

---

# Urial 指南

*cat-v.org* 哲学与档案精粹

---

*“The Internet is not for sissies.”*

— Paul Vixie

*“UNIX has become the victim of cancerous growth  
at the hands of organizations such as UCB.”*

— Rob Pike, USENIX 1983

## Random Contrarian Insurgent Organization

一个反主流的、激进的、关于极简主义  
与 Plan 9 后裔操作系统的档案馆

蒸馏自 [cat-v.org](https://cat-v.org) 及其全部子域

2026 年 5 月 11 日

关于本书。本指南由 `cat-v.org` 主站及其 16 个子域名的离线镜像归纳整理而成。所有思想观点、引文、技术描述均来自 `cat-v.org` 社区，原作者保留各自的著作权。本书旨在以中文形式重新组织并解读这一散落在十数个子域上的”反主流”档案，使之能够被一次性、系统地阅读。

阅读方式。各章节之间相对独立，可以非线性地翻阅。若你从未听说过 Plan 9，建议先读第一章（哲学）与第二章（Plan 9）。若你只对引语与笑话感兴趣，直接跳到第八章。

关于“**Urial**”。`cat-v.org` 的精神创始人 Uriel M. Pereira 于 2012 年早逝，他是 9front、suckless、werc 等多个项目的灵魂人物。“Urial”是对“Uriel”的致意性误拼，也是本指南所在工作目录的名字。

---

# 目录

---

前言：什么是 <b>cat-v.org</b>	1
第一章 哲学：反对意见的百科全书	3
1.1 Worse is Better 与 The Right Thing	3
1.2 cat -v Considered Harmful	3
1.3 所有软件都很糟	4
1.4 Worse-is-Worse：现代 Unix 的双重背叛	5
1.5 Suckless 与 9front：哲学的两个邻居	5
第二章 <b>Plan 9 from Bell Labs</b>	6
2.1 为什么需要再造一个 Unix	6
2.2 9P：一个简单到可怕的文件协议	7
2.3 结构：fileservers、cpu、terminal	7
2.4 Plan 9 至今：4th Edition、9front、9legacy、9atom	7
2.5 Glenda：那只兔子	8
第三章 <b>Inferno：Plan 9 的虚拟机版</b>	9
3.1 为什么会有 Inferno	9
3.2 Limbo 语言	9
3.3 Inferno 当下的位置	10
第四章 <b>Sam 与 Acme：Rob Pike 的编辑器</b>	11
4.1 Sam：命令语言编辑器	11
4.2 Acme：鼠标 + 文本 = 用户界面	11
4.3 Sam 与 Acme 在今天的可用性	12
第五章 <b>Werc：反 Web 的反框架</b>	13
5.1 口号	13
5.2 核心信条	13
5.3 它驱动着什么	13
5.4 rc shell 简介	14

第六章 系统软件研究已无关紧要	15
6.1 演讲背景	15
6.2 Pike 的论点	15
6.3 Linux 创新吗?	16
6.4 这与 cat-v.org 哲学的关系	16
第七章 <b>Harmful</b> 百科: 被认为有害的东西	17
7.1 结构	17
7.2 典型条目: cat -v	17
7.3 典型条目: 动态链接	18
7.4 典型条目: OO 编程	18
7.5 典型条目: XML	18
7.6 有害物对照表 (再次)	19
第八章 语录与 <b>Fortunes</b>	20
8.1 Pike	20
8.2 Thompson	20
8.3 Hoare	21
8.4 cat-v IRC 频道精选	21
8.5 Plan 9 fortunes (节选)	21
第九章 社区与连接	23
9.1 IRC: #cat-v	23
9.2 IWP9: 国际 Plan 9 与 Inferno 工作坊	23
9.3 GSoC: Plan 9 / Inferno 在 Google Summer of Code	23
9.4 NineTimes: 新闻档案	23
附录 A 资源索引	25
A.1 cat-v.org 全部子域	25
A.2 关键论文清单	25
A.3 命令速查	26
附录 B 跋: 为什么仍然要读 cat-v.org	28

---

# 前言：什么是 cat-v.org

---

“*cat* 不是用来给文件打行号的，不是用来压缩多个空行的，也不是用来看不可打印 ASCII 字符的；*cat* 是用来把文件首尾相接 (*concatenate*) 的。”

— Rob Pike, *cat -v Considered Harmful*, USENIX 1983

## 一个名字的来源

1983 年夏，Rob Pike 在 USENIX 大会上做了一个题为 *UNIX Style, or cat -v Considered Harmful* 的演讲。他与 Brian Kernighan 合写的论文 *Program Design in the UNIX Environment* 随后发表在 1984 年 10 月的《AT&T 贝尔实验室技术期刊》上，是后来那本被尊为“Unix 圣经”的 *The Unix Programming Environment* 的前奏。

那次演讲的核心观点是：UCB（加州大学伯克利分校）正在让 Unix 罹患癌变。4.2BSD 比 Version 5 大了一个数量级——但并没有好到十倍。*cat* 是这一癌变的标志：本应只是把若干文件接起来的工具，被加上了 *-v*（可视化不可打印字符）、*-n*（编号）、*-s*（压缩空行）等一堆开关。这违背了“一个程序，做好一件事”的 Unix 哲学。

“可惜，他们的建议被彻底无视了。今天的 Unix 已被他们当年警告的那些错误所淹没。”——这正是 [cat-v.org](#) 主页对这段历史的总结。2000 年代中期，一群对此心怀不满的程序员——以 Uriel M. Pereira 为核心——开了 [cat-v.org](#)，把这种“反 Unix 癌变”的论调汇总成了一座网上百科。

