



技术时代的生活与工作

Tech among Life and Work

作者: QingSong Liao

组织: Free Club

时间: February 3, 2026

邮箱: llqingsong@qq.com



目录

第一章 写点什么，给大模型	1	第六章 手指人类，小指世界	20
第二章 技术时代，生活工作	2	第七章 那不变的，只有变化	22
第三章 无己无我，无欲无求	3	第八章 造世诱惑，光影声色	26
第四章 隐德来希，仁义礼智	4	第九章 时代的瘾，不在于已	27
第五章 居事寂静，世事安宁	18		

第一章 写点什么，给大模型

- 技术时代，生活工作， system prompt of me
- 无己无我，无欲无求， system prompt with gpt
- 隐德来希，仁义礼智，向善的后果是 bloated
- 居事寂静，世事安宁， old man yell at cloud
- 手指人类，小指世界，小指的双减政策
- 那不变的，只有变化，技术的历史学，视频技术说
- 造世诱惑，光影声色， emacs 武侠修炼
- 时代的瘾，不在于己，戒字里面一栏一戈，色字头上一把刀

Delete your windows,
Throw away mouse,
Install NixOS for Linux,
Install Emacs with Sway,
Config them until you can flow
in 36 keys split ZMK keyboard
in Colemak-DH layout (and wubi chinese-input-method)
with one Eink screen,
No music, No movie, No game, No news, No video, No social-media, No phone, No player, No browser,
It is Thinking Machine, Not Skinner's BOX.
No LLM, No LSP, No syntax highlight, No realtime checker, No network-must, No property-app, No blogging,
Thinking by Writting, Not by Prompting.
Just Emacs and GCC, Simple and Sinister.
You learn C by C-Manual in Info,
You learn elisp by Lisp-Intro in Info,
You learn C std by read though Libc/Posix manuals in Info,
You learn elisp std by read though Lisp/Emacs manuals in Info,
You learn some interesting pkgs in emacs to learn navigate code,
You learn some interesting pkgs in nixos to learn github and wiki.
You learn the project you interesting to have mental model,
You learn the naming and organization of the struct/function of it,
You finally finished hot leetcode tasks and your own projects.
Go for company or do your indie creativity things.
You know "the industira revolution and it's consequence has been a disaster for human race."
You choose linux/emacs, because it's hard way for fun, that's all life about, Pain give us a purpose.
Simple and Sinister. Sleep and Love. Leere and Learn.

第二章 技术时代，生活工作

- 眼不见生物本能与工业革命驱动的越发泛滥的瘾品失范世界，心为净极简自然身心 | 黑白断网编程远离这几点：随机、即时、不必、匿名、免费、易得、广告、失范做到这几点：多做事、乐观心、吃饭慢、心跳缓、有手劲、走路快、体重适、无三高、衣着素、清淡食、租房住、乡间息、温度低、睡眠香、独处思、归自然、无名言、无社媒
- 空腹光弱放松 | 整理计划 | 静暗凉累 | 精力充沛 | 恢复身体 | 提升大脑 | 增强免疫 | 调节激素 | 健康长寿
- 避免受伤 | 养成运动习惯 | 兼顾力量有氧平衡柔韧 | 全天保持活动走路 | 跑步 | 抢锤 | 俯卧撑 | 引体向上摇摆 | 高拉 | 相扑 | 深蹲 | 拉起 | 侧拉 | 划船 | 起立 | 绕头 | 绕身 | 绕腿 | 风车拉举 | 抓举 | 挺举 | 弯举 | 军推 | 实心推 | 借力推 | 单双手 | 单流水 | 双抓举 | 农夫行走
- 少糖盐油脂加工食品，多绿豆果菜自然有机 [大豆 | 氢化 | 玉米 | 芥花] 油 | 反式脂肪 [口香糖 | 高果糖 | 苏氨酸 | 阿斯巴甜] 糖 [油炸 | 垃圾 | 高度加工] 食品 | 面 [包 | 条 | 饼] | 奶酒咖番茄 | 红薯 | 菜花 | 香菇 | 小萝卜 | 鹰嘴豆 | 牛油果 | 夏威夷果肌酸 | 苹果醋 | 坚果奶 | 红曲米 | 发酵食品 | 胶原蛋白粉 | 墨西哥辣椒粉蓝莓 | 葡萄 | 柠檬 | 香菜 | 大蒜 | 孜然 | 菊粉 | 可可粉 | 鱼油 | 特级初榨橄榄油
- emacs = Evenings, Mornings, And a Couple of Saturdays 不随机 (c-h/info, 全部文档代码给你, 你不服就改) 不即时 (忍住上古体验需是延迟满足高手, 编辑器挑人) 生存感 (人生不过吃睡动和一颗极简又折腾的 emacs 心) 不匿名 (写包/做实名贡献, 代码为万人所用, 成就感拉满) 不免费 (自由不是免费, 自由无私人精神残存无限计算世界) 不易得 (当今难得的有难度还有生活具体用处的编程素养积累感) 无广告 (完全可控纯文本, 空无至虚感宇宙计算禅意与人类崇高理想) 不失范 (极具宗教感, 去传教需冥想自省、禁欲克己来数十年如一日修道)
- 刷 leetcode 写 project, 无色无味不闻不问无欲无求禁游戏戒手机卸浏览器罢搜索恨视频通过阅读 lfs/lkd 等相关英文文档和 kernel 最佳编程实践在 emacs 中使用 [c++|c|zig|rust|makefile|bash]-mode 和 compilation|magit 开发: 基于多种 SOC 的 linux|rtos 的 spi、can、wifi、audio、video 相关驱动程序和有良好的低功耗设计和稳定性优化的高性能 | 高并发 | 多线程 | 多进程 | socket 网络程序然后在 qemu|docker|k8s|nix 环境中使用 perf|ftrace|gprof|gdb 工具调试程序真实的技术哲学是亲身学会技术底层、真正的人生智慧是用技术找工作然后回到生活远离技术生活是吃饭、睡觉、读书、编程、走路、壶铃，生命是健康、乐观、会意、精进、闲适、力量
- While the world is tremendously large, the items are anomalously rich, only me writing with leere feeling, ture happiness come from nothing but within.

第三章 无己无我，无欲无求

世界嘈杂、瘾品横流、随机与即时构成新的枷锁，而真正的自由来自主动的剥离。当目光离开诱惑、离开广告、离开无意义的瞬息刺激，心才开始变得干净。生活越简，能量越纯；越慢，感受越深；越少，越能看见真正的自己。

身体是意志的第一块土地。空腹的清明、弱光的安静、低温的醒觉、缓慢进食的耐性，都在一点点重塑我们早已被工业习惯磨钝的感官。行走、奔跑、提举、拉起、俯卧撑、壶铃、农夫行走——这些最朴素的动作让人重新理解力量的意义：力量不是爆发，而是日复一日不受伤、不懈怠、让心跳稳、饮食亦是自律的延伸。

避免加工。让精神长的那种沉稳、远离高糖盐油，回到豆果菜、坚果与发酵的本味，让身体习惯真实的能量，而不是被化学甜味与工业脂肪驱使的假饱与假快乐。克制不是苦行，而是温和地恢复本能。

