**实验报告**

**Lab 0**

姓名：赵笑宇

班级：20信息安全

学号：20307130172

**实验报告填写要求**

1.请在每个exercise之后简要叙述实验原理，详细描述实验过程。可以使用中文进行描述，不对语言做要求。

2.请将你认为的关键步骤附上必要的截图。

3.有需要写代码的实验，必须配有代码、注释以及对代码功能的说明。

4.你还可以列举包括但不局限于以下方面:实验过程中碰到的问题你是如何解决的、实验之后你还留有哪些疑问和感想。

5.如果实验附有练习，请在每个练习之后作答，这是实验报告评分的重要部分。

6.Challenge为加分选作题。每个lab可能有多个challenge,我们会根据完成情况以及难度适当加分，具体情况会在课上说明。这部分的实验过程描述应该比exercise更加详细。（请注意，Lab0为基础环境配置，不设置挑战问题。）

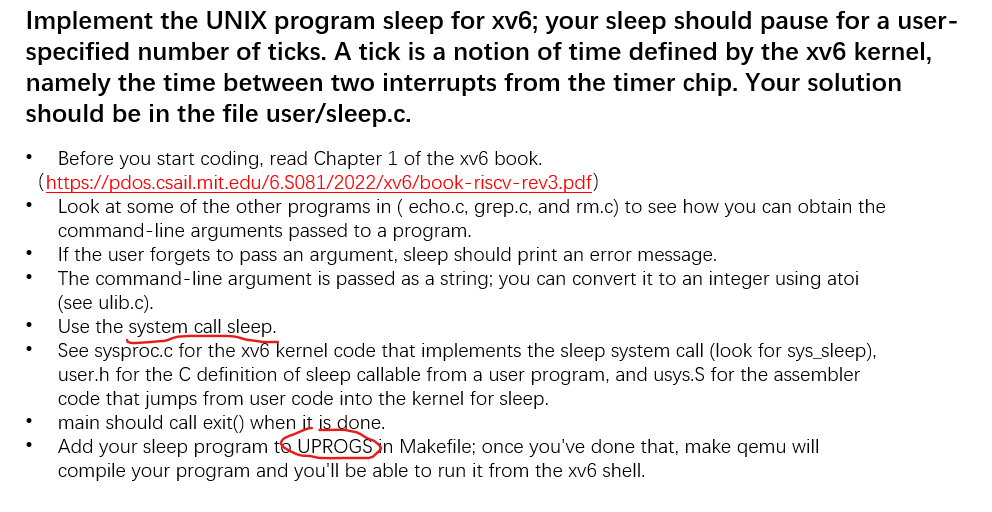
7.切勿抄袭亦或是去互联网复制粘贴答案。

【练习题模板】

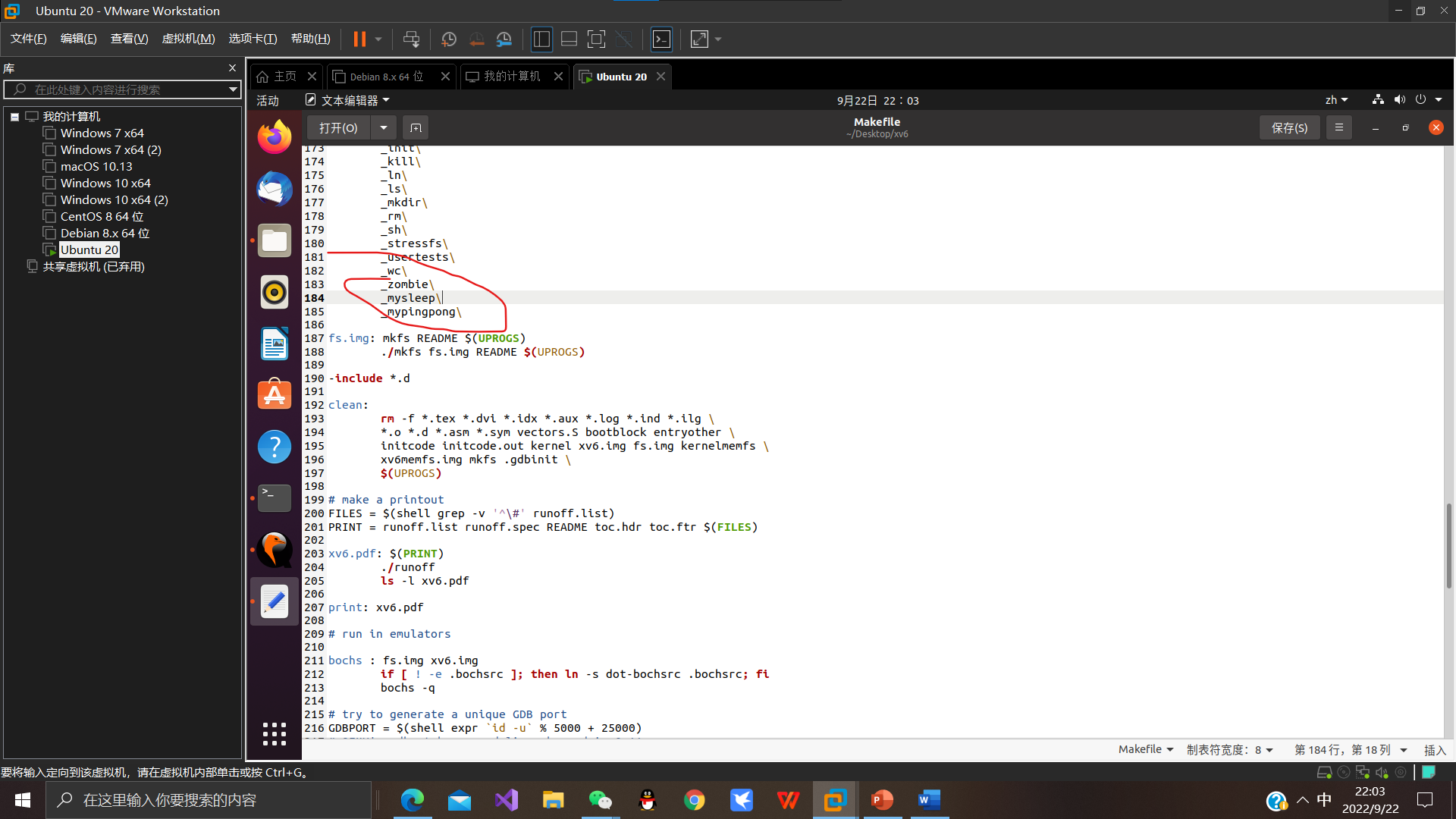
1. Question
2. Screenshot
3. Difficulties and solutions