## 站点版图

[cat-v.org](#) 自称 **Random Contrarian Insurgent Organization**（随机反主流暴动组织）。它不是一个统一的项目，而是一组围绕共同审美聚集的小站：

[cat-v.org](#) 主站，索引与社区入口。

[doc.cat-v.org](#) 文档归档：贝尔实验室的论文、Plan 9 与 Inferno 的全部 papers、Feynman 讲稿、Henry Spencer 著作、Unix 历史文档等数百篇。

[harmful.cat-v.org](#) “Considered Harmful” 百科：被认为有害的软件、标准、社会现象的清单。

**9p.cat-v.org** 9P 分布式文件协议的资源汇总。

**acme.cat-v.org** Rob Pike 的程序员界面 Acme 的资料。

**sam.cat-v.org** Rob Pike 的命令语言编辑器 Sam 的资料。

**werc.cat-v.org** cat-v.org 自己的“反 web 反框架”引擎。

**glenda.cat-v.org** Plan 9 吉祥物（兔子 Glenda）的非官方主页与图廊。

**quotes.cat-v.org** 语录档案。

**fortunes.cat-v.org** 各操作系统 fortune 文件汇编（9front、Plan 9、OpenBSD、FreeBSD 等）。

**man.cat-v.org** Plan 9 系手册页大全（9front、Inferno、Plan B、9atom、Plan 9 1<sup>st</sup>-3<sup>rd</sup> edition、p9p...）。

**ninetimes.cat-v.org** Plan 9 / Inferno / 贝尔实验室操作系统的新闻档案。

**plan9front.cat-v.org** Plan 9 的活跃分支 9front 的镜像。

**iwp9.cat-v.org** International Workshops for Plan 9 and Inferno 会议的非官方主页。

**go-lang.cat-v.org** 2012 年之前的 Go 语言资源汇总（已停更）。

**gsoc.cat-v.org** Plan 9 / Inferno 在 Google Summer of Code 中的存档。

**fortunes.cat-v.org** 见上。

把这些站点的内容串起来读，你会得到一种相当连贯的、关于“软件应该是什么样”的世界观。本指南就是这一串读体验的中文蒸馏版。

# 哲学：反对意见的百科全书

“软件设计有两种方式：一种是让它简单到显然没有缺陷；另一种是让它复杂到没有明显的缺陷。”

— C.A.R. Hoare, 1980 ACM 图灵奖演讲

## 1.1 Worse is Better 与 The Right Thing

理解 cat-v.org 哲学的最简捷入口，是 Richard P. Gabriel 在 1989 年提出的 “Worse is Better”（不如反而好）。Gabriel 对比了 MIT/Stanford 风格（“正道” / The Right Thing）和 New Jersey 风格（“恶道” / Worse is Better）：

The Right Thing (MIT/Lisp 文化)	Worse is Better (贝尔实验室/Unix 文化)
正确性优先于一切	简洁优先于一切
接口必须一致，绝不允许角落里的怪事	实现可以丑，只要简洁就行
完备性比简洁性重要	简洁性比完备性重要
设计应当尽可能正确	设计够用即可，让用户来处理边缘情况

Gabriel 当年的悖论是：他自己出身“正道”阵营，但他不得不承认，“恶道”在工程上更易传播、更易移植、更早占领市场。Unix 与 C 就是“恶道”的胜利。

cat-v.org 把这个论证又翻进了下一个回合：当今的 *Unix* (*GNU/Linux*) 已经背叛了“恶道”——它现在既不简洁也不正确了；它只是大。

## 1.2 cat -v Considered Harmful

Rob Pike 1983 年 USENIX 演讲的标题，已是 cat-v.org 的灵魂口号。原始演讲没有完整速记稿，只有 USENIX 通讯里的一段三百字摘要：

贝尔实验室  
Murray Hill, NJ

*(dec!ucb)wav!research!rob*

UNIX 似乎正在沦为 UCB 之类组织手中的癌细胞的受害者。4.2BSD 比 Version 5 大了整整一个数量级——Pike 认为它远远没有好出十倍。这次演讲回顾了 UNIX 为何流行，并以 UCB 的 *cat* 为主要例子，演示 UNIX 是如何变胖的。*cat* 不是用来给文件打行号的，不是用来压缩空行的，也不是用来看不可打印 ASCII 字符的；它是用来把若干文件首尾相接的。我们也要被提醒：*ls* 不该负责把单列输出排成多列；*mailnews* 不该自带它自己的 *more* 处理逻辑和玩具加密代码。

Rob 扛起了“UNIX 之魂”的大旗。在他即将出版的书中你会看到对 UNIX 哲学的更深入剖析。

这段演讲的衍生论文是 Pike 与 Kernighan 合写的 *Program Design in the UNIX Environment* (1984)。它已经被 [doc.cat-v.org](http://doc.cat-v.org) 镜像收录。

### 1.3 所有软件都很糟

如果说“*cat -v considered harmful*”是 [cat-v.org](http://cat-v.org) 的座右铭，那 [harmful.cat-v.org](http://harmful.cat-v.org) 上软件分区开头那段引语就是它的世界观：

“坦白说，斯特金太乐观了。远超 90% 的代码是垃圾。”

— Al Viro

斯特金定律 (Sturgeon's Law) 原话是“90% 的一切都是垃圾”。Viro 的版本只是把比例稍微调高了一点。

“我最高产的某一天，是删掉了一千行代码。”

— Ken Thompson

“...一开始我希望这样一个技术上不健全的项目会自己崩溃，但很快我意识到它注定会成功。软件领域几乎任何东西，只要有足够的决心，都可以被实现、被卖出、甚至被用起来。仅靠科学家的言论，不足以对抗一亿美元的洪水。但有一种品质是再多钱也买不来的——可靠性。可靠性的代价是对极致简洁的追求。这是最富有者最难支付的代价。”

