



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: November 20, 2025

邮箱: llqingsong@qq.com



目录

第一章 TODO 被技术改变的生活	2	第九章 版本更新历史	19
第二章 学技术的哲学	3	参考文献	22
第三章 本 PDF 仓库简介	5	附录 A 开发中遇见的各种问题	23
第四章 编辑器哲学	6	A.1 TODO upload large files into the github repo	23
第五章 操作系统哲学	8	A.2 TODO learn how to reference in org	23
5.1 Zeroth, get your config in to position. . . .	8	A.3 TODO make Leetcode fresh tasks list... . .	23
第六章 键盘流哲学	9	A.4 FIXME How to modify org-mode's #+IN- CLUDE:, to let it to exclude top and buttom?	23
6.1 nav	9	A.5 KLUDGE pull request to the porn site list project...	23
6.2 num	9	A.6 OKAY try to package real-mono-theme to melpa	24
6.3 mos	9	A.7 OKAY creat new account's github repos for github.io	24
6.4 sym	10	A.8 OKAY upload Purezen to github	24
第七章 屏幕哲学	11	附录 B 嵌入式软件招聘常见要求	25
7.1 Feels:	11	附录 C 嵌入式招聘常见笔试问题	26
7.2 Features:	15	附录 D 嵌入式招聘常见面试问题	27
7.3 Tips for mono:	16		
第八章 TODO fag.h	18		
8.1 Quick Start	18		

序章 技术何为？

内容提要

- ❑ 任务编号：以名为“廖青松”的男婴为身份出生在重庆普通家庭
- ❑ 开始时间：2002-11-04
- ❑ 结束时间：总有一死
- ❑ 完成难度：简单到极难，活着也没 BOSS 啊？
- ❑ 需要小时：一生，大约三万天
- ❑ 本章要点：我对于技术与人生的看法

(如果你进入了 HTML 版本，[请点击此处进入 PDF 版本](#)。)

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、让精神长的那种沉稳。

饮食亦是自律的延伸。避免加工、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 Emacs¹，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，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。

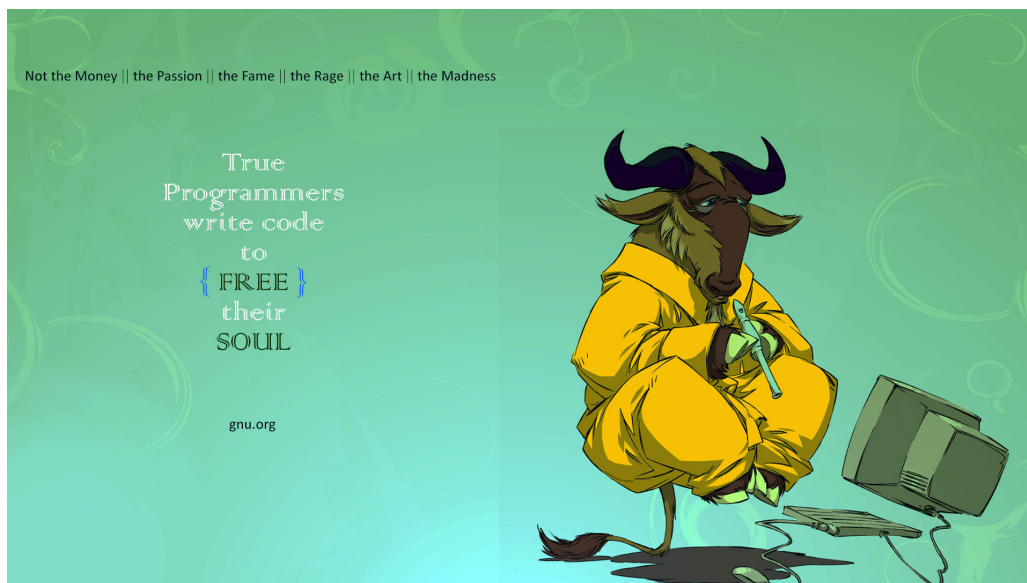


图 1: free-your-soul

¹本书的牛“图片”均出自于 GNU.org 的自由艺术。GNU 计划，又称革奴计划，是由 Richard Stallman 在 1983 年 9 月 27 日公开发起的，目标是创建一套完全自由的操作系统，将牛作为 gnu 和 emacs 的象征是因为 gnu 在英文中本身就有“牛羚、角马”的意思，有趣的是 gnu 本意为 Gnu is Not Unix…再问里面的 gnu 什么意思？还是 Gnu is Not Unix，就像是中国的俗语“山上有座庙”，无穷尽也。

第一章 TODO 被技术改变的生活

待我被裁员了之后再说，我还没工作呢！当然我有很多能写的，但是鉴于时间压力与心情不佳，暂时就不写了。

第二章 学技术的哲学

没有梦想那和咸鱼有什么区别呢？

内容提要

- ❑ 开始时间：2025-11-19

❑ 结束时间：2026 年前

❑ 完成难度：难难难
- ❑ 需要小时：100+

❑ 本章要点：linux/c/zig/rust/rtos

表 2.1: path-to-know-tech

process	url	for
fag	flag/cintro	c, pretend 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 controler
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-去五佰¹那里学修车
2. 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。
3. 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。

¹我父亲那边的车行亲戚

-
4. 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。
 5. 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。
 6. 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/trace/gprof 分析 CPU/GPU 性能瓶颈，优化程序调度。
 7. 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。
 8. 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。
 9. 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。

第三章 本 PDF 仓库简介

This is my blog and p(pdf)log, and emacs/nixos config, code snippets and even more, all in just one repo.