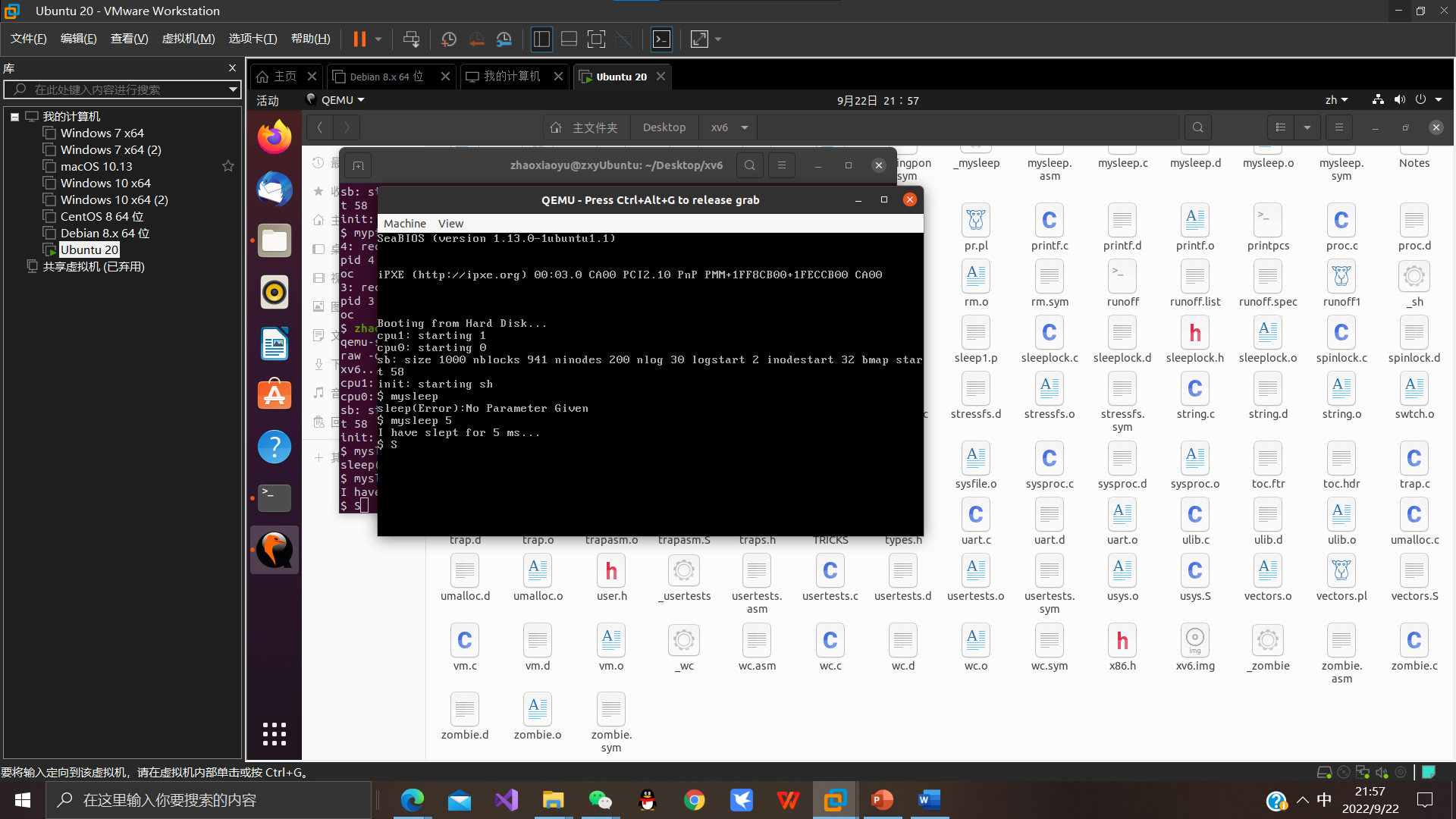
**练习题一：**

1. **Question：**

****

1. **Screenshots:**



****

1. **Difficulties and Solutions**

**<1>C语言有的语法忘掉了以及在xv6中的特定用法**

**比如说printf**

**第一个参数的0/1/2分别代表input/output/error**

