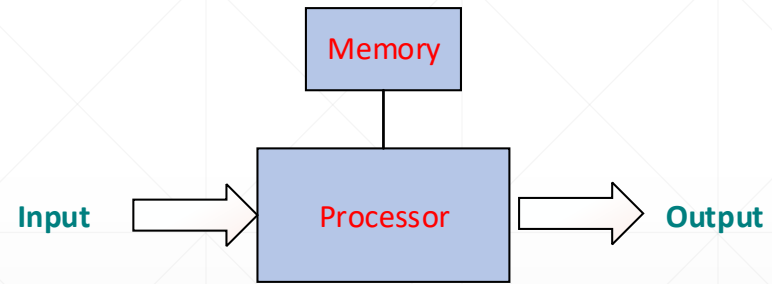


實作案例分享

- 嵌入式條碼掃描系統

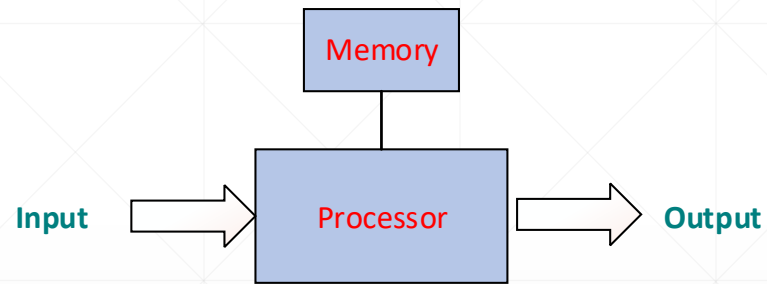


A generic embedded system




實作案例分享

■ 嵌入式條碼掃描系統



A generic embedded system



A photograph of a Raspberry Pi 3 Model B V1.2 board. The board is green with various components visible, including the Broadcom SoC, RAM, and various connectors. A black arrow points from the pin header area of the board to the adjacent pin list table.

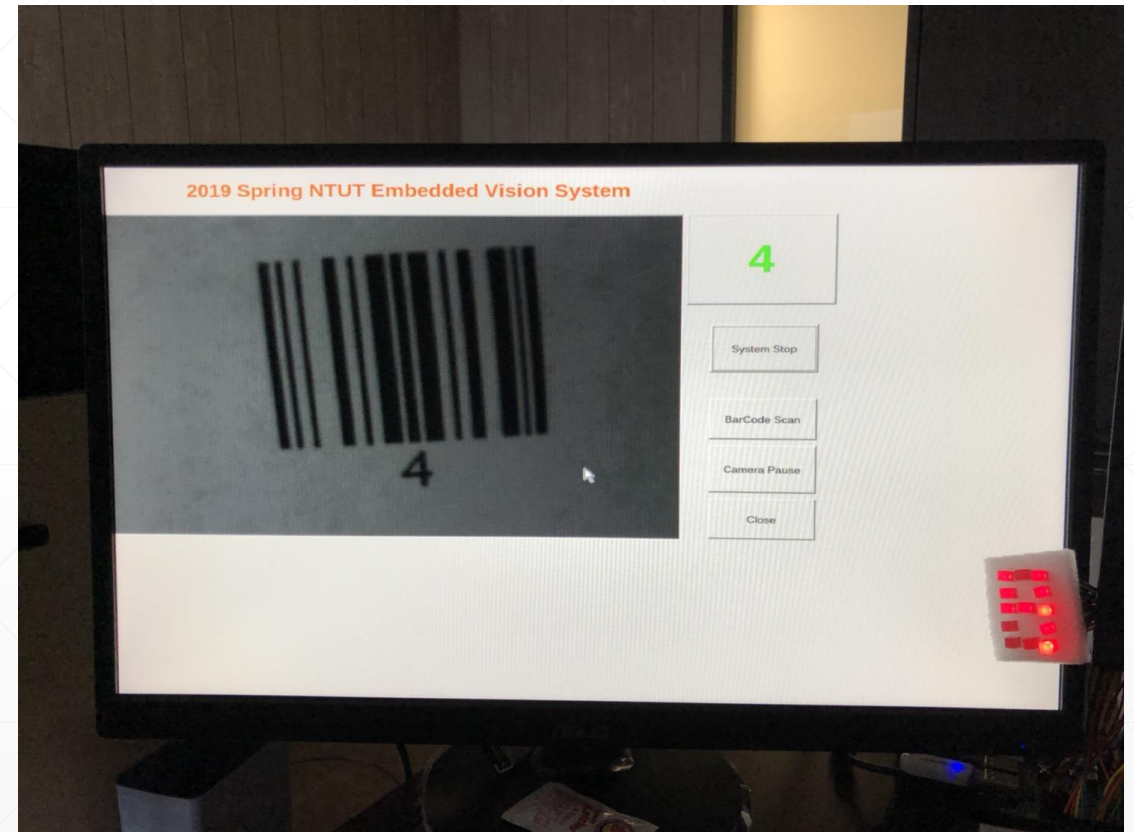
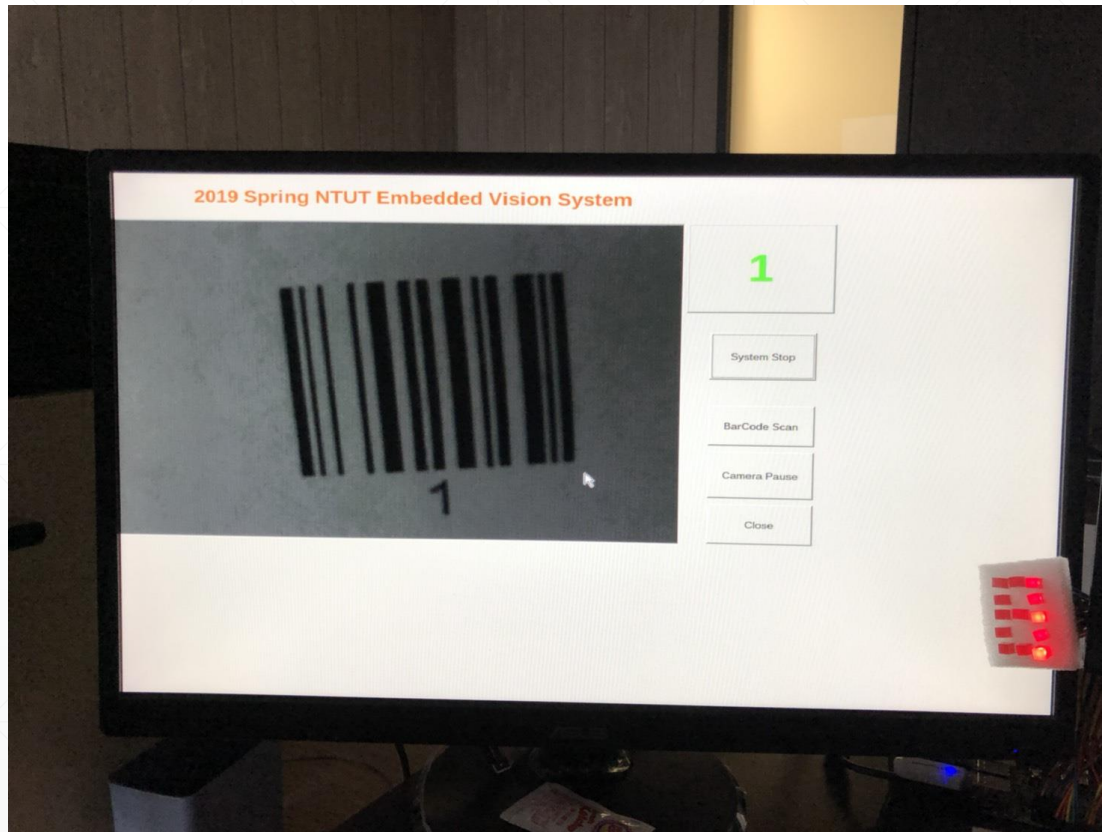
	Pin No.	
3.3V	1	2 5V
GPIO2	3	4 5V
GPIO3	5	6 GND
GPIO4	7	8 GPIO14
GND	9	10 GPIO15
GPIO17	11	12 GPIO18
GPIO27	13	14 GND
GPIO22	15	16 GPIO23
3.3V	17	18 GPIO24
GPIO10	19	20 GND
GPIO9	21	22 GPIO25
GPIO11	23	24 GPIO8
GND	25	26 GPIO7
DNC	27	28 DNC
GPIO5	29	30 GND
GPIO6	31	32 GPIO12
GPIO13	33	34 GND
GPIO19	35	36 GPIO16
GPIO26	37	38 GPIO20
GND	39	40 GPIO21