第四章 编辑器哲学

1. 如果人生还有一次，我还选 emacs

minimal based on the clean and fast minimal-emacs.

purezen pure white and black theme written by me called "purezen".

reproducible package manager is using **nixos**.

simple just a 1500+ lines of post-init.el.

keyboard for my **ZMK** config, the only one soul.

efficient selecting elegantly like queen and editing fastly like king.

zen the world is tremendously large, the items are enormously rich, only me along writing with leere feeling, true happiness come from nothing but within.

2. 我的编辑器邪教之旅：

浏览器里的 vimium C，因为 wsl 学习的 neovim，因为换上 nixos 后抄的 helix 配置，因为大四没找工作太闲学的 emacs。我每用一个就大呼“我草，早知道，还得是 xx!”，然后给同学老师家人都安利安利，给同学说“我有一宝”，给老师说“看我操作”，给家人嘻嘻傻笑像是捡到五百钱。虽然这代表着我非常闲，但是折腾 emacs 确实让我不是很闲以至于抑郁，我还是在用 Vimium C 和 emacs，赋予我闪电切换页面的神奇体验，当然，代价是它们也让我得了一种名为“换页面换 buffer”的 ADHD，再者，其实不用 vim/emacs 也会得，只不过叫作“懒得换页面就直接玩手机”ADHD，“懒得折腾电脑就去玩 CSgo 成了换弹癌”的 ADHD。在这个时代，玩 csgo 和玩 emacs 有什么区别呢？一个能炒皮肤，另一个能折腾皮肤，如是而矣。

3. 我使用 emacs 没有太多原因就只是：

- (a). 依赖少，之前用 vim，我就要关心 vim 键位的 app 适合用来看 pdf，看 epub 看文档，整 vimium 和 vim 一个键位/什么 lazygit/tmux/fcitx/alacritty，还生怕它们之间有什么矛盾，关键 vimscript 也不适合作为灵活的语言，整天很困扰很憋屈，我还折腾 colemak、虎码和 nixos，我的天，对于任何一个大脑决定型的成年人来说，这简直是噩梦！幸好有 emacs 解救我于快捷键地狱中，装包能 list-package 再一点就安装了，pyim 配置虎码就一行，那里还要 fcitx5 怎么怎么样！自从用了 emacs，我再也没有折改过 firefox，一是我没时间，二是我就只用 emacs 的 eww，我也就不在乎什么 vimium 的键位和 vim 不一样了，**能用就行才是真理**！再说用上了 elisp，那才叫美梦成真，想写一个工具简直就比玩手机打游戏等更有意思多了。什么 tmux/lazygit/nixos，现在我只想笑笑，我用 vterm/magit/任何平台都能跑 emacs 不是爽多了吗？只要给我 emacs/编译器，写代码就是轻轻松松的事！emacs 改好了我的强迫症和极多主义，现在我心宽又极简，感谢 emacs！
- (b). 功能多，之前用 helix，虽是开箱即用但是很明显它的功能只限于写代码，快是快但是对我这样的学习困难型人来说，快和慢没有多大的区别，只会让我再难受于“我好慢它好快，我是不是 x 痿了”，helix 是个自带丰富功能工具，但是对于习惯 homerow 的人来说其实用 vim mode 没有什么增益，只会让我更迷茫“我能直接不动手就能上下了，干啥还要这个 hjkl??”。有插件和服务模式这点就足以胜过 helix 了。当然我从 helix 的文档也学到了很多编辑知识，如果没有它的 emacs 主题和打算用 scheme 写插件系统，和让我折腾很久的 helix eink 主题，我也不会有潜移默化现在的编辑技术和极简主题审美了。
- (c). 分心小，之前用 vimium 那就是一个换换换的感觉，总是想去换页面，看看新闻看看新工具，上下跳转，左右腾挪，helix 功能太少不能在里面呆太久，而 vimium 是个浏览器插件在里面呆着又太容易去娱乐，用 vim 主要是去 youtube 参加编辑器神战，helix 也好 vimium 也好。其实我学习 emacs，到能快速掌握 avy 靠得是之前用过 vimium，学习 surround/expand 靠的是 helix，能熟悉 info/man/tldr 靠得是之前在终端里学习 vim。总之 emacs 必然是编辑技术的一个终点站。
- (d). 定制化，没有一个 term emulator 支持比例字体的，只有 emacs，没有一个 editor 的主题是可以自己完全定制的，vscode 那样的使用 javascript 还难以控制还过于复杂资源又大的就算了。elisp/fsf/clean/simple，

学 emacs 之前根本不能理解这话! 有些事情得是自己做了才能得之心而应之于手!

- (e). 历史, 50 年历史的编辑器, 计算机史的一页, 还能写 100 年左右。
- (f). 装 b, (linus/stallman) 开源巨人们/(rust/python/java/ruby/curl/各种函数式…) 之父们钦定编辑器。

第五章 操作系统哲学

1. network, 471mb hosts to block internet out of my pc with dnsmasq and dae proxy tool based on ebf... you know I am ill about internet? because I am in china, and the nsfw/trump/4chan dudes, I love they so much, so I add they into my hosts.
2. emacs editor, + nixos, they both ruin my early adult life.
3. wm: sway, minimal config, pure white look, although miku cursor.
4. shell: fish+foot, with all I need in, direnv/many alias/zoxide/git/and thefxxk or thefxxk updated to "chatgpt"'s codex nowadays?
5. browser: firefox [default uninstall], I use eww because I hate browser.
6. fully reproducible, even with dict/font/anime wallpaper!
7. zig/linux/ccpp doc in ngnix, all kind texinfo can read in emacs.
8. with sops, to store my 0.1 dollar's deepseek account apikey, and chat with my ai girlfriend with my payment password.
9. I am tired of nixos, but I can't leave it, because after using nixos everything is hard to do in other distro, like show off to other distro users "BTW, I use NixOS".