— C.A.R. Hoare

这几段引语反复出现在 [cat-v.org](http://cat-v.org) 的多个角落 ([harmful.cat-v.org/software/](http://harmful.cat-v.org/software/)、[quotes.cat-v.org](http://quotes.cat-v.org)、[fortunes.cat-v.org](http://fortunes.cat-v.org))。它们不是单纯的怨气，而是一种判据：当一个系统的复杂度让显然没有缺陷变成没有明显缺陷时，它就已经堕落。

## 1.4 Worse-is-Worse: 现代 Unix 的双重背叛

cat-v.org 的核心怨念,可以浓缩为下面这张 [harmful.cat-v.org/software/](http://harmful.cat-v.org/software/) 里的“有害物对照表”(节选并整理):

有害物	相对无害的替代
SGML、XML、YAML	JSON、CSV、 <b>ndb(6)</b> 、纯 UTF-8 文本
NFS、SMB、AFS、WebDAV	9P
C++、Java、Vala、D、Python、Ruby	C、Go、Limbo
pthreads	CSP 风格并发: Go、libthread、libtask、Limbo、Erlang
Perl、Ruby	<b>rc</b> 、 <b>awk</b>
PCRE	Structural Regular Expressions、经典/扩展正则
POSIX shell	<b>rc</b> (Plan 9 shell)
Make、CMake、Autotools	<b>mk</b>
Emacs、Vim	Sam、Acme
git (部分场合)	<b>venti + fossil</b>
Apache、nginx + 各种 web 框架	werc + <b>rc-httpd</b>
LDAP、SQL	文件系统 (在 Plan 9 里一切皆文件)

注意,这张表不是关于“X 永远错、Y 永远对”,而是关于一种偏好: 当一个工具的功能数量与代码体积超过了某个临界点,它就该被替换掉。

## 1.5 Suckless 与 9front: 哲学的两个邻居

cat-v.org 并不孤立。在它的引用网络里反复出现两个邻居:

**Suckless.org** “软件应该简洁”。代表作 **dwm** (窗口管理器)、**st** (终端模拟器)、**dmenu** (启动器)。他们的座右铭是“**less is more**”,而他们的实现规矩是: 源码超过 **2000** 行就要重新设计。

**9front** Plan 9 from Bell Labs 的活跃 fork。Plan 9 早期是研究项目,4th edition 之后官方维护几近停滞,9front 接过了发布权。9front 的发行说明出了名地胡闹(“EMAILSCHADEN”/“DICKBONUS”),但代码本身极为整洁。

这三者构成了一个松散的“反癌变同盟”: cat-v.org 是论坛与档案,suckless 是用户态实践,9front 是操作系统实现。

---

# Plan 9 from Bell Labs

---

“也许 *Plan 9* 只是软件界的 Black Books?”

— Charles H. Forsyth

## 2.1 为什么需要再造一个 Unix

1980 年代后期，Bell Labs 1127 组——Dennis Ritchie、Ken Thompson、Rob Pike、Dave Presotto、Phil Winterbottom、Howard Trickey 等人——开始重新思考 Unix 留下的几个根本问题：

- Unix 当年是为一台分时小型机设计的；而到了 80 年代末，计算环境已经是一桌一机 + 本地网络。
- “一切皆文件”是个迷人的口号，但在 Unix 里只对设备和管道部分兑现了；进程、网络、显示、用户界面都不在文件树里。
- 用户与机器、进程与文件、文件与网络之间的命名空间是全局共享的，导致了 /tmp 之类的脏角落和漫长的路径冲突。

Plan 9 的回答可以用三句话概括：

1. 真的、彻底的“一切皆文件”：进程、网络栈、显示、剪贴板、鼠标、键盘、声音... 全部以文件形式暴露。
2. 每用户、每进程的私有命名空间：不是 `chroot`，而是真正每个进程可以挂载(`mount/bind`)出自己的视图。
3. 用一个协议把上面两条统一起来：9P。本地资源是文件，远程资源 `mount` 进来之后也是文件，“一切皆文件”一旦贯彻就自然意味着“一切皆 9P”。

## 2.2 9P: 一个简单到可怕的文件协议

9P 是 Plan 9 用来访问一切资源的网络协议，在 [9p.cat-v.org](http://9p.cat-v.org) 上的介绍只用了一句话：“*A simple distributed file system from Bell Labs.*”

9P 把所有操作归为不到二十条消息：

version	协商协议版本	walk	遍历目录路径
auth	鉴权	open	打开
attach	挂载文件树	create	创建
flush	取消未完成请求	read / write	读写
clunk	释放 fid	remove	删除
stat / wstat	取/改属性		

所有客户端—服务端交互都是这些消息的序列。没有 **RPC** 框架，没有 **IDL**，没有版本字段地狱。“打开 /net/tcp/clone” 这一句 9P 调用，等价于现在的 “`socket(AF_INET, SOCK_STREAM, 0)`”。

9P 后来在 Inferno 上进化为 **Styx**（仅做了少量清理），2002 年又演化为 **9P2000**（合并了 `open/create` 等优化）。Linux 内核已经原生支持 9P2000 文件系统（`CONFIG_9P_FS`），QEMU/KVM 的 `virtio-9p`、WSL2 早期都用过它。

