**实验报告**

**Lab 0**

姓名：施君豪

班级：20 人工智能

学号：20307140008

**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. Code
3. Screenshot
4. Difficulties and solutions

实验练习一：

**Implement the UNIX program sleep for xv6; your sleep should pause for a user-specified number of ticks. A tick is a notion of time defined by the xv6 kernel, namely the time between two interrupts from the timer chip. Your solution should be in the file /sleep.c.**

一、代码：（sleep.c）

#include "types.h"

#include "stat.h"

#include "user.h"

int

main(int argc, char \*argv[])

{

if(argc != 2){

printf(1,"Invalid input!\n");

exit();

}

else{

int length = atoi(argv[1]);

printf(1,"Nothing happens for a while\n");

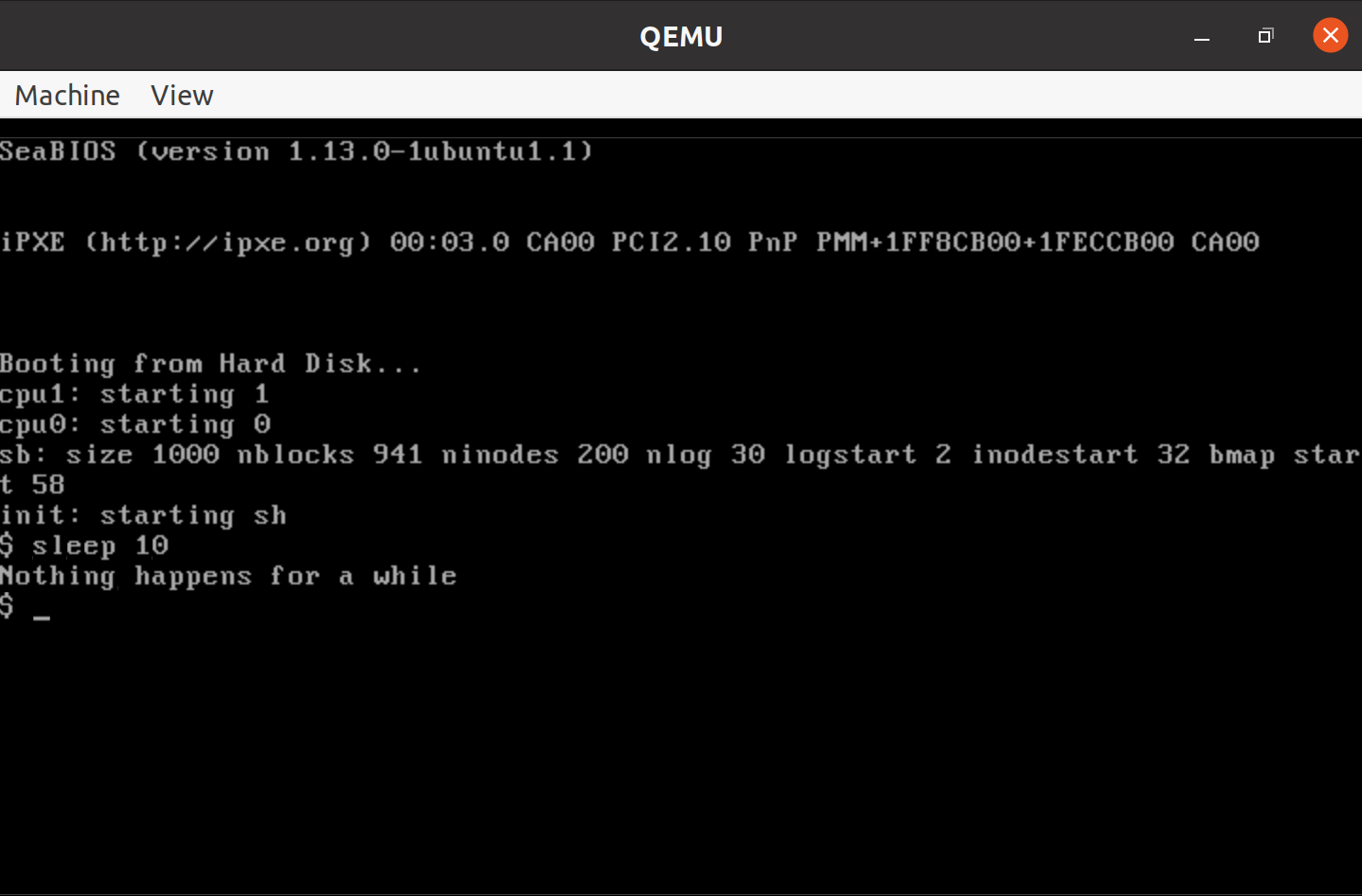
sleep(length);

exit();