论我怎么戒色? 答:nixos。

5.1 Zeroth, get your config in to position.

```
(org-babel-do-load-languages
'org-babel-load-languages '((emacs-lisp . t)(shell . t)))
```

```
mkdir -p $HOME/Leere/
cp -r $PWD $HOME/Leere/
```

```
sudo cp /etc/hardware-configuration.nix $PWD/nixos/device/desktop/hardware-configuration.nix
chmod 777 $PWD/nixos/device/desktop/hardware-configuration.nix
```

```
mkdir -p ~/.config/spos/age/
ln -sf $PWD/nixos ~/.config/
ln -sf $PWD/emacs.d ~/.emacs.d
ln -sf $PWD/resource/keys.txt ~/.config/spos/age/keys.txt
```

```
sudo nix-channel --add https://nixos.org/channels/nixos-unstable
sudo nix-channel --update
```

```
sudo nixos-rebuild switch --flake /home/leeao/.config/nixos#nixos --option substituters 'https://
mirrors.ustc.edu.cn/nix-channels/store https://cache.nixos.org' --cores 6 -j 12 --option
experimental-features "nix-command flakes"
```

第六章 键盘流哲学

- 与原生相合 (win)
- 不冲突 (special prefix)
- 合乎习惯 (stay base)
- home row 效率 (home row effiction)
- general in os(browser, editor)

6.1 nav

```
(q[mark-sexp]w[mark-word]f[mark_defun]p[mark-whole-buffer]b[quick-sdcv-search-at-point])
  (g[er/expand-region])(m[toggle-truncate-lines])
(z[kill-current-buffer]x[delete-other-windows]c[split-window-below]d[switch_other]v[switch-to-buffer-ot
(enter[previous-error]/delete[next-error])

(q[C-M-@]w[M-@]f[C-M-h]p[C-x h]b[c-c c-~ b])
  (g[c-c c-~ g])(m[c-c c-~ m])
(z[c-x 0]x[c-x 1]c[c-x 2]d[c-x o]v[c-c c-~ v])
(enter[M-g p]/delete[M-g n])
```

6.2 num

```
(j[pause]/l[full]/u[down]/y[leftright]/'[open-app]/)
(m[eink])
(k[kill-win]/h[full]/,[down]/.[leftright]//[rotate])

(j[s-shift-j]/l[s-shift-l]/u[s-shift-u]/y[s-shift-y]/'[s-shift-'])
(m[s-shift-m])
(k[s-shift-k]/h[s-shift-h]/,[s-shift-s]/.[s-shift-.]//[s-shift-/])
```

6.3 mos

```
(q[gptel-menu]/w[consult-gh]/f[rg-dwim]/p[disproject-dispatch]/b[ibuffer])(j[font-lock-mode]/l[tldr]/u[
(a[comment]/r[quit]/s[find-file]/t[consult-buffer]/g[magit-status])(m[man])
(z[search-at-point]/x[query-replace-regexp]/c[disproject-compile]/d[dired]/v[multi-vterm-project])(k[ki
(space[indent-rigidly])

(q[c-c c-; q]/w[c-c c-; w]/f[c-c c-; f]/p[c-c c-; p]/b[c-x c-b])(j[c-c c-; j]/l[c-c c-; l]/u[c-c c-; u]
(a[c-x c-;]/r[c-g]/s[c-x c-f]/t[c-x b]/g[c-x g])(m[c-c c-; m])
(z[M-s .]/x[C-M-%]/c[c-c c-; c]/d[c-x d]/v[c-c c-; v])(k[c-x k])
(space[C-X tab])
```

6.4 sym

```
(prompt1[just_give_me_he_code]/prompt2[explain_in_chinese]/prompt3[Think_longer]/prompt4[Search_The_Web]
(m[switch-to-gptel])
(password/id/qq/email/phone)
```

```
(prompt1[just_give_me_he_code]/prompt2[explain_in_chinese]/prompt3[Think_longer]/prompt4[Search_The_Web]
(m[s-m])
(password/id/qq/email/phone)
```

第七章 屏幕哲学

Real Mono Theme, two colors are enough for emacs.

7.1 Feels:

A collection of real monochrome emacs themes in a couple of variants.

/home/leeao/Leere/real-mono-theme
image

real-mono-black.png
real-mono-girl.png
real-mono-oldfasion.png
real-mono-white.png
README.md
real-mono-black-theme.el
real-mono-girl-theme.el
real-mono-oldfasion-theme.el
real-mono-themes.el
real-mono-white-theme.el

```
/**
 * @file
 * @brief Implementation to
 * [Count number of bits to be flipped to convert A to B]
 * (https://www.geeksforgeeks.org/count-number-of-bits-to-be-flipped/
 * in an integer.
 *
 * @details
 * We are given two numbers A and B. Our task is to count the number
 * needed to be flipped to convert A to B.
 *
 * Explanation:
 *
 * A = 01010 B = 10100
 * As we can see, the bits of A that need to be flipped are 01010.
 * If we flip these bits, we get 10100, which is B.
 *
 * Worst Case Time Complexity: O(log n)
 * Space complexity: O(1)
 * @author [Yash Raj Singh](https://github.com/yashrajyash)
 */
#include <cassert> /// for assert
#include <cstdint>
#include <iostream> /// for IO operations
/**
 * @namespace bit_manipulation
 * @brief Bit manipulation algorithms
 */
namespace bit_manipulation {
/**
 * @namesnace count_bits_flip
```

Head: *master* Initial commit

Untracked files (2)

Unstaged changes (4)

modified README.md

@@ -10,7 +10,7 @@ A collection of real
![real-mono-girl](https://raw.githubusercontent.com/leeao/real-mono-oldfasion)(https://raw.g

