

操作系统实验手册：qemu 内核启动

1 实验环境

本实验可在两类环境中完成：服务器环境和本地环境。其中前者使用学校服务器平台完成实验，后者使用个人 PC 完成实验。

本实验应当在服务器环境中进行，本地环境仅作为备用方案。除非特别注明，本文档介绍的所有操作默认在服务器环境中完成。

环境类别	操作系统	CPU 数量/个	内存	磁盘空间
服务器环境	OpenEuler	2 (不可更改)	6.5GiB (不可更改)	40GiB (不可更改)
本地环境	Openkylin	4 (可更改)	8GiB (可更改)	50GiB (不可更改)

2 实验题目

2.1 实验目的

通过本实验的学习，掌握信创操作系统内核裁剪、根文件系统定制以及内核调参技术。

2.2 实验内容

在本次实验中，学生将利用 qemu 模拟器启动并运行一个虚拟机，以下是详细的实验操作步骤：

1. 修改 Linux 内核默认编译配置，移除对 ext4 文件系统的默认支持，并新增对 btrfs 文件系统的支持，随后进行内核编译。
2. 利用 buildroot 工具构建根文件系统（rootfs），配置 Target options 为最终运行该系统的平台类型。配置 Filesystem images 为 btrfs 格式，此操作将生成 btrfs 类型的虚拟磁盘，内含 rootfs。配置 Target packages，向 rootfs 中添加 vim 和 bash。配置 System configuration，将默认 shell 设置为 bash。
3. 通过 qemu，添加共享文件夹和 edu 设备，结合已编译的内核及虚拟磁盘，尝试启动并运行基于 btrfs 虚拟磁盘的虚拟机。
4. 将在 qemu 虚拟机外编译好的 edu 驱动程序复制到虚拟机内，并进行安装和运行测试。
5. 使用默认配置的内核，进入虚拟机后查询内核参数 kernel.shmmax 的值，并尝试修改该参数。修改完毕后，再次查询 kernel.shmmax 的值，以验证修改是否成功。

(本地环境) 在本地环境中，第二点在配置 Target packages 时还要向 rootfs 中添加 openssh；第三点，通过 qemu 添加的不再是共享文件夹，而是添加网络。

3 实验期望结果

虚拟机可以正常启动，edu 设备驱动正常运行，内核参数成功修改。

4 实验前置要求

本实验需要了解 qemu、虚拟机、rootfs、linux 内核相关知识。

5 主要工具与命令介绍

5.1 buildroot

Buildroot 是一个简单、高效且易于使用的根文件系统映像工具，用于构建嵌入式 Linux 系统。广泛应用于嵌入式系统开发，特别是在需要小体积、定制化 Linux 系统的场合。

5.1.1 配置

- 命令：make menuconfig
- 说明：定制根文件系统包含的软件包与特性，使用上述命令可以使用 TUI 对 buildroot 进行快速配置。

5.1.2 构建

- 命令：make -j \$(nproc)
- 说明：在服务器环境下，软件包源代码已提前下载好，上述命令将直接编译相应的软件包，完成根文件系统的构建。
- 说明：在本地环境下，软件包源代码需要下载，上述命令将下载并编译相应的软件包，完成根文件系统的构建。

5.2 kernel

5.2.1 配置

- 命令：make menuconfig
- 说明：定制 linux 内核中包含的驱动与特性，使用上述命令可以使用 TUI 对 linux 内核进行快速配置。

5.2.2 构建

- 命令：在服务器环境下，make -j \$(nproc) Image.gz
- 命令：在本地环境下，make -j \$(nproc) bzImage
- 说明：上述命令将仅编译内核，并进行压缩。Qemu 虚拟机可以使用压缩后的内核。

5.3 Qemu

5.3.1 运行虚拟机

服务器环境：

- 命令：下述命令已在服务器环境中以脚本形式提供，不必手动输入

```
#!/bin/bash
qemu-system-aarch64 \
  -machine virt \
  -cpu cortex-a57 \
  -smp 1 \
  -m 2048 \
  -drive file=rootfs.btrfs,if=virtio,format=raw \
  -kernel Image.gz \
  -append "console=ttyAMA0 root=/dev/vda oops=panic panic_on_warn=1 panic=-1 ftrace_dump_on_oops=orig_cpu debug earlyprintk=serial slub_debug=UZ" \
  -virtfs local,path=/root/share,mount_tag=sharefd,security_model=passthrough \
  -device edu \
  -nographic
```

