

# PA0 实验报告

21124001 潘昕田

2022 年 9 月 11 日

## 1 实验进度

1. Ubuntu 22.04 LTS 安装完成 (真机)
2. 相关环境配置安装完成
3. nemu 可以通过 make 成功构建项目

PA0 相关任务全部完成

## 2 必答题

### 2.1 “对好的提问”以及“通过 STFW 和 RTFM 独立解决问题”之观点

在深入学习计算机的相关知识的过程中，总会遇到各式各样的问题，例如如何设计解决一个问题的算法、如何配置系统的环境或者如何利用一个语言的特性等等。而我们该如何解决这些问题，直觉上最简单的办法就是直接拿着问题去问“大佬”，把问题甩给那些会的人，自己坐等答案。但这样的话，你能从中学到任何有价值的知识吗？显然没有，也许你会争辩“不会啊，拿到答案我也会思考的”，但实际上当你拿的别人已有的答案时，你总是会下意识地把自己的思维和答案的思维靠近，实际上并没有进行什么有意义的思考，往往只会留给你一种“我也就差一点就想到这种做法”的假象。而且，这样的提问只会让大家产生厌恶，因为这样的问题往往对于他人而言都是司空见惯的问题，毫无价值，频繁地询问只会让大家对你的不劳而获产生反感，最终疏远并不再搭理你。就算每次的提问都能得到一个满意的解释，你只会产生一种自己什么都会的假象，到头来到了实际运用的环节，你照样是什么

都不会. 因此这种不加任何自我思考与探索的提问, 可以说, 是所以提问方式中最差的一种.

那么经过自己一番思考, 但是还是无法解决的问题, 是不是就应该马上去询问他人? 事实上 “RTFM” 和 “STFW” 就是对这个问题的回答. 很多问题并不是只有你一个人遇到过, 而是已经在很早就被人所解决. 而当今一个互联网十分发达之时代, 事实上, 大部分这样的问题都曾在各种论坛上被人所提出并得到解决. 还有一部分问题, 例如关于指令集相关的问题以及编程语言中一些库函数的使用方法, 你不明白的地方其实早已被开发者写入文档中. 这样的问题其实只要稍微在网络上检索一下或者去相应的官方文档下查找信息即可. 频繁询问这样重复出现的问题同样也会让被询问者感到厌烦, 没有人愿意把时间和精力消耗在这样重复的问题. 毕竟, 重复回答多次早已被解决的问题 (例如 “Ubuntu 22.04 上如何设置环境变量”, 事实上在网上 (Google) 搜索 “Ubuntu environment variable” 就能找到解决方案了). 与其重复回复这些毫无新意的问题, 不如把时间消耗在更有价值的问题之上. 因此在你已经明确无法通过自己的思考来解决问题的时候, 先不要急着去找人询问, 不妨先去 “RTFM” 和 “STFW”, 以避免过于依赖他人, 同时也可以提高自己的信息检索能力.

当然, 并不是所有的问题都能通过 “RTFM” 和 “STFW” 能够解决的, 当你明确这个问题不论如何自我思考以及 “RTFM” 或者 “STFW” 能够解决时. 确实, 留给你的可能就只有询问他人的方法, 但这也并不意味着提出的问题就可以被别人所热心回答, 你总不能直接抛出一个描述含糊, 可能有多种不同的解答 (在不同人的理解下) 然后期待别人给你解答. 面对每天都可能出现的各种问题, 这类表述含糊的问题自然不会乐意被人们解答. 换作你本人也不可能乐意去解答一个要消耗很长时间理解问题含义的问题, 而且还可能会怀疑提问者是在真的有问题求教还是故意刁难你. 自然, 你也不会愿意去解决这样的问题. 因此, 把问题描述清楚, 不仅提高别人解答问题的概率, 同时也向别人表达了你提问的诚意. 当然, 诚意的体现不仅仅是通过描述清楚问题来表达的, 最好还要在问题之后附上自己对问题的思考这样就表明你已经充分思考过这个问题了, 但你确实没有能力独立解决问题. 这样的话, 人们也更乐意去解答你的问题.

因此, “好的提问” 其实也是自我思考与探索的产物, 只不过你自己无法独立给出问题的解答, 需要他人的帮助. 但这并不等同于 “无脑” 向他人寻求答案, 否则, 你得到的只会是他人的忽视或者像 “RTFM” 或者 “STFW” 这样的回答, 同时还无法提高自身的知识水平.

## 3 实验心得

### 3.1 使用 make 指令构建 nemu 时遇到的问题

一开始使用 `make menuconfig` 构建项目指令执行失败并返回了缺失某些文件配置的报告,虽然按照 **PA0** 实验讲义提示第一次执行 `make menuconfig` 时确实会有错误产生并要求自行解决,但一开始不知道为什么我没能理解错误提示(之前配置其它环境时也有类似关于文件配置缺失的错误提示,但基本上都给出的提示解决了).一顿 STFW 后也只解决了一个文件配置的问题,后来甚至一度以为是自己下载 nemu 时出了问题(第一次执行 `bash init.sh nemu` 因为没有配置 git 的相关信息产生过一次错误).结果开始进行了多次删除 ics2022 再重新下载的操作,自然是不可能解决问题的.后来想到应该是自己没有读懂错误提示导致,然后再次去读错误提示并去网上检索了一下错误提示里面显示我缺失的依赖(例如当时提示的缺少"bison").然后才发现是自己缺失了一些重要的依赖导致的出错.最后按照缺失的依赖提示一步步下载最终成功解决了问题.这着实给了我一个教训,以后这类错误提示不能一眼带过,而是得好好研读(一开始就是因为没有仔细地看错误提示导致没读懂).

### 3.2 设置环境变量的一些思考

提交时按照实验要求要设置 `TOKEN`,`STUID` 和 `STUNAME` 三个环境变量,于是我上网检索了相关信息并找到了 `export` 指令,不过一开始使用时是直接在 terminal 中键入 `export STUID=xxxxxxxxx` 这样的方式实现的,后来发现一旦关闭 terminal 后再使用 `echo $STUID` 就找不到之前设置的 `STUID` 这类的环境变量了.然后继续去读 Stack Oerflow 上的解答,发现需要在 `.bashrc` 文件里设置 `export` 语句才能在每次使用 terminal 时使环境变量生效.后来继续查询相关文档发现 Ubuntu 在启动 bash 时会按照 `.bash_profile`,`.bash_login`,`.profile` 的顺序查找 bash 的相关配置(按优先顺序排列,优先一级的配置文件若检索到便不会执行其它优先级较低的配置文件).我的系统上默认只有 `.profile`,而打开 `.profile` 可以看到执行 `.bashrc` 的相关语句.这也就意味这之所以要在 `.bashrc` 里面设置环境变量才能在所有的 terminal 中生效是因为 `.bashrc` 起到了启动 bash 时的初始化的作用,而在 terminal 中执行 `export` 只是在当前的 terminal 下设置了环境变量,而没有在 bash 启动的初始化阶段去完成环境变量的设置(可以看到 `NEMU_HOME`

以及 `AM_HOME` 之所以每次使用都生效就是因为其设置在 `.bashrc` 中). 而第一次在 terminal 下利用 `vim` 修改 `.bashrc` 要想立即生效需要执行 `source` 命令 (使用 `help source` 发现其解释和 `.` 命令时相同的, 继续查询发现这两个 `bash` 内置命令功能是等价的, 只是 `source` 来源与 `C Shell` 的相关指令, 而 `.` 来自于 `Bourne Shell`). 这方面的探索让我对 `bash` 有了一个更深入的了解.