Compiler Project 4

資工三 409410029 王美綺

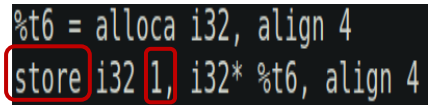
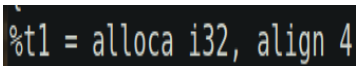
一、功能介紹

1. 支援型態：*struct、int、float、char、整數陣列*

(1)int、float、char

可以從 2 個角度去拆解

- global vs local
- 有初始值及沒有初始值(此都是以 int 為例)

	global	Local
有初值	 在 global 給初值	
沒有初值	 在 global 沒初值預設是 0	

[補]

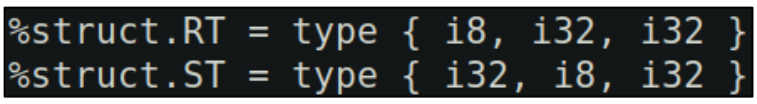
[1] 支援連續宣告(ex. int a = 2, int b = 2..或 float a, b..)

(2)struct(只能支援 3 個變數)

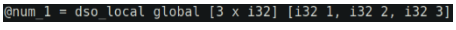
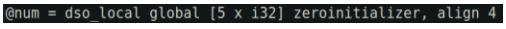
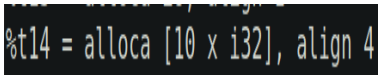
```
struct RT{

        char RT_1;
        int RT_2;
        int RT_3;
};

struct ST{
        int ST_1;
        char ST_2;
        int ST_3;
};
```



(3)整數陣列(只能支援固定 3 格大小)

	global	Local
有初值	 有初值分別是 num[3]={1, 2, 3}	沒做出來
沒有初值	 在 global 沒給初值就全是 0	

2. arithmetic 支援： +、-、*、/

3. comparison 支援： ==、!=、<、>、<=、>=

```
if(A == 0){
        ans = ( 2 + 5) * (A + 1);
}

if(B != 1){
        ans = ans + ans * 2;
}

if( 0 < C){
        ans = ans / 10;
}

if(5 > M){
        ans = ans + e - 5;
}

if(e <= 1){
        ans = ans + (2 + 3) * 2;
}

if(m >= 10){
        ans = ans - 10;
}
```

4. if / if-else（不支援巢狀）

```
if(A == 0){
        ans = ( 2 + 5) * (A + 1);
}

if(B != 1){
        ans = ans + ans * 2;
}

if( 0 < C){
        ans = ans / 10;
}

if(5 > M){
        ans = ans + e - 5;
}

if(e <= 1){
        ans = ans + (2 + 3) * 2;
}

if(m >= 10){
        ans = ans - 10;
}
```

```
if(sum < 50){
        printf("sum is lower than 50!\n");
}
else{
        printf("sum is bigger than 50!\n");
}
```

5. printf 字串及 1 和 2 個變數

```
printf("Please input a number: ");
```

```
printf("%d\n", ans);
```

```
printf("value = %d, ans = %d\n", c, d);
```

[補]

[1] 在一個程式可以支援多個 printf(因需多設定 printf 字串編號)

```
@.str.0 = private unnamed_addr constant [24 x i8] c"Please input a number: \00",
 align 1
@.str.1 = private unnamed_addr constant [3 x i8] c"%d\00", align 1
@.str.2 = private unnamed_addr constant [4 x i8] c"%d\0A\00", align 1
```

6. scanf(一個程式只能有一個變數 scanf 因沒多額外設定編號)

```
scanf("%d", &a);
```

```
declare dso_local i32 @__isoc99_scanf(i8*, ...)
```

```
%t18 = call i32 (i8*, ...) @__isoc99_scanf(i8* getelementptr inbounds ([3 x i8], [3 x i8]* @.str.1, i64 0, i64 0), i32* %t17)
```

7. while

```
while(c > 3){

        c = c -2;

}
```

```
br label %Jump1

Jump1:
%t9 = load i32, i32* %t7, align 4
%cond0 = icmp sgt i32 %t9, 3
br i1 %cond0, label %Ltrue1, label %Lfalse1

Ltrue1:
%t10 = load i32, i32* %t7, align 4
%t11 = sub nsw i32 %t10, 2
store i32 %t11, i32* %t7, align 4
br label %Jump1

Lfalse1:
```