- -machine virt：指定机器类型
- -cpu cortex-a57：指定 CPU 类型
- -smp 1：指定 CPU 核数
- -m：指定内存大小（单位为 MB）
- -kernel：设置内核启动，需提供内核路径
- -driver：添加硬盘设备，需提供硬盘路径
- -append：追加的内核启动参数，如指定 root 路径以及初始化进程路径
- -virtfs：添加共享文件夹。path 参数指定将被共享的，Host 机侧的文件夹；mount_tag 参数指定被共享文件夹在 QEMU 虚拟机内的标志符。
- -device：添加其他设备，此处添加 edu 设备
- -nographic：禁止使用图形界面输出，并使用当前终端作为输出

本地环境：

- 命令

```
qemu-system-x86_64 \
  -m 4096 \
  -kernel bzImage \
  -drive file=./rootfs.btrfs,format=raw \
  -append "console=ttyS0 root=/dev/sda init=/linuxrc" \
  -net nic,model=e1000 -net user \
  -device edu \
  -nographic
```

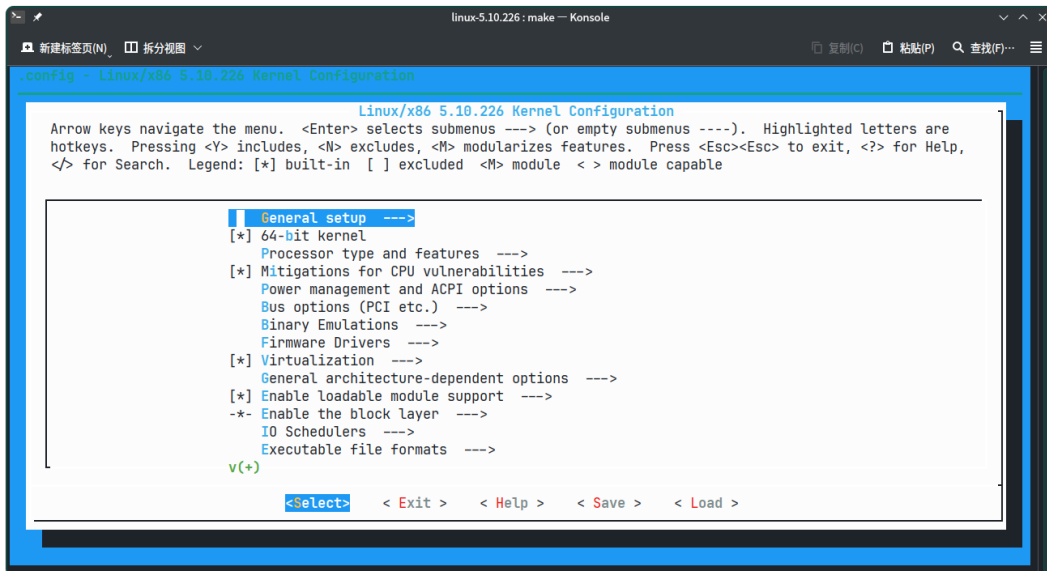
- -m：指定内存大小（单位为 MB）
- -kernel：设置内核启动，需提供内核路径
- -driver：添加硬盘设备，需提供硬盘路径

- `-append`: 追加的内核启动参数, 如指定 `root` 路径以及初始化进程路径
- `-net`: 设置网络设备
- `-device`: 添加其他设备, 此处添加 `edu` 设备
- `-nographic`: 禁止使用图形界面输出, 并使用当前终端作为输出