}

}

二、运行截图：



三、困难与解决办法

第一次上手xv6时对调用的库以及xv6自带的语法不是很熟悉，一开始使用的库为stdio.h，一直出现qemu编译不通过的情况。随后调阅了助教提供的xv6上手教程，专门查阅了其中函数的语法格式，并参照xv6安装时提供的echo.c进行理解。（区别举例：exit() 与 exit(0)，printf(1,”xxx”)与printf(“xxx”) ）。除此以外，在一开始调用atoi函数时未使用正确语法，在上网查阅后得以解决。

实验练习二：

**Write a program that uses UNIX system calls to ''ping-pong'' a byte between two processes over a pair of pipes, one for each direction. The parent should send a byte to the child; the child should print "<pid>: received ping", where <pid> is its process ID, write the byte on the pipe to the parent, and exit; the parent should read the byte from the child, print "<pid>: received pong", and exit. Your solution should be in the file /pingpong.c.**

一、代码：（pingpong.c）

#include "types.h"

#include "stat.h"

#include "user.h"

Int main()

{

int p1[2],p2[2];

pipe(p1);

pipe(p2);

int pid = fork();

if(pid==0){

close(p1[1]);

close(p2[0]);

char son[20];

read(p1[0],son,10);

int id1 = getpid();

printf(1,"%d:received %s\n",id1,son);

write(p2[1],"pong",sizeof("pong"));

close(p2[1]);

exit();

}

else{

close(p1[0]);

close(p2[1]);

write(p1[1],"ping",sizeof("ping"));

close(p1[1]);

char father[20];

read(p2[0],father,10);

int id2 = getpid();

printf(1,"%d:received %s\n",id2,father);

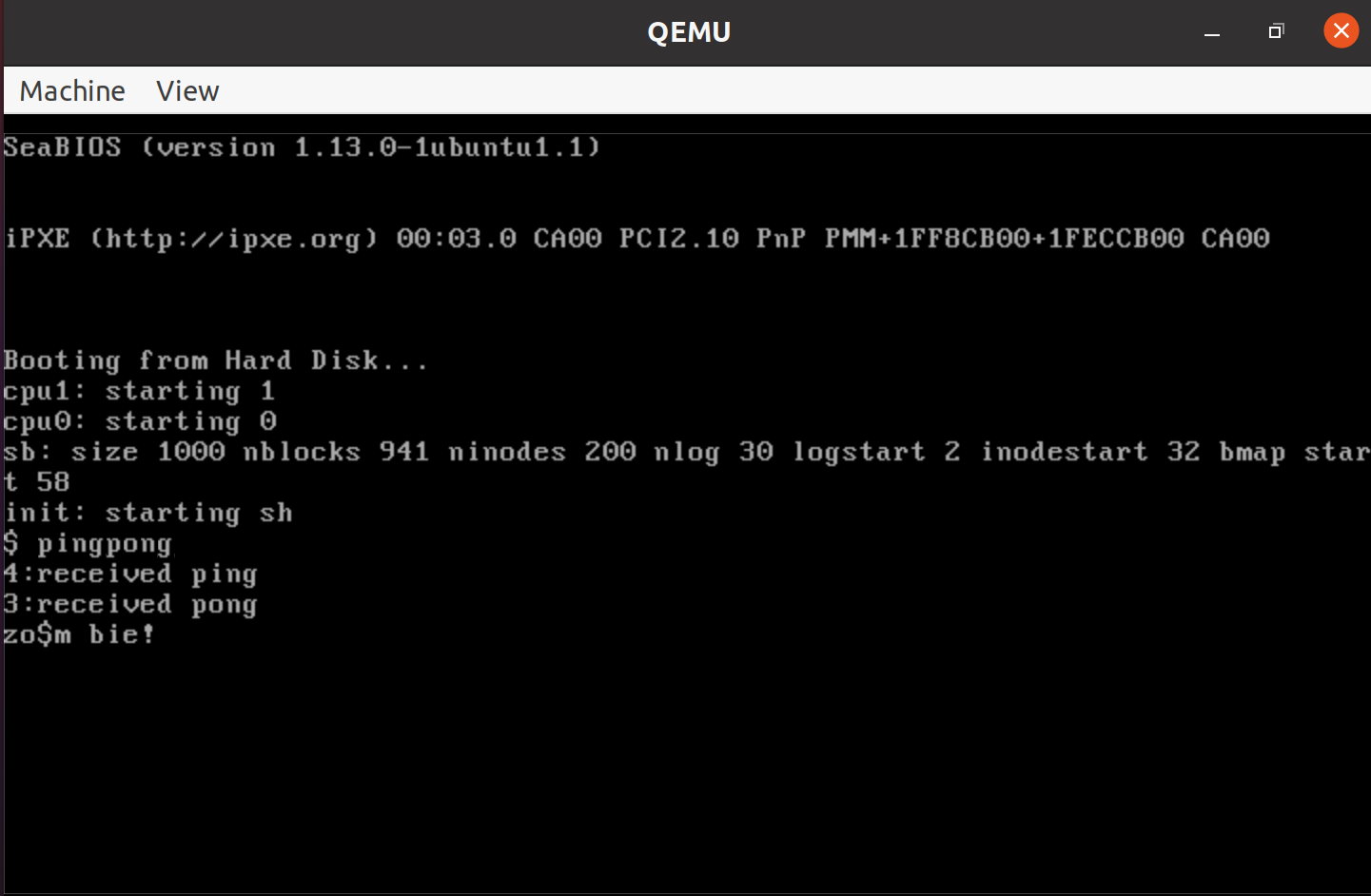
close(p2[0]);

