 printf("F started\n");


OUTPUT
TERMINAL
DEBUG CONSOLE
3: powershell
t8=t6-t7
PS D:\ccc\sp\code\c\02-compiler\00-exp0> gcc EE.c -o EE
PS D:\ccc\sp\code\c\02-compiler\00-exp0> ./EE
E started
F started
F finished
E finished
PS D:\ccc\sp\code\c\02-compiler\00-exp0>
Ln 8, Col 22 (12 selected)
Spaces: 2
UTF-8
CRLF
C
Win32


EE.exe


PS D:\ccc\sp\code\c\02-compiler\00-exp0> gcc EE.c -o EE

sp > code > c > 02-compiler > 00-exp0 > EE.c > F()

```c
#include <stdio.h>
void F();

void E() {
  printf("E started\n");
  // E();
  F();
  printf("E finished\n");
}

void F() {
  printf("F started\n");
  printf("F finished\n");
}

int main() {
```

程式碼:

E 產生 F

Void E( ){

Printf(“E started\n”);

F( );

Printf(“E finished\n”);

Expo.c:處理”+-“

sp > code > c > 02-compiler > 00-exp0 > exp0.c > main(int, char * [])

```c
}

int main(int argc, char * argv[]) {
  printf("argv[0]=%s argv[1]=%s\n", argv[0], argv[1]);
  printf("=== EBNF Grammar =====\n");
  printf("E=F ([+-] F)*\n");
  printf("F=Number | '(' E ')'\n");
  printf("==== parse:%s ========\n", argv[1]);
  parse(argv[1]);
}
```

argc 參數個數　　argv 參數變數(陣列)

./expo→argv[0](參數 0)

‘(3+5)-(2+4)-7’→argv[1]（參數 1）

isNext(“+-”)判斷是不是”+”或”-“

static 變數只會執行一次初始化

nextTemp( )傳回下一個臨時變數的代號

EX: 0 代表 t0,3 代表 t3(會回傳 0,1,2,3……)

F( )函數 F = Number 或’(‘E’)’　E=F([+-]F)*

(‘*’代表括號裡可以出現 0 次以上)

isdigit ( )檢查是否是數字

tokenIdx 顯示目前看到第幾個字

*tokens 表示目前所看到的字串

assert(0)讓程式停掉離開並處理

next( )取得字元，同時進到下一格

ch( )取得目前字元

exp0var:可處理變數

t1 =3 轉乘@3 , D=A;@t1;M=D

t1 =x 轉乘@x , D=M;@t1;M=D

@x 是 D=M

因為@x 是定位 x 變數，這個變數在記憶體裡面

Ex:x 記憶體位址在 100, @x D=M→@100 D=M

(不能用 D=A　因為只會把 100 存入 D)

第三周

exp1.c:處理*/月下層優先順序越高

E=T([+-]T)*語法　　E=3+5*8

T=F([*/]F)*　　T+T

　　F*F

Parse(char*str){

Tokens = str;字串為’3+5*8’

E( );}

Lexer.c:

readText( )

fopen( ),fread( ),fclose( )讀入文字檔

text[len]字串(長度夠長) char[TMax]10000000

argv[1]第一個參數

puts(code)輸出程式　lex(code)解析程式

isAlpha( )變數名稱或關鍵字

isspace( )略過空白

startable = #\0include\0”sum.h”\0int\0main…..

types[tokenTop]設定型態

tokens[tokenTop++]設定詞彙

03-compiler

“compiler.h”標頭檔

Endif 對映到#ifndef _COMPILER_H_

enum{Id,Int,Keyword,Literal,Char}定義每個函數

tokens[ ]詞彙表　　types[ ]所有型態

語法 PROG =STMTS　　STMT=WHILE|BLOCK|ASSIGN

BLOCK={STMTS}　　WHILE =while(E)STMT(有遞迴關係)

STMTS =STMT*　　ASSIGN =id’ = ’E’;

E=F(opE)*　F=(E)|Number|ID

While(E)STMT

emit(“t%d =%s\n”,f,item);產生中間碼

第四周

05-compiler.c:

Ir.h:*op 作加法或乘法運算

IrAssignTs:t=sum

irDump()印出整個中間碼

irNew()創建新的結構

irvm.c:執行中間碼

atoi(p->s)將字串轉為數字

trace= printf 印出指令執行狀況

.c 檔通常放程式碼.h 通常放 define 的那些東西

Map.c 定義了一個可搜尋的結構(雜湊表)，然後

varAdd 就呼叫 mapAdd(&varMap,.....)去新增一個變數到 VarMap 雜湊表裡面

不直接使用 mapAdd 是因為這樣模組化比較好，呼叫時參數會少一點

If(isRun)irRun();執行中間碼

03-asmVm

symAdd(&symMap,label,address);記住每個符號位置

fopen(inFile,”r”)開檔

空行標記……沒有碼 pass1 要記住

Sscanf 輸入指令

FILE*hfp = fopen(hackFile,”w”);開 16 進位輸出檔

FILE*hfp = fopen(binFile,” w”);開 2 進位輸出檔

Code2binary(code,binary);轉 2 進位

Unit16_t　b=c6toi(binary); 轉 16 進位