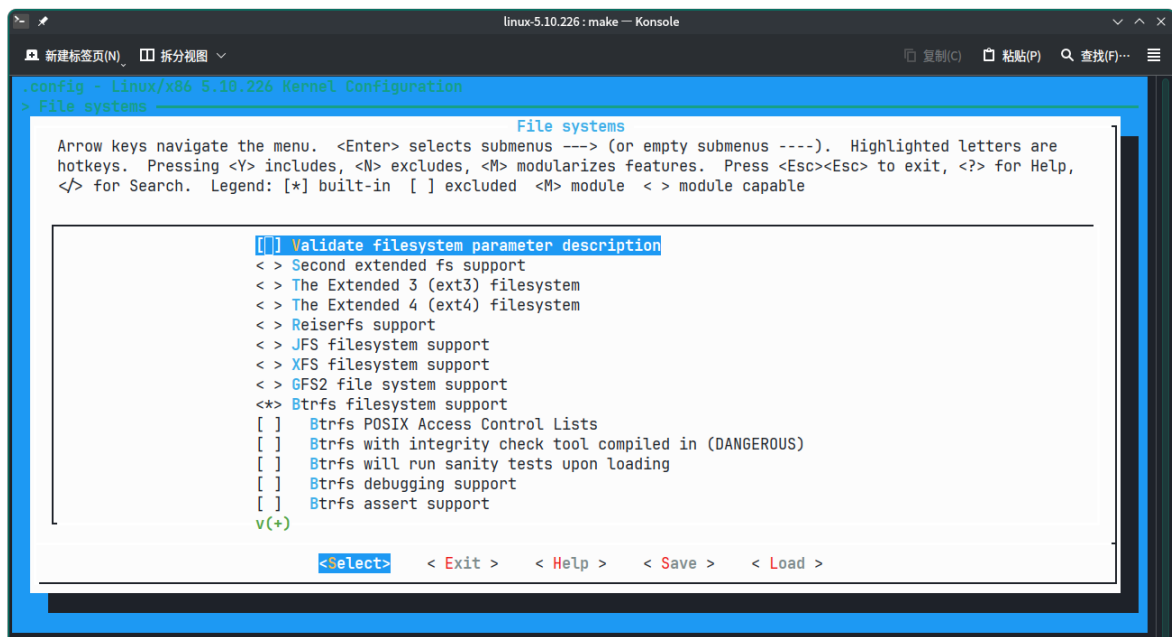
6 主要步骤

6.1 编译内核

首先使用 `make menuconfig` 命令打开内核配置选项界面：



在进入 file systems 子目录后，取消 ext4 支持，并勾选 btrfs 支持：



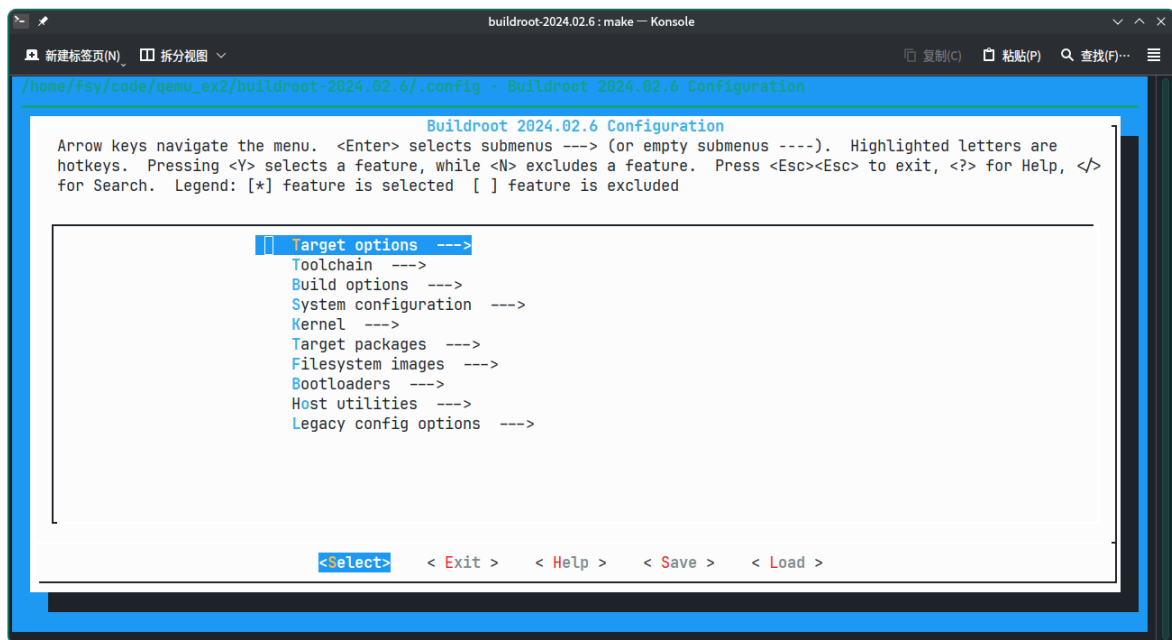
完成上述配置后保存并退出。