-pic's font list: bookerly, ubuntumono
+pictures's font list: bookerly, ubuntu

Features:

1. ****Not Greyscale****, no blur anymore
@@ -19,13 +19,38 @@ pic's font list: bookerly
4. ****Full****, configed 360+ faces, I didn't

Tips for more mono:

-1. global-hide-mode-line-mode, build
-2. no-emoji, alter emacs to be "text ed
-3. fringe specific mode auto hide
-4. turn off show-paren-mode, build y
-5. window-divider-mode, too, build y
-6. font switch, switch between propo
-7. display-line-numbers-mode, too, b
+- (global-hide-mode-line-mode 1), bu
+- (no-emoji 1), alter emacs to be "text
+- (show-paren-mode -1), build your c
+- (window-divider-mode -1), too, bu
+- (display-line-numbers-mode -1), to
+- font switch, switch between propo
+
+ `` ` lisp
+(defvar my-alternate-font "-DAMA-I
+(defvar my-default-font "bookerly")
+(defvar fontfont 1)

```

/home/teeao/Leere/real-mono-themes:
image
  real-mono-black.png
  real-mono-girl.png
  real-mono-oldfasion.png
  real-mono-white.png
README.md
real-mono-black-theme.el
real-mono-girl-theme.el
real-mono-oldfasion-theme.el
real-mono-themes.el
real-mono-white-theme.el

***
* @file
* @brief Implementation to
* [Count number of bits to be flipped]
* (https://www.geeksforgeeks.org/count-number-of-bits-to-be-flipped/)
* in an integer.
*
* @details
* We are given two numbers A and B.
* needed to be flipped to convert A to B.
*
* Explanation:
*
* A = 01010 B = 10100
* As we can see, the bits of A that are different from B are
* If we flip these bits, we get 10100 which is B.
*
* Worst Case Time Complexity: O(log n)
* Space complexity: O(1)
* @author [Yash Raj Singh](https://www.geeksforgeeks.org/author/yash-raj-singh/)
*/
#include <cassert>    /// for assert
#include <cstdint>
#include <iostream>    /// for IO operations

```

```

Head:      main hh
Rebase:    upstream/main Add package-
Push:      origin/main ready to pull
Tag:       1.3.0 (109)

Untracked files (2)
Unstaged changes (4)
modified   custom.el
@@ -4,7 +4,20 @@
    ;; If you edit it by hand, you could lose it.
    ;; Your init file should contain something like this to use the theme:
    ;; If there is more than one, they will be applied in order.
- '(org-agenda-files nil nil nil "C")
+ '(custom-enabled-themes '(real-mono-themes))
+ '(package-selected-packages
+   '(aggressive-indent alert almost
+     compile-anglican consult-gh-pages dired-subtree
+     eshell-toggle-hideshow hide-mode-line
+     marginalia nix-mode no-litter
+     posix-manual saveplace-popup tldr trashed
+     zig-mode ztrace))
+ (custom-set-faces
+   ;; custom-set-faces was added by custom.
+   ;; If you edit it by hand, you could lose it.
modified   post-early-init.el
modified   post-init.el
modified   themes/purezen-theme.el

Unpushed to origin/main (256+)
Unmerged into upstream/main (13)
316e29b main hh
a57bc13 local/main modified: post
8a9027a fix vterm
c7c73b9 fix org theme
428c493 rearrange
27c707c wait I test
e94a505 I don't want to be configer
70bfd82 configing theme is drug
fb81f8e ready to pull
eaff15e new file: nestor.el
d07275c before offline

```



```

/home/leeao/Leere/real-mono-t
image
  real-mono-black.png
  real-mono-girl.png
  real-mono-oldfasion.png
  real-mono-white.png
README.md
real-mono-black-theme.el
real-mono-girl-theme.el
real-mono-oldfasion-theme.el
real-mono-themes.el
real-mono-white-theme.el

/**
 * @file
 * @brief Implementation to
 * [Count number of bits to be
 * (https://www.geeksforgeeks.o
 * in an integer.
 *
 * @details
 * We are given two numbers A a
 * needed to be flipped to conv
 *
 * Explanation:
 *
 * A  = 01010 B  = 10100
 * As we can see, the bits of A
 * If we flipthese bits, we get
 *
 * Worst Case Time Complexity:
 * Space complexity: O(1)
 * @author [Yash Raj Singh](htt

```

```

Head:      main hh
Rebase:    upstream/main Add pa
Push:      origin/main ready to
Tag:       1.3.0 (109)

Untracked files (2)
Unstaged changes (4)
modified   custom.el
@@ -4,7 +4,20 @@
    ;; If you edit it by hand, y
    ;; Your init file should con
    ;; If there is more than one
- '(org-agenda-files nil nil n
+ '(custom-enabled-themes '(re
+ '(package-selected-packages
+   '(aggressive-indent alert
+                                     compil
+                                     consul
+                                     dired-
+                                     eshell
+                                     hide-m
+                                     margin
+                                     nix-mo
+                                     posix-
+                                     savepl
+                                     tldr t
+                                     zig-mo
(custom-set-faces
  ;; custom-set-faces was adde
  ;; If you edit it by hand, y
modified   post-early-init.el
modified   post-init.el
modified   themes/purezen-them

```

```

Unpushed to origin/main (256+)
Unmerged into upstream/main (1
316e29b main hh
a57bc13 local/main modified:
8a9027a fix vterm
c7c73b9 fix org theme