**（这个是看xv6技术手册得到的）**

**以及退出时直接调用exit()即可**

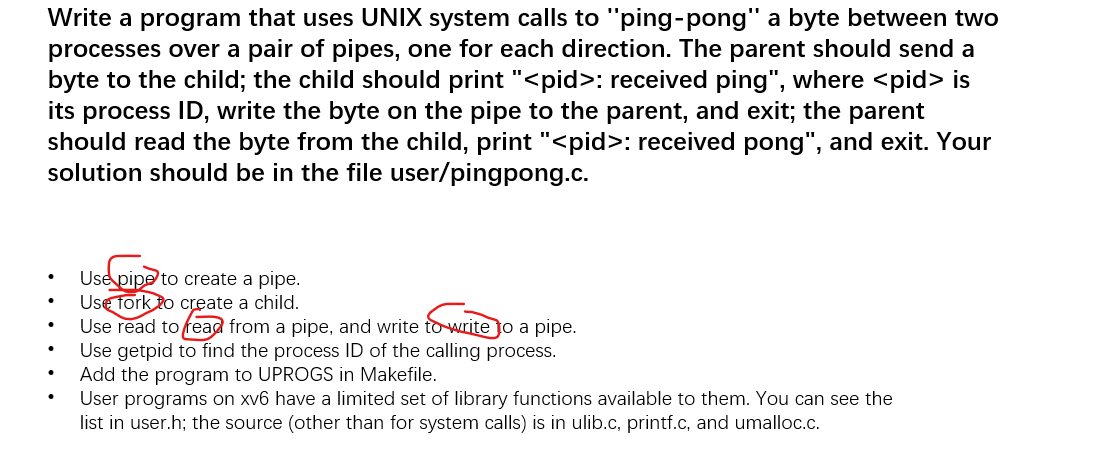
**<2>以及main函数的写法应为**

****

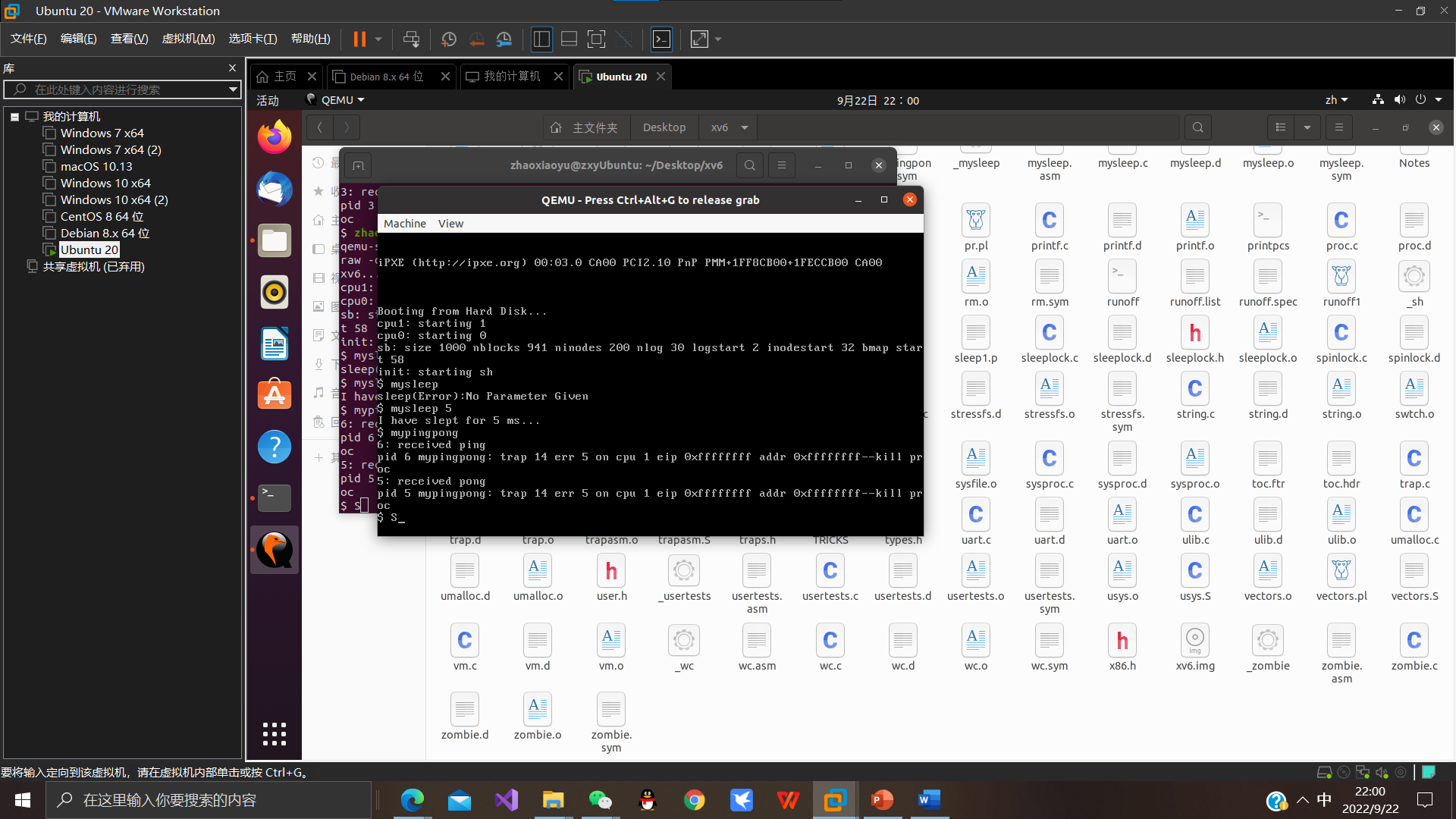
**而非int[] argv (java写习惯了容易写成String[]args)**