}

exit();

}

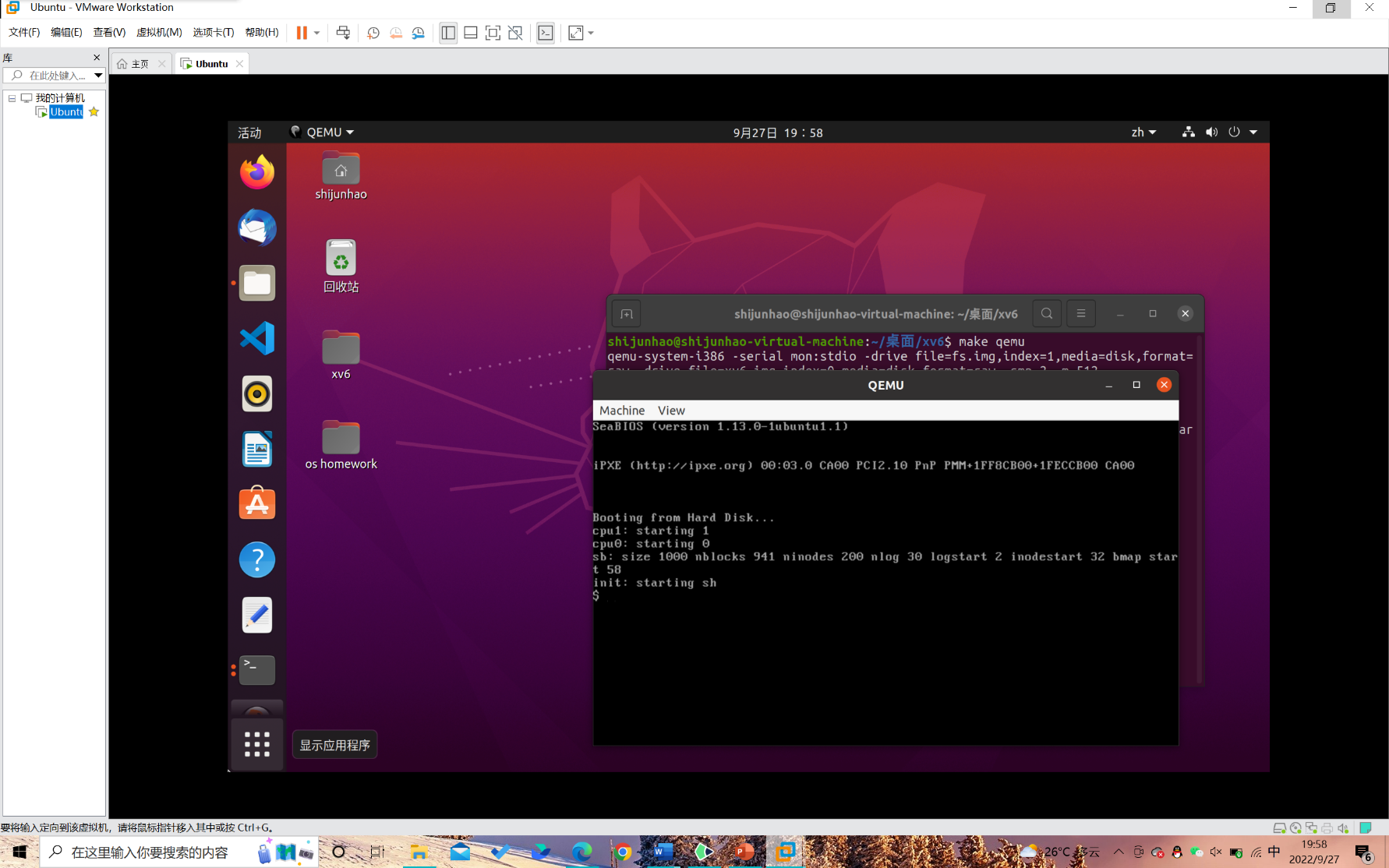
二、屏幕截图



三、困难与解决办法

刚开始时，对于pipe函数使用的设想是通过一根管道实现父进程与子进程之间的数据传输。但是在实践中容易出现某个进程在管道一端输入ping或者pong之后自己读取了输入的数据。一个可行的解决办法是在写入和读取之间加入sleep函数暂停进程，但同时也拖慢了运行速度。最终选择使用两根管道分别负责两个方向的传输。

**Part 1：Environment Configuration**

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

1、下载并安装vmware

2、下载Ubuntu的iso镜像，并在vmware中创建虚拟机

3、在虚拟机中安装make，gcc，以及qemu

4、在qemu硬件模拟器上安装xv6操作系统

**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？

最初下载的版本为Ubuntu22（最新版），22版本相较于20版本不再预装make命令与gcc编译器，需要自己下载。除此以外，在安装qemu时会报错“don‘t know how to make ‘xxxx‘“，经查为不同版本之间的兼容性问题。最终没有选择修改相关内核文件，重新安装了Ubuntu20版本，问题顺利解决。

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

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

操作系统虚拟化是一种服务器虚拟化技术，指将操作系统所管理的计算机资源，包括进程、文件、设备、网络等分组，然后交给不同的应用程序使用，以便它可以一次在一台计算机上运行由多个用户处理的不同应用程序，同时它们也不会相互干扰，从而更加充分地利用计算机资源。

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

QEMU是一种通用的开源计算机仿真器和虚拟器，有全系统仿真，用户模式仿真两种操作模式，支持多种架构。

QEMU优点：

1、默认支持多种架构

2、可扩展，开源，可自定义新的指令集，可移植，仿真速度快。

3、通过纯软件实现的虚拟化模拟器，几乎可以模拟任何硬件设备。

4、增加了模拟速度，某些程序甚至可以实时运行。

5、可以储存及还原运行状态(如运行中的程序)，可以虚拟网络卡。

6、在特定条件下模拟速度可以达到主机速度。

QEMU缺点：

1、对微软Windows及某些主机操作系统的支持不完善(某些模拟的系统仅能运行)