```

```

/home/leeao/Leere/real-mono-theme
image
  real-mono-black.png
  real-mono-girl.png
  real-mono-oldfasion.png
  real-mono-white.png
README.md
real-mono-black-theme.el
real-mono-girl-theme.el
real-mono-oldfasion-theme.el
real-mono-themes.el
real-mono-white-theme.el

/**
 * @file
 * @brief Implementation to
 * [Count number of bits to be flipped]
 * (https://www.geeksforgeeks.org/count-number-of-bits-to-be-flipped-in-an-integer/)
 *
 * @details
 * We are given two numbers A and B. We need to find the number of bits
 * needed to be flipped to convert A to B.
 *
 * Explanation:
 *
 * A = 01010 B = 10100
 * As we can see, the bits of A that are different from B are the 1st, 3rd, 4th, and 5th bits.
 * If we flip these bits, we get 10100 which is equal to B.
 *
 * Worst Case Time Complexity: O(N)
 * Space complexity: O(1)
 * @author [Yash Raj Singh](https://www.geeksforgeeks.org/author/yash-raj-singh/)

```

```

Head:      main hh
Rebase:    upstream/main Add package
Push:      origin/main ready to push
Tag:       1.3.0 (109)

Untracked files (2)
Unstaged changes (4)
modified   custom.el
@@ -4,7 +4,20 @@
    ;; If you edit it by hand, you
    ;; Your init file should contain
    ;; If there is more than one, then
- '(org-agenda-files nil nil nil
+ '(custom-enabled-themes '(real-
+ '(package-selected-packages
+ '(aggressive-indent alert alarm
+                               compile-async
+                               consult-completers
+                               dired-subprocess
+                               eshell-terminal
+                               hide-mode
+                               marginalia
+                               nix-mode
+                               posix-marks
+                               saveplace
+                               tldr-tracker
+                               zig-mode
    (custom-set-faces
      ;; custom-set-faces was added to
      ;; If you edit it by hand, you
modified   post-early-init.el
modified   post-init.el
modified   themes/purezen-theme.el

Unpushed to origin/main (256+)
Unmerged into upstream/main (13)
316e29b main hh
a57bc13 local/main modified: package
8a9027a fix vterm
c7c73b9 fix org theme

```

<pre> /home/leeao/Leere/real-mono-theme image real-mono-dark.png real-mono-eink.png real-mono-girl.png real-mono-old.png README.md real-mono-dark-theme.el real-mono-eink-theme.el real-mono-girl-theme.el real-mono-old-theme.el real-mono-sea-theme.el real-mono-themes.el /** * @file * @brief Implementation to * [Count number of bits to be flipped] * (https://www.geeksforgeeks.org/c * in an integer. * * @details * We are given two numbers A and B * needed to be flipped to convert A to * * Explanation: * * A = 01010 B = 10100 * As we can see, the bits of A that need * If we flip these bits, we get 10100, u </pre>	<pre> Head: master Initial commit Untracked files (7)... Unstaged changes (10) modified README.md @@ -1,52 +1,102 @@ Real mono themes ===== -Just A Real Mono Theme +Real Mono Theme, two colors are e ## Feels: A collection of real monochrome en -![real-mono-white](https://raw.gi -![real-mono-black](https://raw.gi +![real-mono-eink](https://raw.gi +![real-mono-dark](https://raw.gi +![real-mono-girl](https://raw.gith -![real-mono-oldfashion](https://ra +![real-mono-old](https://raw.gith +![real-mono-sea](https://raw.gith -pic's font list: bookerly, ubuntumono +pictures's font list: bookerly, ubun ## Features: -1. Not Greyscale, no blur anymore +1. Not Greyscale, no blur anymore 2. Easy to customize, can set the -3. Distraction-free, no extra info </pre>
---	---

pictures's font list: bookerly, ubuntumono, firacode, terminess, bookerly blod Italic.

7.2 Features:

1. **Not Greyscale**, no blur anymore, it's much better to use eink screen for pure black and white!
2. **Easy** to customize, can set the only two colors by config the default face's foreground/background color.
3. **Distraction-free**, no extra info-overwhelming causing by font-lock, only few font diversity in magit/dired etc for better function recognize.

4. **Full**, configed 370+ faces, I didn't see any monochrome theme can reach that much, as my daily driver, it's good enough.

7.3 Tips for mono:

- (global-hide-mode-line-mode 1), build your own brain memory
- (no-emoji 1), alter emacs to be "text editor" instead of discord
- (show-paren-mode -1), build your own eye insight
- (window-divider-mode -1), too, build your memory
- (display-line-numbers-mode -1), too, build your own eye insight

```
(defvar my-alternate-font "-DAMA-UbuntuMono Nerd Font-regular-normal-normal--13---m-0-iso10646-1") (def-
var my-default-font "bookerly") (defvar fontfont 1) (defun my-toggle-font () "Toggle between UbuntuMono and bookerly
fonts." (interactive) (if (= fontfont 1) (progn (set-face-attribute 'default nil :font my-default-font :height 160) (setq fontfont
0)) (progn (set-face-attribute 'default nil :font my-alternate-font :height 210) (setq fontfont 1)))) #+end_src
```

- fringe specific mode auto hide

```
(defun my-set-fringe-face ()
  "auto_hide_fringe_face Depending on major mode."
  (if (derived-mode-p '(occur-mode gud-mode))
      (set-face-attribute 'fringe nil
                          :background (face-attribute 'default :background)
                          :foreground (face-attribute 'default :foreground))
      (set-face-attribute 'fringe nil
                          :background (face-attribute 'default :background)
                          :foreground (face-attribute 'default :background))))
(add-hook 'after-change-major-mode-hook #'my-set-fringe-face)

- elisp for toggling paperlike-hd to switch between read and watch.
``` elisp
(defvar monitor-state 0
 "Current_monitor_state, either 0 for read or 1 for watch.")
(defun monitor ()
 "switch_monitor from read mode to watch mode"
 (interactive)
 (let((monitorpath "-i2c_/dev/i2c-4")
 (monitorcli "paperlike-cli")
 (monitorarg '("_contrast_" "_speed_" "_mode_" "_clear_"))
 (mode-state '(("9" "5" "1") ("2" "1" "3"))))
 (split-window-below)
 (other-window 1)
 (switch-to-buffer "*Shell_Command_Output*")
 (split-window-below)
 (other-window 1)
 (switch-to-buffer "*Async_Shell_Command*")
 (progn
 (dotimes (number 3)
 (shell-command (concat
 monitorcli
```