Key

Power +	UART
GND	SPI
I ² C	GPIO

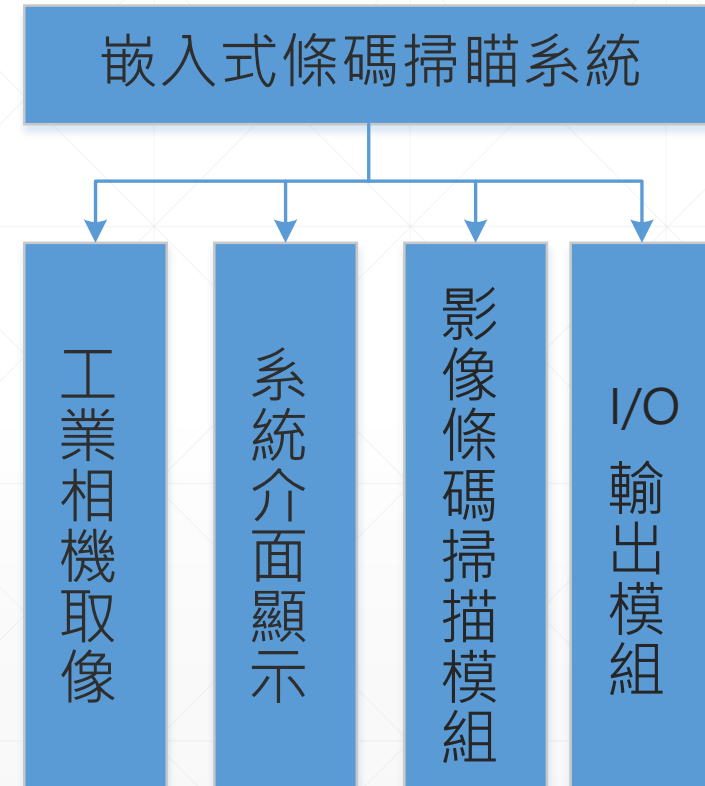
實作案例分享

■ 嵌入式條碼掃描系統



實作案例分享

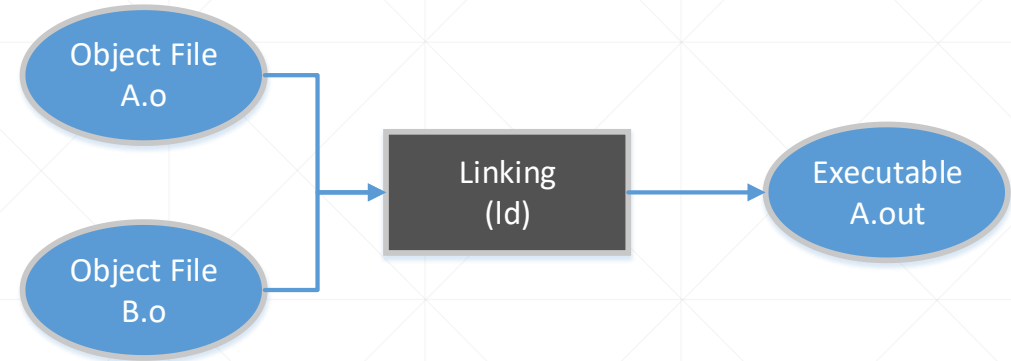
- 嵌入式條碼掃描系統
 - 多工系統 (工業相機取像執行緒、系統介面顯示執行緒)
 - 影像條碼掃描模組
 - I/O 輸出
- 物聯網應用：
 - 導入遠端監控與操作系統



Embedded Software Porting

```
/* a.cpp */  
#include <stdio.h>  
int func(int a, int b);  
int main()  
{  
    int value = func(10,20);  
    printf("value = %d \n", value);  
    retrun 0;  
}
```

```
/* b.cpp */  
int func(int a ,int b)  
{  
    return a+b;  
}
```