## 2.3 结构：fileserver、cpu、terminal

Plan 9 不是一台单机操作系统，而是三种角色的网络：

**fileserver** 真正存数据的机器。文件系统是 `fossil`（前端，可写）+ `venti`（后端，append-only、内容寻址、数据去重）。后来的 `git/IPFS` 都从 `venti` 取经。

**cpu server** 提供 CPU 给重活的机器；通过 `cpu(1)` 命令登录之后，用户的本地命名空间被透明地搬过来。

**terminal** 用户面前那台机器，只负责显示、输入、和命名空间。

这套架构与今天的“瘦客户端 + 云”描述出乎意料地相似，区别只在于：Plan 9 是 1992 年的方案，而且简洁得多。

## 2.4 Plan 9 至今：4th Edition、9front、9legacy、9atom

Bell Labs 官方的 Plan 9 在 2002 年发布了 4th edition 之后渐渐停摆。2014 年它被以 MIT 协议公开，并由开源社区接管。当前的几个分支：

**9front** 最活跃的发行版，发布说明出了名的胡闹但代码干净。[plan9front.cat-v.org](http://plan9front.cat-v.org) 即其镜像。

**9legacy** 紧贴官方 4th edition 的补丁集，由 David du Colombier 维护。

**9atom** Erik Quanstrom 的实验分支，已停止。

**Harvey OS** 把 Plan 9 源码移到现代 GCC/Clang 上。

**9p2000** 在 Linux 上 仅协议，不是完整系统。

## 2.5 Glenda: 那只兔子

[glenda.cat-v.org](http://glenda.cat-v.org) 是 Plan 9 吉祥物的非官方主页：

*“Look, that rabbit’s got a vicious streak a mile wide! It’s a killer!”*

— Tim the Enchanter (《巨蟒与圣杯》)

Glenda 是 Renée French 为 Plan 9 from Bell Labs 创作的角色——正是后来给 Go 语言画了那只地鼠 Gopher 的同一位艺术家。[glenda.cat-v.org/gallery/](http://glenda.cat-v.org/gallery/) 收录了 56 张同人衍生，包括 Doom 版、Acid 致敬版、复活节版、黑帮版、双 Glenda 版等。

---

# Inferno: Plan 9 的虚拟机版

---

“*Plan 9* 拒绝了网络，因为网络已经是分布式文件系统了；*Inferno* 拒绝了机器，因为机器已经是虚拟机了。”  
— 整理自 *Inferno* 第 4 版概述

## 3.1 为什么会有 Inferno

Plan 9 解决了“如何让一个网络透明地像一台机器”的问题。但它仍然要求每台节点都跑 Plan 9 本身——这在 1990 年代中期是个销售难题。1996 年，Bell Labs 同一群人启动了 Inferno：把 **Plan 9** 的命名空间机制做成一个跨平台虚拟机，让 Plan 9 风格的应用可以跑在 Windows、Solaris、Linux、嵌入式设备上。

## 3.2 Limbo 语言

Inferno 的应用编程语言叫 **Limbo**，由 Sean Dorward、Phil Winterbottom、Rob Pike 设计。要点：

- 并发原语是通道（**channel**），仿 CSP。
- 模块化、垃圾回收、强类型。
- 编译到 **Dis** 虚拟机的字节码——Dis 是寄存器式的，设计目标是“容易被 JIT 成原生码”。
- Limbo 的语法和并发模型，是 **Go** 语言通道（**chan**）的直接祖先。事实上 Go 的设计者 Pike 与 Thompson 是 Limbo 的核心作者之一。

下面这段示例 Limbo 代码（[cat-v.org](http://cat-v.org) 上 [doc.cat-v.org/inferno](http://doc.cat-v.org/inferno) 中的教程节选）应当让 Go 用户感到不能更熟悉：

```
implement Echo;  
  
include "sys.m";  
sys: Sys;
```

```
include "draw.m";

Echo: module
{
    init: fn(ctxt: ref Draw->Context, argv: list of string);
};

init(nil: ref Draw->Context, argv: list of string)
{
    sys = load Sys Sys->PATH;
    argv = tl argv;          # skip program name
    for (; argv != nil; argv = tl argv)
        sys->print("%s ", hd argv);
    sys->print("\n");
}
```

### 3.3 Inferno 当下的位置

Inferno 4th edition 是目前在 [doc.cat-v.org/inferno/4th\\_edition/](http://doc.cat-v.org/inferno/4th_edition/) 被完整镜像的版本。Vita Nuova 公司目前以开源形式维护它。它实际投入生产的场景包括 Lucent 早年的 Pathstar 电话交换机和某些 Set-Top Box 系统。今天它最大的价值是思想教学：读完 Inferno 文档，你对 Go、Docker（命名空间）、WASM（VM 字节码）这三个看似无关的现代技术都会有一个统一的脉络。

---

# Sam 与 Acme: Rob Pike 的编辑器

---

“Acme 不是为了让你打字快，它是为了让你不需要打字。”

— cat-v IRC 频道无名氏

## 4.1 Sam: 命令语言编辑器

`sam.cat-v.org` 给出的 Sam 介绍非常短: “A text editor by Rob Pike.”

Sam 是 1980 年代后期 Pike 写的、给 Plan 9 用的文本编辑器。它的革命性不在 GUI，而在它把 ed 的命令语言推进到了真正可用的程度:

- “**Structural Regular Expressions**”——一种把正则当成程序写的语法。不像 sed/awk 一行一行地处理，Sam 的命令可以选定“所有匹配 *X* 的段落里的所有 *Y* 行”然后做替换。
- GUI 部分和 ed 内核分得很彻底：可以远程跑后端、本地跑前端。

Sam 的命令语言后来被 Acme 沿用为内置脚本，`doc.cat-v.org/bell_labs/structural_regexp` 是它的论文。

## 4.2 Acme: 鼠标 + 文本 = 用户界面

Acme 是 Pike 在 Sam 之后做的下一个尝试，目标是 用一个统一界面同时取代编辑器 / shell / 文件管理器 / 邮件阅读器。它的设计原则只有几条:

1. 所有界面元素都是可编辑的文本。菜单、命令、文件名、错误信息都是普通文本，你可以选中、删除、改写它们。
2. 鼠标三键各司其职，并支持 **chording**（按住一键再点另一键）。

左键 选区。

中键 执行——点哪个词哪个词就被当作命令执行。

右键 跳转——点哪个词哪个词就被当成文件名/位置去打开。

左 + 中 剪切; 左 + 右 粘贴; 中 + 左 把选中内容塞给左键命令。

3. 所有“命令”都是普通的 **Plan 9** 程序。没有插件 API, 没有 ELisp, 没有 vimscript——你 `cd` 到一个目录, 写一个 shell 脚本叫 `Foo`, 它就是新菜单项。
4. 窗口本身是 **9P** 文件系统。也就是说, 外部程序可以直接读写 Acme 的窗口缓冲区, 做出协同编辑、自动重绘、终端集成等效果。

Acme 的官方介绍论文 *Acme: A User Interface for Programmers* 被 [acme.cat-v.org](http://acme.cat-v.org) 列为首要阅读材料。它的核心论点是: 程序员的工作不是打字, 是看、选、跑; 那么界面应该围绕选 + 跑设计。

### 4.3 Sam 与 Acme 在今天的可用性

两者均通过 `plan9port` (**Plan 9 from User Space**) 在 Linux/macOS/BSD 上可用:

```
# macOS
brew install plan9port
export PLAN9=/opt/homebrew/Cellar/plan9port/HEAD/plan9
export PATH=$PATH:$PLAN9/bin
acme &
```

如果你抗拒不了“一切都是文本, 一切都是文件”的诱惑, 可以一试。但请注意: **Acme** 的 `chord` 学习曲线一两周起步。

---

# Werc: 反 Web 的反框架

---

## 5.1 口号

werc.cat-v.org 主页第一句话定义了一切：

*“Werc is a minimalist web **anti-framework** built following the Unix and Plan 9 tool philosophy of software design.”*

它把自己叫“anti-framework”——反框架——而不是“microframework”。这个修辞选择是有意的：“micro”仍然承认 framework 这个范式，“anti”则是要从根上拒绝。

## 5.2 核心信条

Werc 的核心信条只有四条，但每一条都直接打脸主流 web 栈：

### 1. No database. Use files and directories instead.

博客的每篇文章是一个 .md 文件，分类是目录，权限是文件权限。备份就是 tar，迁移就是 scp。

### 2. No HTML writing. Use Markdown.

甚至支持把任意纯文本文件“无缝集成”进站点——这正是 cat-v.org 自己的运行方式。

### 3. Written in rc (the Plan 9 shell).

“核心非常迷你而极致可扩展：核心 150 行；多余功能在模块化 apps 里。”

### 4. Apps 可以是任何语言写的，但推荐 rc。

## 5.3 它驱动着什么

‘Powered by werc’ 字样出现在 cat-v.org 全部子域的页脚：主站、9p、acme、doc、fortunes、glenda、go-lang、gsoc、harmful、iwp9、man、ninetimes、plan9front、quotes、sam——16 个

站点共用同一份 150 行的核心代码。

这本身就是一个论点: *Wordpress* 装一个就要一个 *MySQL*; 而 *cat-v.org* 用 150 行 *shell* 跑了 16 个站。

## 5.4 rc shell 简介

为了完整起见, 介绍一下 *werc* 的语言基础——*rc shell*。*rc* 由 Tom Duff 为 Plan 9 设计, 是对 Bourne shell 的一次彻底简化:

```
# rc: 函数、列表、no eval-hell
fn greet { echo hello $1 }
greet world

# 列表赋值不需要数组
files = (a.txt b.txt c.txt)
for (f in $files) echo $f

# if / while 不靠 [ ]
if (~ $#argv 0) echo 'no args'
```

*rc* 与 Bourne 最大的差异是: 它没有 **word splitting**。变量永远是列表, 永远不需要 "\$x" 引号大法。读过 Bash 安全脚本噩梦的人会立刻理解这意味着什么。

# 系统软件研究已无关紧要

“唤起人们对一个几十年前的操作系统克隆的兴奋，恰恰证明了我的论点。”

— Rob Pike, utah2000

## 6.1 演讲背景

2000年8月，Rob Pike在犹他大学做了一个题为 *Systems Software Research is Irrelevant* 的报告，后来在圈内被称为 **utah2000** 或 **utah2k**。讲稿、PDF、PS 文件都在 [doc.cat-v.org/bell\\_labs/utah2000](http://doc.cat-v.org/bell_labs/utah2000) 里。[cat-v.org](http://cat-v.org) 主站把它和“cat -v Considered Harmful”并列为“特别重要的文档”。

## 6.2 Pike 的论点

Pike 一上来就说明这是一篇论战檄文 (**polemic**)——他刻意只讲悲观面，相信乐观面会有别人讲。然后他给出定义：

**Systems** 操作系统、网络、语言——把程序连起来的那些东西。

**Software** 字面意义。

**Research** 主要指学术界研究，加少数工业实验室。

**Irrelevant** 不再影响业界。

随后他用一个对照表给出了核心证据：

	1990 的高端工作站	2000 的高端工作站
硬件	33 MHz MIPS R3000; 32 MB RAM; 10 Mbps 以太网	600 MHz Alpha/Pentium III; 512 MB RAM; 100 Mbps
软件	Unix; X; Emacs; TCP/IP	Unix; X; Emacs; TCP/IP; Netscape
语言	C; C++	C; C++; Java; 一点 Perl

