

Perl 的文化

本書是 Perl 文化的一部分，所以我們並不指望能夠對 Perl 文化有個完整的交代。我們只能端出兩道開胃小菜（一段簡史，以及一點藝術的展現 — 有些人可能會說：「只有這麼一釘點藝術作品呀！」）。接下來的 Perl 文化大餐請到 www.perl.org 以及 www.perl.com 享用。（Larry 將他個人正式對外公開的講話（瘋言瘋語）擺在 www.wall.org/~larry。）或者你可以跟某些 Perl 程式員打交道。我們無法告訴你這些 Perl 程式員是哪些人 — 這些 Perl 程式員都有一個通病，就是太雞婆了。

27.1 歷史讓 Perl 成真

想知道 Perl 為什麼會變成這付德性，必須先了解 Perl 出現的原因。讓我們搬出塵封已久的歷史書 …

1986 年，Larry 是「多層安全廣域網路」(multilevel-secure wide-area network) 發展計劃的系統程式員。他負責在美國西岸安裝三台 VAXen 以及三台 Sun 主機，並利用一條加密過的、速率為 1200-baud 的串列連線，連接到東岸具有類似配置的地方。因為 Larry 的主要任務是技術支援（他並非該計畫的程式員，只是一位系統專家而已），正好能將自己的三個德性（懶散、不耐煩以及傲慢）發揚光大。於是他發展以及改進了各種有用的工具 — 比如 *rn*、*patch* 以及 *warp*。【註】

註 大約在這個時候 Larry 被「功能蔓延主義」(feeping creaturism)【譯註】迷了心竅，他一直以這是生命的基本需要為藉口，強烈地主張「在加入一個功能就好了」。畢竟，假使生命原本就很複雜的話，程式不也是這樣嗎？特別是像 *rn* 這樣的程式（可以為你讀取新聞），應該被當成高等人工智慧計劃來發展。當然，有些人說 *patch* 程式已經太聰明了。

譯註 feeping creaturism 其實就是 feature creep；這是強調語氣的寫法。

有一天，就在 Larry 把 *rn* 程式分解成許多程式片段並且放入目錄之後，大經理走到他面前說：「Larry，我們需要能夠處理這六台 VAXen 以及六台 Sun 之架構的控管系統。一個月之內就用得著，開始動手吧！」

雖然 Larry 不是個逃避責任的人，不過他心裡盤算著：最好能夠利用現有的程式搞出一個可以兼管東西兩岸機器的控管系統，讓東西兩岸的系統管理員可以檢視問題報告，以便進行認可與控制的動作。答案只有一個字：B-news。【註】

Larry 著手為這些機器進行 news 系統的安裝，並且加上兩支控制命令：*append* 命令可以為現有的文章附加內容，*synchronize* 命令可以維持東西二岸機器之文章數量的一致性。控管功能可以透過 RCS (Revision Control System) 來達成，而認可與遞交的動作則可以透過 news 系統與 *rn* 來達成。到目前為止，一切都很美好。

然後，大經理要求 Larry 產生報告。網路新聞分別存放在主要伺服器上的各個檔案中，而這些檔案之間有許多交互參照的情形。Larry 的第一個念頭是「使用 *awk* 好了。」遺憾的是，那時的 *awk* 無法根據檔案裡的資訊，處理多個檔案的開啟與關閉。Larry 並不想去設計一個特殊用途的工具。結果，一個新的程式語言於焉誕生。

這個工具最初並不叫 Perl。Larry 曾跟同事以及好友們 (Dan Faigin，這篇歷史的撰寫人；以及 Larry 的連襟 Mark Biggar，他在設計之初幫了很大的忙) 交換過意見。Larry 經過深思熟慮之後，決定將字典裡長度在三或四個字母的單字排除在外。最早期曾經一度命名為 Gloria，也就是日後 Larry 的親蜜愛人 (以及老婆)。不過他很快就體認到，這個名字可能會引發不少的家庭糾紛。

然後這工具的名稱成為了 Pearl，這名稱後來突變成我們今日的 Perl。部分原因是 Larry 發現已經存在一個稱為 PEARL 的語言，不過最主要的因素是 Larry 懶得打那麼多的字。於是就順理成章地使用四個字母的 Perl。(在你心中是否還殘存一個過去的假象，以為它是 Practical Extraction And Report Language [實用性摘要與報表式語言] 等單字第一個字母所拼成的。)