方法1 :

```
# g++ -c a.cpp  
# g++ -c b.cpp  
# g++ a.o b.o  
# ./a.out
```

方法2 :

```
# g++ a.cpp b.cpp  
# ./a.out
```

```
# g++ a.cpp b.cpp -o practice  
# ./practice
```

```
root@hank-X302LJ:/home/hank# ./practicce  
value = 30
```

Embedded Software Porting

■ Program2

- 請設計 + - * / 四則運算的函式，撰寫於b.cpp中，並在program2.cpp中進行呼叫計算，並將數值列印出來。

```
int main()
{
    int addvalue = addfunc(10,20);
    int subvalue = subfunc(30,5);
    int mulvalue = mulfunc(3,2);
    double divvalue = divfunc(50,4);

    printf("add func value = %d \n",addvalue);
    printf("sub func value = %d \n",subvalue);
    printf("mul func value = %d \n",mulvalue);
    printf("div func value = %1.2f \n",divvalue);
    return 0;
}
```

program2.cpp

```
# g++ program2.cpp b.cpp -o program2
# ./program2
```

```
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# ./program2
add func value = 30
sub func value = 25
mul func value = 6
div func value = 12.50
```

請觀看執行檔中的符號表或是查看b.o中自己設計的函式？有啥不同之處？

```
# readelf -s program2
```

```
# g++ -c b.cpp
```

```
# readelf -s b.o
```

Embedded Software Porting

program2.cpp

```
#include <stdio.h>

int addfunc(int a, int b);
int subfunc(int a, int b);
int mulfunc(int a, int b);
double divfunc(double numerator, double denominator);

int main()
{
    int addvalue = addfunc(10,20);
    int subvalue = subfunc(30,5);
    int mulvalue = mulfunc(3,2);
    double divvalue = divfunc(50,4);

    printf("add func value = %d \n", addvalue);
    printf("sub func value = %d \n", subvalue);
    printf("mul func value = %d \n", mulvalue);
    printf("div func value = %1.2f \n", divvalue);

    return 0;
}
```

b.cpp

```
int addfunc(int a, int b)
{
    return a+b;
}

int subfunc(int a, int b)
{
    return a-b;
}

int mulfunc(int a, int b)
{
    return a*b;
}

double divfunc(double numerator, double denominator)
{
    return numerator/denominator;
}
```



```

int addfunc(int a, int b);
int subfunc(int a, int b);
int mulfunc(int a, int b);
double divfunc(double numerator, double denominator);

```

_z7addfuncii

_z7subfuncii

_z7mulfuncii

_z7divfuncdd

```

root@hank-X302LJ: /home/hank/2019_NTUT_embedded_course/practice
35: 0000000000400500      0 FUNC    LOCAL   DEFAULT   14 frame_dummy
36: 0000000000600e10      0 OBJECT  LOCAL   DEFAULT   19 __frame_dummy_init_ar
ray
37: 0000000000000000      0 FILE    LOCAL   DEFAULT   ABS program2.cpp
38: 0000000000000000      0 FILE    LOCAL   DEFAULT   ABS b.cpp
39: 0000000000000000      0 FILE    LOCAL   DEFAULT   ABS crtstuff.c
40: 0000000000400908      0 OBJECT  LOCAL   DEFAULT   18 __FRAME_END__
41: 0000000000600e20      0 OBJECT  LOCAL   DEFAULT   21 __JCR_END__
42: 0000000000000000      0 FILE    LOCAL   DEFAULT   ABS
43: 0000000000600e18      0 NOTYPE   LOCAL   DEFAULT   19 __init_array_end
44: 0000000000600e28      0 OBJECT  LOCAL   DEFAULT   22 __DYNAMIC
45: 0000000000600e10      0 NOTYPE   LOCAL   DEFAULT   19 __init_array_start
46: 0000000000400740      0 NOTYPE   LOCAL   DEFAULT   17 __GNU_EH_FRAME_HDR
47: 0000000000601000      0 OBJECT  LOCAL   DEFAULT   24 __GLOBAL_OFFSET_TABLE__
48: 00000000004006c0      2 FUNC    GLOBAL   DEFAULT   14 __libc_csu_fini
49: 0000000000000000      0 NOTYPE   WEAK     DEFAULT   UND __ITM_deregisterTMClone
etab
50: 0000000000601028      0 NOTYPE   WEAK     DEFAULT   25 data_start
51: 0000000000601038      0 NOTYPE   GLOBAL   DEFAULT   25 _edata
52: 00000000004006c4      0 FUNC    GLOBAL   DEFAULT   15 _fini
53: 0000000000000000      0 FUNC    GLOBAL   DEFAULT   UND printf@@GLIBC_2.2.5
54: 0000000000400616     19 FUNC    GLOBAL   DEFAULT   14 _Z7mulfuncii
55: 0000000000000000      0 FUNC    GLOBAL   DEFAULT   UND __libc_start_main@@GL
IBC_
56: 0000000000601028      0 NOTYPE   GLOBAL   DEFAULT   25 __data_start
57: 0000000000000000      0 NOTYPE   WEAK     DEFAULT   UND __gmon_start__
58: 0000000000601030      0 OBJECT  GLOBAL   HIDDEN   25 __dso_handle
59: 00000000004006d0      4 OBJECT  GLOBAL   DEFAULT   16 _IO_stdin_used
60: 0000000000400650     101 FUNC    GLOBAL   DEFAULT   14 __libc_csu_init
61: 0000000000601040      0 NOTYPE   GLOBAL   DEFAULT   26 __end
62: 0000000000400430     42 FUNC    GLOBAL   DEFAULT   14 _start
63: 0000000000601038      0 NOTYPE   GLOBAL   DEFAULT   26 __bss_start
64: 0000000000400526     202 FUNC    GLOBAL   DEFAULT   14 main
65: 0000000000400629     26 FUNC    GLOBAL   DEFAULT   14 _Z7divfuncdd
66: 0000000000000000      0 NOTYPE   WEAK     DEFAULT   UND __rtld_initialize
67: 0000000000400604     18 FUNC    GLOBAL   DEFAULT   14 _Z7subfuncii
68: 0000000000601038      0 OBJECT  GLOBAL   HIDDEN   25 __TMC_END__
69: 0000000000000000      0 NOTYPE   WEAK     DEFAULT   UND __ITM_registerTMCloneT
able
70: 00000000004005f0     20 FUNC    GLOBAL   DEFAULT   14 _Z7addfuncii
71: 00000000004003c8      0 FUNC    GLOBAL   DEFAULT   11 _init