**练习题二：**

1. **Question：**

****

1. **Screenshot:**

****

**3. Difficulties and Solutions**

**<1>pipe的用法以及fork的用法**

**pipe[0]用于读 pipe[1]用于写**

**fork复制创建一个子进程从调用位置执行与父进程相同的代码段以及调用fork输出图的画法**

**<2>一定要弄清先后顺序！！！否则运行不了**

**父进程先发一个ping 子进程收到后 回复一个pong之后**

**父进程方可打印出pong**

**关于printf 如果父进程不等子进程结束后就printf**

**然而同时由于printf不是原子操作**

**父子进程的printf交替进行 会打印一团糟**

**所以要在父进程打印之前父进程调用wait()**

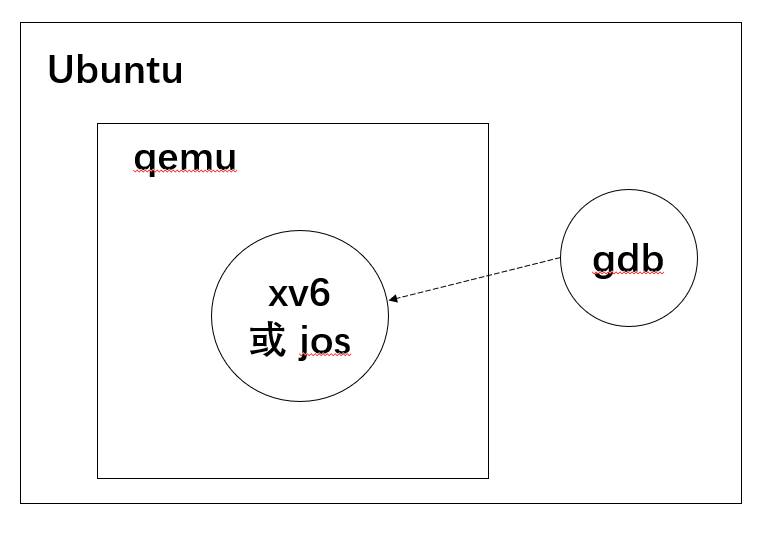
**同时子进程读的时候要关闭他写的那个pipe的读端口**

**以及父进程写的pipe的写端口（父进程类似）**

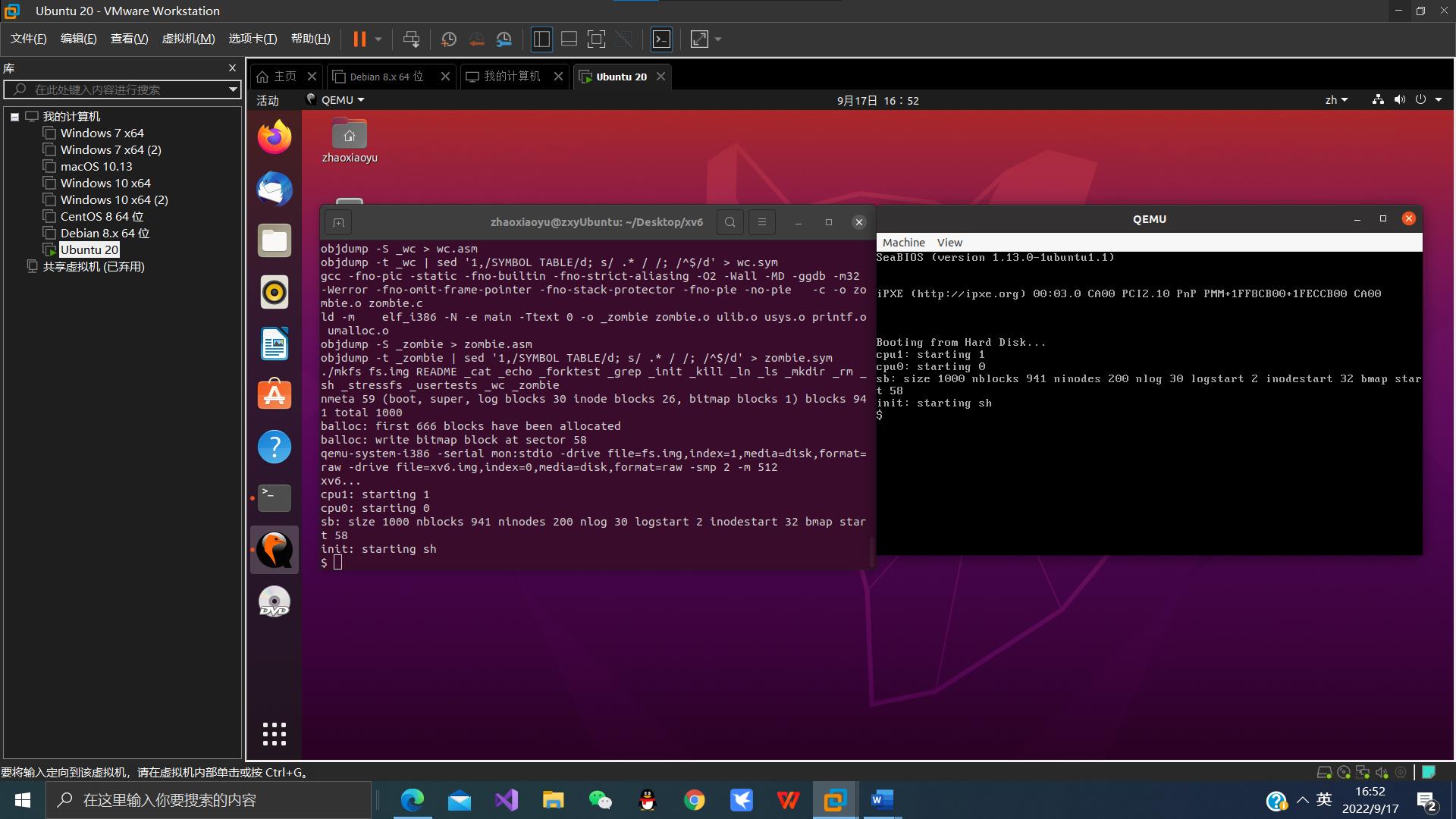
**Part 1：Environment Configuration**

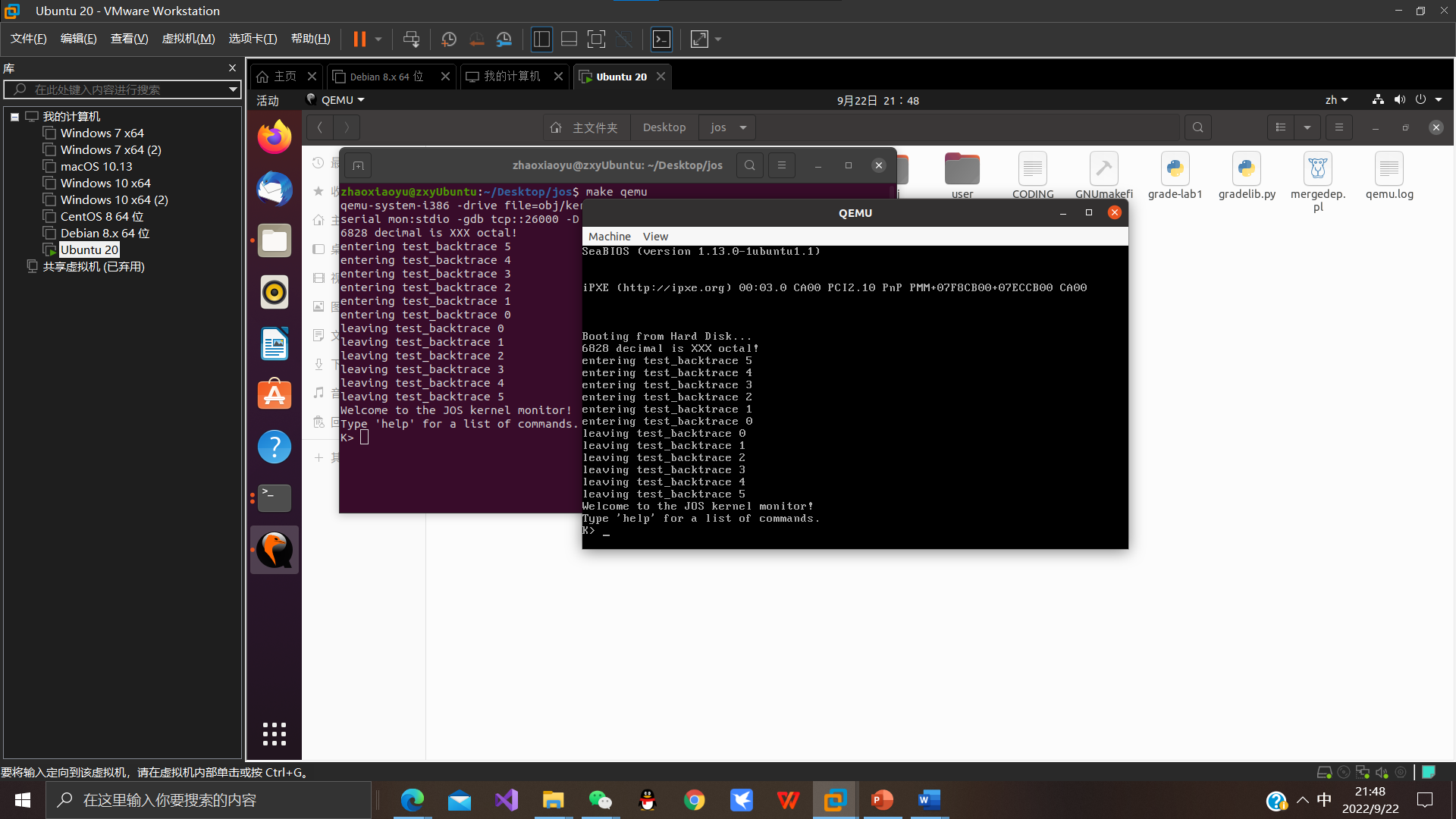
**Exercise 1：**Please install your own environment, attach a screenshot of the process and describe it in detail.