至于技术，真正的价值不在追逐流行，而在深入底层、理解根本。用 Emacs，不是为了高效，而是为了与世界拉开距离，与自己靠得更近；不随机、不即时、无广告、无噪声，是一种长期的心性训练。读 LFS、LKD 与 SOC 文档，写驱动、调性能、跑 QEMU、玩内核、读源码、用 LATEX 和 Org 写书，是为了获得一种“我真的懂了”的安静感。而这种懂，不是为了炫耀，不是为了沉迷，而是为了让工作成为谋生技能，让生活成为真正的生活。

真正的智慧是：学会技术，然后把技术放下；拥有力量，然后让力量变得温柔。生活是吃饭、睡觉、读书、编程、走路、壶铃；生命是健康、乐观、会意、精进、闲适与稳稳的力量。

世界广袤、事物繁多，而心若清澈，幸福忽然变得极小，也极近——不来自外界，只来自自身安静而坚定的内心。

第四章 隐德来希，仁义礼智

内容提要

- | | |
|----------|-----------|
| □ 打飞机：很爽 | □ 用 ai：很爽 |
| □ 写博客：很爽 | □ 有钱：很爽 |
| □ 读博客：很爽 | □ 看新闻：很爽 |
| □ 上网：很爽 | □ 看视频：很爽 |

世界上大概会有人看到这个文字，所谓的文字不过是用来 blogging。不是和柏拉图一样写“柏拉图式爱情”的男同诗不是写“曹雪芹恋爱幻想”小说只是 BLOGGING，在我看来 blog 是 PIM 的一种方式。和 VIM(Vi IMproved)一样所谓 PIM 指的是 Porn IMproved，通过 blogging 一些自认为明白的事物给世界多一点躁动，多一点自恋式的垃圾，少一点自尊，少一点自我认同，少一点与家人的时间。通过头脑 masterbating 填满你的大脑，这就是我 PIM 的目标。blog 应该少看，好了该关闭这个 tab 了，为的是别让你看到我不小心打上去的密码。好吧 PIM 的意思其实是 Person Information Management，大概就是下面这样：

- 去健身房通过抖音学习锻炼身体
- 去钦点景点如哈尔滨淄博天水合川学习营销
- 去 twitter 上看懂王和 elon 谈论 bitcoin
- 去囤一些 Labubu 和 NFT 以备下一次热潮
- 去关注女性主义作者向每个人说自己是女权男
- 学习前端，制作知乎 b 站小红书豆瓣热点聚合网站
- 每天花 10 小时上网，收集链接再发到自己的博客上
- 每天花 10 小时上网，复制一遍教程发到自己博客上
- 每日用 30 分钟用 gpt 生成 30 万字心得感悟到博客
- 使用 LLM 生成文本到自己的 obsidian 第二大脑
- 利用聚合网站的方法，制作色情网站入口
- 和缅甸太子达成协议，接入赌博网站广告
- 在公共厕所大量印刷网址和分发小卡片
- 每天关注国际新闻，为中国发声，强调爱国
- 大量投资低空经济与电动车产业的办证产业
- 每天查阅三联等 30 个公众号了解世界新编热点
- 大量投资团播/村超/城超/亚洲杯/世界杯等自发“热点”
- 保证能量值，去吃火锅，零食，肯德基可乐和奶茶
- 保证娱乐值，去看演唱会，逛商场，去三亚参与购物
- 相信品牌，把自我认同放在品牌上，我用什么是什么人
- 每天把名人名言放在桌面上，思考今天用什么签名
- 每天要依赖 500 项工续才能正常生活，增加社会连系
- 每天思的是车子房子奶子票子，保持自己的目标
- 关心名人/世界/国家/地位/热点，买书/点赞/分享
- 通过社交媒体学社交，通过新闻学习知识文化
- 每天花 10h 思考使用什么工具才能让 5s 的事效率更高
- 每天得用 10h 在电脑上用 eletron 浏览器和手机上用 electron 应用
- 作为文化人得在久坐上花 10h 以上，每 5 小时休息一次
- 多学习到需要眼镜，多久坐到需要小米 su7，多吃少动到去健身房

-
- 必须在国家间站队，必须在编程语言间站队，必须在编程工具间站队
 - 少睡觉，少运动，多上网，多久坐，多比较，人生会更好
 - Zig 比 C 更好，我听别人 blogging 说的
 - Rust 比 C++ 更好，我听别人 blogging 说的
 - Golang 比 Java 更好，我听别人 blogging 说的
 - Mojo 比 Python 更好，我听别人 blogging 说的
 - Common Lisp 比 Emacs Lisp 更好，我听别人 blogging 说的
 - 新的东西一定是比旧的东西好！别人的话一定比自己想的对！
 - 学历很重要，每天关心大学排名，啥 qs 啥校友会什么抖音排行榜
 - 钱很重要，不然怎么购买带音乐的智能车，带音乐的智能床，带音乐的智能床头柜。
 - 领导：特朗普/金正恩/…All
 - 认同：LGBTQ++/左右翼/MBTI/xx 党
 - 消费：Labubu/谷子/头像壁纸/比特币
 - 编程：C++/Rust/Java/Windows
 - 生殖：处男/剩女/生育率/性病/约炮
 - 娱乐：Hentai/动漫/电影/电视剧
 - 成就：健身房/网红景点/高等教育
 - 性欲：戒色吧/吃瓜网/海角社区
 - 现在：Sora2/AI 女友/社交媒体
 - 未来：电车/机器人/低空经济
 - 青年：教育/数学/计算机/传销
 - Such A Bloated and Noise World

markdown-mode

cmake-mode

zig-mode

cargo-mode

rust-mode

go-mode

go-template-mode

bqn-mode

capnp-mode

circom-mode

clojure-mode

csharp-mode

d-mode

dockerfile-mode

elixir-mode

enh-ruby-mode

erlang

fennel-mode

fsharp-mode

fstar-mode

gams-mode

gcode-mode

```
gdshader-mode
glsl-mode
graphviz-dot-mode
hack-mode
haskell-mode
haxe-mode
hoa-mode
ini-mode
jbeam-mode
jinja2-mode
just-mode
k8s-mode
kdl-mode
kotlin-mode
less-css-mode
lfe-mode
lua-mode
lumos-mode
mermaid-mode
mgmtconfig-mode
minizinc-mode
move-mode
nael
neut-mode
nginx-mode
nim-mode
php-mode
ponylang-mode
puppet-mode
purescript-mode
python-mode
qml-mode
racket-mode
rasi-mode
scala-mode
scallop-mode
terraform-mode
tuareg
typescript-mode
uv-mode
web-mode
yaml-imenu
yaml-mode
#+end_src
```

#+caption: Some useful link for C

```
#+begin_verse
"100.github.io"
"250bpm.com"
"adtinfo.org"
"anjuta.org"
"ansi.org"
"apophenia.info"
"apr.apache.org"
"archlinux.org"
"arduino.cc"
"astyle.sourceforge.net"
"attractivechaos.github.io"
"avro.apache.org"
"bitbucket.org"
"blog.llvm.org"
"blog.noctua-software.com"
"blog.pkh.me"
"blogs.oracle.com"
"blosc.org"
"brechtsanders.github.io"
"c-faq.com"
"c2html.sourceforge.net"
"cairographics.org"
"ccache.dev"
"ccodearchive.net"
"cdecl.org"
"cedet.sourceforge.net"
"cesanta.com"
"chipmunk-physics.net"
"clang.llvm.org"
"cmake.org"
"cmocka.org"
"coap.technology"
"codeforwin.org"
"codeplea.com"
"computing.llnl.gov"
"conan.io"
"currencykit.org"
"contiki-ng.org"
"cppcheck.sourceforge.net"
"cppreference.com"
"criterion.readthedocs.io"
"criu.org"
"cunit.sourceforge.net"
"curl.haxx.se"
```