（服务器环境）使用 `make -j $(nproc) Image.gz` 编译内核。

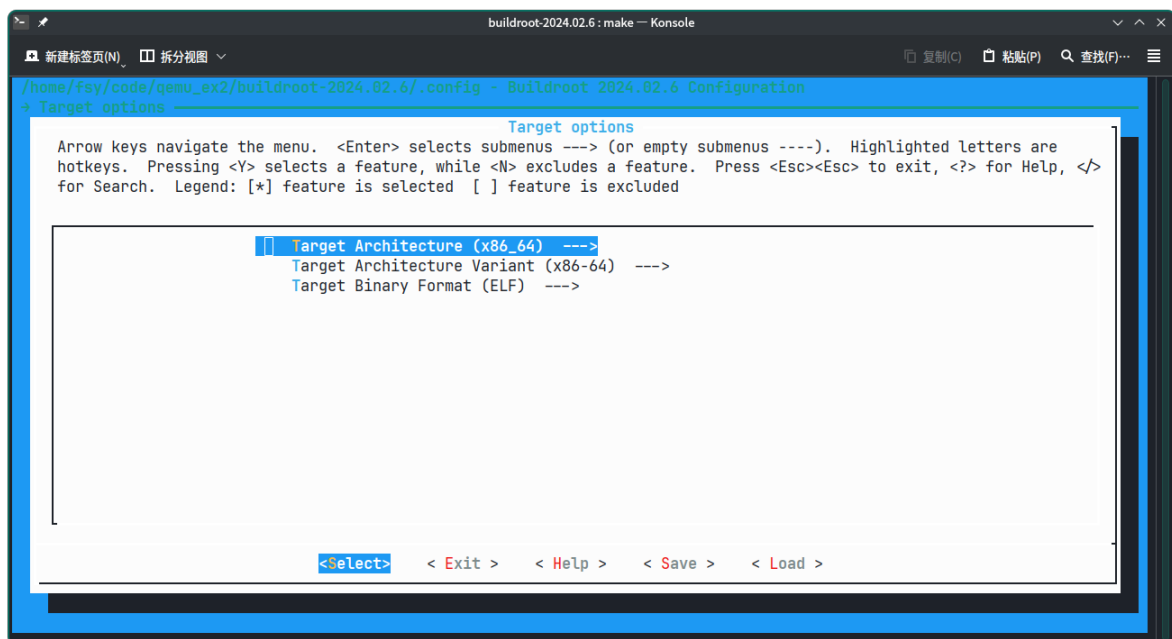
(本地环境) 使用 `make -j $(nproc) bzImage` 编译内核。

6.2 构造 rootfs

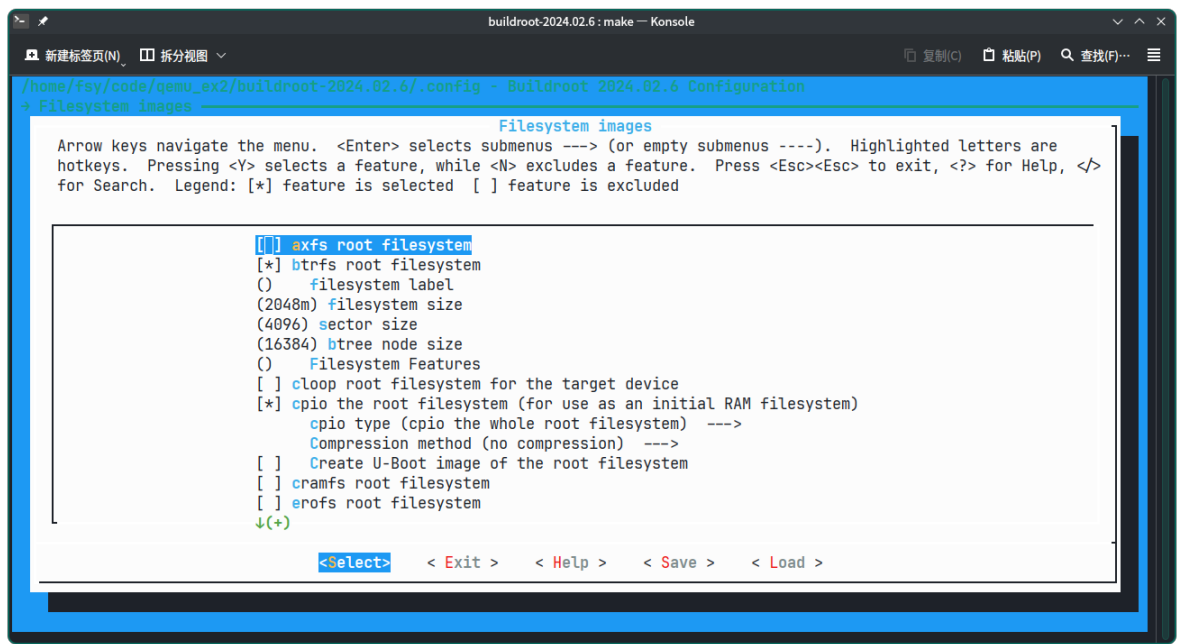
首先使用 `make menuconfig` 命令打开 buildroot 配置选项界面：