**配置的结构如下**

****

**安装完成20版本并配置成功的截图如下（xv6与jos类似的**）





**Exercise 2 ：**What problems did you encounter, how did you solve them, and if you didn't encounter any problems, what did you learn during the installation process？

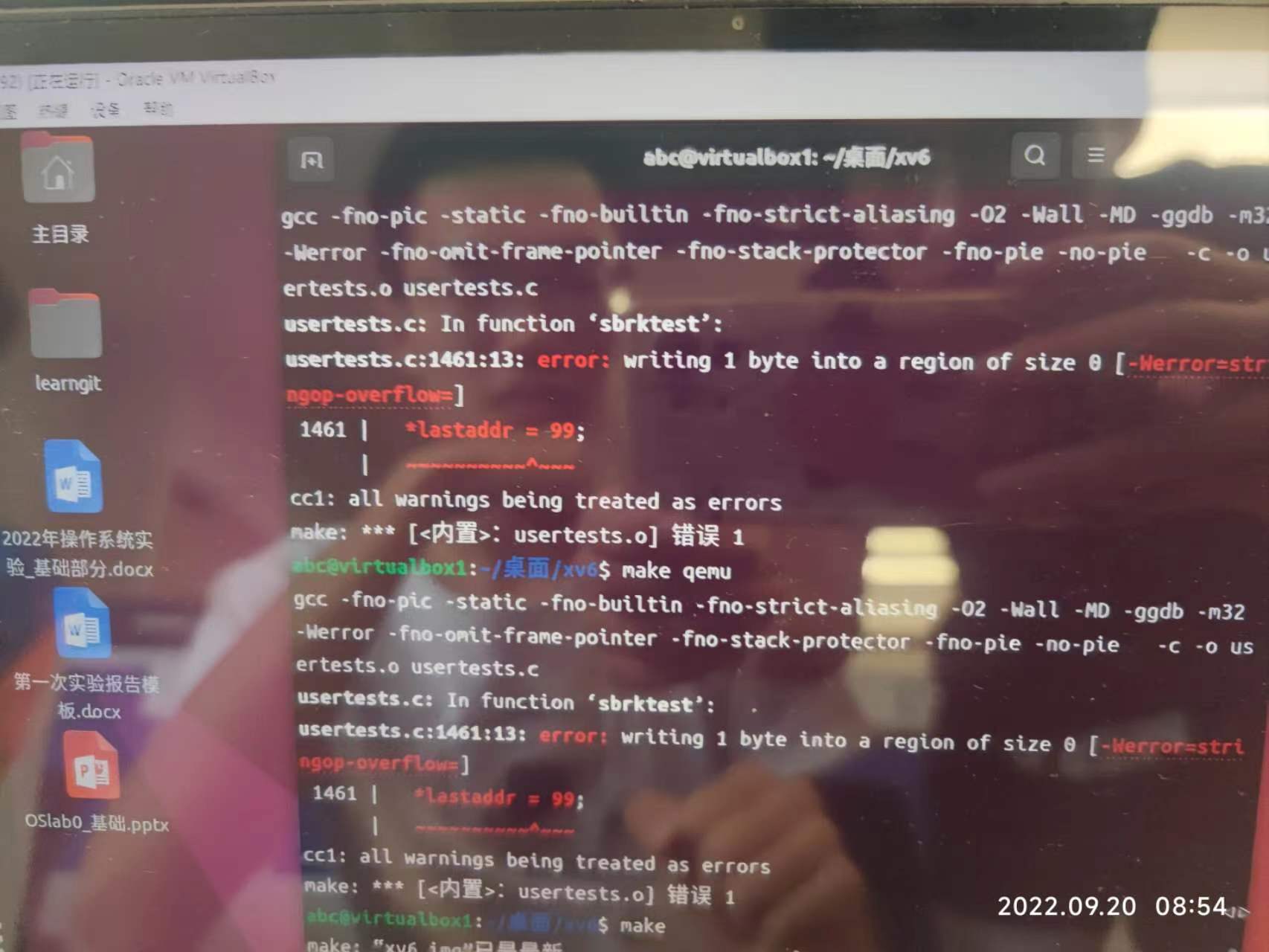
**心路历程：**

**一开始下载了20与22版本的Ubuntu源**

**但安装的时候脑子一抽筋 想着新的或许更好用 倒腾了半天**

**22在make qemu的时候一直报错**

**（忘记截图了就只好用群里相同问题的图啦**

****

**关键助教之前还发过**

**“如果make qemu遇到编译问题，请下载ubuntu20版本。”**

**安装的时候我也看到了 但还是脑子抽**

**心得体会：**

**助教师兄轻易不说话 一字千金 吾当字字斟酌**

**Part 2：Some More Questions about Configuration**

**Exercise 3 ：**What does “operating system virtualization” mean and what important uses does it have？

**<1>操作系统的虚拟性指的是，操作系统会把一个物理上的实体变为若干个逻辑上的对应物，物理实体是实际存在的，而逻辑对应物是用户感受到的.