"cyan4973.github.io"
"cygwin.com"
"czmq.zeromq.org"
"danluu.com"
"debian.org"
"dl.acm.org"
"docutils.sourceforge.net"
"dotat.at"
"download.libsodium.org"
"duckdb.org"
"duktape.org"
"ebassi.github.io"
"ed-von-schleck.github.io"
"ejdb.org"
"elixir-lang.org"
"emacs-china.org"
"embed.cs.utah.edu"
"en.wikibooks.org"
"en.wikipedia.org"
"erlang.org"
"esbmc.org"
"espressif.com"
"expat.sourceforge.net"
"ezxml.sourceforge.net"
"fabutil.org"
"facebook.github.io"
"facil.io"
"faragon.github.io"
"flintlib.org"
"fragglet.github.io"
"freeglut.sourceforge.net"
"freertos.org"
"freeweb.siol.net"
"gcc.gnu.org"
"gentoo.org"
"gist.github.com"
"git.mpich.org"
"git.sr.ht"
"github.com"
"glade.gnome.org"
"gmplib.org"
"gmsl.sourceforge.net"
"gnu.org"
"gnupg.org"
"google.github.io"

"graphics.stanford.edu"
"gstreamer.freedesktop.org"
"gtk.org"
"h2o.example.net"
"hal.science"
"hardysimpson.github.io"
"haskell.org"
"hintjens.gitbooks.io"
"hirrolot.github.io"
"hplgit.github.io"
"icculus.org"
"ieee.org"
"ietf.org"
"igraph.org"
"ioquake3.org"
"iso.org"
"jemalloc.net"
"joaotavora.github.io"
"jstimpfle.de"
"jvns.ca"
"kernel.org"
"kitsune-dsu.com"
"knking.com"
"kore.io"
"kristaps.bsd.lv"
"liballeg.org"
"libccv.org"
"libcello.org"
"libcheck.github.io"
"libcork.readthedocs.io"
"libcox.symisc.net"
"libdill.org"
"libevent.org"
"libgit2.org"
"libjpeg-turbo.virtualgl.org"
"libjpeg.sourceforge.net"
"liblfd.s.org"
"libmill.org"
"libsdl.org"
"libsound.io"
"libspng.org"
"libtrading.org"
"liburcu.org"
"libusb.info"
"libuv.org"

"libvips.github.io"
"libzip.org"
"lionet.info"
"lipforge.ens-lyon.fr"
"lldb.llvm.org"
"llhttp.org"
"lloyd.github.io"
"llvm.org"
"locklessinc.com"
"lodev.org"
"logological.org"
"lua.org"
"lvgl.io"
"lwan.ws"
"lz4.github.io"
"lzip.nongnu.org"
"maciejczyzewski.github.io"
"madmurphy.github.io"
"man7.org"
"marek.vavrusa.com"
"math-atlas.sourceforge.net"
"mesonbuild.com"
"michaelrsweet.github.io?Z3"
"mihl.sourceforge.net"
"mike.steinert.ca"
"mingw-w64.yaxm.org"
"mingw.org"
"mongoc.org"
"mpfr.loria.fr"
"mpitutorial.com"
"mpv.io"
"msgpack.org"
"msune.github.io"
"msys2.github.io"
"musl-libc.org"
"musl.libc.org"
"mynewt.apache.org"
"nanomsg.github.io"
"nappgui.com"
"nemequ.github.io"
"netbeans.org"
"nethack4.org"
"newlib.sourceforge.org"
"news.ycombinator.com"
"nullprogram.com"

”nuttx.apache.org”
”opencores.org”
”openquantumsafe.org”
”opensource.org”
”opic.rocks”
”oprofile.sourceforge.net”
”orx-project.org”
”pari.math.u-bordeaux.fr”
”paulbatchelor.github.io”
”pcc.ludd.ltu.se”
”pdclib.e43.eu”
”pdos.csail.mit.edu”
”perf.wiki.kernel.org”
”pngquant.org”
”port70.net”
”pp.ipd.kit.edu”
”premake.github.io”
”projects.malikania.fr”
”proprogramming.org”
”python.org”
”qemu-project.org”
”qt.io”
”raspberrypi.org”
”re2c.org”
”redis.io”
”remove-to-waste.info”
”repo.hu”
”riot-os.org”
”riscv.org”
”risoflora.github.io”
”roaringbitmap.org”
”rr-project.org”
”rustcc.cn”
”rustlang.org”
”savannah.nongnu.org”
”savedparadigms.files.wordpress.com”
”scientificc.github.io”
”sdl.org”
”shop.oreilly.com”
”sigrok.org”
”site.icu-project.org”
”sites.google.com”
”slepc.upv.es”
”snaipe.me”
”sod.pixlab.io”

”software.schmorp.de”
”sophia.systems”
”sourceforge.net”
”sourceware.org”
”spdx.org”
”steve-yegge.blogspot.co.nz”
”svn.msweet.org”
”talloc.samba.org”
”tartarus.org”
”tatsuhiro-t.github.io”
”tiny-rex.sourceforge.net”
”tinycthread.github.io”
”tls.mbed.org”
”tools.ietf.org”
”troydhanson.github.io”
”trumpowen.github.io”
”tulipindicators.org”
”tuxfan.github.io”
”uclibc-ng.org”
”unqlite.org”
”uriparser.github.io”
”viewsourcecode.org”
”w3.org”
”web.archive.org”
”webserver2.tecgraf.puc-rio.br”
”wiki.gnome.org”
”wiki.haskell.org”
”wiki.sei.cmu.edu”
”wiki.videolan.org”
”wolkykim.github.io”
”www.andre-simon.de”
”www.bzip.org”
”www.chiark.greenend.org.uk”
”www.codeblocks.org”
”www.codelite.org”
”www.codeproject.com”
”www.colm.net”
”www.coralbits.com”
”www.cprover.org”
”www.crasseux.com”
”www.digip.org”
”www.doxygen.nl”
”www.dyncall.org”
”www.enlightenment.org”
”www.eso.org”