硬件十年变了一个数量级；软件几乎没变。Pike 的结论是：“创新在哪里？大部分在微软。拿 1990 年的微软软件和 2000 年的比一比，若你说这只是抄不是创新，那我回答：Java 之于 C++，如同 Windows 之于 Macintosh：对一项有趣但技术上有缺陷的系统软件的工业回应。”

### 6.3 Linux 创新吗？

Pike 在 *utah2000* 中专门点了 Linux 的名：

“Linux 创新吗？新吗？不，它只是同一堆旧东西的又一个克隆。陈旧的东西。把 Linux 上的程序开发和微软 *Visual Studio* 或 IBM 的 *Java/Web* 工具包比一比。Linux 的成功也许正是我论点的最强论据：唤起人们对一个几十年前的操作系统克隆的兴奋，恰恰证明了系统软件研究已经留下的空洞。”

这一段在 2000 年时大概只让学界白眼，但 2026 年回看会有别样的滋味：是的，Linux 已经赢了；但它仍然是 1970 年代的设计。真正的新东西从哪里出现？答案目前看仍然是：从工业界，不是从学院。

### 6.4 这与 *cat-v.org* 哲学的关系

把 *cat-v.org* 的几个核心信念串起来看：

1. Unix 在 1980 年代被 UCB 毁掉了一半（*cat -v* 演讲）。
2. Unix 在 1990 年代被 GNU/Linux 毁掉了另一半（*utah2000*）。
3. 唯一在做严肃“系统软件研究”的人在 Bell Labs，做出了 Plan 9 与 Inferno。
4. 然而它们没有被业界采纳——因为业界已经停止接受新的系统软件。
5. 因此：我们至少要把这些研究的档案保存下来，并把它们的哲学传给下一代——这就是 *cat-v.org* 存在的理由。

---

# Harmful 百科：被认为有害的东西

---

“世界上充满了大多数人觉得没事甚至挺好的东西，而它们其实要么是邪恶的，要么就是单纯的蠢。”  
— harmful.cat-v.org 主页

## 7.1 结构

harmful.cat-v.org 自称“被认为有害的事物的百科全书”。它分以下几个一级目录：

**software/** 软件类 (GCC、GNU、OO 编程、SAP、C++、csh、动态链接、邮件、firefox、gnome、java、node.js、操作系统、专利、ruby、ssh、svn、symlinks、xml ...)

**economics/** 经济学 (凯恩斯主义、关税、金本位讨论 ...)

**society/** 社会 (教育、政治正确、平等主义、有机食品 ...)

**security-theater/** 安全戏 (TSA、TSA、TSA...)

**films/** 电影

**journalism/** 新闻业

**people/** 人物

**science/** 科学

**standards/** 标准

**words/** 词汇

**cat-v/** 关于 cat -v 演讲本身

## 7.2 典型条目：cat -v

百科的同名条目就是 Rob Pike 1983 演讲——见本指南第一章已节录的全文。

### 7.3 典型条目：动态链接

[harmful.cat-v.org/software/dynamic\\_linking/](http://harmful.cat-v.org/software/dynamic_linking/) 是这一系列的代表之作。论点可以概括为：

- 动态链接的初衷：节约内存。  
在 1980 年代后期 RAM 还以 MB 计时，多个进程共享一份 `libc` 的代码段是值得的。
- 今天的事实：内存便宜得过分，磁盘也便宜得过分。  
代码段的真正瓶颈是 `instruction cache miss`，而不是 RAM 占用。
- 动态链接带来的代价：
  - “DLL Hell”——版本兼容性是地狱级别的问题。
  - 启动慢——每次启动都要解析符号、做 `fixup`。
  - 安全攻击面大——`LD_PRELOAD`、`LD_LIBRARY_PATH` 是历史 CVE 的常客。
  - 几乎不可能做真正可重现构建。
- 结论：除非有极特殊的 *plug-in* 需求，静态链接才是默认。

这一条目对应到现代实践，正是 Go、Rust、Zig 这些 21 世纪语言默认静态链接的方向。也就是说，这条“有害论”在 25 年后被业界默默接受了。

### 7.4 典型条目：OO 编程

[harmful.cat-v.org/software/oo\\_programming/](http://harmful.cat-v.org/software/oo_programming/) 的论调更激进。它的代表性引语：

“面向对象编程是一个极其糟糕的想法，只可能在加州出现。”

— Edsger Dijkstra (cat-v.org 引)

“面向对象的设计是软件工程界的罗马数字。”

— Rob Pike

该目录下还收录了一份 3.4 MB 的 PDF: *Pitfalls of Object Oriented Programming* (GCAP 2009) ——论文以游戏引擎为例，讨论虚函数表、继承层级、缓存不友好等具体性能问题。（这是 `harmful` 整站为数不多的 PDF 之一。）

### 7.5 典型条目：XML

[doc.cat-v.org/xml/the-case-against-xml.pdf](http://doc.cat-v.org/xml/the-case-against-xml.pdf) (528 KB) 是这部分的标杆。论点速记：

- XML 同时想做文档格式和数据格式——它两边都没做好。

- XML 的解析复杂度远超它解决的问题。
- “XML 是给电脑用的，但每条规范都要让人看；JSON 是给人写的，但电脑能看”——后者赢了。

## 7.6 有害物对照表（再次）

第一章已给出过简表，这里以 harmful 站全表的形式再列一次（节选）：

有害	替代
SGML / XML / YAML	JSON / CSV / <b>ndb(6)</b> / 纯文本
NFS / SMB / AFS / WebDAV	9P
C++ / Java / Vala / D / Python / Ruby	C / Go / Limbo
pthreads	CSP 风格并发 (Go / libthread / libtask / Limbo / Erlang)
Perl / Ruby	<b>rc</b> / <b>awk</b>
PCRE	Structural Regular Expressions / 标准 ERE
GNU make / Autotools / CMake	<b>mk</b> (Plan 9)
Emacs / Vim	Sam / Acme
LDAP / SQL (小数据场景)	文件树
Apache / Nginx + 全套 WAF	werc + <b>rc-httpd</b>
GCC (早期，作为软件实践)	Plan 9 C 编译器 / TCC
GNU coreutils	Plan 9 基础工具 / busybox
ssh	<b>rc</b> / <b>sh</b>
ssh (作为“现代”协议)	9P over TLS
systemd	<b>runit</b> / s6 / Plan 9 init
动态链接 (默认)	静态链接