**

**<2>让CPU看起来更多-时分复用技术**

**如，虚拟处理器技术，借助多道程序设计技术为每个程序建立进程，多个程序分时复用处理器；虚拟设备技术，物理设备虚拟为多个逻辑设备，每个程序占用一个逻辑设备，多个程序通过逻辑设备并发访问。**

**让内存看起来更大-空分复用技术**

**如：虚拟磁盘技术，物理磁盘虚拟为多个逻辑磁盘，如C、D、E等逻辑盘，使用起来更加安全、方便；虚拟内存技术，在逻辑上扩大程序的存储容量，使用比实际内存更大的容量，大大提升编程效率。**

**Exercise 4：**What is Qemu? What are its advantages and disadvantages? What are xv6 and jos?

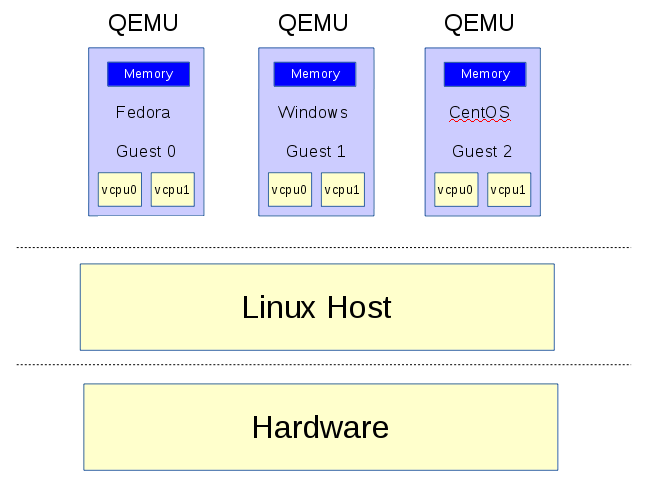
**<1>QEMU浅分析**

**QEMU是一个托管的虚拟机，它通过动态的二进制转换，模拟CPU，并且提供一组设备模型，使它能够运行多种未修改的客户机OS，可以通过与KVM一起使用进而接近本地速度运行虚拟机（接近真实电脑的速度）。**

**QEMU还可以为user-level的进程执行CPU仿真，进而允许了为一种架构编译的程序在另外一种架构上面运行（借由VMM的形式）。**

**QEMU是一款开源的模拟器及虚拟机监管器(Virtual Machine Monitor, VMM)。QEMU主要提供两种功能给用户使用。一是作为用户态模拟器，利用动态代码翻译机制来执行不同于主机架构的代码。二是作为虚拟机监管器，模拟全系统，利用其他VMM(Xen, KVM, etc)来使用硬件提供的虚拟化支持，创建接近于主机性能的虚拟机。**