```


Embedded Software Porting

```
int addfunc(int a, int b);  
int subfunc(int a, int b);  
int mulfunc(int a, int b);  
double divfunc(double numerator, double denominator);
```

`_z7addfuncii`

`_z7subfuncii`

`_z7mulfuncii`

`_z7divfuncdd`

```
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# g++ -c b.cpp  
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# readelf -s b.o
```

Symbol table '.symtab' contains 12 entries:

Num:	Value	Size	Type	Bind	Vis	Ndx	Name
0:	0000000000000000	0	NOTYPE	LOCAL	DEFAULT	UND	
1:	0000000000000000	0	FILE	LOCAL	DEFAULT	ABS	b.cpp
2:	0000000000000000	0	SECTION	LOCAL	DEFAULT	1	
3:	0000000000000000	0	SECTION	LOCAL	DEFAULT	2	
4:	0000000000000000	0	SECTION	LOCAL	DEFAULT	3	
5:	0000000000000000	0	SECTION	LOCAL	DEFAULT	5	
6:	0000000000000000	0	SECTION	LOCAL	DEFAULT	6	
7:	0000000000000000	0	SECTION	LOCAL	DEFAULT	4	
8:	0000000000000000	20	FUNC	GLOBAL	DEFAULT	1	_Z7addfuncii
9:	0000000000000014	18	FUNC	GLOBAL	DEFAULT	1	_Z7subfuncii
10:	0000000000000026	19	FUNC	GLOBAL	DEFAULT	1	_Z7mulfuncii
11:	0000000000000039	26	FUNC	GLOBAL	DEFAULT	1	_Z7divfuncdd

Embedded Software Porting

- `extern "C"` 是C++特有的組合關鍵字
 - 被它修飾的目標是"`extern`"的
 - 被它修飾的目標是"`C`"的
 - 被用來實現C/C++混合程式設計

- C++中有函式重載(overloading)：

```
void func(int i, int j);
```

```
void func(double i, double j);
```

```
void func(int i, int j);
```

```
void func(double i, double j);
```



```
_funcii
```

```
_funcdd
```

- 如果C++調用一個C語言編寫的.DLL時，其.DLL標頭檔或聲明介面函數時，需加`extern "C" { }`。
- 在C中引用C++語言中的函數和變數時，C++的標頭檔需添加`extern "C"`，但是在C語言中不能直接引用聲明了`extern "C"`的該標頭檔，應該僅將C檔中將C++中定義的`extern "C"`函式宣告為`extern`類型。

Embedded Software Porting

program2.cpp

```
#include <stdio.h>

extern "C"
{
    int addfunc(int a, int b);
    int subfunc(int a, int b);
    int mulfunc(int a, int b);
    double divfunc(double numerator, double denominator);
}

int main()
{
    int addvalue = addfunc(10,20);
    int subvalue = subfunc(30,5);
    int mulvalue = mulfunc(3,2);
    double divvalue = divfunc(50,4);

    printf("add func value = %d \n", addvalue);
    printf("sub func value = %d \n", subvalue);
    printf("mul func value = %d \n", mulvalue);
    printf("div func value = %1.2f \n", divvalue);

    return 0;
}
```

readelf -s program2

b.cpp

```
extern "C" int addfunc(int a, int b)
{
    return a+b;
}

extern "C" int subfunc(int a, int b)
{
    return a-b;
}

extern "C" int mulfunc(int a, int b)
{
    return a*b;
}

extern "C" double divfunc(double numerator, double denominator)
{
    return numerator/denominator;
}
```

g++ -c b.cpp
readelf -s b.o


```

root@hank-X302LJ: /home/hank/2019_NTUT_embedded_course/practice
35: 0000000000400500      0 FUNC    LOCAL   DEFAULT   14 frame_dummy
36: 0000000000600e10      0 OBJECT  LOCAL   DEFAULT   19 __frame_dummy_init_ar
ray_
37: 0000000000000000      0 FILE     LOCAL   DEFAULT   ABS program2.cpp
38: 0000000000000000      0 FILE     LOCAL   DEFAULT   ABS b.cpp
39: 0000000000000000      0 FILE     LOCAL   DEFAULT   ABS crtstuff.c
40: 0000000000400908      0 OBJECT  LOCAL   DEFAULT   18 __FRAME_END__
41: 0000000000600e20      0 OBJECT  LOCAL   DEFAULT   21 __JCR_END__
42: 0000000000000000      0 FILE     LOCAL   DEFAULT   ABS
43: 0000000000600e18      0 NOTYPE   LOCAL   DEFAULT   19 __init_array_end
44: 0000000000600e28      0 OBJECT  LOCAL   DEFAULT   22 __DYNAMIC
45: 0000000000600e10      0 NOTYPE   LOCAL   DEFAULT   19 __init_array_start
46: 0000000000400740      0 NOTYPE   LOCAL   DEFAULT   17 __GNU_EH_FRAME_HDR
47: 0000000000601000      0 OBJECT  LOCAL   DEFAULT   24 __GLOBAL_OFFSET_TABLE_
48: 00000000004006c0      2 FUNC    GLOBAL  DEFAULT   14 __libc_csu_fini
49: 0000000000400604      18 FUNC    GLOBAL  DEFAULT   14 subfunc
50: 0000000000000000      0 NOTYPE   WEAK     DEFAULT   UND __ITM_deregisterTMClone
eTab
51: 0000000000601028      0 NOTYPE   WEAK     DEFAULT   25 data_start
52: 0000000000400616      19 FUNC    GLOBAL  DEFAULT   14 mulfunc
53: 00000000004005f0      20 FUNC    GLOBAL  DEFAULT   14 addfunc
54: 0000000000601038      0 NOTYPE   GLOBAL  DEFAULT   25 __edata
55: 00000000004006c4      0 FUNC    GLOBAL  DEFAULT   15 __fini
56: 0000000000000000      0 FUNC    GLOBAL  DEFAULT   UND printf@GLIBC_2.2.5
57: 0000000000400629      26 FUNC    GLOBAL  DEFAULT   14 divfunc
58: 0000000000000000      0 FUNC    GLOBAL  DEFAULT   UND __libc_start_main@GL
IBC_
59: 0000000000601028      0 NOTYPE   GLOBAL  DEFAULT   25 __data_start
60: 0000000000000000      0 NOTYPE   WEAK     DEFAULT   UND __gmon_start__
61: 0000000000601030      0 OBJECT  GLOBAL  HIDDEN   25 __dso_handle
62: 00000000004006d0      4 OBJECT  GLOBAL  DEFAULT   16 __IO_stdin_used
63: 0000000000400650      101 FUNC    GLOBAL  DEFAULT   14 __libc_csu_init
64: 0000000000601040      0 NOTYPE   GLOBAL  DEFAULT   26 __end
65: 0000000000400430      42 FUNC    GLOBAL  DEFAULT   14 __start
66: 0000000000601038      0 NOTYPE   GLOBAL  DEFAULT   26 __bss_start
67: 0000000000400526      202 FUNC    GLOBAL  DEFAULT   14 main
68: 0000000000000000      0 NOTYPE   WEAK     DEFAULT   UND __Jv_RegisterClasses
69: 0000000000601038      0 OBJECT  GLOBAL  HIDDEN   25 __TMC_END__
70: 0000000000000000      0 NOTYPE   WEAK     DEFAULT   UND __ITM_registerTMCloneT
able
71: 00000000004003c8      0 FUNC    GLOBAL  DEFAULT   11 __init
root@hank-X302LJ: /home/hank/2019_NTUT_embedded_course/practice#