```
monitorpath
(car (nthcdr number monitorarg))
(car (nthcdr number (car (nthcdr monitor-state mode-state)))))
))
(sleep-for 1))
(setq monitor-state (if (= 0 monitor-state) 1 0))
(sleep-for 1.5)
(sleep-for 0.5)
(async-shell-command (concat monitorcli monitorpath "_clear"))
(let ((async (get-buffer-window "*Async_Shell_Command*"))
 (shell (get-buffer-window "*Shell_Command_Output*")))
 (when async (delete-window async))
 (when shell (delete-window shell)))
(sleep-for 0.5)
(kill-buffer "*Async_Shell_Command*")
(kill-buffer "*Shell_Command_Output*")
(donothing)))
```

## 第八章 TODO fag.h

Copied from tsoding's flag module: <https://github.com/tsoding/flag.h>

### 8.1 Quick Start

Check `example.c`

```
cc -o example example.c
./example -help
```



## 第九章 版本更新历史

2025-11-19 更新：晴

- ① 天气好，玩了一下午结果踩在沥水槽腿掉下去了
- ② leetcode 让 chatgpt 写 leetgo 的 elisp 工具，写到最后凌晨没网了!
- ③ 后悔的一天 reddit 里的 china\_irl 我常常去看，实在是没意思，但总有一种引力让我去刷，莫名的又想看片，于是乎又成了看片加键政，什么高市什么献忠，吃瓜吃到 10 年的兽兽门，再看现在的其他事件，感觉 10 年代清新多了，那时候的主角还是 87 年“小妹”，现在则是 0 几年的“小妹”，真是世风日下，历史重是重复又重复，技术重是加强又如强啊，献忠机器人，何时到来？假韩炳哲的话改下：“互联网让道德世风内涵成了空气，一切除了炫/爽/性没有他用”。
- ④ 折腾 nixos/emacs 精简所有文件到博客这一个仓库 nixos 只用一个 flake.nix 和 secrets.yaml 再把一串 key 放到一个位置就能用了，简直是极简完了。emacs 只用一个 post-init.init，再 git clone 下 minimal-emacs 到位，org 再 src 运行一下，就能用了。这两都是 2000 行的配置。总之我有了一个 blog/pdf/emacs/nixos/snippet/hosts 集于一体的仓库。完全可重现，从配置到经历，其它的代码也可以从其它 repo 里 clone 下来，但总之目前这些就是我大学大概折腾的所有玩具了吧。

2025-11-18 更新：阴

- ① 沉迷看书<sup>1</sup> 错过了我设定的上网时间二点到五点<sup>2</sup>...明天再测试吧<sup>3</sup>。
- ② C 语言刷一遍 flag 代码，发现自己还是不太懂 C 语言。今天又没刷力扣，感觉自己很废，就这样还想搞什么？

2025-11-17 更新：雨

- ① 开始刷 Leetcode 完成 0 题，折腾 leetgo 去了，使用 async-shell-command 与 emacs 不全好用，两者的时间不一致问题，shell-command 又卡 emacs 本身，两者同样不能管理弹出的 shell 输出。只能用完整的 shell 命令和在 async 里 eval emacs function 套娃看看。
- ② 将本“博客”的 html 和 pdf 上传了 github.io 只是不只为何几十分钟了 qingsongliao.github.io 还没有，明天再看吧。
- ③ 完成了 real-mono-themes 的 emacs 主题包从此 emacs 又多了个别具一格的主题，不过也许 70 年代的 emacs 就长这样吧，oldfashion never die。

2025-11-16 更新：阴

- ① 开始写作
- ② 使用 elegant 模板搭建自己博客？写书还是写博客？干脆一起写吧。

2025-11-19 更新：晴

- ① 天气好，玩了一下午结果踩在沥水槽腿掉下去了
- ② leetcode 让 chatgpt 写 leetgo 的 elisp 工具，写到最后凌晨没网了!

<sup>1</sup>把什么现代诗选和中国皇帝传都丢了吧，再看下去这辈子就毁了！只留下两本书，互联网浅薄与雅思 7 天词汇，一是提醒我互联网对大脑的“伤害”另一个是提升下我的大脑，最近几年或十年特别是高考后，感觉自己的脑子雾雾的。