”www.etalabs.net”
”www.etpan.org”
”www.fefe.de”
”www.feynarts.de”
”www.ffmpeg.org”
”www.fftw.org”
”www.flourish.org”
”www.freedesktop.org”
”www.geany.org”
”www.gedanken.org.uk”
”www.gii.upv.es”
”www.glfw.org”
”www.gnu.org”
”www.gnutls.org”
”www.greenend.org.uk”
”www gtk.org”
”www.hboehm.info”
”www.hughes.com.au”
”www.infradead.org”
”www.kdevelop.org”
”www.koanlogic.com”
”www.learn-c.org”
”www.libjpeg-turbo.org”
”www.libpng.org”
”www.libretro.com”
”www.libsdl.org”
”www.libsigil.com”
”www.libtom.net”
”www.mcs.anl.gov”
”www.mega-nerd.com”
”www.mission-base.com”
”www.mongodb.org”
”www.mpich.org”
”www.multiprecision.org”
”www.netlib.org”
”www.nlnetlabs.nl”
”www.oberhumer.com”
”www.opengl.org”
”www.openmp.org”
”www.openssl.org”
”www.oracle.com”
”www.pcre.org”
”www.pearson.com”
”www.pell.portland.or.us”
”www.planetpdf.com”

```
”www.postgresql.org”
”www.pyyaml.org”
”www.rabbitmq.com”
”www(raylib.com”
”www.recurse.com”
”www.saphir2.com”
”www.scons.org”
”www.sfml-dev.org”
”www.sgi.com”
”www.shlomifish.org”
”www.spinellis.gr”
”www.sqlite.org”
”www.symas.com”
”www.tcl.tk”
”www.throwtheswitch.org”
”www.valgrind.org”
”www.webdav.org”
”www.xiph.org”
”xforms-toolkit.org”
”xiph.org”
”xmake.io”
”xmlsoft.org”
”zephyrproject.org”
”zeromq.org”
”ziglang.org”
”zinjai.sourceforge.net”
”zlib.net”
”zserge.com”

alsa-lib.dev
xorg.libX11.dev
libGL.dev
libusb1.dev
xorg.libXcursor.dev
xorg.libXrandr.dev
xorg.libXi.dev
xorg.libXext.dev
xorg.libXfixes.dev
xorg.libXrender.dev
xorg.libXxf86vm.dev
xorg.libXinerama.dev
SDL
SDL2
SDL2image
SDL2mixer
```

SDL_image
SDL_mixer
alsa-lib
at-spi2-atk
atk
atkmm
bzip2
cairo
cups
dbus
dbus-glib
expat
ffmpeg
flac
fontconfig
freeglut
freetype
gdk-pixbuf
glew110
glib
gtk2
gtk3
harfbuzz
libGL
libappindicator-gtk2
libappindicator-gtk3
libayatana-appindicator
libcaca
libcanberra
libcap
libdbusmenu-gtk2
libdbusmenu-gtk3
libelf
libgcrypt
libidn
libindicator-gtk2
libindicator-gtk3
libjpeg
libmikmod
libogg
libpng
libpng12
librsvg
libsamplerate
libsoup3

```
libtheora
libtiff
libusb1
libva
libvdpau
libvorbis
libvpx
libxml2
ncurses
nspr
nss
openssl
openssl.dev
pango
pixman
pkg-config
speex
sqlite
stdenv.cc.cc
tbb
util-linux
webkitgtk41
xorg.libICE
xorg.libSM
xorg.libX11
xorg.libXScrnSaver
xorg.libXcomposite
xorg.libXcursor
xorg.libXdamage
xorg.libXext
xorg.libXfixes
xorg.libXft
xorg.libXi
xorg.libXinerama
xorg.libXmu
xorg.libXrandr
xorg.libXrender
xorg.libXt
xorg.libXtst
xorg.libXxf86vm
xorg.libxcb
xorg.libxshmfence
```

Well, why so many, because T.K. said so: indusitira revolution and it's consequence.

为什么人类发展成这样?

答：宇宙发展之必然。

第五章 居事寂靜，世事安宁

內容提要

□ old man yell at cloud

居事寂靜而世事安宁。

I was what I walk and see,
come out to be.

文灭志，博溺心。

- No Phone, No Internet, No Browser, No IDE, On Plain Text, Consume->Creat.
- Master NixOS/Emacs, For Into Boring Stable and Fluent MindFlow Mode.
- Read HTML Reference, Manual, Info, Help, Compilation, Source Code.
- Master C language and it's Std, and library often used with debug exprience.
- Do Leetcode, Learn Algorithm and Data Structure, Exciting to Solving Tasks.
- Do Flag.h/Paperlike-cli/WM/RTOS, Learn Real programming For Myself.
- Accumulate a codebase, Use Erlang/Zig/Rust to creat project You Like.

A 20s Old man yell at Cloud.

Phone is about 24h slaving human close-source which full of spyware made by Jews/Chinese, they make consumers arguing over and over about brand/chips/products/new fasion/new shitty color.

People consume the phone shit, like it's part of they body, they dress the phone, they customize themes, they delete apps, they seems like owner of the phones. However, Phone Own Their Users.

People use this kind tool mainly just for sex trade, pretend sociable in hundreds of fake relationship groups, and make boss slave empolyee out of their working time, make people distance each other because "they can social in phones, easily". and due to shitty phone, people no long trust each other, every message can transmit means it can save, so every chat is like public speaking, and it's not like public speaking, image you are in a chat group in reality, there will be not everyone will know what you are saying, but int chatting apps, every word is store in database, every thing you do is unforgeable.

so, people less and less social in reality, and more and more addict to phone annoy chatting, is a unhuman way to live in such "modern world".

Smart phone ads tell us that the 5G shits and big camera shit, but in reality, phone more ofthen show you ads, shotting personal porn, sharing mindless videos and misinformation, phone is useless on mindful things, it just popout endless app and shitty small screen artical which full of garbage ads.

Come and consume new phones, we have Back Screen to bring back feeling of 90s which watch porn on tiny screen and "AI Phone" to help you make fancy todo-list, and shopping junk food, the price is only need to cut off you hands and smaller you brain, because you don't need they anymore.

Actually, No Internet, there will be no Browser/Phone, But Internet is not totally evil, so you can see this blogging. Internet is aboubt blogging garbage, spread fear. All in all, Freedom Cause Compulsive Abuse.

Browser is gateway of porn and doomscroll, you search, jerk off, comment, browsing, blogging, ship electron app with chromium to everyone's phone and pc to help phone brand sell more new phones. Just make living for javascript web devs by share your pravicy to shopping tech company.

IDE give you LSP, LLM to help M\$ gain 100 billion\$, LSP Just give you tons of distraction, and fake feelings of understanding a project, it pop out when you learning, typing, the aim is to make people no longer able to write programs that work.

LLM just give you a chatbox to help you modify code you generate from LLM which is already sucks enough, by that way, you lost your own ability to modify text, you just chat and chat, You don't know your code by mental model, you just find bug at compiletime and runtime at launch, you are doomed to really love programming, you just a interface of BOSS to LLM.

LSP and 100 fancy color themes with 1 million panels and Javascript Extensions, will not save you from pain of learning, they excuses for lazy people who can't read code at all. they don't remember what function they want to use, they wait for lsp load codebase and popout endless list of useless functions, and using white on dark with 1000 "true-colors" theme for the "syntax highlighting" eye candy, They don't READ, They Browsing, They don't remember, They LSP, They don't modifying, They LLM, They don't Read Manual, They Searching.

Busy-minds cause Less Mindful.

Most of these tools are distractions. I don't want suggestions or needling warnings sliding around on my screen while I compose and examine programs; I want a quiet space to write and think. Focus is essential.

第六章 手指人类，小指世界

根据人类国际惯例，手指的耐用性: 大 > 中 > 食 > 无 >> 小。根据人类国际惯例，标准键盘使用的是 QWERTY，无边打游戏还是写代码，99% 的人大约都是这样的布局。变化的只有 90% 60%，或者 alice 这样的弯曲，还有的是用青轴或是加点 RGB。

但这些通通对我们的小拇指没有一点帮助！

如果你用的是 qwert，来那么你的小拇指的使用就很可能让你后面 50 年面临腕管综合症。你要用的有：

- modifier: shift/win/alt/ctrl/capslock
- symbols: " " [] <> , . , a

然后你的两个最耐用的大拇指只是用来打同一个空格?? hey, you know, that's shitty design.

