25/9/30 单次系统状态快照

完成cat /proc/meminfo | grep MemAvailable和cat /proc/loadavg的.c文件编译。说实话，这两个命令就弄了6个多小时，而且涉及的知识点不多，还是太菜了，最主要还是这差不多是我第一次以项目驱动的方式来学习，由输出的最终结果决定我怎么学，但是这样基础不牢，做出来倒是有成就，虽然只是前进的0.00001步，这看视频来得实在但是不知道效率相比的话怎么样，才刚开始，基础才是胜利。

升级这个为系统状态监控器，搞了一晚上终于改好了，虽然代码是ai写的，但是每一步我都理解了，代码的结构，用到的函数什么的都能清楚作用及用法，这么看来这种学习方法还不错。

25/10/1 15：44，终于打通了，现在开发板能成功在Mobaxterm上显示了，问题有：忘了把usb-ttl插到电脑上，引脚接错了两次，一次插到了13pin那边发现没有tx，一次插到了26pin的那里。然后是有显示但是是乱码，原因是波特率低，就从115200调到了1500000，最后成功。（后面重新烧录开机两次，但波特率是115200就可以了，不清楚为什么）

下面学：在宿主机和目标板之间传输文件，慢了，过了差不多两个小时。

服了，就为了把这开发板接个网就弄了4个小时，结果是因为手机热点只给 WiFi 接入的设备分配 IP，不会转发到你台式机的网口。然后在 Windows 上开启网络共享就可以了，麻了，浪费了这么久。

19；40 已经把监控器复制到开发板上并运行成功了。唉不容易啊

复盘：今天昨天都是按照ai的思路一步步走，我觉得有问题最后先自己思考再问，而且引导着走肯定能走下去，但是基础很不扎实，先自己弄弄吧。

25/10/2 自己做了个cpu的监控器，很多很多的问题，c的代码就写了近7小时，好在最后写出来了，备份很重要，而且我的基础太薄弱了，直接拿来用还是比较欠缺的。

25/10/3 今天稀里糊涂的就结束了，还是缺乏警觉和专注力，主要还是不知道怎么学，确实像ai说的那样即使愿意花几个小时排查bug也不愿意自己好好深入了解一下c语言和计算机系统。每天都得做改变。

25/10/4 早上解决按方向键却在输入数字/逗号，弄了有点久，然后是该环境，用本地IDE连接虚拟机，了输入back却直接显示^H，弄了半天是解决了，但是p键输入变成了%s,服了，还是直接用原始的吧。

我靠b键打不了了，我靠单引号也打不了了，用的deepseek的bashrc极简版，麻了。  
asudo apt update

asudo apt install --reasinstall ash

重装bashrc恢复，还是没恢复。

14：00，已经崩死了，新建用户剩下的不管了。重新

现在在更新系统，网老是断，弄了几次了，

16:02 终于更新完了，今天又浪费了很久

晚上，排查cpu usage逻辑问题，学习gbd

这两天光是调教环境配置了，而json文件的配置我还不会，虽然能看懂。

2025/10/5 今天主要学了系统调用函数和片面的一些驱动知识，还在用系统调用重构一遍代码，现在已经完成两个，但是也该睡觉了。中午开始的时候开始刷了leetcode三道简单题，第一题直接不会，看了之后还是觉得很简单的，之后得多刷题啊。学基础知识有点学不进去啊，趁早改吧，早上起来先好好看看。

2025/10/6 早上先看一小时c primier plus ，然后是把系统调用版本做完，到中午现在1点钟，刷leetcode，打算是5-10分钟没思路或者没做不完就看答案，多刷题。学数据结构，顺序表，单链表，动态分配