<sup>2</sup>我有点网瘾，所以通过家里的路由器限制一下。速度 200kps，限制热门社交媒体的域名，时间上只有下午二点到五点能用。每次无限制上网都有一种沉迷的感觉，看黄片刷新闻作为瘾症生意在互联网真是完美载体，总之，戒断的痛苦真是难受啊!!!

<sup>3</sup>晚上又用母亲的手机上网了，怎么试都发现这个 github.io 是 404，下载下来好像是 html 本身的问题。切换了一下账号发现看不了那个号了，问了下 chatgpt 发现 github 有新号防 bot 功能，没法只能重新用回我的 NestorLiao 账号 [https://github.com/orgs/community/discussions/55609?utm\\_source=chatgpt.com](https://github.com/orgs/community/discussions/55609?utm_source=chatgpt.com)

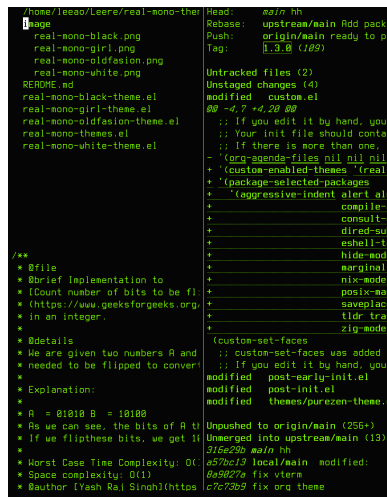


图 9.1: real-mono-old

- ③ 后悔的一天 reddit 里的 china\_irl 我常常去看，实在是没意思，但总有一种引力让我去刷，莫名的又想看片，于是乎又成了看片加键政，什么高市什么献忠，吃瓜吃到 10 年的兽兽门，再看现在的其他事件，感觉 10 年代清新多了，那时候的主角还是 87 年“小妹”，现在则是 0 几年的“小妹”，真是世风日下，历史重是重复又重复，技术重是加强又如强啊，献忠机器人，何时到来？假韩炳哲的话改下：“互联网让道德世风内涵成了空气，一切除了炫/爽/性没有他用”。
- ④ 折腾 nixos/emacs 精简所有文件到博客这一个仓库 nixos 只用一个 flake.nix 和 secrets.yaml 再把一串 key 放到一个位置就能用了，简直是极简完了。emacs 只用一个 post-init.init，再 git clone 下 minimal-emacs 到位，org 再 src 运行一下，就能用了。这两都是 2000 行的配置。总之我有了一个 blog/pdf/emacs/nixos/snippest/hosts 集于一体的仓库。完全可重现，从配置到经历，其它的代码也可以从其它 repo 里 clone 下来，但总之目前这些就是我大学大概折腾的所有玩具了吧。

2025-11-18 更新：阴

- ① 沉迷看书<sup>4</sup> 错过了我设定的上网时间二点到五点<sup>5</sup>…明天再测试吧<sup>6</sup>。
- ② C 语言刷一遍 flag 代码，发现自己还是不太懂 C 语言。今天又没刷力扣，感觉自己很废，就这样还想搞什么？

2025/11/17 更新：雨

- ① 开始刷 Leetcode 完成 0 题，折腾 leetgo 去了，使用 async-shell-command 与 emacs 不全好用，两者的时间不一致问题，shell-command 又卡 emacs 本身，两者同样不能管理弹出的 shell 输出。只能用完整的 shell 命令和在 async 里 eval emacs function 套娃看看。
- ② 将本“博客”的 html 和 pdf 上传了 github.io 只是不只为何几十分钟了 qingsongliao.github.io 还没有，明天再看吧。
- ③ 完成了 real-mono-themes 的 emacs 主题包从此 emacs 又多了个别具一格的主题，不过也许 70 年代的 emacs 就长这样吧，oldfashion never die。

<sup>4</sup>把什么现代诗选和中国皇帝传都丢了吧，再看下去这辈子就毁了！只留下两本书，互联网浅薄与雅思 7 天词汇，一是提醒我互联网对大脑的“伤害”另一个是提升下我的大脑，最近几年或十年特别是高考后，感觉自己的脑子雾雾的。

<sup>5</sup>我有点网瘾，所以通过家里的路由器限制一下。速度 200kps，限制热门社交媒体的域名，时间上只有下午二点到五点能用。每次无限制上网都有一种沉迷的感觉，看黄片刷新闻作为瘾症生意在互联网真是完美载体，总之，戒断的痛苦真是难受啊!!!

<sup>6</sup>晚上又用母亲的手机上网了，怎么试都发现这个 github.io 是 404，下载下来好像是 html 本身的问题。切换了一下账号发现看不了那个号了，问了下 chatgpt 发现 github 有新号防 bot 功能，没法只能重新用回我的 NestorLiao 账号 [https://github.com/orgs/community/discussions/55609?utm\\_source=chatgpt.com](https://github.com/orgs/community/discussions/55609?utm_source=chatgpt.com)

