# Report

21302010042

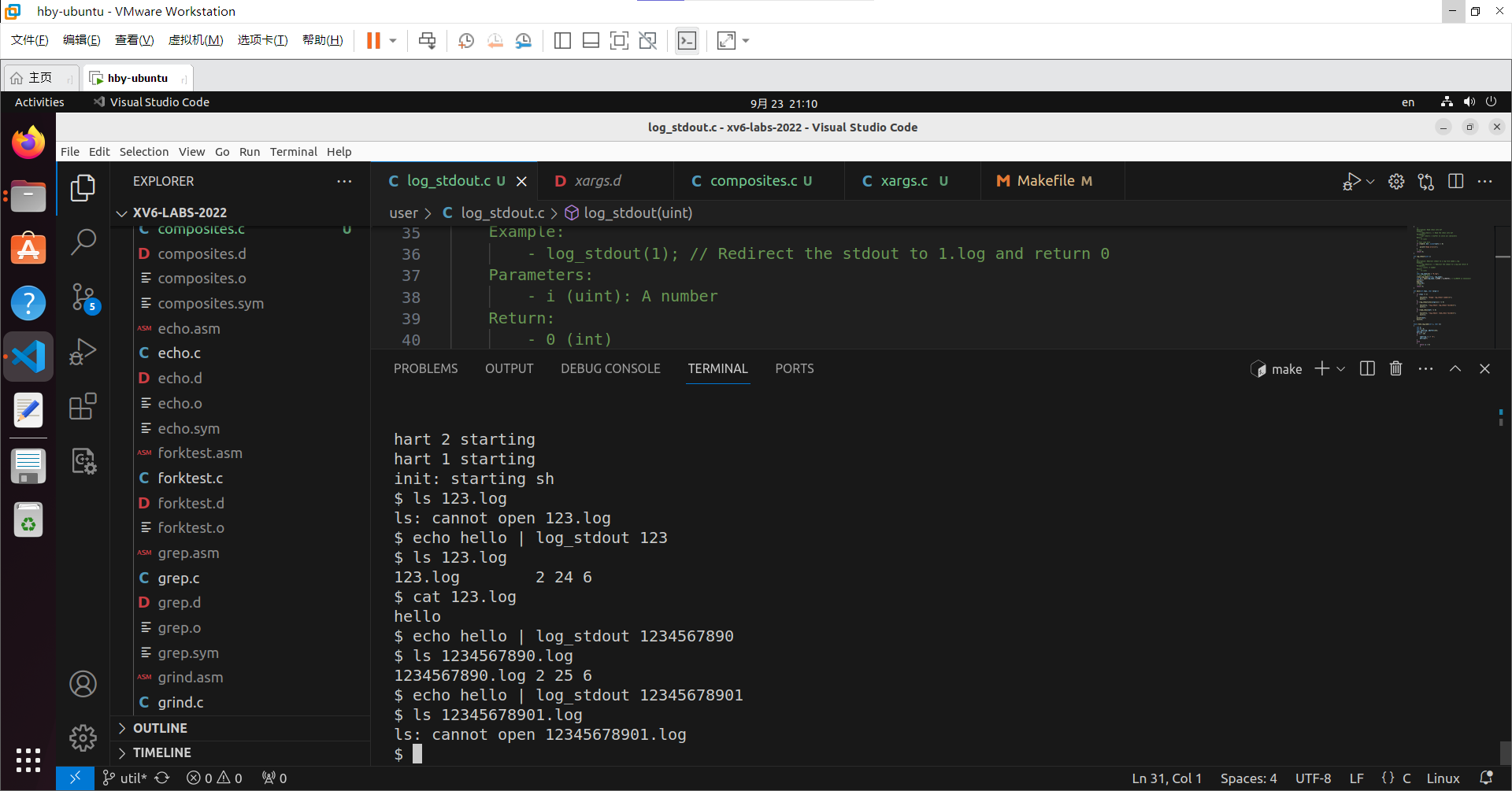
侯斌洋

①实验思路及测试结果

***Log\_stdout.c:***

要实现将标准输出重定向到某个文件，则只需要打开该文件，close（1），然后再dup即可实现。这里利用了dup自动选择最低的文件描述符的机制。本题中比较复杂的是组合log\_name的过程，这里要将uint转为char数组并加上“.log”，我将该过程封装在了函数creat\_log\_name中。