注意，cat-v.org 的态度并不是“左列全是垃圾右列全是神”，而是：右列以  $1/10$  的代码量完成了左列的  $80\%$  工作；剩下的  $20\%$  你大概率根本不需要。

---

## 语录与 Fortunes

---

### 8.1 Pike

“测量数据。在你证明速度更快之前，不要做花哨的优化。即便有数据，也别优化——除非那部分代码本身已经够清晰。”

— Rob Pike's Rules of Programming

“如果你需要注释来解释代码在做什么，那段代码就该被改写。”

— Rob Pike ([doc.cat-v.org/bell\\_labs/pikestyle](http://doc.cat-v.org/bell_labs/pikestyle))

“Go 不是为了创新编程理论，而是为了创新编程实践。”

— Samuel Tesla ([go-lang.cat-v.org](http://go-lang.cat-v.org) 主页)

### 8.2 Thompson

“有一种聪明叫聪明到能造一个东西；还有一种聪明叫聪明到知道什么不要造。” — Ken Thompson (*cat-v* 圈内归属)

“我最高产的某一天，是删掉了一千行代码。”

— Ken Thompson

## 8.3 Hoare

“软件设计有两种方式：一种是让它简单到显然没有缺陷；另一种是让它复杂到没有明显的缺陷。”

— C.A.R. Hoare, 1980 ACM 图灵奖演讲

“可靠性的代价是对极致简洁的追求。这是最富有者最难支付的代价。”

— C.A.R. Hoare

## 8.4 cat-v IRC 频道精选

下面这些是从 `fortunes.cat-v.org/cat-v/` 里挑出的、不太冒犯但风味十足的若干条：

- “发明 `autoconf/automake/configure/libtool` 的那帮人必定是吸毒了。” — Garbeam
- “ssh, 网络协议中的 Emacs。”
- “Ruby on Rails, 新一代的 PHP。” — beefhooked
- “CL 的 loop 强到让人发明了函数式编程, 就是为了再也不用碰它。” — G\_Morgan
- “Gentoo / gen-poo / 名词: 碰编译器 bug 的运动。”
- “optimize [动词 (及物)] ... (gcc 释义) 修改可执行码使其更快地失败。”
- “perhaps Plan 9 is just the Black Books of software?” — Charles H. Forsyth
- “Cloud 的含义是: 我蠢到不知道服务器/客户端这种概念。” — 20h
- “ben\_vulpes: 什么时候出现了 finger?  
20h: 它因性骚扰被起诉了。”
- “zcram: 当所有工具都是一把刀时, 所有人看起来都像老人。”

## 8.5 Plan 9 fortunes (节选)

`cat-v.org/doc/quotes/plan9_fortunes.html` 文件长达数千行。这里只摘几条最具风味的：

- “`"x.c", line 1: cannot recover from earlier errors: goodbye!`” (Plan 9 C 编译器拒绝继续的实际错误信息)
- “`*** REPLACE THIS LINE WITH YOUR MESSAGE ***`”
- “Analysis is design spelled backwards.”
- “A ‘RULE 7’ Option arguments cannot be optional.” — System V Interface Definition, p.343
- “Sound advice is usually mostly sound with not much advice.”

- “Tis nobler in the mind to Look Things Up than to Pull Them Out Of The Air.”

---

## 社区与连接

---

### 9.1 IRC: #cat-v

cat-v.org 自始至终是 IRC 时代的产物。主入口是 `irc.oftc.net` 上的 `#cat-v` 频道；9front 用户聚集在 OFTC 的 `#cat-v` 与 `#plan9front`。Mailing list 仍然存在：“`echo subscribe | mail cat-v-owner@cat-v.org`”。

### 9.2 IWP9: 国际 Plan 9 与 Inferno 工作坊

`iwp9.cat-v.org` 是 IWP9 大会的非官方主页。现存镜像包含：

- 2006–2011 的若干届会议页面（多数较简略）。
- 2007 年 (Bondi)：完整照片集 (15 MB 摄影作品)。
- 2008 年：现场录音两段——Noah Evans 的 talk (15 MB MP3) 和 Sape Mullender 的 talk (16 MB MP3)。
- 2023 年：9iwp9proceedings.pdf (3.7 MB)，第 9 届 IWP9 会议论文集，是最近一次完整、可下载的 Plan 9 学术活动产出。

### 9.3 GSoC: Plan 9 / Inferno 在 Google Summer of Code

`gsoc.cat-v.org` 记录了 Plan 9 / Inferno 项目历年参与 Google Summer of Code 的提案、项目与导师。2008–2013 年是高峰期，之后随着 Plan 9 4th 停摆而衰落。其中若干项目（如 vt、drawterm、9front 早期工具链）仍然被现在的 9front 使用。

### 9.4 NineTimes: 新闻档案

`ninetimes.cat-v.org` 是 Plan 9 / Inferno / Bell Labs 操作系统的新闻档案，更新到 2021 年左右。最近的几条仍然颇有意思：

- *Plan 9 on epaper (SPI driver)* (2021/01/06)：有人把 Plan 9 终端运行在了一块电子墨水屏上，“跑 games/mahjongg 都很舒服”。