```
/home/leao/Leere/real-mono-theme
#
real-mono-black.png
real-mono-girl.png
real-mono-oldfashion.png
real-mono-white.png
README.md
real-mono-black-theme.el
real-mono-girl-theme.el
real-mono-oldfashion-theme.el
real-mono-theme.el
real-mono-white-theme.el

Head: main hh
Rebase: upstream/main Rdd packt
Push: origin/main ready to pu
Tag: 1.3.0 (189)

Untracked files (2)
Unstaged changes (4)
modified custom.el
00 -d,? +d,20 00
:: If you edit it by hand, you
:: Your init file should contain
:: If there is more than one, t
- '(org-agenda-files nil nil nil
+ (custom-enabled-themes '("real-
+ (package-selected-packages
+ '(aggressive-indent alert els
+ compile-e
+ consult-t
+ dired-sub
+ eshell-tc
+ hide-mode
+ marginali
+ nix-mode
+ posix-mar
+ saveplace
+ tldr tras
+ zig-mode

(custom-set-faces
:: custom-set-faces was added b
:: If you edit it by hand, you
modified post-early-init.el
modified post-init.el
modified themes/purezen-theme.e

Unpushed to origin/main (255+)
Unmerged into upstream/main (13)
316e20b main hh
a57bc13 local/main modified: p
0a9027a fix vterm
c7c72e3 fix org theme
```

图 9.2: real-mono-old

2025/11/16 更新：阴

- ① 开始写作
- ② 使用 **elegant** 模板搭建自己博客? 写书还是写博客? 干脆一起写吧。

## 参考文献

- [1] Charles T Carlstrom and Timothy S Fuerst. “Agency Costs, Net Worth, and Business Fluctuations: A Computable General Equilibrium Analysis”. In: *The American Economic Review* (1997), pp. 893–910. ISSN: 0002-8282.
- [2] 方军雄. “所有制、制度环境与信贷资金配置”. In: *经济研究* 12 (2007), pp. 82–92. ISSN: 0577-9154.
- [3] Qiang Li, Liwen Chen, and Yong Zeng. “The Mechanism and Effectiveness of Credit Scoring of P2P Lending Platform: Evidence from Renrendai.com”. In: *China Finance Review International* 8.3 (2018), pp. 256–274.
- [4] 刘凤良, 章潇萌, and 于泽. “高投资、结构失衡与价格指数二元分化”. In: *金融研究* 02 (2017), pp. 54–69. ISSN: 1002-7246.
- [5] 吕捷 and 王高望. “CPI 与 PPI “背离” 的结构性解释”. In: *经济研究* 50.04 (2015), pp. 136–149. ISSN: 0577-9154.
- [6] Vincenzo Quadrini. “Financial Frictions in Macroeconomic Fluctuations”. In: *FRB Richmond Economic Quarterly* 97.3 (2011), pp. 209–254.

## 附录 A 开发中遇见的各种问题

### A.1 TODO upload large files into the github repo

use sed to filter, use what to split into small files.

### A.2 TODO learn how to reference in org

[<empty citation>] org-cite-insert unable to insert problem

### A.3 TODO make **Leetcode** fresh tasks list...

- have already?
- yes/no, yes, edit that file, no, get the files(testcase/question.md/solution.xxx)
- edit the files question to include src in end, solution.xxx to have mode line in header.
- download image to local, turn md in org format, and delete md one.
- add a allorg, to include all under src dir's question.org
- 支持无网本地测试，在源代码文件头部中加入 mode line，在题目文件尾部中加入 include 源代码
- 根据题号在相同的目录下抓取不同题目和几个指定语言，并下载至本地图片，再转化 md 为 org 格式
- 支持在根目录中得到 src 的所有 org file 的 include
- md to org 的 == 问题
- based on leetgo to creat a leetcode elisp pkg
- test leetcode emacs pkg instead

### A.4 FIXME How to modify org-mode's **#+INCLUDE:**, to let it to exclude top and buttom?

it have: top specific line to end specific line line to top specific top specific line to end contribute to the org-mode's include function to let it

exclude from top specific line to end specific line but count from buttom

- study org-mode, try to contribute to it.

### A.5 KLUDGE pull request to the porn site list project...

by clone and compare the problem is that "I include a lost non-porn site in to it". not just "non-porn" sites, I also get a lot cdn server to block, which fetch from a lot of m3u8 streaming server. it may also block some movies which using the same server as porn sites... so, there is only KLUDGE for thing like blocking porn site, first question: what is porn? really? to be, the social media like twitter which spread sexual clip? the pornhub teaching mathmatic? the reddit/4chan/zhihu/weibo/... even the taobao can sell sex toys with sexual pictures, so blocking internet is blocking porn really? I don't know, it's a question for everyone.

## A.6 **OKAY** try to package real-mono-theme to melpa

minic almost-mono-theme, creat recipe, make pr wait for mainter check. I am familaring with github, feeling awesome!

## A.7 **OKAY** creat new account's github repos for github.io

have a username.github.io public repo, have index.html in it, in [url](#)

## A.8 **OKAY** upload Purezen to github

- description: theme that really monochrome A collection of real monochrome emacs themes in a couple of variants.
- clone a monochrome and study it <https://github.com/cryon/almost-mono-themes>
- minic a theme to creat a repo <https://github.com/qingsongliao/real-mono-themes>



## 附录 B 嵌入式软件招聘常见要求

- 本科及以上学历，嵌入式软件工作经验，电子、通信或计算机相关专业；
- 有 ARM Linux 软件的开发经验，熟练使用 C++ 和 C 语言；
- 熟悉 Linux 常用设备操作，有 spi、can、WiFi、audio、video 相关驱动开发经验优先；
- 熟悉多线程、多进程和 socket 网络编程，有并发程序开发经验和良好的设计思路；
- 有机器人相关嵌入式设计或者实时系统经验优先。
- 负责 yocto & android 的构建 (Build) 与升级 (OS Upgrade)，确保系统的稳定性和兼容性。
- 参与新硬件平台的 Bringup，包括 CPU、GPU、Memory 等核心组件的初始化和调试。
- 分析和解决系统稳定性问题，优化系统架构设计，提升整体性能和可靠性。
- 研究客户系统功能需求，定制和优化系统功能，满足客户特定场景的需求。
- 解决系统性能问题，包括 CPU、GPU、Memory 等资源的优化与调度。
- 主导基线升级 (Baseline Upgrade)，确保系统与最新技术和安全标准同步。
- 具备操作系统 (如 Android、Linux) 的构建与升级经验，熟悉系统启动流程和内核开发。
- 有硬件平台 Bringup 经验，熟悉 CPU、GPU、Memory 等核心组件的初始化和调试。
- 具备系统架构设计能力，能够优化系统性能并解决稳定性问题。
- 有基线升级经验，熟悉版本管理和代码合并流程。
- 具备客户需求分析和功能定制能力，能够根据客户需求优化系统功能。
- 熟悉性能优化工具 (如 perf, ftrace, gprof 等)，能够解决 CPU、GPU、Memory 相关的性能问题。
- 有嵌入式系统开发经验，熟悉低功耗设计和优化。
- 熟悉虚拟化技术 (如 KVM, QEMU) 和容器化技术 (如 Docker, Kubernetes)。
- 有 MTK 芯片平台开发经验

## 附录 C 嵌入式招聘常见笔试问题

## 附录 D 嵌入式招聘常见面试问题