

[回頁頂 \(/gnu-make/naive-](#)[makefile/#main-content\)](#)[回首頁 \(/\)](#)

## Makefile 教學：即使是新手也可以

le

[sourcedoc.com/\)](#)

[sourcedoc.com/gnu-make/\)](#)

/ 即使是新手也可以馬上開始寫 Makefile

最後修改日期為 MAY 27, 2018

我們這裡展示一個初階的 (naive) Makefile，本例摘自筆者先前練資料結構的微型程式。在這裡，我們刻意地少用 Makefile 特有的語法，只用先前提到的基本概念來寫 Makefile，讓整個 Makefile 看起來更單純：

```
all: dynamic

dynamic:
    gcc -fPIC -c deque_int.c
    gcc -shared -o libalgodequei.so deque_int.o

static: deque_int.o
    ar rcs -o libalgodequei.a deque_int.o

test: test_deque_int.out
    ./test_deque_int.out
    echo $$?

test_deque_int.out: deque_int.o test_deque_int.o
    gcc -o test_deque_int.out test_deque_int.o deque_int.o

test_deque_int.o:
    gcc -c test_deque_int.c

deque_int.o:
    gcc -c deque_int.c

clean:
    rm -f test_deque_int.out *.o *.so *.a
```

贊助商連結

Google 已關閉廣告

贊助商連結

Google 已關閉廣告

專案的使用方式如下：

make 或 make dynamic，制作動態函式庫

**ServerBank**  
資訊設備升級必買，企業所需一次購足



- `make clean`：清除由編譯器生成的檔案

預設的任務是 `all`，而 `all` 相依於 `dynamic`，故我們不加入任何參數時，預設動作為製作動態函式庫。節錄程式碼如下：

回首頁()

`all: dynamic`



```
int.o deque_int.c
dequei.so deque_int.o
```

`static`，而 `static` 相依於 `deque_int.o`。節錄程式碼如

```
static: deque_int.o
    ar rcs -o libalcodequei.a deque_int.o

deque_int.o:
    gcc -c deque_int.c
```

使用檔案做為任務名稱的好處在於，當我們第二次執行相關的任務時，`make` 會略過該任務。這樣的設計是由於早期的電腦運行速度較慢，編譯中大型程式往往需花數十分鐘至數小時，若能略去不必要的任務，就可以省下一些編譯程式的時間。

執行測試程式的任務為 `test`，該任務有兩層的相依性。節錄程式碼如下：

```
test: test_deque_int.out
    ./test_deque_int.out
    echo $$?

test_deque_int.out: deque_int.o test_deque_int.o
    gcc -o test_deque_int.out test_deque_int.o deque_int.o

test_deque_int.o:
    gcc -c test_deque_int.c

deque_int.o:
    gcc -c deque_int.c
```

贊助商連結

Google 已關閉廣告

贊助商連結

Google 已關閉廣告

看起來比先前的任務略長，但其實只是以 GCC 依序編譯 C 程式碼，如果讀者從任務相依性的源頭追蹤任務流程，就知道這段程式碼所代表的意義。(可由上閱讀此段程式碼。)

其中唯一比較有點小技巧的地方在於 `echo $$?`。在類 Unix 系統中，`$?` 是一個內建環境變數，表示最後一個程式執行後所回傳的錯誤碼，若為 0 表示程式沒有錯誤，若為其他數字表示程式有某種錯誤。`echo $?` 表示印出上一個程式的錯誤碼；由於在 Makefile 中，`$` (錢字號) 有特殊意義，所以要用 `$$` 表示該字號為一個 `$` 字串而非 Makefile 的特殊符號。

**ServerBank**  
資訊設備升級必買，企業所需一次購足



```
clean:
    rm -f test_deque_int.out *.o *.so *.a
```

回頁頂 (/gnu-make/naive-  
一般在網路上看到  
makefile/#main-content)

一般在網路上看到的 **Makefile** 都充滿著各種難以理解的符號和規則，越強的高手寫的 **Makefile** 越難理解；甚至有些 **Makefile** 是用 **Autotools** 或 **CMake** 自動生成的，而自動生成的 **Makefile** 往往很冗長而難以閱讀。但即使我們不用特殊

寫出可用的 Makefile。



版本控制系統，寫 **Makefile** 時並不需要追求一步到  
le 沒那麼漂亮，只要能夠正確運作，之後再將  
(oring) 得漂亮一些即可。**Makefile** 畢竟是軟體專案  
重要的是能夠正確地編譯專案，而不用過度追求語

分享本文

 (<https://www.facebook.com/sharer.php?>

u=https%3a%2f%2fopensourcedoc.com%2fgnu-make%2fnaive-makefile%2f) 


(<https://twitter.com/intent/tweet?url=https%3a%2f%2fopensourcedoc.com%2fgnu-make%2fnaive->

makefile%2f&text=%5bGNU%20Make%5d%20Makefile%20%e6%95%99%e5%ad%b8%ef%bc%9a%e5%


 (<https://www.linkedin.com/shareArticle?>

mini=true&amp;url=https%3a%2f%2fopensourcedoc.com%2fgnu-make%2fnaive-

makefile%2f&title=%5bGNU%20Make%5d%20Makefile%20%E6%95%99%E5%AD%B8%EF%BC%9A%E5%

 (<https://lineit.line.me/share/ui?url=https%3a%2f%2fopensourcedoc.com%2fgnu-make%2fnaive->

makefile%2f&text=%5bGNU%20Make%5d%20Makefile%20%e6%95%99%e5%ad%b8%ef%bc%9a%e5%

 (<https://web.skype.com/share?url=https%3a%2f%2fopensourcedoc.com%2fgnu-make%2fnaive->

贊助商連結

贊助商連結

Google 已關閉廣告

efile%2f&text=%5bGNU%20Make%5d%20Makefile%20%e6%95%99%e5%ad%b8%  
(http://www.evernote.com/clip.action?

<https://opensource.com/gnu-make-naive>

efile%2f&title=%5bGNU%20Make%5d%20Makefile%20%e6%95%99%e5%ad%b8%  
(https://mail.google.com/mail/?

.../cm&to=&su=%5bGNU%20Make%5d%20Makefile%20%e6%95%99%e5%ad%b8%  
...%2fnaive-makefile%2f) Y! (http://compose.mail.yahoo.com/?

subject=%5bGNU%20Make%5d%20Makefile%20%e6%95%99%e5%ad%b8%ef%b  
e%2fnaive-makefile%2f) (mailto:?)

```
ject=%5bGNU%20Make%5d%20Makefile%20%e6%95%99%e5%ad%b8%ef%bc%9
:e%2fnaive-makefile%2f)
```

本站

Google 已關閉廣告

# ServerBank

## 資訊設備升級必買，企業所需一次購足



What do you think?

0 Responses

👍

🤔

❤️

😮

😡

😞

Upvote

Funny

Love

Surprised

Angry

Sad

回首頁 (/)

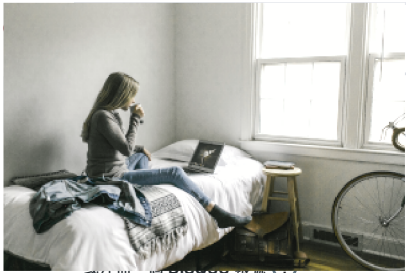
0則留言

開源技術教學網

🔒

1 登錄

按最佳排序



姓名

來做第一個留言的人吧！

Copyright (c) 2014-2022 Michelle Chen. All Rights Reserved.

贊助商連結

Google 已關閉廣告

贊助商連結

Google 已關閉廣告