2、由于为软件模拟，模拟速度通常情况下比不过其他虚拟软件，如VMware

3、不支持跨平台运行

4、在安装及使用操作上比其他模拟软件较为繁琐。

Xv6是由麻省理工学院(MIT)为操作系统工程课程（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)。目前最新版为RISC-V版本，本实验使用的为xv6-public版本。

Jos 也是由麻省理工学为操作系统工程课程开发的操作系统，使用时间较早（2006），目前以使用xv6居多。

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

Qemu是linux内核的虚拟机，并且灵活性很大。由于是使用代码来模拟cpu的，导致其速度可能相对剩余二者略慢一些，但是适应性最好，可以在系统上模拟别的内核。缺点是是运行winows虚拟机时性能并不好。虽然qemu-kvm也可以用硬件虚拟化技术与pci直通，缺点是配置复杂，同时缺少可视化图形界面，操作较为繁琐。

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各方面性能都更好。相比vmware虽然有着开源免费的优点，同时2d性能强悍，但相比之下功能相对较少，不过也足够日常使用。

vmware是闭源软件，有vmware workstation，vmware player等诸多细分应用，功能强大。它对比于virtualbox主要是3d性能提升，但2d性能相对较差。除此以外，vmware对于系统的适配相对更好，如linux系统会直接提供 open-vm-tools 和 open-vm-tools-desktop 软件包，安装较为便利。

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

make是GNU Make中的一条计算机指令，指令的操作是读入一个名为makefile 的文件，然后执行这个文件中指定的指令。

Makefile 文件描述了整个工程的编译、连接等规则，规定了要编译的内容与顺序，最终集成为一个文件。在完成之后能够在shell中使用一行make命令来直接完成“自动化编译”，极大提升了效率。

**Part 3：The Ending**

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

通过本实验我了解了虚拟机安装的流程，同时对linux系统，GNU make，以及本实验所使用的xv6有了一定了解。除此以外，在实现sleep与pingpong函数的过程中了解了xv6的语法，体验了调试的流程，为以后的实验打下基础。