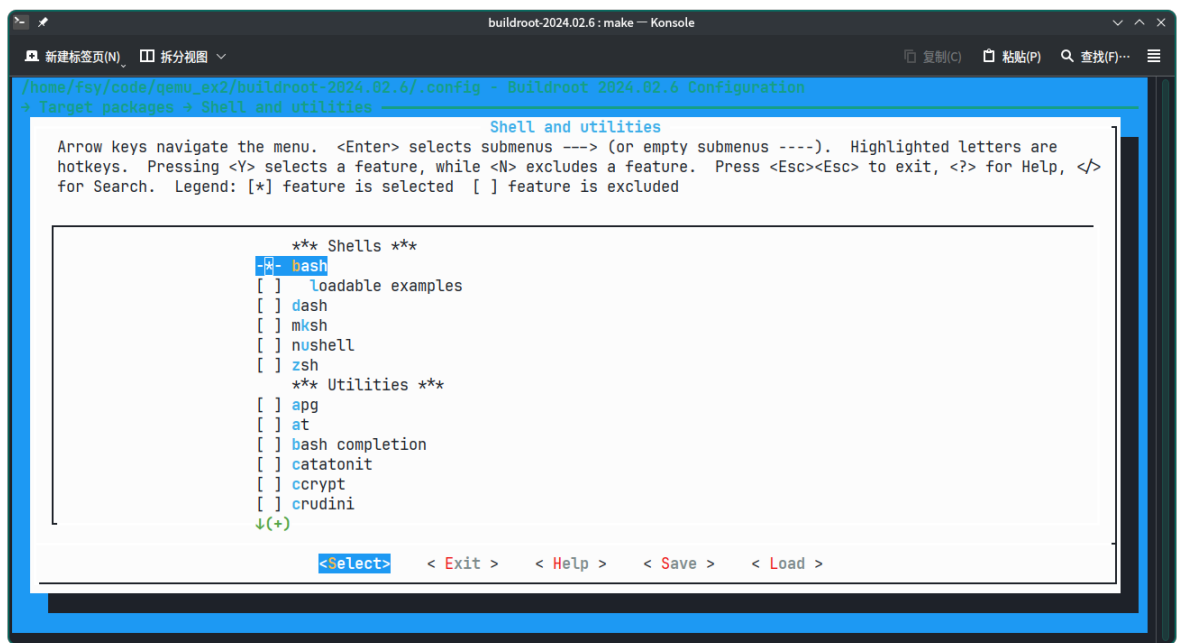
配置 Target options 为 x86_64（仅供参考，此处应使用 `uname -r` 命令提前查看当前机器指令集类型）：



配置 Filesystem images 为 btrfs 格式¹:



配置 Target packages, 向 rootfs 中添加 vim、openssh（服务器环境无需配置 openssh）以及 bash:



完成上述配置后保存并退出。使用 `make -j $(nproc)` 构造根文件系统。

¹ btrfs 格式要求文件系统大小不低于 109MB。

6.3 启动虚拟机

（服务器环境）同样按照 5.3.1 运行虚拟机的命令启动虚拟机，不过需要在服务器环境中提前创建/root/share 文件夹。

（本地环境）按照 5.3.1 运行虚拟机的命令启动虚拟机，此处不再赘述。

6.4 加载并测试 edu 设备驱动

6.4.1 服务器环境

由于服务器环境提供的 qemu 版本较低，无法配置 user 网络，所以改用共享文件夹方式复制文件。

首先，在 Qemu 虚拟机内部创建一个空文件夹，比如/root/qemu-share。

然后，在 Qemu 虚拟机内部执行命令挂载 Host 侧的/root/share 文件夹到 Qemu 侧的/root/qemu-share。

```
mount -t 9p -o trans=virtio [mount tag] [mount point]
```

下一步，将前一个实验中编译的 edu 设备驱动及测试程序的二进制文件复制到 Host 侧的/root/share 文件夹下。

最后，进入 Qemu 虚拟机内部，以相同的方法加载设备驱动，并运行测试程序以验证 edu 设备的阶乘功能。

注：在 Qemu 虚拟机中使用 9p 协议实现共享文件夹的详细描述可以参考 <https://wiki.qemu.org/Documentation/9psetup>。

6.4.2 本地环境

使用 scp 命令将前一个实验中编译好的 edu 设备驱动及测试程序的二进制文件复制到 Qemu 虚拟机内部，并以相同的方法加载设备驱动，随后运行测试程序以验证 edu 设备的阶乘功能。

6.5 确认必要程序已安装

在 Qemu 虚拟机内部确认 vim 和 bash 程序已经安装，并且默认登陆 shell 是 bash。

6.6 修改内核参数

查看 kernel.shmmax 内核参数。随后使用 sysctl 修改该内核参数，并确认该参数已被修改。

6.7 任务结束

至此，本实验的全部内容已结束。