xv6 实验项目介绍及环境搭建

2351882 王小萌

Tongji University,2025 Summer

代码仓库链接：https://github.com/wxmxmw/Tongji-University-xv6-labs-2021.git

项目介绍

xv6 是一个用于教学目的的类 Unix 操作系统，基于 RISC-V 架构实现。它由 MIT 开发并广泛应用于操作系统课程的教学中。xv6 的设计简洁、结构清晰，涵盖了现代操作系统的基本功能，如进程管理、内存管理、文件系统、中断处理等，是学习操作系统原理与实践的理想平台。

本实验基于 2021 年版本的 xv6 Labs，共包含以下 10 个实验模块:

实验编号	实验名称	实验目标简述
Lab 1	Utilities（实用工具）	实现用户态命令如 sleep、pingpong、primes、find、xargs 等
Lab 2	System calls（系统调用）	添加系统调用追踪功能和 sysinfo 系统调用
Lab 3	Page tables（页表操作）	实现页表遍历、访问位检测等功能
Lab 4	Traps（中断处理）	学习 RISC-V 汇编语言及异常/中断处理机制
Lab 5	Copy-on-Write（写时复制）	实现 COW 机制优化 fork 性能
Lab 6	Multithreading（多线程）	支持内核级多线程调度与同步
Lab 7	Network driver（网卡驱动）	实现 e1000 网卡驱动支持网络通信

实验编号	实验名称	实验目标简述
Lab 8	Lock（锁机制）	优化并发控制，减少锁竞争
Lab 9	File system（文件系统）	扩展文件系统功能，理解 inode 与目录结构
Lab 10	mmap（内存映射）	实现 mmap 及 munmap 系统调用

通过完成这些实验，不仅掌握了操作系统底层机制的实现方式，还提升了对 RISC-V 架构、C 语言指针操作、并发编程、内存管理等方面的能力。

实验开发环境

组件	版本/说明
操作系统	Ubuntu 20.04 （运行于 WSL2）
虚拟化支持	已启用 Hyper-V 和虚拟机平台功能
编译工具链	gcc-riscv64-linux-gnu, binutils-riscv64-linux-gnu
调试器	gdb-multiarch
模拟器	QEMU (qemu-system-misc)
实验源码仓库	mit-pdos/xv6-labs-2021

环境配置/Tools

1. 安装 WSL 并启用虚拟化,启动 Ubuntu:

步骤 1 - 启用适用于 Linux 的 Windows 子系统

```
PS C:\Windows\system32> dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart

部署映像服务和管理工具
版本: 10.0.22621.2792

映像版本: 10.0.22631.5472

启用一个或多个功能
[==========================100.0%==========================]
操作成功完成。
PS C:\Windows\system32>
```

步骤 2 - 启用虚拟机功能

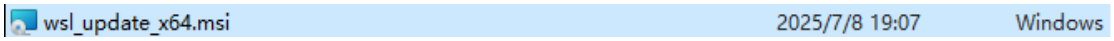
```
PS C:\Windows\system32> dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart

部署映像服务和管理工具
版本: 10.0.22621.2792

映像版本: 10.0.22631.5472

启用一个或多个功能
[==========================100.0%==========================]
操作成功完成。
PS C:\Windows\system32>
```

步骤 3 - 下载 Linux 内核更新包

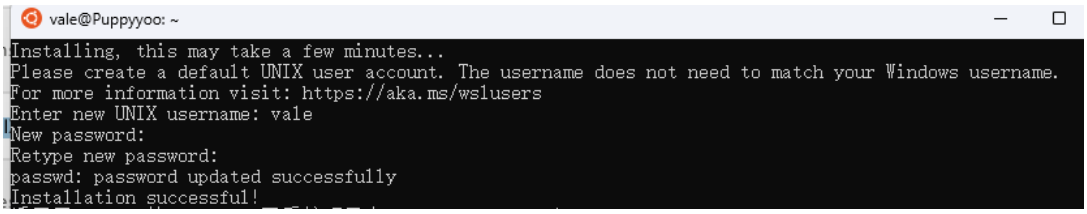


步骤 4 -将 WSL 2 设置为默认版本:

```
PS C:\Windows\system32> wsl --set-default-version 2
有关与 WSL 2 的主要区别的信息，请访问 https://aka.ms/wsl2
操作成功完成。
PS C:\Windows\system32>
```

步骤 5 -下载并手动安装 Linux 分发版：Ubuntu 20.04

安装完成后，系统提示用户输入新 UNIX 用户名和密码。输入过程中的信息如下：（密码隐藏）



2. 软件源更新和环境准备:

安装本项目所需的所有软件，运行:

```
$ sudo apt-get update && sudo apt-get upgrade

$ sudo apt-get install git build-essential gdb-multiarch qemu-system-misc gcc-riscv64-linux-gnu binutils-riscv64-linux-gnu
```

```
vale@Puppyyoo: $ sudo apt-get install git build-essential gdb-multiarch qemu-system-misc gcc-riscv64-linux-gnu binutils-riscv64-linux-gnu
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.25.1-1ubuntu3.14).
git set to manually installed.
The following additional packages will be installed:
  acl adwaita-icon-theme at-spi2-core binutils binutils-common binutils-x86-64-linux-gnu cpp cpp-9 cpp-9-riscv64-linux-gnu
  cpp-riscv64-linux-gnu dpkg-dev fakeroot fontconfig g++ g++-9 gcc gcc-10-cross-base-ports gcc-9 gcc-9-base gcc-9-cross-base-ports
  gcc-9-riscv64-linux-gnu gcc-9-riscv64-linux-gnu-base gdb gdbserver gstreamer1.0-plugins-base gstreamer1.0-plugins-good
  gstreamer1.0-x gtk-update-icon-cache hicolor-icon-theme humanity-icon-theme ibverbs-providers ipxe-qemu libaal
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatk-bridge2.0-0 libatk1.0-0 libatk1.0-data
  libatomic1 libatomic1-riscv64-cross libatspi2.0-0 libavahi-client3 libavahi-common-data libavahi-common3 libavc1394-0
  libbabeltrace1 libbinutils libboost-iostreams1.71.0 libboost-thread1.71.0 libbrlapi0.7 libc-dev-bin libc6-dbg libc6-dev
  libc6-dev-riscv64-cross libc6-riscv64-cross libcaca0 libcacard0 libcairo-gobject2 libcairo2 libccl-0 libcdparanoia0 libcolord2
  libcrypt-dev libctf-nobfd0 libctf0 libcups2 libdatrie1 libdpkg-perl libdv4 libdw1 libepoxy0 libfakeroot libfdt1
  libfile-fcntllock-perl libgbm1 libgcc-9-dev libgcc-9-dev-riscv64-cross libgcc-s1-riscv64-cross libgdk-pixbuf2.0-0
```

```
Setting up gcc-riscv64-linux-gnu (4:9.3.0-1ubuntu2) ...
Setting up librbd1 (15.2.17-0ubuntu0.20.04.6) ...
Setting up g++-9 (9.4.0-1ubuntu1~20.04.2) ...
Setting up librsvg2-common:amd64 (2.48.9-1ubuntu0.20.04.4) ...
Setting up g++ (4:9.3.0-1ubuntu2) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
Setting up build-essential (12.8ubuntu1.1) ...
Setting up libiscsi7:amd64 (1.18.0-2) ...
Setting up qemu-block-extra:amd64 (1:4.2-3ubuntu6.30) ...
Setting up qemu-system-common (1:4.2-3ubuntu6.30) ...
Created symlink /etc/systemd/system/multi-user.target.wants/qemu-kvm.service → /lib/systemd/system/qemu-kvm.service.
Setting up qemu-system-misc (1:4.2-3ubuntu6.30) ...
Setting up qemu-utils (1:4.2-3ubuntu6.30) ...
Setting up adwaita-icon-theme (3.36.1-2ubuntu0.20.04.2) ...
update-alternatives: using /usr/share/icons/Adwaita/cursor.theme to provide /usr/share/icons/default/index.theme (x-cursor-theme) in
auto mode
Setting up humanity-icon-theme (0.6.15) ...
Setting up ubuntu-mono (19.04-0ubuntu3) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for libglib2.0-0:amd64 (2.64.6-1~ubuntu20.04.9) ...
Setting up libgtk-3-0:amd64 (3.24.20-0ubuntu1.2) ...
Processing triggers for libc-bin (2.31-0ubuntu9.18) ...
/sbin/ldconfig.real: Can't link /usr/lib/wsl/lib/libnvoptix_loader.so.1 to libnvoptix.so.1
Setting up libgtk-3-bin (3.24.20-0ubuntu1.2) ...
Setting up libvte-2.91-0:amd64 (0.60.3-0ubuntu1~20.5) ...
Setting up qemu-system-gui:amd64 (1:4.2-3ubuntu6.30) ...
Processing triggers for libgdk-pixbuf2.0-0:amd64 (2.40.0+dfsg-3ubuntu0.5) ...
Processing triggers for libc-bin (2.31-0ubuntu9.18) ...
/sbin/ldconfig.real: Can't link /usr/lib/wsl/lib/libnvoptix_loader.so.1 to libnvoptix.so.1
```