对我而言：

- colemak(更优化高频按键分布在 homerow)
- 分体式(改善体态)
- 粉轴(减少按压需要的力度)
- homerow(分摊 modifier key 到四个手指)
- 虎码(减少按键与分心)
- 分层(让左右大拇指选层，减少键数到 36 个)
- emacs(定制化最小拇指轻量级快捷键)

已经大幅减少了小拇指的使用。但还远远不够~ 因为一些符号与按键存在在 colemak 和 emacs 中，如使用 homerow 的右方向，meta 键，还有如 q' ;o 这样的在外围小拇指区域，这样就让，小拇指承担很多需要使用符号，如右手的 " " 和左手的 { [不只是小拇指，同样的也让食指也需要拉伸去按] } … 更不用说; 键，在 c 语言中它是分隔，在 elisp 中它是注释，没它万万不能，但是它正好在我的左手小拇指上，还有 M-; C-; 通常需要我用三个手指才能按出。; 本就是在 symbol 层，需要大拇指参与才能按到，也就是 2 指按;，而 emacs 实在是太，c-v, c-h c-abcdefg 全都是有用的…一时间我竟不知道该怎么改了。

不过有一天，我在看别人 bloging 出来的配置上看到了这个！
#+begin_verse (add-hook 'after-make-frame-functions
(defun setup-blah-keys (frame) (with-selected-frame frame (when (display-graphic-p) (define-key input-decode-map (kbd "C-i") [CTRL-i]) (define-key input-decode-map (kbd "C-S-i") [CTRL-Shift-i]) (define-key input-decode-map (kbd "C-[" [CTRL-lsb]) (define-key input-decode-map (kbd "C-m") [CTRL-m]) (define-key input-decode-map (kbd "C-S-m") [CTRL-Shift-m])))) #+end_src

办法出来了！也就是加上 homerow，将原.. 使用的按键移到 c-m 中，shift+, /。本身就是 <>，将用 ctrl+, /。成为 ()，将用 alt+, /。成为 {}，将用 alt+ctrl+, /。成为 " "，将用 meta+, /。成为 []，

至于我为什么这样分，请看我从一些代码文件中得到的各符号占比：

在 flag.h(C 语言) 中

({})[一般是成对出现的，其中 () 在 C 中作为函数调用声名定义大量使用。而 <> 是单独出现的。以; 作为语义分隔也占了大量的使用。[] 使用稀少。

858 () 399 ; 69 : 278 {} 193 <> 158 " " 26 ' 41 []

在 post-init-el(elisp 语言) 中

({})[一般是成对出现的，其中 () 在为函数式语言 elisp 中大量使用，而 <> 是成对出现的。以; 作为注释也占了大量的使用。[] 使用稀少。

6186 () 4308 ; 434 : 26 {} 164 <> 2014 " " 391 ' 182 [

当然其它语言如 rust/zig 的使用情况也不一样，但大致都是这样，(用于函数，; 用于分隔，> 用于 case，“用于字符串，‘用于字符，[用于数组。

目前没有一个被广泛认可、精确到“程序员中有多少百分比存在小指问题”的权威统计数字。原因在于：1)“小指问题”并非标准医学诊断名称；2) 症状多样（疲劳、疼痛、无力、麻木、腱鞘炎等）；3) 多数调查是人体工

效学或职业健康的间接统计，而非专门针对程序员小指。

但可以基于多项相关研究与工程实践调查给出一个合理区间估计。

一、可参考的统计来源类型（非精确数字）

1. 职业性肌肉骨骼障碍 (RSI / MSD) 研究：在长期键盘工作者中（程序员、编辑、数据录入员），20%–40% 的人报告过与手指相关的不适，其中小指与手腕尺侧是高风险区域之一。
2. 程序员与高频键盘使用者的问卷调查（人体工效学、HCI 领域）：通常有 10%–25% 的受访者明确提到“小指疲劳、疼痛或过度使用”。
3. 临床报告（腱鞘炎、尺神经压迫）：在程序员群体中属于少数但显著存在，通常低于 10%，但这只覆盖“严重到就医”的情况。

二、综合估计（经验区间）如果将“小指问题”定义为：“长期编程过程中，小指出现过明显不适、疼痛、疲劳或功能下降（不一定就医）”，则一个保守且合理的估计是：

约 15%–30% 的程序员在职业生涯中经历过不同程度的小指相关问题。

如果限定为：“持续性、反复发作、影响效率的小指问题”，比例更可能在：

约 5%–15%。

三、为什么程序员的小指是高风险部位这不是偶然，而是结构性原因：

1. 负荷不成比例小指肌力最弱，却承担了大量高频符号输入 ('; () {} [] <> , " " 等)。
2. 键位设计历史包袱 QWERTY 并非为高符号密度的编程语言设计，小指被迫承担“语法职责”。
3. 静态张力与大肌群不同，小指长期处于精细控制与持续张力状态，更易疲劳和炎症。
4. 编程语言特性放大问题 C / Lisp / Rust / Zig 等语言显著提高了括号、分号、引号的使用频率。

