LLVM C subset description

Data type: 此程式***只支援int*** 的型態。

arithmetic

1. 支援int 的四則運算，還有mod (%)，j規則比照數學上的四則運算。

2. condition arithmetic 只有 > >= == < <= != 這四種，

目前還不支援 && || !，且只支援binary operation:

Example: a> b, a<= 0, a!= b etc.

可編譯的statements:

1. assignment:

Var = arithmeticExpr;

2. declarations:

可以單獨宣告: int a;

或是帶有初始值的宣告: int a = 7;

也可以一次宣告多個 int a, b;

Example: int a = 12, b, c = 3;

在這個statement 裡面，當一個變數被宣告之後，除了會被記在Hashmap 裡面，我有把他們的初始值都統一先給0 (store i32 0, %var, align 4)。

3. printf :

此程式可以處理 printf函數。

一樣，在formatstring 裡面的type 只有支援%d。

在formatstring 裡面的%d 數量要與後面給出的參數一樣多。

此printf也可以處理’\n’，可以印出換行(實作時使用\A0取代)

可以處理多個參數的printf ，也可以處理0 個的:

Example:

printf(“aaaa\n”);

printf(“bbbb”);

printf(“a = %d, b = %d, c= %d, d= %d\n”, a,b,c,d);

4. condition:

支援:

單個if ,

If-else,

此程式也可以執行多個else if 的結構，例如:

if(a == 1){

a = 3;

}

else if(a == 1){

a = 5;

}

else if (a == 2){

a = 7;

}

else{

a = 9;

}

test2.c中有類似的範例

我在實作condition statement時，我是先把labelCount當前的值作為回到main branch 的label:

L1:

…

L2:

…

L0:

Keep main function

然後，每一個if statement (以及else if )

都會產生出兩個Labels :

條件true : current labelCount:

最終若是執行此block，則會jump 到main branch 繼續執行

條件: false: current labelCount+1:

當condition false的時候會進來，會去在判斷branch 的情況，看要去哪一個branch。

如果之後沒有branch了，則會jump 到main branch 繼續執行

\*\* else 的block最後會jump 到main branch 繼續執行

Example:

If(a > b){

//L1 stmt

}

else if(a < b){

//L3 stmt

}

else{

//L4 stmt

}

則.ll 檔會類似:

condi1 = sgt a,b

br condi1 L1, L2

L1: ; a>b ture

L1 stmt

br L0

L2: ; a>b false

condi2 = slt a, b

br condi2 L3, L4

L3: ; a<b ture

L3 stmt

br L0

L4: ;a<b false

L4 stmt

br L0

L0:

Keep main

如果說沒有else ，則最後一個if false condition 多出來的標籤，就直接jump到main。

雖然這樣會多一個沒用的label，但為了要考慮那個if如果有false Label stmt存在的可能性，這是我目前想到的解法。

Example2:

If(a > b){

//L1 stmt

}

else if (a == b){

//L3 stmt

}

condi1 = sgt a,b

br condi1 L1, L2

L1: ; a>b ture

L1 stmt

br L0

L2: ; a>b false

condi2 = slt a, b

br condi2 L3, L4

L3: ; a==b ture

L3 stmt

br L0

L4: ;a == b false

br L0

L0:

Keep main

也支援巢狀結構，可以執行test1.c 的相關測試