```

```

root@hank-X302LJ: /home/hank/2019_NTUT_embedded_course/practice# g++ -c b.cpp
root@hank-X302LJ: /home/hank/2019_NTUT_embedded_course/practice# readelf -s b.o

Symbol table '.symtab' contains 12 entries:
Num:  Value              Size Type      Bind   Vis      Ndx Name
 0: 0000000000000000      0 NOTYPE   LOCAL   DEFAULT   UND
 1: 0000000000000000      0 FILE     LOCAL   DEFAULT   ABS b.cpp
 2: 0000000000000000      0 SECTION  LOCAL   DEFAULT    1
 3: 0000000000000000      0 SECTION  LOCAL   DEFAULT    2
 4: 0000000000000000      0 SECTION  LOCAL   DEFAULT    3
 5: 0000000000000000      0 SECTION  LOCAL   DEFAULT    5
 6: 0000000000000000      0 SECTION  LOCAL   DEFAULT    6
 7: 0000000000000000      0 SECTION  LOCAL   DEFAULT    4
 8: 0000000000000000      20 FUNC    GLOBAL  DEFAULT    1 addfunc
 9: 0000000000000014      18 FUNC    GLOBAL  DEFAULT    1 subfunc
10: 0000000000000026      19 FUNC    GLOBAL  DEFAULT    1 mulfunc
11: 0000000000000039      26 FUNC    GLOBAL  DEFAULT    1 divfunc
root@hank-X302LJ: /home/hank/2019_NTUT_embedded_course/practice#

```

```

# g++ -c b.cpp
# readelf -s b.o

```

```
# readelf -s program2
```


Embedded Software Porting

▪ Program3

- 請使用自行設計的四則運算函式設計出三角形、矩形、圓形面積計算函式，撰寫於c.cpp中，並在program3.cpp中進行呼叫面積計算，並將面積數值列印出來。

```
#include <stdio.h>
#include "c.h"

int main()
{
    double baseSide = 3;
    double height = 3;
    double tri_area = Area_Triangle(baseSide, height);

    double radius = 5;
    double circle_area = Area_Circle(radius);

    double width = 6;
    height = 8;
    double rect_area = Area_Rectangle(width, height);

    printf("triangle area = %1.2f \n", tri_area);
    printf("circle area = %1.2f \n", circle_area);
    printf("rectangle area = %1.2f \n", rect_area);
    return 0;
}
```

program3.cpp

```
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# ./program3
triangle area = 4.50
circle area = 75.00
rectangle area = 48.00
```

Embedded Software Porting

■ Program3

- 請使用自行設計的四則運算函式設計出三角形、矩形、圓形面積計算函式，撰寫於c.cpp中，並在program3.cpp中進行呼叫面積計算，並將面積數值列印出來。

```
#include <stdio.h>
#include "c.h"

int main()
{
    double baseSide = 3;
    double height = 3;
    double tri_area = Area_Triangle(baseSide, height);

    double radius = 5;
    double circle_area = Area_Circle(radius);

    double width = 6;
    height = 8;
    double rect_area = Area_Rectangle(width, height);

    printf("triangle area = %1.2f \n", tri_area);
    printf("circle area = %1.2f \n", circle_area);
    printf("rectangle area = %1.2f \n", rect_area);
    return 0;
}
```

program3.cpp

```
#pragma once
```

c.h

```
extern "C" double Area_Triangle(double a_base_side, double a_height);
extern "C" double Area_Circle(double a_radius);
extern "C" double Area_Rectangle(double a_width, double a_height);
```

c.cpp

```
#include "b.h"

extern "C" double Area_Triangle(double a_base_side, double a_height)
{
    return divfunc(mulfunc(a_base_side, a_height), 2);
}
extern "C" double Area_Circle(double a_radius)
{
    return mulfunc(mulfunc(a_radius, a_radius), 3.1415926);
}
extern "C" double Area_Rectangle(double a_width, double a_height)
{
    return mulfunc(a_width, a_height);
}
```

g++ program3.cpp c.cpp b.cpp -o program3

Embedded Software Porting

- 類別 (class)：積木
- class是C++中用來封裝資料的關鍵字，當您使用類別來定義一個物件 (Object) 時，您考慮這個物件可能擁有的「屬性」 (Property) 與「方法」 (Method) 。
 - 屬性是物件的靜態描述
 - 方法是物件上的動態操作。
- 在類別封裝時，有一個基本原則是：資訊的最小化公開，是基於安全性的考量，避免程式設計人員隨意操作屬性成員而造成程式的錯誤。

public 公開 VS private 私有

Embedded Software Porting

first_class.h

```
#pragma once
class Area_Calculator
{
public:
    Area_Calculator();
    ~Area_Calculator();

private:
    double m_radius;

public:
    void setRadius(double a_radius);
    double GetArea();
};
```

公開方法
建構子、解構子

私有屬性

公開方法

first_class.cpp

```
#include "first_class.h"

Area_Calculator::Area_Calculator()
{
    m_radius = area = 0;
}

void Area_Calculator::setRadius(double a_radius)
{
    m_radius = a_radius;
}

double Area_Calculator::GetArea()
{
    return m_radius*m_radius*3.1415926;
}
```