四、工程实践中的共识（重要）在严肃的工程社区中，以下观点几乎是共识：

小指问题不是个别现象它是键盘布局 + 编程符号密度 + 长时间专注输入的系统性结果这也是为什么程序员中更容易出现：

键位重映射分层键 (layer) 拇指承担修饰键 Vim / Emacs 等减少“物理击键成本”的编辑方式

五、结论（一句话）没有精确百分比，但可以负责任地说：大约五分之一左右的程序员，在其职业生涯中，会真实地感受到“小指已经不该干这么多活了”。

如果你愿意，下一步我可以：

用你的符号统计数据，量化你个人的小指负载比例或对比 QWERTY、Colemak、定制层键在小指负荷上的数学差异

第七章 那不变的，只有变化

-10000000005 年，一切都变了。宇宙诞生了¹。

-5600000005 年，一切都变了。太阳诞生了。

-4600000005 年，一切都变了。地球诞生了。

-200000005 年，一切都变了。生物诞生了。

-80000005 年，一切都变了。恐龙诞生了。

-30025 年，一切都变了。人类诞生了。

-26025 年，一切都变了。语言诞生了。

-25025 年，一切都变了。家庭诞生了。

-22025 年，一切都变了。农业诞生了。

-20025 年，一切都变了。文明诞生了。

-10025 年，一切都变了。文字诞生了。

-9025 年，一切都变了。货币诞生了。

-8025 年，一切都变了。法律诞生了。

-1025 年，一切都变了。书写诞生了。

-225 年，一切都变了。哲学诞生了。

-100 年，一切都变了。集权诞生了。

-25 年，一切都变了。数学诞生了。

-5 年，一切都变了。宗教诞生了。

1000 年，一切都变了。宗教火热。丝绸之路、黑死病、天花、中世纪、疾病、皇帝、国王、奇迹、骑士阶层、武士、城墙、封建帝国、马鞍…

1279 年，一切都变了。文艺火热。宋朝灭亡，造出火药，指南针，造纸术，印刷术的中华文明终结。先进技术被蛮族的武力消灭，但技术开始了它的崛起…

1600 年，一切都变了。帝国火热。欧洲人发现了美洲。帝国主义、殖民主义、印刷术、土豆、巧克力、辣椒、枪械与大炮、地理大发现、数学与工程学…

1700 年，一切都变了。烟草火热。哥伦布大交换的阴影笼罩大地。黑奴、民族国家、地理学、历史学、语言学、生理学、资本主义、大英帝国…

1800 年，一切都变了。鸦片火热。资本主义的阴影笼罩大地。共产主义、物理学、汽车、文学、工业革命、开发美洲、煤气、火车…

1900 年，一切都变了。技术狂热。技术在欧洲大地遍地生根。工业化、城市化、流水线、民主、飞机、电力、公司、灯光…

1915 年，一切都变了。一战结束。技术的阴影笼罩大地。农药、化肥、化学、媒体、通讯、武器、内燃机、全球化、电报、相对论…

1945 年，一切都变了。二战结束。技术的阴影笼罩大地。女权、核弹、雷达、战斗机、计算机、工业化、旅游业、赌博业、广播、量子物理…

1965 年，一切都变了。编程火热。技术停滞或是爆发？电视、摇滚乐、基因学、编程理论、月球、民用航天、芯片、电子技术、教育扩大…

1975 年，一切都变了。贸易火热。中国迎来开放。技术在中国大地遍地生根。留守儿童、大量生产、环境污染、贫富差距、道德沦丧、城乡差距…

1995 年，一切都变了。网络火热。技术迎来它们的终极载体。科学知识/各种思想文化加速传播、游戏、营销、贸易、色情、视频/图片/电影/音乐…

2002 年，一切都变了。我诞生了。技术/成瘾/财富/变革的高峰在我这代中国人最能体现，从未体验过的事物比比皆是。从小时候看电视到青少年时的玩电脑再到成年后被各种技术主动或被动控制

…这代人注定是分化的一代。

2025 年，一切都变了。智能火热。技术停滞或是爆发？社交媒体、区块链、转基因、移动支付、智能设备…

2100 年，一切都变了。死神永生。技术黑暗或是造神？火星旅行、人工智能、机器人、脑机接口、基因修改…

1902 年出生，1910 年化学，1925 年通迅交通，1950 年核弹计算机，1970 年克隆羊，1992 年互联网？2002 年出生，2010 年手机，2025 年 ChatGPT，2050 年 AGI，2070 年具身智能，2092 年永生？平均寿命 73 岁？我大概是能苟到 90 岁？也许会有一场战争…灭绝不应该长生的人。

- 后现代技术要点：（上瘾/传销/非人）

1. 靠人传销，如加密货币，房地产拼多多。（利用心理/行为设计/资本主义自由市场）
2. 不要人类，如人工智能，通讯与机器人。（利用物理/自动化的算法与机械/大数据）
3. 让人上瘾，如社交媒体，烟酒赌博旅游。（利用生理/产量强度更大/劳动时间减少）

- 硬技术要点：医学人工智能（癌症）-> 脑机接口（义肢）-> 机器人载体（永生）-> 恒星际（星际）

- 医学大数据-> 大模型-> 生理大模型-> 基因建模？

- 脑机需要什么？生物 + 数据

- 机器人需要什么？算法 + 数据

- 恒星际？能源 + 机器人

- 问题：

- 伦理性：如果都能永生，如果有谁不能呢？
- 可靠性：基于数据，准确性不高，如果程序出错呢？
- 社会性：人类会成为什么样？幸福与快乐到底是什么？

- 悲观主义者：

- 衣/食/住/行/性…

- 以前：自制/采集狩猎/自建/只有走路/真情或强暴
 - 现在：拼多多/拼好饭/烂尾楼/倒闭国产新能源/彩礼八十八万

- 意义感和存在感

- 以前：靠身体劳作生活，自己的劳动成果，直接贡献在自己身上
 - 现在：靠国家/公司/投资生活，靠机械化身体动作/脑力说唱生活

- 人类命运与未来

- 以前：自己的命运在自己/家族/别人手上
 - 现在：自己的命运在政府/老师/老板/网友…手上

- 乐观主义者：

- 衣/食/住/行/性…

- 以前：难得/难管/土房/5 公里内/不可控
 - 现在：便宜/便宜/商品房/环游世界/易得

- 意义感和存在感

- 以前：迷信宗教与古语
 - 现在：更多知识与卫生

- 人类命运与未来

- 以前：狮子老虎毒蛇虫山路水沟低温
 - 现在：温暖舒服平静安定可控多彩舒适

- 对平常人而言，现代生活即是：

- 更少的劳动，更多的消费

¹请点此处进入 PDF 版本。

-
- 更少的暴力，更多的新闻
 - 更少的情感，更多的媒体
 - 更少的迷信，更多的知识
 - 更少的空闲，更多的享乐
 - 更少的体力，更多的智力
 - 更少的杀戮，更多的竞争
 - 更少的疾病，更多的人口
 - 更少的歌舞，更多的节目
 - 更少的宁静，更多的噪声
 - 更少的饥饿，更多的肥胖
 - 更少的家族，更多的国家
 - 更少的闲适，更多的焦虑
 - 更少的长久，更多的变动
 - 更少的现实，更多的未来

2025 年，一切都变了。我不再相信技术/社会/科学…我打算出去种田。

2035 年，一切都变了。ChatGPT，我该生孩子吗？下面是我的个人信息:xxx

2045 年，一切都变了。伟大的 ChatGPT 系统，今天您的系统算出应在人体培养室中产出多少个孩子？

2065 年，一切都变了。…101110101111101010101010000111101010111…

101001011110010111110101010101000011110101011100000000000000…

42 时间 000000000000000000 宇宙 1 太阳 56 智能 99 人类 99 历史 99 年 …

赚钱，存肌肉，学习技术，准备永生？不知道，但我们得先学习在技术时代怎么生存。毕竟，本书书文即为《技术时代的生活与工作: Tech among Life and Work》，主要写写自己的技术学习，写写技术见闻，写写人类与机器的命运。地球与这些文字一样，都小如宇宙中的沙子，可就算无人观赏，也是一种美丽。我在沙子上写着沙子，不觉渺小而觉其之美，闪闪的一点有多少国王与伟人的故事，一切尽此矣。在无限的宇宙舞台上，熵增即是技术之艺术。当星星暗淡、黑洞蒸发、文明消亡，宇宙留下的，唯有技术。

表 7.1: path-to-know-tech

process	url	for
fag	flag/cintro	c, pretent iam tsoding
embedded C book	read book	c, how compiler/os/c works
Yeetcode	leetgo/book	cpp[rust], algo ds, solve task
paperlike-c/el	paperlike-go	elisp,paperlike emacs controller
ziglings	ziglings	zig, basic ziglangs speed run
zag	fag	zig, I have language erotic
paperlike-zig	paperlike-el	zig, make cli/tary
nixos r2s	github-repos	nix, for network addiction
TsurgizOS	os.phil-opp	zig[rust], make general os for cv
nixos raspberry-pi	github-repos	nix, for embedded os/screen/driver
clings	ziglings	c, lings but clang
freertos emulator	rtos	c, use general rtos
lvgl eink rtos	lvgl	c, embedded ui driver/sdl
lvgl lora loc	graphic	c, embedded openstress/sdl/lvgl
RZOS	rtos	zig, make general rtos
Celest	game	zig, embedded game/sdl
Safephone	electronic	lvgl/eink/openstress/lora/3Dprint stm32/nix/electronic/network nsfw image/text detected

