

Lab5: I/O 设备与网卡驱动 - 用户手册

一、实验环境准备

1.1 环境要求

- (1) Ubuntu/Linux 系统
- (2) Python 3.x (用于测试脚本)
- (3) 至少 2 个终端窗口
- (4) 确保在 xv6 源码根目录下操作

1.2 进入实验环境

- (1) 进入 xv6 实验目录
`cd ~/xv6-labs-2024`
- (2) 清理之前的编译文件
`make clean`

二、编译与启动 xv6 系统

2.1 编译 xv6 内核

在第一个终端窗口中

```
make qemu
```

2.2 成功启动的标志

当看到以下输出时，表示 xv6 系统已成功启动：

```
xv6 kernel is booting
```

```
hart 2 starting
```

```
hart 1 starting
```

```
init: starting sh
```

```
$
```

此时出现 \$ 提示符，表示可以输入命令。

三、测试准备

3.1 测试脚本位置

所有测试脚本位于 xv6 源码根目录下：

nettest.py: Python 测试脚本

user/nettest.c: xv6 内的测试程序

3.2 需要两个终端窗口

终端 1: 用于运行 xv6 系统 (make qemu)

终端 2: 用于运行 Python 测试脚本

四、三个核心测试

4.1 测试 1: 发送功能测试 (e1000_transmit)

测试目的: 验证网卡发送数据包功能是否正常。

测试步骤:

(1) 终端 2 - 启动测试服务器:

```
python3 nettest.py txone
```

可能的输出示例: (也有可能不会有输出)

```
listening on lo port 25193
```

```
Waiting for packet...
```

(2) 终端 1 - 在 xv6 中运行发送测试:

```
$ nettest txone
```

观察结果:

成功: 终端 2 显示 txone: OK

失败: 终端 2 显示错误信息, 或 xv6 中无响应

4.2 测试 2: 接收功能测试 (e1000_recv)

测试目的: 验证网卡接收数据包功能是否正常。

测试步骤:

(1) 终端 1 - 确保 xv6 正在运行 (如果退出了需要重新 make qemu)

(2) 终端 2 - 发送测试数据包:

```
python3 nettest.py rxone
```

观察结果:

成功: 终端 1 (xv6) 显示:

```
arp_rx: received an ARP packet
```

```
ip_rx: received an IP packet
```

失败: 无输出

4.3 测试 3：完整功能测试（UDP 系统调用）

测试目的：验证完整的 UDP 收发功能，包括系统调用。

测试步骤：

（1）终端 2 - 启动完整测试：

```
python3 nettest.py grade
```

可能的输出示例：

```
listening on lo port 27643
```

```
Waiting for packet...
```

（2）终端 1 - 在 xv6 中运行完整测试：

```
$ nettest grade
```

观察结果：

成功：终端 2 显示一系列测试通过，最后显示 ALL TESTS PASSED

终端 1 最后显示 free: OK

失败：测试过程中出现错误信息