写完读取cpu温度功能，突然usb-ttl没起作用，把杜邦线拔掉又装上就好了

写代码就用了10多20分钟吧，很简单现在，哈哈哈，后面是先是把测试环境换到开发板上，然后再看下怎么远程传输文件。

靠，突然被git搞得一惊一跳的，还是多备份多提交啊，幸好是commit了再pull，都拉了什么东西进来，先清理下电脑和git仓库。

2025/10/7 早上看操作系统，发现没学真的很多，还有得早睡早起。把错误信息提示改进之后刷题，链表真的理解太浅了，刷题都不太看得懂，还都是简单题，有点困又不想学，先看视频吧。

我好菜啊

菜就多练！！！

重新慢慢的看视频学链表，从单链表开始，还是必须得自己一步步的敲出来啊，清晰多了。

注意力不够持续集中，效率低。 链表细节很多，多学多理解，尤其是链表创建。单链表学了3小时不止，主要是理解，这个是基础中的基础了吧，虽然以前学过，但是都三年了差不多忘完了。

25/10/8 第一次早起走走。精神多了，主要还是吃早餐吧，希望能养成习惯，后面冷了不知道什么样。先看书，看了边链表和assert库，把文件标准IO重新看看。接下来把链表视频看完。考虑下无头结点的情况。先做题吧，打好基础。相当于是学完了一遍链表，但是对于怎么应用几乎毫无头绪，初步想法是把多种监控功能做成链表，但是接下来呢？诶，突然有点累了。

看着ai发的重构项目架构，真的很懵，虽然能够初步地简单了解，但是这样对我这样的初学者还是太难受了，而且我不知道这样投入合不合适，我现在的当务之急是找工作而不是掌握好每个知识点，这对我现在来说是不可能实现的，虽然链表确实很重要。

开始做日志功能，还是很懵，休息下重新开始吧，要有自己的理解。

25/10/9 今天主要是对过往知识的复习吧，但是只复习了一部分和把leetcode上的题目看了，感觉其实我学的东西也还算多的，但是掌握度不够，而且不会串不起来，还是得加油啊。今天pu活动，上台分享还是很紧张，背个两三百字的内容还是会很卡，面试不能这样。

25/10/10， 今天晚起了20分钟，8点50差不多才起床，早上出去走走拿了c与指针，然后先看了函数指针，初步弄懂了，还解决了int \*(\*(\*f)())()是什么，从内到外其实很好理解，最终是一个函数指针，因为是int \*(X)()类型，如果是int X()那就是指针函数，下面是对套娃的理解：  
我的理解是(\*f)假设是指针1，(\*f)（）则是指针1指向了一个函数，假设这个函数为函数1那就是说指针1指向了函数1，而\*(\*f)()其实就是说函数1返回的是一个指针，假设他是指针2，(\*(\*f)())()这个其实就是指针2指向了新的函数，假设他指向函数2，int \*(\*(\*f)())();函数2返回的是一个整型指针，完成。

继续：  
指针数组其实和这个是差不多的，类比就很好理解了 下面是指针，函数与数组的三者结合了 int (\*f[])();由内而外，首先f是数组，数组元素是指针，再结合（），数组f的元素应该是函数指针，把f[]拿掉就看的很清楚了 然后是int \*(\*f[])()，在刚刚的基础上，数组f的元素应该是函数指针，再来看这个其实就有点怪了，数组f的元素是函数指针，指针难道还要指向int \*吗？那就是说这样是错误的 书上说这是声明了一个指针数组，指针指向的类型是返回值为整型指针的函数，我的理解就不够了 等一下，如果用刚刚的方法的话，先从内而外，\*f[]是数组，元素都是指针，替换之后是int \*(X)()，有点明确了，这就是指针函数，返回值是整型指针的函数，（X）是指针数组，最后结合就是书上说的指针数组，指针指向的类型是返回值为整型指针的函数，应该是这样，所以一切都先看变量是先和什么结合的，一开始它就差不多定性了

开始重新写日志吧，一定要多敲啊，自己思考的慢慢敲。

重构结构，把采集文件里的打印函数都去掉，到主函数统一处理，而且规整多了