1. 终端行编辑设置和粘贴缓冲的典型问题：复制较长的命令粘贴到终端他就会把复制的命令直接右移，而且方向键的左右越过时就会乱跳行

问题根源：行长度限制：终端认为当前行已满，长命令粘贴时自动右移，编辑模式冲突：默认的行编辑器（如 line discipline）对特殊字符（如方向键转义序列）处理异常，串口延迟：串口通信的缓冲延迟导致光标定位错误

解决：在长命令的适当位置手动添加 \ 然后换行，换用更稳定的串口终端工具，直接建立SSH连接

最后解决：stty size 确定终端宽度，stty cols 120 rows 80 变宽终端

1. 串口连接，每次把usb-ttl拔了之后原来那个串口就连不上去了

正常操作流程的一部分。在嵌入式开发中，频繁拔插调试串口是常态

1. 内存（RAM）上放的数据有哪些，文件都是放在硬盘上吗，linux系统情况一不一样呢

3.1内存中的数据包括：

程序代码：操作系统和应用程序的可执行代码通常会加载到内存中，以便快速执行。

栈（Stack）：用于存储函数调用、局部变量、返回地址等。栈的空间较小，且有固定的分配方式。

堆（Heap）：用于动态分配内存，在程序运行时分配大小不固定的内存区域，如通过 malloc 或 new 动态分配的内存。

操作系统数据结构：操作系统使用内存来存储进程控制块（PCB）、文件系统缓存、网络缓冲区等。

缓冲区/缓存：操作系统通常会将磁盘文件、数据库等的数据缓存到内存中，提高访问速度。

3.2硬盘上的数据包括：

操作系统文件：操作系统的核心文件、配置文件、库文件等存储在硬盘中。

用户文件：文档、图片、视频、音频等用户数据都存放在硬盘上。

程序和应用文件：安装的应用程序、软件工具等会存放在硬盘上。

3.3文件系统用于管理硬盘上的数据。常见的文件系统有：

NTFS（Windows系统常用）

FAT32（较旧的文件系统，兼容性好）

ext4（Linux系统常用）

APFS（Apple设备常用）

3.4在 Linux 系统中，内存和硬盘的存储机制与其他操作系统基本相同，但 Linux 对内存和文件系统的管理有一些独特的特点：

虚拟内存：Linux 使用虚拟内存管理技术，将硬盘上的一部分空间（交换空间，swap）当作虚拟内存。当物理内存不足时，操作系统会将部分内存数据交换到硬盘的交换分区，从而避免内存溢出。虽然硬盘的速度较慢，但通过交换空间可以提高系统的稳定性。

文件系统缓存：Linux 会使用内存缓存文件系统的数据。当你读取文件时，Linux 会将文件内容缓存到内存中，下一次读取相同文件时，直接从内存中获取数据，减少硬盘的读取操作，提高访问速度。

I/O调度：Linux内核有一套I/O调度器（如CFQ、Deadline、Noop等），通过优化硬盘的读写操作，提高磁盘的性能，尤其是在多任务并发读写时。

进程与内存管理：Linux采用了精细的进程和内存管理机制，每个进程都有独立的地址空间（虚拟内存），并通过内存页表管理虚拟内存与物理内存的映射。硬盘上的交换空间和虚拟内存可以动态扩展内存空间，从而优化内存使用。

挂载文件系统：在 Linux 中，所有的硬盘、磁盘分区或网络文件系统（NFS）都会通过 mount 命令挂载到统一的文件系统树（/）。这样，不同的存储设备都可以被当作文件系统中的目录来访问。