早期的 Perl 欠缺許多今日的功能。樣式比對 (pattern matching)、檔案代號 (filehandle)、純量 (scalar) 以及格式 (format) 都是當初就有的功能，不過只有一丁點的函式，當時還沒有關連式陣列 (associative array)，而且只提供了一個蹩腳的正規表示式 (regular expression) 實作 (借用自 *rn*)。雖然 manpage 只有 15 頁長，不過 Perl 在當時的執行速度比 *sed* 以及 *awk* 還快，於是我們開始在計劃裡的其他應用中使用 Perl。

●.....
註 這是 Usenet (傳送網路新聞的邏輯網路) 傳輸軟體的第二個實作版本。

後來有其他單位想借調 Larry。有一天，出現另一位大經理，前來要求說：「Larry 來支援研發的工作吧。」於是 Larry 把 Perl 也帶了去，而且他發現 Perl 漸漸被發展成系統管理的好工具。Larry 從 Henry Spencer 那邊取得了一份出色的正規表示式套件，並且將它改得面目全非。而 Larry 所加入的功能大部分是他自己想要的，有少部分是其他人想要的。然後 Larry 在網路上發表 Perl。【註一】接下來又是一連串的歷史事件。【註二】

Perl 的發行年表如下：Perl 1.0 的發行日為 1987 年 12 月 18 日；有些人還會紀念 Perl 的生日。Perl 2.0 的發行日在 1988 年 6 月之後，並由 Randal Schwartz 設計出傳說中的 Just Another Perl Hacker 簽名。1989 年，Tom Christiansen 首先在 Baltimore（巴爾的摩港市）的 Usenix 大會發表 Perl 的教學課程。Perl 3.0 的發行日在 1989 年 10 月，這是 Perl 第一次以 GNU 公眾版權流通的版本。

1990 年 3 月，Larry 寫下了第一首 Perl 詩（參閱下一節）。Larry 與 Randal 編寫出本書的第一版（Pink Camel；粉紅皮駱駝書）並於 1991 年初出版。同時也發行了 Perl 4.0；這是個以 Artistic License 以及 GPL 兩種版權流通的版本。

在眾人的期盼之下 Perl 5 於 1994 年 10 月問世。這個版本將 Perl 完全改寫，並且納入了 object（物件）與 module（模組）的特性。Perl 5 的問世甚至連《Economist》也加以報導。1995 年，CPAN 被正式引見給 Perl 社群。1996 年，Jon Orwant 發行《Perl Journal》雜誌。經過長期的醞釀，本書第二版（Blue Camel；藍皮駱駝書）於同年秋天出版。1997 年夏天，第一屆的 O'Reilly Perl 研討會（TPC）在 San Jose, California 舉辦。現在令人矚目的事件幾乎每天都會發生，所以其餘的歷史事件，請看 CFAST（Comprehensive Perl Arcana Society Tapestry，位於 <http://history.perl.org>）所提供的及時新聞。

27.2 Perl 詩

附文展示的是 1990 年愚人節在 Usenet 出現的偽造文件。我們並未加上任何的註解，目的只是在告訴讀者，一般的程式語言多麼沒人性。對 Larry 而言，文學之美處處可見。附文中有提到一首名為 Black Perl 的佳作，它最初是用 Perl 3 寫成的，不過已經無法用 Perl 5 剖析了。

●.....
註一 更令人驚訝的是，Larry 在 JPL（Jet Propulsion Lab）任職時，依然持續為 Perl 發表新的版本。而今，由其他人接手大部分的工作，而 Larry 則宣稱自己在 O'Reilly & Associates（一個出版電腦以及其他書籍的小公司）任職。

註二 我在此處為這段歷史做個註解。在 Perl 草創之際，由於正想對 *rn* 進行大改，所以已經將它分解成許多的程式片段。自從 Larry 開始投身於 Perl 的發展，就再也沒有碰過 *rn* 程式了。到目前為止它還是一些程式片段。有時想到了，Larry 還揚言要以 Perl 來改寫 *rn* 程式，但是他從未認真過。

