

xv6-labs-2024 实验汇总

- 2351289周慧星

本项目是关于 xv6-labs-2024的实验，包含了 9 个实验，涵盖了 Unix 工具、系统调用、页表、陷阱处理、写时复制、网络、锁、文件系统和内存映射等多个方面。以下是每个实验的详细信息：

目录

- [xv6-labs-2024 实验汇总](#)
 - [目录](#)
 - [仓库地址：](#)
 - [实验列表](#)
 - [环境搭建和tools](#)
 - [Lab1: Xv6 and Unix utilities](#)
 - [Lab2: System Calls](#)
 - [Lab3: Page Tables](#)
 - [Lab4: Traps](#)
 - [Lab5: Copy-on-Write Fork for xv6](#)
 - [Lab6: networking](#)
 - [Lab7: locks](#)
 - [Lab8: file system](#)
 - [Lab9: mmap](#)
 - [实验准备步骤](#)

仓库地址：

[2351289周慧星-xv6-labs-2024](#)

实验列表

详细实验内容见各对应pdf文件。

环境搭建和tools

使用VMware、Ubuntu和VS Code完成xv6实验并搭建环境，具体见文档Tools。

Lab1: Xv6 and Unix utilities

- **实验内容：** 本实验旨在让用户熟悉 xv6 及其系统调用，包含 sleep、pingpong、primes、find 和 xargs 等子实验。
- **对应分支：** util
- **测试（跑分）方法：**

make grade

Lab2: System Calls

- **实验内容：**包含 Using gdb、System call tracing 和 Attack xv6 等子实验，涉及系统调用的调试、跟踪和安全漏洞测试。
- **对应分支：**syscall
- **测试（跑分）方法：**

```
make grade
```

Lab3: Page Tables

- **实验内容：**包含 Inspect a user-process page table 和 Speed up system calls 等子实验，主要涉及用户进程页表的检查和系统调用的加速。
- **对应分支：**pgtbl
- **测试（跑分）方法：**

```
make grade
```

Lab4: Traps

- **实验内容：**包含 RISC-V assembly、Backtrace 和 Alarm 等子实验，涉及 RISC-V 汇编、回溯和警报机制的实现。
- **对应分支：**traps
- **测试（跑分）方法：**

```
make grade
```

Lab5: Copy-on-Write Fork for xv6

- **实验内容：**实现写时复制（Copy-on-Write, COW）机制的 `fork` 系统调用，优化 xv6 中进程创建的性能。
- **对应分支：**cow
- **测试（跑分）方法：**

```
make grade
```

此外，还可以进行以下测试：

```
$ cowtest  
$ usertests -q
```

Lab6: networking

- **实验内容：**包含 Part One: NIC 和 Part Two: UDP Receive 等子实验，涉及网卡驱动和 UDP 接收的实现。
- **对应分支：**net
- **测试（跑分）方法：**

```
make grade
```

Lab7: locks

- **实验内容：**包含 Memory Allocator 和 Buffer Cache 等子实验，主要涉及内存分配器和缓冲区缓存的锁机制优化。
- **对应分支：**lock
- **测试（跑分）方法：**

```
make grade
```

此外，还可以进行以下测试：

```
$ make qemu  
$ bcachetest  
$ usertests -q
```

Lab8: file system

- **实验内容：**包含 Large files 和 Symbolic links 等子实验，涉及大文件和符号链接的实现。
- **对应分支：**fs
- **测试（跑分）方法：**

```
make grade
```

此外，还可以进行以下测试：

```
$ bigfile  
$ usertests -q
```

Lab9: mmap

- **实验内容：**实现 mmap 和 munmap 系统调用，涉及内存映射和文件系统的操作。
- **对应分支：**mmp
- **测试（跑分）方法：**

```
make grade
```

此外，还可以进行以下测试：

```
$ mmaptest  
$ usertests -q
```

实验准备步骤

在进行每个实验之前，通常需要进行以下准备步骤：

切换到实验对应的代码分支：

```
git fetch  
git checkout <branch_name>  
make clean
```

其中 `<branch_name>` 是每个实验对应的分支名称。