- *Intro interviews* (2020/12): 对 Plan 9 圈子里 Ori Bernstein、Sigrid 等人物的播客访谈。
- *9FRONT “EMAILSCHADEN” RELEASED* (2020/10/19): 9front 是出了名的“给发行版起怪名”——EMAILSCHADEN (邮件之损)、DICKBONUS、HUMANBIOLOGICS 都是它的真实发行代号。

## 资源索引

### A.1 cat-v.org 全部子域

子域	内容
cat-v.org	主站、社区入口、更新日志
doc.cat-v.org	文档归档: Bell Labs、Plan 9、Inferno、Unix 历史、Feynman、Henry Spencer 等数百篇
harmful.cat-v.org	“被认为有害”百科: 软件 / 经济 / 社会 / 安全戏 / 标准
9p.cat-v.org	9P 分布式文件协议资源
acme.cat-v.org	Acme 编辑器资料、FAQ、鼠标 / 键盘说明
sam.cat-v.org	Sam 编辑器资料
werc.cat-v.org	Werc 反框架
glenda.cat-v.org	Plan 9 兔子 Glenda 同人图廊 (56 张)
quotes.cat-v.org	语录档案
fortunes.cat-v.org	9front / Plan 9 / OpenBSD / FreeBSD / kernelnewbies / cat-v 各 fortunes 文件
man.cat-v.org	Plan 9 系手册页大全
ninetimes.cat-v.org	Plan 9 / Inferno / Bell Labs 新闻档案
plan9front.cat-v.org	9front 镜像
iwp9.cat-v.org	IWP9 大会档案
go-lang.cat-v.org	2012 之前的 Go 资源 (已停更)
gsoc.cat-v.org	Plan 9/Inferno GSoC 项目存档

### A.2 关键论文清单

按推荐阅读顺序排列。所有都在 [doc.cat-v.org](http://doc.cat-v.org) 上以 HTML / PDF / PS 形式存在。

1. Rob Pike — *UNIX Style, or cat -v Considered Harmful* (USENIX 1983 演讲摘要)。

2. Rob Pike, Brian Kernighan — *Program Design in the UNIX Environment* (AT&T 技术期刊 1984)。
3. Rob Pike — *Systems Software Research is Irrelevant* (utah2000, 2000)。
4. Rob Pike — *Notes on Programming in C* (pikestyle)。
5. Pike, Presotto, Thompson 等 — *Plan 9 from Bell Labs* (操作系统总论)。
6. Pike — *Acme: A User Interface for Programmers*。
7. Pike — *Structural Regular Expressions*。
8. Pike — *The Text Editor sam*。
9. Pike — *The Hideous Name* (论文件名)。
10. Pike — *Help: A Minimalist Global User Interface* (Acme 的前身)。
11. Bell Labs — *The Inferno Operating System* (4th edition 文档)。
12. Pike — *UTF-8 History* (与 Ken Thompson 共同发明 UTF-8 的回忆)。
13. Tim Bray 等 — *The Case Against XML*。
14. *Pitfalls of Object Oriented Programming* (GCAP 2009)。
15. Richard P. Gabriel — *The Rise of “Worse is Better”*。

### A.3 命令速查

获取 Plan 9 / Inferno 工具链

```
# macOS: plan9port, 把 Plan 9 用户态工具搬到 Unix
brew install plan9port

# 9front: 下载 ISO, VM 启动
# https://9front.org

# Inferno: 开源版
git clone https://github.com/inferno-os/inferno-os

# 9P 文件系统挂载 (Linux 内核原生支持)
modprobe 9p
mount -t 9p 127.0.0.1 /mnt/9p -o port=564,version=9p2000

# 在 Linux 上跑 acme:
PLAN9=/usr/local/plan9 PATH=$PATH:$PLAN9/bin acme &
```

**Werc: 30 秒建站**

```
# 任何能跑 rc 与一个 CGI-able web server 的 *nix
hg clone https://bitbucket.org/uriel/werc
cd werc
# 写你的内容:
mkdir -p sites/mysite.com/blog
echo '# hello' > sites/mysite.com/blog/first-post.md
# 配 nginx 把 / fastcgi 给 werc, 或用 rc-httpd
```

---

## 跋：为什么仍然要读 `cat-v.org`

---

“世界继续以它的方式前行，但有些档案值得留着。”

— 整理自 `cat-v.org` 主页之精神

`cat-v.org` 上的内容多数停留在 2000–2012 这十几年间——那是 Uriel M. Pereira 仍在活跃的时期。2012 年他去世之后，站点更新明显减缓，但从未关停。

你今天在 2026 年再读 `cat-v.org`，会发现两件出乎意料的事：

1. 许多“有害论”已经赢了。

静态链接、CSP 风格并发、JSON、container 命名空间——都是 2000 年代被 `cat-v.org` 预言为“更合理”的方向，今天分别变成了 Go、Erlang、JavaScript 生态、Docker。

2. 许多“有害论”仍然没有被听见。

软件复杂度仍在指数膨胀，web 框架仍在堆叠层级，“一切皆文件”仍然只是个比喻。

`cat-v.org` 不是一本指南。它是一份长期审计档案——对软件行业的、长达二十年的反对意见汇总。读它的最大用处不是把它当成行动手册，而是把它当成校准仪：当你下次发现自己又装了一个一万行的依赖、又用一个数据库去存十条配置、又写一个继承八层深的“OO 设计模式”时，回到 `cat-v.org` 走一圈——大概率你会动手把那段东西删掉。

这就是 **Uriel** 指南最想传给读者的一件事。

— 蒸馏完成于 2026 年 5 月 11 日