3. 测试安装

```
$ qemu-system-riscv64 --version
```

```
vale@Puppyyoo:~$ qemu-system-riscv64 --version
QEMU emulator version 4.2.1 (Debian 1:4.2-3ubuntu6.30)
Copyright (c) 2003-2019 Fabrice Bellard and the QEMU Project developers
```

```
vale@Puppyyoo:~$ riscv64-linux-gnu-gcc --version
riscv64-linux-gnu-gcc (Ubuntu 9.4.0-1ubuntu1~20.04) 9.4.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

4. 编译内核

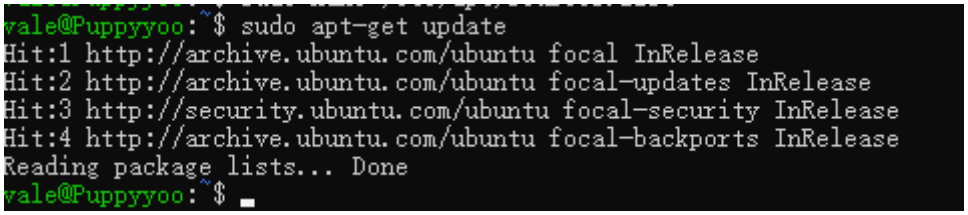
下载 xv6 内核源码

```
$ git clone git://github.com/mit-pdos/xv6-riscv.git
```

更新镜像源

```
$ sudo nano /etc/apt/sources.list

$ sudo apt-get update
```



Guidance

理解 C 语言和指针

确保你理解 C 语言和指针。Kernighan 和 Ritchie 的《C 程序设计语言（第二版）》是 C 语言的简洁描述。一 些有用的指针练习在这里。除非你已经非常熟悉 C 语言，否则不要跳过或略读上述指针练习。如果你不能 真正理解 C 语言中的指针，你将在实验中遭受难以形容的痛苦和折磨，然后最终通过艰难的方式理解它们。相信我们；你不想知道“艰难的方式”是什么。

一些指针的常见习语特别值得记住： 如果 int *p = (int*)100，那么 (int)p + 1 和 (int)(p + 1) 是不同的数字：前者是 101，而后者是 104。当向指针添加整数时，如第二种情况，整数隐式乘以指针指向的对象的大小。 p[i] 定义为 *(p+i)，指的是 p 指向的内存中的第 i 个对象。上述加法规则有助于在对象大于一个字节时使此定义工作。&p[i] 与(p+i)相同，生成 p 指向的内存中第 i 个对象的地址。 虽然大多数 C 程序从不需要在指针和整数之间进行类型转换，但操作系统经常需要。每当你看到涉及内存地址的加法时，问问自己这是整数加法还是 指针加法，并确保添加的值适当乘以或不乘以。

检查进度

如果你的练习部分工作，请通过提交代码来检查你的进度。如果稍后出现问题，可以回滚到检查点并以较 小的步骤前进。要了解更多关于 Git 的信息，请查看 Git 用户手册，或者，你可能会发现这个面向计算机科 学的 Git 概述很有用。

测试失败

如果测试失败，请确保你了解为什么代码失败。插入打印语句直到你了解发生了什么。你可能会发现打印 语句会生成大量你想搜索的输出；一种方法是在 script 中运行 make qemu（在你的机器上运行 man script），它将所有控制台输出记录到一个文件中，然后你可以搜索。不要忘记退出 script。

使用 GDB 调试

在许多情况下，打印语句就足够了，但有时能够逐步执行一些汇编代码或检查堆栈上的变量是有帮助的。 要使用 gdb 调试 xv6，请在一个窗口中运行 make qemu-gdb，在另一个窗口中运行 gdb（或 riscv64 linux-gnu-gdb），设置断点，然后输入 c（继

续），xv6 将运行直到命中断点。（参见使用 GNU 调试器 获取有用的 GDB 提示。） 如果你想查看编译器为内核生成的汇编代码或找到特定内核地址的指令，请参见 kernel.asm 文件，该文件 在编译内核时由 Makefile 生成。（Makefile 还为所有用户程序生成.asm 文件。）

内核崩溃

如果内核崩溃，它将打印一条错误消息，列出崩溃时程序计数器的值；你可以搜索 kernel.asm 以查找程序 计数器在崩溃时所在的函数，或者你可以运行 addr2line -e kernel/kernel pc-value（运行 man addr2line 获取详细信息）。如果你想获取回溯，请重新启动使用 gdb：在一个窗口中运行 make qemu gdb，在另一个窗口中运行 gdb（或 riscv64-linux-gnu-gdb），在 panic 中设置断点（b panic），然后 输入 c（继续）。当内核命中断点时，输入 bt 获取回溯。

内核挂起

如果内核挂起（例如，由于死锁）或无法继续执行（例如，由于执行内核指令时的页面错误），你可以使 用 gdb 找出挂起的地方。在一个窗口中运行 make qemu-gdb，在另一个窗口中运行 gdb（riscv64-linux gnu-gdb），然后输入 c（继续）。当内核似乎挂起时，在 qemu-gdb 窗口中按 Ctrl-C 并输入 bt 获取回溯。

QEMU 监视器

QEMU 有一个“监视器”，可以让你查询仿真机器的状态。你可以通过输入 control-a c（“c”表示控 制台）来访问它。一个特别有用的监视器命令是 info mem，用于打印页表。你可能需要使用 cpu 命令来选 择 info mem 查看的核心，或者你可以使用 make CPUS=1 qemu 来启动 qemu，以使其只有一个核心。