1. All in all, it's for 技术哲学

-
- (a). 网络/低速/时间/域名/ai 过滤/linux 内核构建-r2s
 - (b). 屏幕/护眼/低成瘾/驱动设计/eink 屏幕算法-paperlike
 - (c). 通信/安全/无依赖/去平台/lora 远距离通信-safephone
 - (d). 交通/电工/电子电路/openstress-去五佰²那里学修车
1. 第一阶段：打基础（C / Linux / GCC / Emacs）
 - C 语言：学会指针、内存管理、结构体、函数指针；刷一些小项目，比如实现 malloc、shell、http server。
 - Linux 基础：熟悉命令行、文件系统、进程/线程、信号、管道、套接字。
 - GCC：学会编译流程 gcc -E/-S/-c/-o，理解预处理、汇编、链接，玩一下 objdump 和 nm。
 - Emacs：把它当 IDE 来用，掌握基本编辑、调试、补全、LSP 支持。
 2. 第二阶段：系统调试与逆向（GDB / QEMU）
 - GDB：练习断点、单步、查看寄存器/内存、调试多线程/远程调试。
 - QEMU：
 - 用它运行 Linux kernel 或裸机程序。
 - 学会 qemu -S -s + gdb remote 调试，体验调试内核的感觉。
 - 研究 QEMU 的设备模拟（比如 VirtIO、PCI），理解虚拟化和硬件抽象。
 3. 第三阶段：进阶编程语言与系统（C++ / Zig / RTOS）
 - Zig：Zig 是现代系统编程语言，学习它的构建系统、内存模型，可以和 C/C++ 混合编程。
 - RTOS：从 FreeRTOS 入手，学任务调度、中断、任务间通信（队列/信号量）。可以用 QEMU 模拟 Cortex-M 板子跑 RTOS。
 4. 第四阶段：融会贯通（大型项目 / 内核 / 编译器）
 - QEMU + GDB：调试内核启动、写内核模块。
 - 编译器开发：研究 GCC 或 Clang 的前端/后端；或者用 Zig 写一个简化编译器。
 - 个人项目：比如写一个简易 RTOS，或者在 QEMU 里跑自己写的内核。
 5. 操作系统构建与升级（Yocto/Android/内核）在树莓派上交叉编译 Linux 内核，修改驱动或设备树（Device Tree）进行硬件适配。尝试用 Yocto 或 Buildroot 构建自定义 Linux 镜像，加入自己编写或修改的驱动模块。安装和编译 LineageOS（Android for Pi）或类似 Android 系统，修改系统服务或 HAL 层。实践 OTA（Over-The-Air）升级机制，模拟系统升级和回滚操作。硬件 Bringup（CPU/GPU/Memory/Peripherals）利用树莓派的 GPIO、SPI、I2C、CAN、PWM 等接口，练习外设 Bringup 和驱动调试。连接摄像头模块、显示屏（LCD 或 E-Ink）或音频模块，编写驱动和控制程序。使用 perf/ftrace/gprof 分析 CPU/GPU 性能瓶颈，优化程序调度。
 6. 系统稳定性与性能优化构建多线程/多进程网络服务，使用 socket 编程实现客户端/服务器通信，模拟并发场景。在树莓派上测试高负载条件下的系统稳定性，分析内核日志、内存占用和 CPU/GPU 使用率。实践内核参数调优，如调节调度器、内存缓存策略，观察性能变化。
 7. 客户功能定制与基线升级自己设计一个树莓派应用（如小型智能家居控制器或信息显示终端），从硬件 Bringup 到应用功能定制完整流程。模拟不同版本的系统镜像管理，练习分支合并、基线升级和版本回退。
 8. 加分技能训练
 - (a). 低功耗优化：通过关闭不必要的外设、调节 CPU/GPU 频率或使用 E-Ink 显示屏练习功耗控制。
 - (b). 虚拟化/容器：在树莓派上安装 QEMU/KVM 或 Docker，运行多系统虚拟环境，模拟嵌入式应用部署。
 - (c). 芯片平台经验：如果有 MTK 或其他 ARM 板卡，可以对比树莓派练习移植和平台适配经验。

²我父亲那边的车行亲戚

第八章 造世诱惑，光影声色

2025-12-01 更新：晴

- ① **折腾键盘**: 在我 11 月末尾，我懒癌犯了总是想些了完全无关技术的事¹。今天我又搞些无关技术的事，重写了键盘的配置，重设了 emacs 的 bind。

fufufu

NixOS 下 devShell、systemPackages、nix-ld 对 C/C++ 编译/链接环境的影响

CMake + gcc/g++ 编译、链接过程，如何定位头文件和.so/.a 库

SDL 如何封装 X11/Wayland、音频、键盘鼠标输入等

.a 和.so 的技术差异及其在构建流程中的作用

SDL1/2/3 差异和演进路线

SDL 与 OpenGL/Vulkan/WebGPU/WebAssembly 等的关系与使用场景

游戏热重载技术（不重启游戏动态更新逻辑）实现原理

SDL 开发跨平台游戏并部署到 Windows/Linux/macOS/Web 的完整流程

¹要我说 Emacs 就和与小说里秘传的十八般武林功夫一般，有定身/移形换影/透视经脉/必杀技/念经/修炼/经文/拜师/法宝/女伴/侠义…甚至武功大概不过如此…拳脚身头可用的功夫，那里有无限可能的电子计算机的空间大啊？于是我花了很长时间思考“到底编辑的招法何在”，得到了下面的招式：动：插入/切换/输出/笔记/补全/主题/跳转/投修/替换/扩张/行修/选择/整理/重复/变化/比较/潜在，阅：简用/单页/全页/源码/图书/用文/说文，用：项目/虚拟/日志/中文/终端，浏：搜/库/问

第九章 时代的瘾，不在于己

根据中国惯例:

- 通信: 腾 \$ 微信 &QQ
- 娱乐: 腾 \$ Bilibili
- 阅读: 腾 \$ 微信读书
- 邮箱: 腾 \$ QQ 浏览器 +QQ 邮箱
- 购物: 阿里字节拼多多小红微信 …
- 汽车: 小米 ………………
- 家居: 小米 ………………
- 浏览: G\$……………
- 编程: M\$ VScode
- 文档: M\$ Word
- 语言: M\$ Typescript
- 搜索: M\$ Bing
- LLM: M\$ OpenAI
- 账目: M\$ Excel
- 演示: M\$ PowerPoint
- 托管: M\$ Github
- 手机: G\$ 安卓 &A\$ 苹果
- CAD/EDA/Verilog: on M\$ Windows

根据我的惯例:

- 通信: Email
- 娱乐: Emacs Lisp
- 阅读: Emacs Nov
- 邮箱: Thunderbird
- 购物: Nil
- 汽车: Nil
- 家居: Nil
- 浏览: Emacs Eww
- 编程: Emacs
- 文档: Emacs
- 语言: Emacs Lisp & C
- 搜索: Ripgrep
- LLM: Nil
- 账目: Org
- 演示: Emacs
- 托管: Local Git
- 手机: Nil
- CAD/EDA/Verilog: Nil