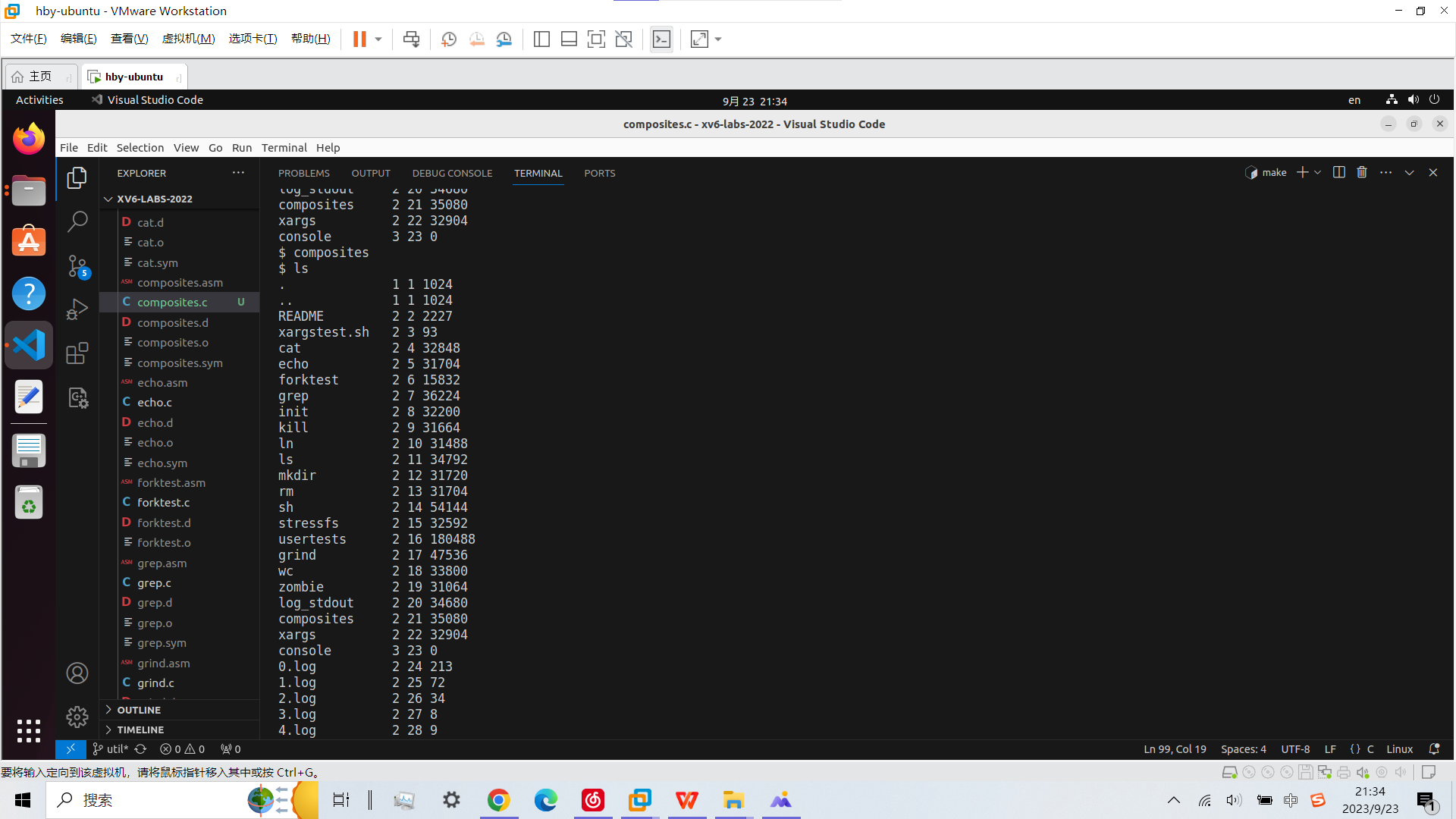
测试结果截图：

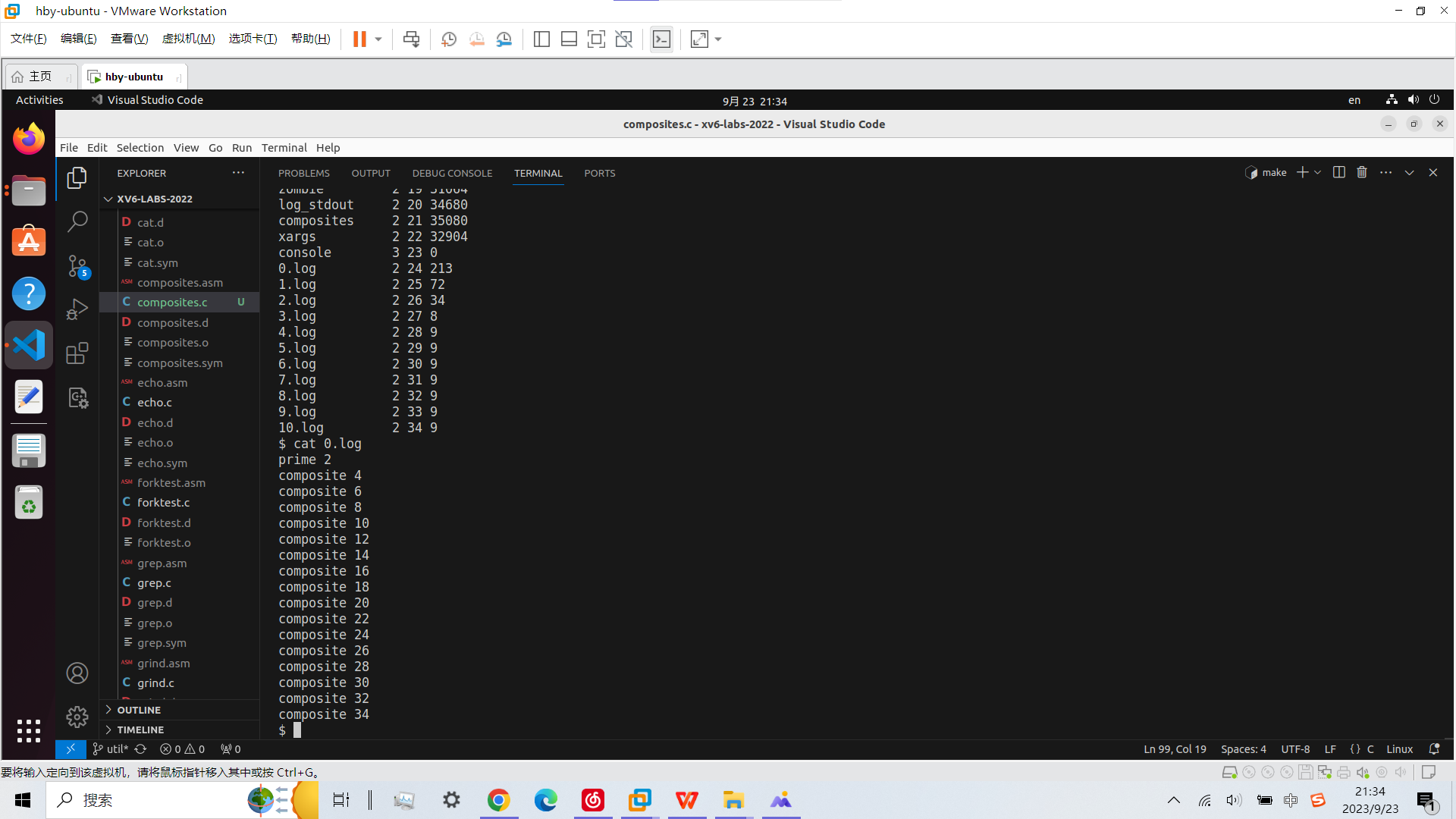


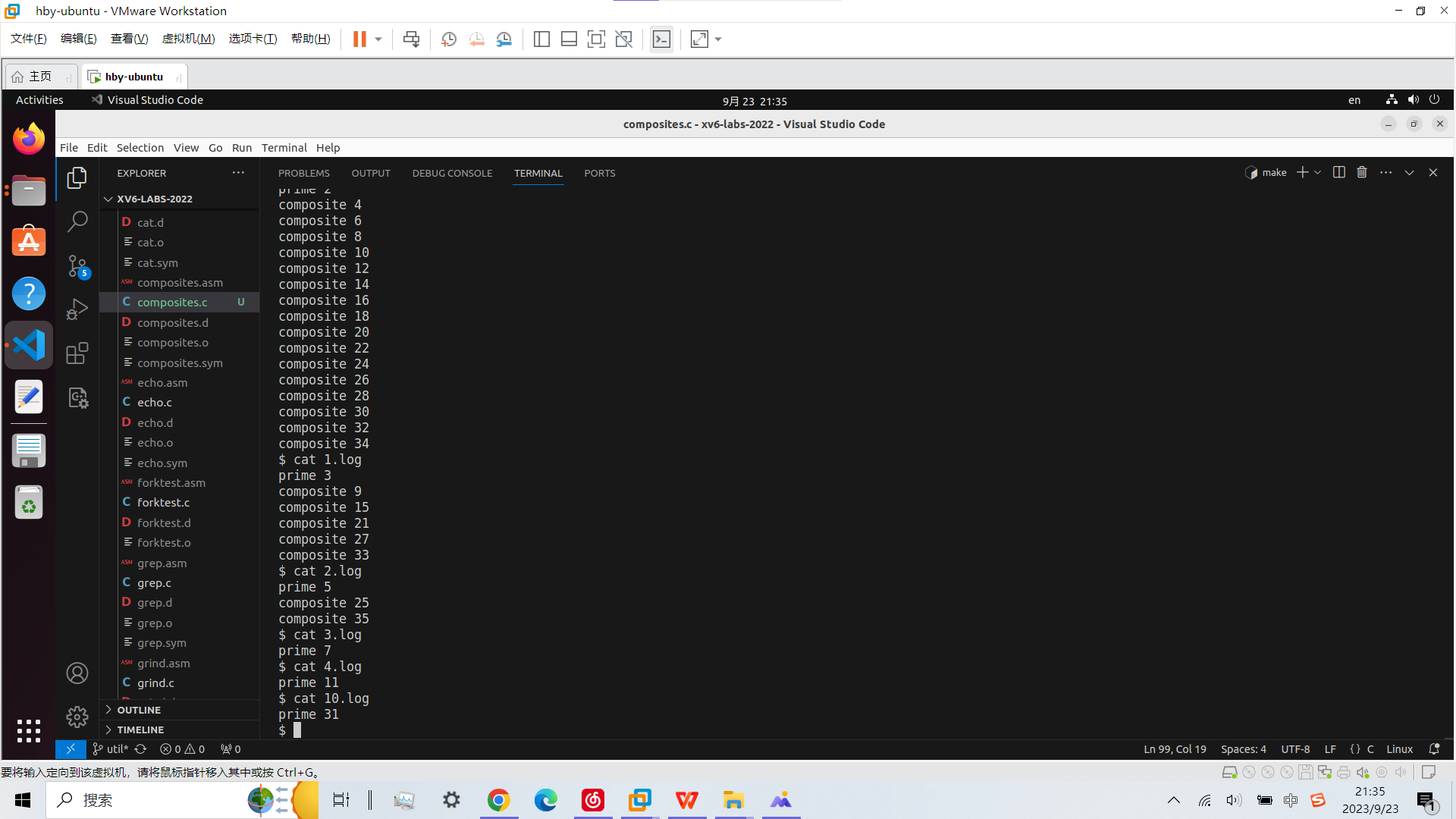
***Composites.c:***

本题中，程序利用pipe的机制实现一个素数过滤器。比如如果要区分2-35的素数，则首先向第一个子进程从管道依次输入2-35，子进程默认取从管道中第一个输入的数为素数，然后依次取管道中剩下的数并检测这些数是否能够整除该素数，如果能整除则说明不为素数，直接drop并打印；如果不能整除则该数通过管道进入下一个子进程继续进行筛选。通过在第一个进程与第二个进程之间建立管道，在第二个进程与第三个进程之间建立管道……在每个父进程与子进程之间建立管道，从而逐层筛选出素数。