作品數量能夠勝過 Larry 的僥倖者應推當代 Perl 女詩人，Sharon Hopkins。Sharon 是一個多產的 Perl 詩人，她曾在 Usenix Winter 1992 技術研討會上發表過一篇有關 Perl 詩的論文，標題為《Camels and Needles: Computer Poetry Meets the Perl Programming Language》（駱駝與針：當電腦詩遇上 Perl 程式語言）。除了是一位最多產的 Perl 詩人外，Sharon 也是發行範圍最廣的作家，你甚至可以在《Economist》以及《Guardian》上看到她的詩篇：

```
#!/usr/bin/perl

APPEAL:

listen (please, please);

open yourself, wide;
    join (you, me),
connect (us,together),

tell me.

do something if distressed;

    @dawn, dance;
    @evening, sing;
    read (books,$poems,stories) until peaceful;
    study if able;

    write me if-you-please;

sort your feelings, reset goals, seek (friends, family, anyone);

    do*not*die (like this)
    if sin abounds;

keys (hidden), open (locks, doors), tell secrets;
do not, I-beg-you, close them, yet.

                                accept (yourself, changes),
                                bind (grief, despair);

require truth, goodness if-you-will, each moment;

select (always), length(of-days)

# listen (a perl poem)
# Sharon Hopkins
# rev. June 19, 1995
```

Perl 詩

Article 970 of comp.lang.perl:
 Path: jpl-devvax!pl-dexxav!lwall
 From: lwall@jpl-dexxav.JPL.NASA.GOV (Larry Wall)
 Newsgroups: news.groups,rec.arts.poems,comp.lang.perl
 Subject: CALL FOR DISCUSSION: comp.lang.perl.poems
 Message-ID: <0401@jpl-devvax.JPL.NASA.GOV>
 Date: 1 Apr 90 00:00:00 GMT
 Reply-To: lwall@jpl-devvax.JPL.NSAS.GOV (Larry Wall)
 Organization: Jet Propulsion Laboratory, Pasadena, CA
 Lines: 61

我注意到人們迫切需要一個地方來抒發情感以及探討技術。我發現有些人寄給我的文章並不適合發表在任何新聞群組上。或許是因為我最近曾在 `comp.lang.perl` 和 `rec.arts.poems` 發表過文章，但似乎有人想要以 **Perl** 來寫詩，並問我應該在何處發表他們的作品。下面是一些例子：

一位研究生（於期末週）寫了下面的三行俳句詩（**haiku**）：

```
study, write, study,
do review (each word) if time.
close book. sleep? what's that?
```

羅德岱堡（**Fort Lauderdale**）有人寫了這麼一首詩：

```
sleep, close together,
sort of sin each spring & wait;
50% die
```

有位無名氏寫了一首《**Black Perl**》。（無疑的，就連《珍珠篇》的詩人 [Pearl poet] 也會為之撼動。）

```
BEFOREHAND: close door, each window & exit; wait until time.
open spellbook, study, read (scan, select, tell us);
write it, print the hex while each watches,
reverse its length, write again;
kill spiders, pop them, chop, split, kill them.
unlink arms, shift, wait & listen (listening, wait),
sort the flock (then, warn the "goats" & kill the "sheep");
kill them, dump qualms, shift moralities,
values aside, each one;
die sheep! die to reverse the system
you accept (reject, respect);
next step,
kill the next sacrifice, each sacrifice,
wait, redo ritual until "all the spirits are pleased";
do it ("as they say").
do it(*everyone***must***participate***in***forbidden**s*e*x*).
return last victim; package body;
exit crypt (time, times & "half a time") & close it,
```

（續下頁）

```
select (quickly) & warn your next victim;
AFTERWORDS: tell nobody.
wait, wait until time;
wait until next year, next decade;
    sleep, sleep, die yourself,
die at last
```

我試過了，儘管它符合 Perl 的語法，可是它似乎一點用處也沒有。不過，我反倒很高興 ... 因為這樣我就可以順水推舟地成立 `comp.lang.perl.poems`，好讓這些文章有地方發表，所以我們就不必為 `perl` 和詩是否要放在一起而爭論不休了。或者，我應該成立一個 `rec.arts.poems.perl` 來放這些詩，比如以上這些詩只是語法符合而已，根本沒有任何用處。（畢竟，在 `rec.arts.poems` 中已有前例可循。）既然我們也要為詩成立 `comp.lang.perl.poems`，就讓我野人獻曝發表一首我的三行俳句詩：

```
print STDOUT q
Just another Perl hacker,
unless $spring
```

Larry Wall lwall@jpl-devvax.jpl.nasa.gov