建構進行初始化

公開方法探訪私有屬性

建構子：常用於初始化類別內的屬性資料或建立類別類所需之記憶體
解構子：常用於釋放類別內之物件或所使用的記憶體

Embedded Software Porting

■ Program4

請使用類別語法將上述四則運算與面積計算函式進行面積計算類別設計，其中計算參數(寬、高...等等)請設計公用函式進行參數設定，將計算面積類別撰寫於d.cpp中，並在program4.cpp中進行計算面積類別呼叫，並將面積數值列印出來。

program4.cpp

```
#include <stdio.h>
#include "d.h"

int main()
{
    TArea *calculator = new TArea();

    calculator->setTriangleParam(4,4);
    double tri_area = calculator->getTriangleArea();

    calculator->setCircleParam(10);
    double circle_area = calculator->getCircleArea();

    calculator->setRectangleParam(6,8);
    double rect_area = calculator->getRectangleArea();

    printf("triangle area = %1.2f \n", tri_area);
    printf("circle area = %1.2f \n", circle_area);
    printf("rectangle area = %1.2f \n", rect_area);

    delete calculator;
    return 0;
}
```

```
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# g++ program4.cpp
d.cpp b.cpp -o program4
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# ./program4
triangle area = 8.00
circle area = 300.00
rectangle area = 48.00
```

```
#include <stdio.h>
#include "d.h"
```

program4.cpp

```
int main()
{
    TArea *calculator = new TArea();

    calculator->setTriangleParam(4,4);
    double tri_area = calculator->getTriangleArea();

    calculator->setCircleParam(10);
    double circle_area = calculator->getCircleArea();

    calculator->setRectangleParam(6,8);
    double rect_area = calculator->getRectangleArea();

    printf("triangle area = %1.2f \n", tri_area);
    printf("circle area = %1.2f \n", circle_area);
    printf("rectangle area = %1.2f \n", rect_area);

    delete calculator;
    return 0;
}
```

```
#pragma once
```

d.h

```
class TArea
{
private:
    double m_width;
    double m_height;
    double m_baseSide;
    double m_radius;

public:
    void setTriangleParam(double a_base_side, double a_height);
    void setCircleParam(double a_radius);
    void setRectangleParam(double a_width, double a_height);

    double getTriangleArea();
    double getCircleArea();
    double getRectangleArea();
};
```



d.cpp

```
#include "b.h"
#include "d.h"

void TArea::setTriangleParam(double a_base_side, double a_height)
{
    m_baseSide = a_base_side;
    m_height = a_height;
}

void TArea::setCircleParam(double a_radius)
{
    m_radius = a_radius;
}

void TArea::setRectangleParam(double a_width, double a_height)
{
    m_width = a_width;
    m_height = a_height;
}

double TArea::getTriangleArea()
{
    return divfunc(mulfunc(m_baseSide, m_height), 2);
}

double TArea::getCircleArea()
{
    return mulfunc(mulfunc(m_radius, m_radius), 3.1415926);
}

double TArea::getRectangleArea()
{
    return mulfunc(m_width, m_height);
}
```

Embedded Software Porting

■ Program5

請使用自行設計之面積計算類別(d.h, d.cpp)，建立靜態連接檔(*.a)，並於program4.cpp呼叫使用。



```
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# g++ program4.cpp libd.a -o program5
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# ./program5
triangle area = 8.00
circle area = 300.00
rectangle area = 48.00
```

ldd : 查看依賴的函式庫

```
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# ldd program5
linux-vdso.so.1 => (0x00007fffd65f1000)
libstdc++.so.6 => /usr/lib/x86_64-linux-gnu/libstdc++.so.6 (0x00007f55b5a7c000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f55b56b2000)
libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007f55b53a9000)
/lib64/ld-linux-x86-64.so.2 (0x00007f55b5dfe000)
libgcc_s.so.1 => /lib/x86_64-linux-gnu/libgcc_s.so.1 (0x00007f55b5193000)
```

經過靜態連接檔後，執行檔已經不需要d.cpp即可執行~

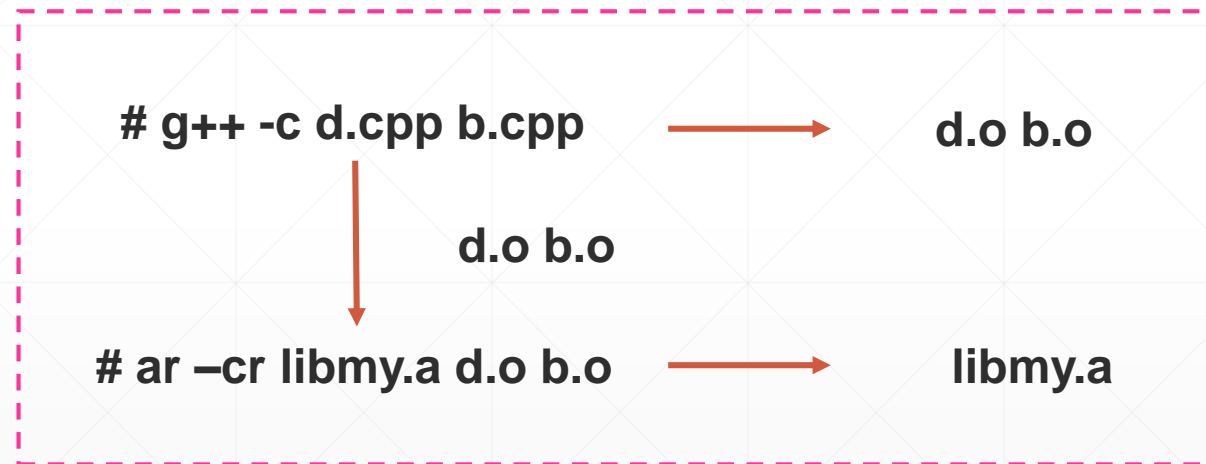
Embedded Software Porting

- 靜態函式庫 (static library)