测试结果截图：



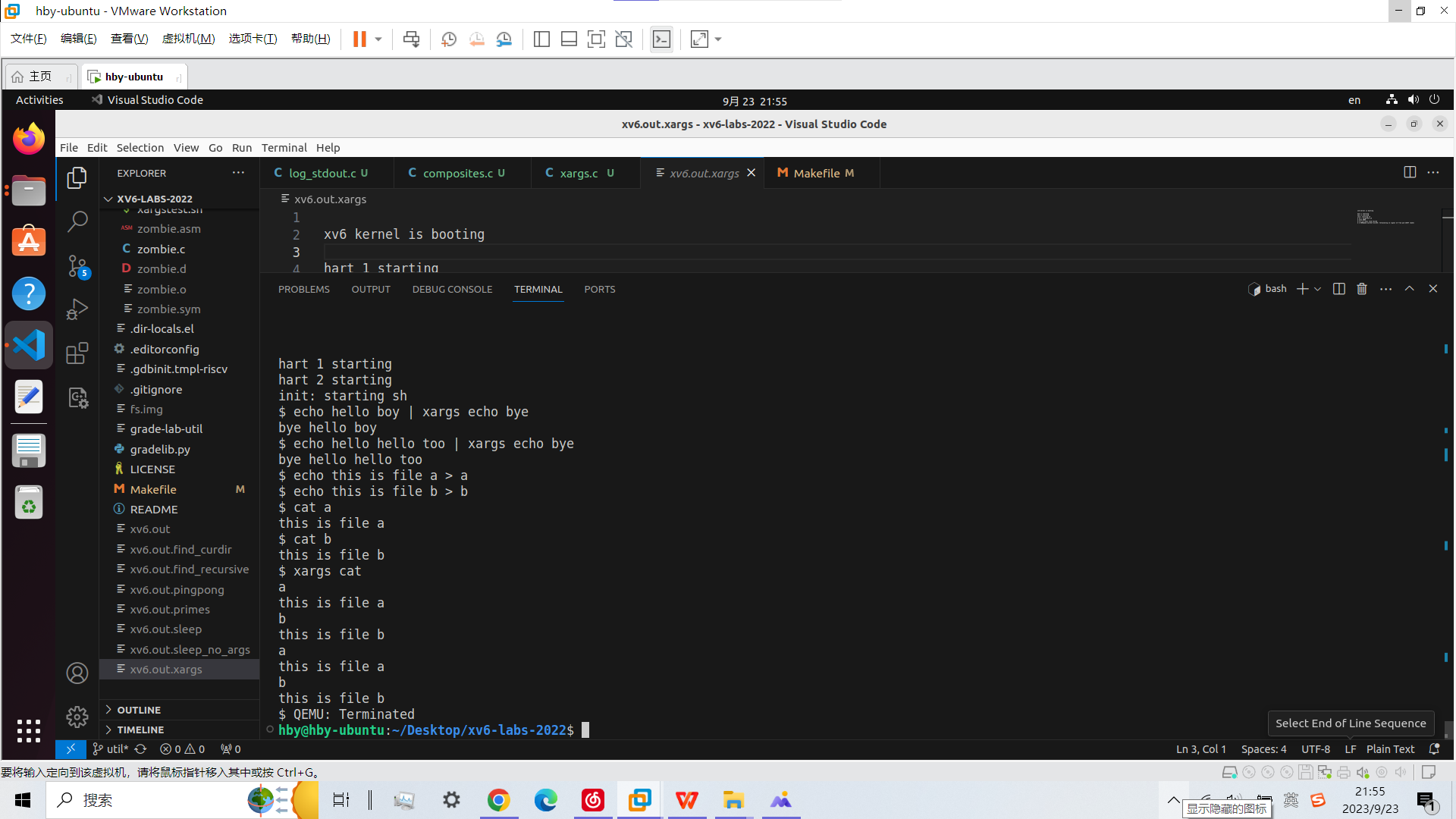




***Xargs.c:***

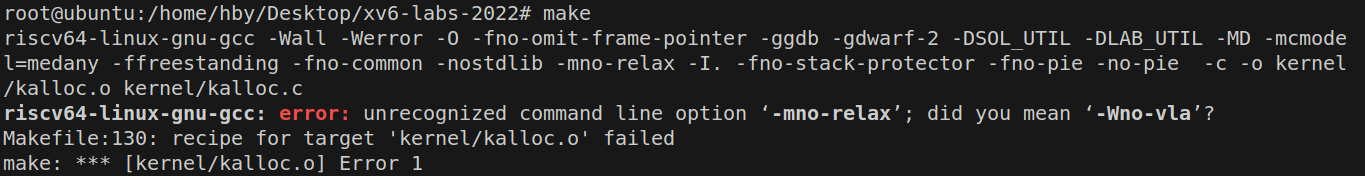
本题主要是实现对一个命令，其参数可以从stdin读入并根据该参数运行该命令。具体来说，如果我要cat 1,txt，通常做法是在shell中直接输入该行并执行。而使用xargs，便可实现指定命令而暂时不指定参数，参数从stdin读入，比如可以先输入xargs cat这样就指定了命令cat，之后如果输入1.txt并换行，则执行cat 1.txt，如果输入2.txt则执行cat 2.txt。在具体的代码实现上，主要是对于命令行的一些处理：拼接参数以形成完整命令，并在子进程中执行该命令。

测试结果截图：



②一些问题：

在第一次make的时候报错：



这是由于riscv64-linux-gnu-gcc编译工具链版本太低，可通过以下操作解决：

$ sudo apt install gcc-8-riscv64-linux-gnu

$ sudo update-alternatives --install /usr/bin/riscv64-linux-gnu-gcc riscv64-linux-gnu-gcc /usr/bin/riscv64-linux-gnu-gcc-8 8

在make qemu时出错，Ubuntu版本过低（18.04），qemu-system-misc中没有qemu-system-riscv64。之后安装ubuntu22.04可以直接编译成功qemu和xv6，但ubuntu22.04需要手动添加VMware文件夹下的linux.iso盘以安装vmware-tools，并且需要sudo apt install open-vm-tools和open-vm-tools-desktop。安装过后可以正常实现复制粘贴，但无法拖拽文件，需要再安装nemo等程序。

<https://blog.csdn.net/qq_41866334/article/details/125626778?utm_source=app&app_version=6.1.5>

（ubuntu中apt依赖严重，软件包关系错综复杂，且很多包不向下兼容，导致配置运行环境上的困难）

Xv6中read的最大字节数不可超过2048，否则会报错。在read（0, buf, 2048）时发现。