8. switch(只能支援 3 個 case)

```
switch(c){

        case 0:
                d = sum(c);
                break;
        case 1:
                d = sum(c);
                break;
        case 2:
                d = sum(c);
                break;
}
```

```
switch i32 %t20, label %Jump5 [
   i32 0, label %Jump2
   i32 1, label %Jump3
   i32 2, label %Jump4
]

Jump2:
%t15 = load i32, i32* %t7, align 4
%t16 = call i32 @sum(i32 %t15)
store i32 %t16, i32* %t8, align 4
br label %Jump5

Jump3:
%t17 = load i32, i32* %t7, align 4
%t18 = call i32 @sum(i32 %t17)
store i32 %t18, i32* %t8, align 4
br label %Jump5

Jump4:
%t19 = load i32, i32* %t7, align 4
%t20 = call i32 @sum(i32 %t19)
store i32 %t20, i32* %t8, align 4
br label %Jump5

Jump5:
```

9. for(我覺得這個最難 XD)

```
for(i = 0; i < 10; i ++){
        sum = sum + i;
}
```

A:

先判斷是否有大於 10

B:

True 則坐迴圈內容

C:

把 i++並跳回 A 檢查是否大於 10

```
br label %Jump1

Jump1:
%t3 = load i32, i32* %t2, align 4                    A
%cond0 = icmp slt i32 %t3, 10
br i1 %cond0, label %Ltrue1, label %Lfalse1

Ltrue1:
%t4 = load i32, i32* %t1, align 4
%t5 = load i32, i32* %t2, align 4                    B
%t6 = add nsw i32 %t4, %t5
store i32 %t6, i32* %t1, align 4
br label %Jump2

Jump2:
%t7 = load i32, i32* %t2, align 4
%t8 = add nsw i32 %t7, 1                             C
store i32 %t8, i32* %t2, align 4
br label %Jump1

Lfalse1:
```

10. 副程式(傳入一個整數變數並回傳一個整數變數)

```
int sum(int a){

        int b;
        b = 5 * a;

        return b;
}
```

```
define dso_local i32 @sum(i32 %t1) {
%t2 = alloca i32, align 4
store i32 %t1, i32* %t2, align 4
%t3 = alloca i32, align 4
%t4 = load i32, i32* %t2, align 4
%t5 = mul nsw i32 5, %t4
store i32 %t5, i32* %t3, align 4
%t6 = load i32, i32* %t3, align 4
ret i32 %t6
}
```

```
%t16 = call i32 @sum(i32 %t15)
```

二、tokens

RETURN:'return';

INT:'int';

CHAR:'char';

FLOAT:'float';

STRUCT:'struct';

VOID: 'void';

LT_OP:'<';

GT_OP:'>';

LE_OP:'<=';

GE_OP:'>=';

EQ_OP:'==';

NE_OP:'!=';

PLUS_OP:'+';

MINUS_OP:'-';

MULTIPLE_OP:'*';

DIVID_OP:'/';

PP_OP: '++';

MM_OP: '--';

IF: 'if';

ELSE: 'else';

BREAK: 'break';

WHILE: 'while';

EOF_: 'EOF';

FOR: 'for';

DO: 'do';

SWITCH: 'switch';

CASE: 'case';

CONTINUE: 'continue';

DEFAULT: 'default';

MAIN: 'main';

SCANF: 'scanf';

PRINTF: 'printf';

DEC_NUM: ('0' |

('1'..'9')(DIGIT)*);

ID:

(LETTER)(LETTER|DIGIT)*;

fragment LETTER : 'a'..'z' | 'A'..'Z'

| '_';

fragment DIGIT : '0'..'9';

FLOAT_NUM: FLOAT_NUM1 |

FLOAT_NUM2 |

FLOAT_NUM3;

fragment FLOAT_NUM1:

(DIGIT)+'.'(DIGIT)*;

fragment FLOAT_NUM2:

'.'(DIGIT)+;

fragment FLOAT_NUM3:

(DIGIT)+;

STRING_LITERAL: ""

(EscapeSequence | ~('\\'|'"'))* "";

NULL: 'null'|

'\\0'{$channel=HIDDEN;};

WS: (' '|'\r'|'\t'|'\n')+

{$channel=HIDDEN;};

COMMENT:'/*' .* '*/'

{$channel=HIDDEN;};

fragment EscapeSequence: '\\'

('b'|'t'|'n'|'f'|'r'|'\"'|'\''|'\\');

三、測試檔案分析

test1.c:

1. 包含各式 type 的宣告(初始、未初始、global、local)
2. if
3. comparison
4. arithmetic
5. printf
6. scanf

test2.c:

1. for
2. printf
3. if-else

test3.c:

1. 副程式
2. While
3. Switch
4. printf