#include <stdio.h>

void my\_int\_func(int x)

{

printf( "%d\n", x );

}

int main()

{

void (\*foo)(int);

foo = &my\_int\_func;

/\* call my\_int\_func (note that you do not need to write (\*foo)(2) ) \*/

foo( 2 );

/\* but if you want to, you may \*/

(\*foo)( 2 );

return 0;

}

|-FunctionDecl 0x1b53fa0 </home/labaccount/11-2-17/mlsa/IG/example/funcPointer/funcPointer.c:2:1, line:5:1> line:2:6 used my\_int\_func 'void (int)'

| |-ParmVarDecl 0x1b53ee0 <col:18, col:22> col:22 used x 'int'

| `-CompoundStmt 0x1b54190 <line:3:1, line:5:1>

| `-CallExpr 0x1b54110 <line:4:5, col:23> 'int'

| |-ImplicitCastExpr 0x1b540f8 <col:5> 'int (\*)(const char \*, ...)' <FunctionToPointerDecay>

| | `-DeclRefExpr 0x1b54050 <col:5> 'int (const char \*, ...)' Function 0x1b45a28 'printf' 'int (const char \*, ...)'

| |-ImplicitCastExpr 0x1b54160 <col:13> 'const char \*' <BitCast>

| | `-ImplicitCastExpr 0x1b54148 <col:13> 'char \*' <ArrayToPointerDecay>

| | `-StringLiteral 0x1b54078 <col:13> 'char [4]' lvalue "%d\n"

| `-ImplicitCastExpr 0x1b54178 <col:21> 'int' <LValueToRValue>

| `-DeclRefExpr 0x1b540a8 <col:21> 'int' lvalue ParmVar 0x1b53ee0 'x' 'int'

`-FunctionDecl 0x1b541d0 <line:8:1, line:19:1> line:8:5 main 'int ()'

`-CompoundStmt 0x1b55690 <line:9:1, line:19:1>

|-DeclStmt 0x1b54410 <line:10:5, col:21>

| `-VarDecl 0x1b543b0 <col:5, col:20> col:12 used foo 'void (\*)(int)'

|-BinaryOperator 0x1b54498 <line:11:5, col:12> 'void (\*)(int)' '='

| |-DeclRefExpr 0x1b54428 <col:5> 'void (\*)(int)' lvalue Var 0x1b543b0 'foo' 'void (\*)(int)'

| `-UnaryOperator 0x1b54478 <col:11, col:12> 'void (\*)(int)' prefix '&'

| `-DeclRefExpr 0x1b54450 <col:12> 'void (int)' Function 0x1b53fa0 'my\_int\_func' 'void (int)'

|-CallExpr 0x1b54520 <line:14:5, col:12> 'void'

| |-ImplicitCastExpr 0x1b54508 <col:5> 'void (\*)(int)' <LValueToRValue>

| | `-DeclRefExpr 0x1b544c0 <col:5> 'void (\*)(int)' lvalue Var 0x1b543b0 'foo' 'void (\*)(int)'

| `-IntegerLiteral 0x1b544e8 <col:10> 'int' 2

|-CallExpr 0x1b55628 <line:16:5, col:15> 'void'

| |-ImplicitCastExpr 0x1b55610 <col:5, col:10> 'void (\*)(int)' <FunctionToPointerDecay>

| | `-ParenExpr 0x1b545b0 <col:5, col:10> 'void (int)':'void (int)'

| | `-UnaryOperator 0x1b54590 <col:6, col:7> 'void (int)':'void (int)' prefix '\*'

| | `-ImplicitCastExpr 0x1b54578 <col:7> 'void (\*)(int)' <LValueToRValue>

| | `-DeclRefExpr 0x1b54550 <col:7> 'void (\*)(int)' lvalue Var 0x1b543b0 'foo' 'void (\*)(int)'

| `-IntegerLiteral 0x1b545d0 <col:13> 'int' 2

`-ReturnStmt 0x1b55678 <line:18:5, col:12>

`-IntegerLiteral 0x1b55658 <col:12> 'int' 0