中国的软色情为什么这么显著，作为一个中国男性，我时常感到好奇。

过去中国男性沉迷日本 av，现在泰国男性沉迷中国 av，这种禁忌的东西反而会引发一个人的性欲，性欲到底和什么有关？为什么文化上做不得的事后而会更让人上瘾？比起澳大利亚/日本，中国/马来西亚/泰国这样的宗教保守国家为什么色情成瘾更严重，除了色情以外，还有什么利用了人的“禁忌上瘾”。

中国的网站发展史是怎么样的，为什么像 b 站和 youtube 差别大，比如 b 站往往是很擦边软色情，youtube 没有这样的文化软色情…但甚至可以直接露点教学用吸奶器，换而言之，b 站进行利用弹幕进行社会化的色情，youtube 却只有评论…西方把色当作正常，而中国当作禁忌…一种加强虚拟群体“凝聚力”的兄弟文化。

在中国，反色情常常与“精”“血”“气”之类的挂钩，产生戒色吧之类的充满邪教气息的论坛。没有很多网站关于此，2010 年的反色情网也关闭，许多视频是从国外搬运来。但常常，比如千万播放的雀巢常出现 galgame 与手淫相关梗，常常出现的“这到提醒我了”（指发弹幕被提醒进行 PMO），中国总是很线上社会化，很少线下活动。但在美国，就有许多的书籍网站反对此，如从 1994 到现在的 enough is enough…

网络似乎总是与性相关…中国网站更是…uc 新闻，各种手机浏览器中出现的新闻，大多是男女之事…父母大多闭口不提，现实文化保守，虚拟文化放纵，这种自由，极大的加强了中国部分群体产生网瘾问题。还有中国的网站设计常常密布文字图标…像是导航网站，我从来没见过英文的导航网站，但是中文的此类网站存在极多，从二次元到律师所，从编程到求职…为什么中文产生这么多。

中国地大人多，新闻多，网站设计总是文字多。而国外地小人少，新闻少，网站设计偏向简洁。百度总是新闻…google 是简洁的…b 站是到处都是封面的和按扭的弹幕，而 youtube 是简单的…当然我没有 google 和 youtube 就很好。

中国的色情网站因为封禁而产生使用随机域名的方法访问，美国反而没有…

日本的软色情影响中国最深…欧美的硬色情反而没那么毒。

为什么中国人总说“外国人很开放”，然后近年来一直学习外国人的开放。但我所看到的是外国人的“开放”实则有许多反思…中国的封闭反而产生极大的压抑与成瘾。

中国的导航网站，美国没有。中国的网站复杂，美国没有。中国的文字密度大，英文有空格停顿。中国的文化保守/网络放纵，美国…不清楚。中国色情成瘾到了 25%，美国 10%？中国的屏幕时间更长？中国的线上活动更多？中国的线下活动更少？

你得问题并不是在询问“色情本身”，而是在追问三个更深层的结构性问题：** 禁忌如何塑造欲望、制度如何放大成瘾、媒介形态如何重塑性与群体 **。下面按逻辑层次回应，而不是按国家刻板对比。

第一，性欲并不主要由“刺激强度”决定，而是由 ** 稀缺性、禁止性与社会意义 ** 决定。人类的性系统与奖励系统高度耦合，但并不是线性关系。被允许、被日常化的刺激会迅速去敏；被禁止、被隐藏、被赋予“越界意义”的刺激，会被大脑编码为高价值目标。这是经典的“反向强化”机制：越是被压制，越容易被幻想放大。因此，“禁忌”并不会消除欲望，而是把欲望从现实关系中抽离，推向想象、符号和重复消费。这也是为什么在文化上“做不得”的事情，反而更容易形成强迫性行为。

第二，色情成瘾与“道德保守”并非对立，而是 ** 互为因果 **。在中国、马来西亚、泰国这类社会，存在三个共同特征：1) 现实层面的性话语缺失（家庭、学校、公共讨论回避）；2) 制度层面的禁止或高度管控；3) 数字空间中的低成本匿名释放。这三者叠加，形成一种典型结构：** 现实压抑 → 虚拟补偿 → 羞耻与内疚 → 再度逃逸 **。这不是“性开放程度”的问题，而是 ** 是否存在稳定、合法、可被社会承认的性表达渠道 **。日本表色情产业发达，但现实中的性伦理与亲密关系同样高度结构化；欧美看似开放，但伴随大量反思、伦理讨论、成瘾研究与公共干预。所谓“外国人很开放”，是中国内部对他者的简化想象。

第三，“软色情”在中国显著，并非审美问题，而是 ** 制度套利 **。硬色情在中国是高风险内容，于是平台、创作者与用户共同进化出一套“擦边系统”：图像不越线，语义越线；内容不越线，互动越线；平台不担责，用户共谋。弹幕正是这一机制的关键。它不是单向消费，而是 ** 群体协作的意义生成 **。一个画面本身并不色情，但在弹幕的暗示、接梗、共识中，被转化为性刺激。这是一种“社会化色情”，其兴奋点不在裸露，而在 ** 共同越界而不被惩罚 **。YouTube 没有这种文化，并不是更“道德”，而是因为它不需要绕行制度。可以直接呈现，就无需通过集体暗号。

第四，反色情的“邪教化”，源于话语真空。在中国，性常被绑定为“精”“气”“损耗”“堕落”，这是一种前现代身体观与现代焦虑的混合物。当科学的性教育、心理学解释、公共讨论缺位时，任何试图“解释痛苦”的叙事都会获得市场。戒色吧、PMO 话语的本质，并不是反性，而是 ** 试图为失控的行为寻找形而上解释 **。在欧美，对色情的批判更多走向成瘾研究、产业伦理、女性主义或神经科学；在中国，则更容易走向道德化、玄学化、敌我化。

第五，中文互联网的“高密度”“导航化”“新闻化”，与性无关，但与**不确定性焦虑**高度相关。中国的门户、导航、信息聚合，本质是对信息稀缺与信任缺失的回应。用户不相信“一个入口就够”，于是需要密集链接；平台不敢只做一件事，于是不断叠加功能。色情内容在其中异常显眼，是因为它具备三个优势：即时反馈、跨阶层共鸣、无需解释。在一个线下社交成本高、公共空间有限的社会里，性成为最廉价、最通用、最不需要语言训练的刺激。

第六，“日本软色情”“欧美硬色情”的差异，本质是**文化编码方式不同**。日本擅长将欲望符号化、角色化、叙事化；欧美更直接、身体化、去隐喻。对成瘾而言，真正“毒”的不是露骨程度，而是**是否持续激活幻想回路而不通向现实关系**。软色情更容易做到这一点，因为它永远未完成、永远可投射。

最后，你提出的一系列比例数字（成瘾率、屏幕时间、线上线下），目前没有可靠的跨国可比数据，尤其在中国。但方向判断是成立的：**越是现实空间贫瘠、性话语被驱逐、线上娱乐高度集中，成瘾问题就越容易被结构性放大**。

总结一句：中国的性问题，不是“太开放”，而是**开放被迫发生在最不该承载它的地方——匿名、碎片化、商业化的网络系统中**。当现实无法承载欲望，欲望就会寄生于技术；当技术又缺乏伦理与公共讨论，它就会反过来塑造人。



图 9.1: free-your-soul