**QEMU作为系统模拟器时，会模拟出一台能够独立运行操作系统的虚拟机。如下图所示，每个虚拟机对应主机(Host)中的一个QEMU进程，而虚拟机的vCPU对应QEMU进程的一个线程。**



**<2>优缺点分析**

**qemu – 优点**

**\*可以模拟 IA-32 (x86)个人电脑，AMD64个人电脑， MIPS R4000，升阳的 SPARC sun3 与 PowerPC（PReP 及 Power Macintosh）架构**

**\* 支持其他架构，不论在主机或虚拟系统上（请参看QEMU主页以获取完整的清单）**

**\* 增加了模拟速度，某些程式甚至可以实时运行**

**\* 可以在其他平台上运行Linux的程式**

**\* 可以储存及还原运行状态（如运行中的程式）**

**\* 可以虚拟网络卡**

**\* 可模拟多CPU**

**qemu – 缺点：**

**\*对微软视窗及某些主机操作系统的支持还不完善（某些模拟的系统仅能运行）**

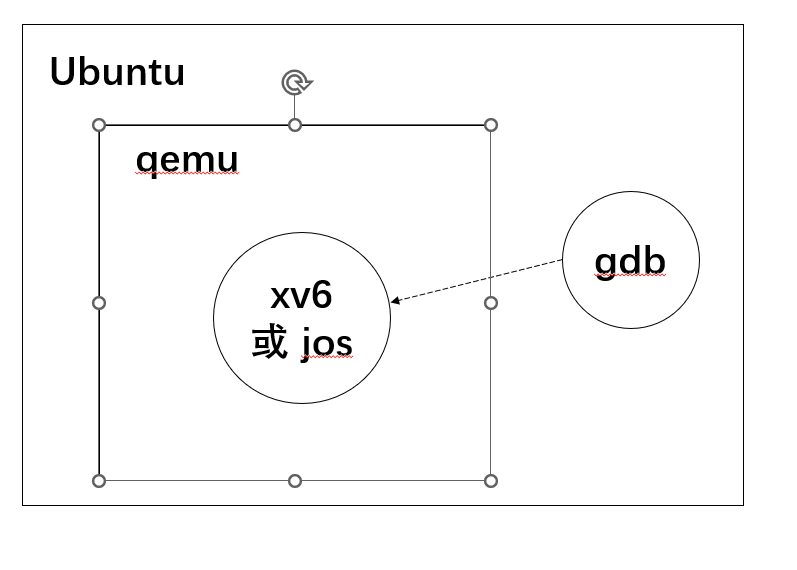
**对不常用的架构的支持并不完善**

**\*除非使用kqemu（或qvm86）加速器，否则其模拟速度仍不及其他虚拟软件，如VMware**

**\*比其他模拟软件难安装及使用**

**<3>xv6与jos**

**Xv6与jos是两个操作系统 xv6比较完整全面 jos简单 jos是xv6技术基础上的改写**

****

**Xv6是由麻省理工学院(MIT)为操作系统工程的课程（代号6.828）,开发的一个教学目的的操作系统。Xv6是在x86处理器上(x即指x86)用ANSI标准C重新实现的Unix第六版(Unix V6，通常直接被称为V6)。**

**JOS则选择微内核结构，将fs作为一个独立的具有较高权限的守护进程在用户态实现**

**两者的区别详见**

[6.828 - Lab5：xv6 Log Filesystem vs. JOS Filesystem - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/365257100)

**Exercise 5：**What's the difference between Qemu and VMware or Virtual Box?

**qemu-kvm可以只用于Linux平台是linux内核的虚拟机，并且灵活性很大，唯一不足是运行winows虚拟机性能并不好，尤其是虚拟3d性能支持不佳（virgl驱动只被linux支持，windows系统需额外编译且只有opengl），当然，qemu-kvm可以用硬件虚拟化技术（包括vGPU）也可以用pci直通，可以进行intel核显虚拟化(igvt)缺点是配置复杂，在桌面版上由于virt-manager的连接方式性能受限。如果你是运行linux虚拟机可以使用qemu-kvm，但若运行windows虚拟机最好不要用它，除非需要配置igvt和显卡直通。**

**virtualbox是开源虚拟机平台，由于其开源特性可以很好地定制化，尤其适合Android系统的虚拟机。virtualbox相比于qemu-kvm，各方面性能都更好（很好的2d性能以及可接受的3d性能）。virtualbox的缺陷是3d性能不佳，仅支持到DirectX9以及OpenGL3.0且支持特性不完整，优点是2d性能强悍（能有约50%主机性能，但是2d性能仅能用于windows系统）。**

**vmware是闭源软件，一般用vmware workstation或player，现在用vmware player 16即可（桌面需求的功能都有了，快照还是不被支持），它对比于virtualbox主要是3d性能提升，但2d性能较差。它支持到opengl3.3和DirectX11，最大8G虚拟显存可以发挥独显的作用，经测试性能大约是原来的50%左右（理论性能更高约66%），可以配置桥接网络。**

**Exercise 6：**What does the "make" directive mean? What is a makefile？

**在linux系统中make是一个非常重要的编译命令，不管是自己进行项目开发还是安装应用软件，我们都经常要用到make或makeinstall。利用make工具，我们可以将大型的开发项目分解成为多个更易于管理的模块，一个工程中的源文件不计数，其按类型、功能、模块分别放在若干个目录中，makefile定义了一系列的规则来指定，哪些文件需要先编译，哪些文件需要后编译，哪些文件需要重新编译，甚至于进行更复杂的功能操作，因为makefile就像一个Shell脚本一样，其中也可以执行操作系统的命令。makefile 带来的好处就是“自动化编译”，一旦写好，只需要一个make命令，整个工程完全自动编译，极大的提高了软件开发的效率。**

**当make 命令被执行时，它会扫描当前目录下Makefile或makefile文件找到目标以及其依赖。如果这些依赖自身也是目标，继续为这些依赖扫描Makefile 建立其依赖关系，然后编译它们。一旦主依赖编译之后，然后就编译主目标,假设你对某个源文件进行了修改，你再次执行make 命令，它将只编译与该源文件相关的目标文件，因此，编译完最终的可执行文件节省了大量的时间。**

**Part 3：The Ending**

**Exercise 7：**Please tell me what you learned from this experiment, or give a summary of the experiment.

**需要了解的东西太多了**

1. **整个实验如果完全做下来做的东西还是比较简单的 无脑做**

**但是如果想要深入理解整个实验背后的逻辑 涉及的东西还是比较多的**

**包括Linux各种指令的用法啦**

**以及Vmware 主机与虚拟机传文件啦**

**以及整个操作系统究竟干了啥东西（目前还在挣扎看xv6英文手册）**

1. **挣扎一学期看看能蹦跶个什么结果吧！同志仍需努力…**