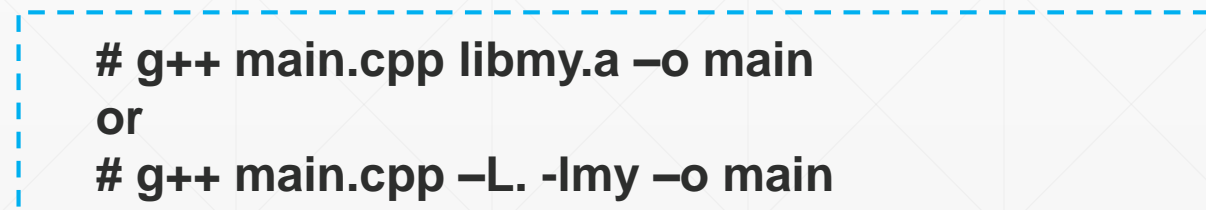
由物件檔案 (object files) 所構成的封裝檔，檔案名稱以 lib 開頭，副檔名則為 .a 。

- 透過ar工具打包所需要的物件檔封裝成靜態函式庫

編輯靜態函式庫



使用靜態函式庫編輯執行檔




```

root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# g++ -c d.cpp b.cpp
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# ar -cr libd.a d.o b.o
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# readelf -s libd.a

```

File: libd.a(d.o)

Symbol table '.symtab' contains 17 entries:

Num:	Value	Size	Type	Bind	Vis	Ndx	Name
0:	0000000000000000	0	NOTYPE	LOCAL	DEFAULT	UND	
1:	0000000000000000	0	FILE	LOCAL	DEFAULT	ABS	d.cpp
2:	0000000000000000	0	SECTION	LOCAL	DEFAULT	1	
3:	0000000000000000	0	SECTION	LOCAL	DEFAULT	3	
4:	0000000000000000	0	SECTION	LOCAL	DEFAULT	4	
5:	0000000000000000	0	SECTION	LOCAL	DEFAULT	5	
6:	0000000000000000	0	SECTION	LOCAL	DEFAULT	7	
7:	0000000000000000	0	SECTION	LOCAL	DEFAULT	8	
8:	0000000000000000	0	SECTION	LOCAL	DEFAULT	6	
9:	0000000000000000	49	FUNC	GLOBAL	DEFAULT	1	_ZN5TArea16setTrianglePar
10:	0000000000000032	30	FUNC	GLOBAL	DEFAULT	1	_ZN5TArea14setCircleParam
11:	0000000000000050	48	FUNC	GLOBAL	DEFAULT	1	_ZN5TArea17setRectanglePa
12:	0000000000000080	84	FUNC	GLOBAL	DEFAULT	1	_ZN5TArea15getTriangleAre
13:	0000000000000000	0	NOTYPE	GLOBAL	DEFAULT	UND	mulfunc
14:	0000000000000000	0	NOTYPE	GLOBAL	DEFAULT	UND	divfunc
15:	00000000000000d4	69	FUNC	GLOBAL	DEFAULT	1	_ZN5TArea13getCircleAreaE
16:	0000000000000011a	56	FUNC	GLOBAL	DEFAULT	1	_ZN5TArea16getRectangleAr

File: libd.a(b.o)

Symbol table '.symtab' contains 12 entries:

Num:	Value	Size	Type	Bind	Vis	Ndx	Name
0:	0000000000000000	0	NOTYPE	LOCAL	DEFAULT	UND	
1:	0000000000000000	0	FILE	LOCAL	DEFAULT	ABS	b.cpp
2:	0000000000000000	0	SECTION	LOCAL	DEFAULT	1	
3:	0000000000000000	0	SECTION	LOCAL	DEFAULT	2	
4:	0000000000000000	0	SECTION	LOCAL	DEFAULT	3	
5:	0000000000000000	0	SECTION	LOCAL	DEFAULT	5	
6:	0000000000000000	0	SECTION	LOCAL	DEFAULT	6	
7:	0000000000000000	0	SECTION	LOCAL	DEFAULT	4	
8:	0000000000000000	20	FUNC	GLOBAL	DEFAULT	1	addfunc
9:	0000000000000014	18	FUNC	GLOBAL	DEFAULT	1	subfunc
10:	0000000000000026	19	FUNC	GLOBAL	DEFAULT	1	mulfunc
11:	0000000000000039	26	FUNC	GLOBAL	DEFAULT	1	divfunc

Embedded Software Porting

■ Program6

請使用自行設計之面積計算類別(d.h, d.cpp)，建立動態連接檔(*.so)，並於program4.cpp呼叫使用。

d.cpp



libd.so

```
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# g++ program4.cpp libd.so -o program6
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# cp libd.so /usr/lib
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# ./program6
triangle area = 8.00
circle area = 300.00
rectangle area = 48.00
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice#
```

ldd : 查看依賴的函式庫

```
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# ldd program6
linux-vdso.so.1 => (0x00007ffec373e000)
libd.so => /usr/lib/libd.so (0x00007f458577d000)
libstdc++.so.6 => /usr/lib/x86_64-linux-gnu/libstdc++.so.6 (0x00007f45853fb000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f4585031000)
libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007f4584d28000)
/lib64/ld-linux-x86-64.so.2 (0x00007f458597f000)
libgcc_s.so.1 => /lib/x86_64-linux-gnu/libgcc_s.so.1 (0x00007f4584b12000)
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice#
```

執行檔執行時，不需要d.cpp，但需要有動態連接檔才可順利執行~

Embedded Software Porting

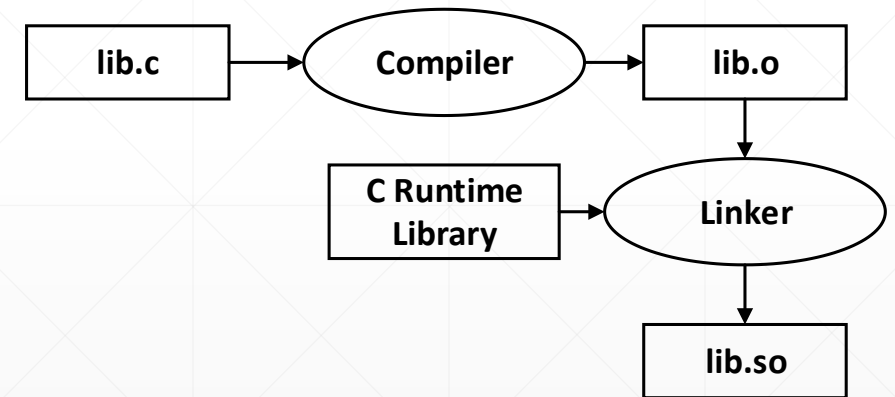
■ 動態函式庫、共享函式庫 (shared library)