1. 跨环境传输c文件时：

把windows的文件传输到linux时应该检查换行符和回车符的格式

V im编辑器进入.c文件后：Set ff?检查格式 dos是windows，若是dos，可以set ff=unix解决

注意中英文的分号。

5.编码问题，隐藏字符

检查文件编码和换行符file main.c Makefile

显示所有特殊字符cat -A main.ccat -A Makefile

检查文件权限ls -la main.c Makefile

查看文件hex，找异常字符

hexdump -C main.c | head -20

检查git状态，看文件是否被修改git statusgit diff

6.查看当前网络连接情况：nmcli dev status

有线连接：确认下网先和端口的连接:ethtool end0 有：Link detected: yes 就没问题

如果上面不行，：sudo /sbin/ethtool end0 用root查看

若有连接，就是线通了，但是DHCP 没给 IP，需要手动触发或配置静态 IP。

检查DHCP：尝试重新申请一个地址

sudo dhclient -v -r end0 # 释放旧的

sudo dhclient -v end0 # 重新申请

输出里有bound to 192.168.x.x 就可以

看路由表：ip route 如果只有 wlan0 的 default，而没有 end0，说明还是没成功

如果有，检查网关配置。

我的情况：开发板有线连接台式，台式开启ICS（网络共享），有一次是设置静态ip，开发板仍然是 192.168.104.x

因为手动设的静态 IP 在顶替，ICS DHCP 根本没发挥作用。正确做法是 清除手动 IP → 开 DHCP，让台式机给开发板分配地址。

一旦看到开发板的有线口变成 192.168.137.x 或 10.42.x.x，再测试网关，就能真正通过台式机的 Wi-Fi 热点上网了。

若需设置优先级，可以将有线的end0的metric调低，无线的wlan0调高，低的优先。

7.vim编辑器出现方向错乱，更改shell提示符颜色

通过编辑文件bashrc和文件vimrc可以分别对shell提示符的格式，颜色等和vim的常用功能如折叠，插件等进行更改

在vim里，一定要看清当前的编辑模式之后再操作，尤其是复制粘贴的时候。  
8.按方向键却在输入数字/逗号

原因是终端被切进了“应用光标/小键盘模式，方向键被当成数字键（←=4，→=6），VS Code 的集成终端里，跑完某些程序/任务后偶发不复位，就会一直这样。通过更改linux里的bashrc文件使每次回到提示符都自动复位，设置vscode避免“按任意键关闭”残留状态。根因其实在 Readline（bash 的行编辑器）：它每次准备读一行时会执行 smkx（应用小键盘），把终端切到特殊模式。

1. 按back直接显示^H

这和8一起讲，这两个问题由于我的疏忽导致了我在各种更改bashrc文件的时候突然发现p键已经不能正常显示了输出在终端的是%s，而b键，单引号键也不能正常使用，即使是我重装了bashrc文件也没有得到解决，我逐一排查了情况，不是键盘输入问题，不是驱动问题，因为在另一个系统里键盘输入正常，大概率就是我更改bashrc文件导致。

放弃在虚拟机里直接跑VSCode！

性能差：虚拟机图形界面+VSCode双重开销

终端问题：你现在遇到的就是典型例子

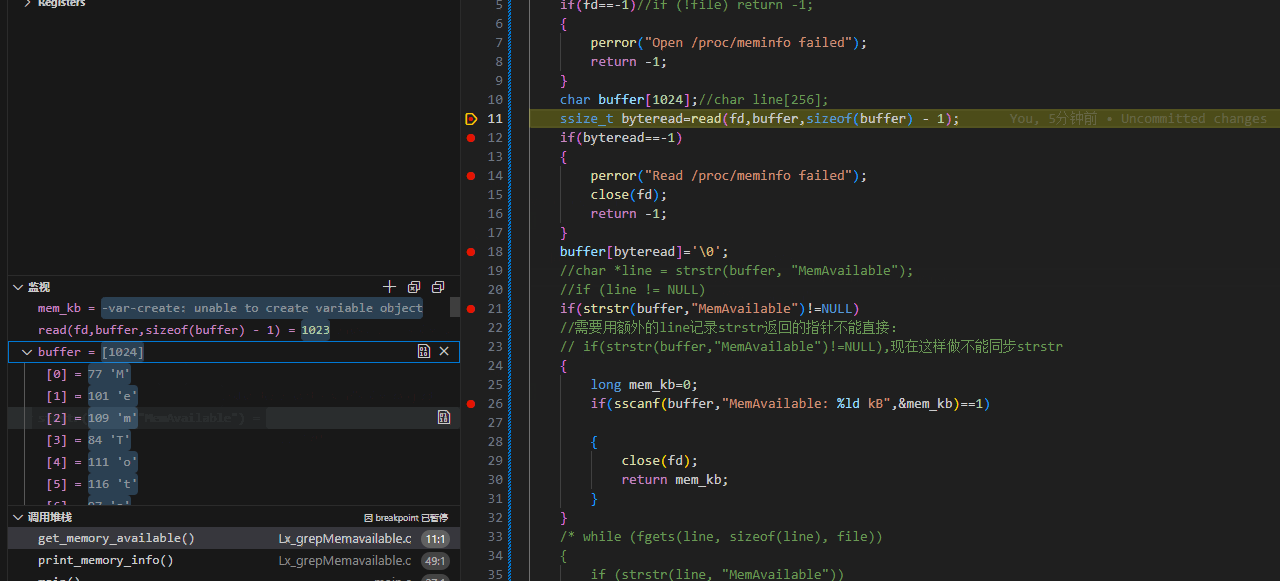
Remote-SSH体验更好：Windows端响应更快，终端稳定

1. 10.编译配置混乱

tasks.json中命令拼写错误（-Mextra应为-Wextra）使用正确的编译选项

文件依赖缺失（main.c调用但未编译对应的.c文件）确保所有被调用的函数都有对应的.c文件在编译列表中

1. 在第11行，buffer成功读取到了stat文件从第一个字符开始的内容，但是我调试到下一行的时候我发现buffer的首元素就变了，变成了stat文件里后面的内容



缓冲区污染：文件描述符重用冲突 - 同一个fd被多个函数使用，缓冲区未正确初始化，字符串终止符问题，其他函数修改了缓冲区

1.检查全局变量和缓冲区冲突 2.缓冲区初始化3。检查其他函数是否修改了这个缓冲区，确保没有指针越界访问