程式實際開始執行時，才會被載入的函式庫。檔案名稱以 lib 開頭，副檔名則為 .so。

編輯動態函式庫

```
# g++ -fPIC -shared d.cpp b.cpp -o libd.so → libmy.so
or
# g++ -fPIC -c d.cpp
# g++ -fPIC -c b.cpp
# g++ -shared d.o b.o -o libd.so → libmy.so
```

可透過 `readelf -s` 觀察 .so 檔輸出符號是否正確



使用動態函式庫編輯執行檔

```
# g++ main.cpp libmy.so -o main
or
# g++ main.cpp -L. -lmy -o main
```



Embedded Software Porting

```

33: 0000000000000000 0 FILE LOCAL DEFAULT ABS d.cpp
34: 0000000000000000 0 FILE LOCAL DEFAULT ABS b.cpp
35: 0000000000000000 0 FILE LOCAL DEFAULT ABS crtstuff.c
36: 00000000000000bc8 0 OBJECT LOCAL DEFAULT 14 __FRAME_END__
37: 00000000000200e50 0 OBJECT LOCAL DEFAULT 17 __JCR_END__
38: 0000000000000000 0 FILE LOCAL DEFAULT ABS
39: 00000000000200e58 0 OBJECT LOCAL DEFAULT 18 _DYNAMIC
40: 00000000000201030 0 OBJECT LOCAL DEFAULT 21 __TMC_END__
41: 00000000000201028 0 OBJECT LOCAL DEFAULT 21 __dso_handle
42: 000000000000009e0 0 NOTYPE LOCAL DEFAULT 13 __GNU_EH_FRAME_HDR
43: 00000000000201000 0 OBJECT LOCAL DEFAULT 20 __GLOBAL_OFFSET_TABLE__
44: 000000000000009ab 26 FUNC GLOBAL DEFAULT 10 __divfunc
45: 00000000000000972 20 FUNC GLOBAL DEFAULT 10 __addfunc
46: 0000000000000000 0 NOTYPE WEAK DEFAULT UND __cxa_finalize
47: 00000000000000986 18 FUNC GLOBAL DEFAULT 10 __subfunc
48: 000000000000006b8 0 FUNC GLOBAL DEFAULT 7 __init
49: 0000000000000000 0 NOTYPE WEAK DEFAULT UND __ITM_registerTMCloneTable
50: 00000000000000870 48 FUNC GLOBAL DEFAULT 10 __ZN5TArea17setRectanglePa
51: 00000000000000852 30 FUNC GLOBAL DEFAULT 10 __ZN5TArea14setCircleParam
52: 00000000000000820 49 FUNC GLOBAL DEFAULT 10 __ZN5TArea16setTrianglePar
53: 0000000000000000 0 NOTYPE WEAK DEFAULT UND __ITM_deregisterTMCloneTab
54: 000000000000008f4 69 FUNC GLOBAL DEFAULT 10 __ZN5TArea13getCircleAreaE
55: 00000000000201030 0 NOTYPE GLOBAL DEFAULT 22 __bss_start
56: 000000000000009c8 0 FUNC GLOBAL DEFAULT 11 __fini
57: 00000000000000998 19 FUNC GLOBAL DEFAULT 10 __mulfunc
58: 00000000000201030 0 NOTYPE GLOBAL DEFAULT 21 __edata
59: 00000000000201038 0 NOTYPE GLOBAL DEFAULT 22 __end
60: 0000000000000093a 56 FUNC GLOBAL DEFAULT 10 __ZN5TArea16getRectangleAr
61: 000000000000008a0 84 FUNC GLOBAL DEFAULT 10 __ZN5TArea15getTriangleAre
62: 0000000000000000 0 NOTYPE WEAK DEFAULT UND __Jv_RegisterClasses
63: 0000000000000000 0 NOTYPE WEAK DEFAULT UND __gmon_start__

```

readelf -s libd.so

Embedded Software Porting

- 注意動態連接檔路徑
 - 通常程式執行時，動態連接檔的搜尋路徑為：`/usr/lib`。

如果`/usr/lib`中沒有`libd.so`.....

```
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# ldd program6
linux-vdso.so.1 => (0x00007fffdbde1000)
libd.so => not found
libstdc++.so.6 => /usr/lib/x86_64-linux-gnu/libstdc++.so.6 (0x00007fdc6effe000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fdc6ec34000)
libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007fdc6e92b000)
/lib64/ld-linux-x86-64.so.2 (0x00007fdc6f380000)
libgcc_s.so.1 => /lib/x86_64-linux-gnu/libgcc_s.so.1 (0x00007fdc6e715000)
```

→ `cp libd.so /usr/lib`

```
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# ldd program6
linux-vdso.so.1 => (0x00007ffec373e000)
libd.so => /usr/lib/libd.so (0x00007f458577d000)
libstdc++.so.6 => /usr/lib/x86_64-linux-gnu/libstdc++.so.6 (0x00007f45853fb000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f4585031000)
libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007f4584d28000)
/lib64/ld-linux-x86-64.so.2 (0x00007f458597f000)
libgcc_s.so.1 => /lib/x86_64-linux-gnu/libgcc_s.so.1 (0x00007f4584b12000)
```

Embedded Software Porting

- 注意動態連接檔路徑
 - 亦可於執行執行檔時，將搜尋路徑改道當下資料夾。

```
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# ldd program6
linux-vdso.so.1 => (0x00007ffffdbde1000)
libd.so => not found
libstdc++.so.6 => /usr/lib/x86_64-linux-gnu/libstdc++.so.6 (0x00007fddc6effe000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fddc6ec34000)
libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007fddc6e92b000)
/lib64/ld-linux-x86-64.so.2 (0x00007fddc6f380000)
libgcc_s.so.1 => /lib/x86_64-linux-gnu/libgcc_s.so.1 (0x00007fddc6e715000)
root@hank-X302LJ:/home/hank/2019_NTUT_embedded_course/practice# LD_LIBRARY_PATH=. ./program6
triangle area = 8.00
circle area = 300.00
rectangle area = 48.00
```

LD_LIBRARY_PATH=. ./program6