

浙江大学



本科实验报告

姓名：

学院： 生物医学工程与仪器科学学院

系： 生物医学工程系

专业： 生物医学工程

学号：

指导教师： 周凡

2024 年 11 月 18 日

浙江大学实验报告

课程名称：嵌入式系统 实验类型：

实验项目名称：嵌入式系统的启动

同组学生姓名：

指导老师：田翔

实验地点：创客空间 实验日期：2024 年 11 月 18 日

一、实验目的和要求

通过本实验，掌握 Linux 内核的基本操作，学习内核编译及其在嵌入式系统中的应用。实验目标包括：

配置交叉编译工具链，编译适用于树莓派的 Linux 内核。

将自编译内核替换树莓派的默认内核，并完成启动测试。

探索交叉编译的概念和应用场景，深入理解 Linux 内核启动原理。

二、实验内容和原理

交叉编译是指在一个平台上生成适用于另一平台的可执行程序。通过编译适配树莓派架构的 Linux 内核，并将其替换原有内核，从而完成内核更新和启动。

安装交叉编译工具链：在 Ubuntu 环境中安装适用于 ARM 架构的编译器（aarch64-linux-gnu-gcc）。

配置内核并编译：下载 Linux 内核源码，解压并通过工具链对其进行编译。

替换内核镜像：将编译生成的内核镜像文件写入 SD 卡替代树莓派默认内核。

启动树莓派并验证内核版本：通过 SSH 远程登录树莓派，检查新的内核版本。

三、主要仪器设备

PC、树莓派开发板。

四、操作方法和实验步骤

1、安装交叉编译工具链

```
sudo apt-get install git bison flex libssl-dev zip libncurses-dev make
```

2、配置内核并编译

```
cd lab2_zju
tar -xvf gcc-linaro-7.5.0-2019.12-x86_64_aarch64-linux-gnu.tar.tar

tar -xvzf linux.tgz
cp make.sh mount.sh umount.sh linux/
cd linux
chmod +x make.sh mount.sh umount.sh

./make.sh 3
./make.sh 4
./make.sh 0
```

（这里需要把 make.sh 里面 make Image modules dtbs 改成 make -j8 Image modules dtbs，不然 4 个小时也编译不完。改完之后重新 make clean 然后重复）

3、让 WSL2 读取 SD 卡

将.wslconfig 文件修改为：

```
[wsl2]

kernel=c:\\users\\bubblevan\\bzImage
```

然后把.wslconfig 和 bzImage 放到 C://user/bubblevan 目录下。

```
// powershell
wsl --shutdown
wsl
```

```
winget install --interactive --exact dorssel.usbipd-win

// ubuntu
uname -a
sudo apt install linux-tools-generic hwdata
sudo update-alternatives --install /usr/local/bin/usbip usbip
/usr/lib/linux-tools/*-generic/usbip 20
```

将 SD 卡插入电脑。

```
// powershell(SP)
usbipd list
usbipd attach --wsl --busid 1-3
// ubuntu
lsblk
```

4、安装内核镜像

```
sudo mkdir /mnt/fat32
sudo mkdir /mnt/ext4
sudo ./mount.sh
./make.sh 6
./make.sh 7
vim /mnt/fat32/config.txt
```

在文中增加两行：

```
arm_64bit=1
kernel=kernel8.img
```

```
./umount.sh
```

5、启动树莓派

五、实验数据记录和处理

1、编译内核信息

2、内核版本验证

```
PS C:\Windows\system32> usbipd list
Connected:
BUSID  VID:PID    DEVICE                                STATE
1-6     048d:c988    USB 输入设备                          Not shared
1-11    04f2:b7b8    Integrated Camera, Tobii Experience   Not shared
1-13    14cd:1212    USB 大容量存储设备                    Not shared
1-14    8087:0033    英特尔(R) 无线 Bluetooth(R)         Not shared

Persisted:
GUID                                DEVICE
```

```
PS D:\home> usbipd list
Connected:
BUSID  VID:PID    DEVICE                                STATE
1-3     14cd:1212    USB 大容量存储设备                    Shared
1-6     048d:c988    USB 输入设备                          Not shared
1-11    04f2:b7b8    Integrated Camera, Tobii Experience   Not shared
1-14    8087:0033    英特尔(R) 无线 Bluetooth(R)         Not shared

Persisted:
GUID                                DEVICE

PS D:\home> usbipd attach --wsl --busid 1-3
usbipd: info: Using WSL distribution 'Ubuntu-24.04' to attach; the device will be available in all WSL 2 distributions.
usbipd: info: Using IP address 172.24.160.1 to reach the host.
```

```
bubblevan@Bubbles:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0 388.6M  1 disk
sdb          8:16   0    8G    0 disk [SWAP]
sdc          8:32   0    1T    0 disk /snap
                               /mnt/wslg/distro
                               /
sdd          8:48   1 14.8G   0 disk
└─sdd1       8:49   1  512M   0 part
└─sdd2       8:50   1 14.3G   0 part
```

```
bubblevan@Bubbles: ~/lab2  ×  +  v
rm: cannot remove '/mnt/ext4/lib/modules/4.19.127-v8+/kernel/fs/nls': Permission denied
rm: cannot remove '/mnt/ext4/lib/modules/4.19.127-v8+/kernel/fs/efivarfs': Permission denied
rm: cannot remove '/mnt/ext4/lib/modules/4.19.127-v8+/kernel/fs/reiserfs': Permission denied
rm: cannot remove '/mnt/ext4/lib/modules/4.19.127-v8+/kernel/fs/ocfs2/dlmfs': Permission denied
rm: cannot remove '/mnt/ext4/lib/modules/4.19.127-v8+/kernel/fs/ocfs2/dlm': Permission denied
rm: cannot remove '/mnt/ext4/lib/modules/4.19.127-v8+/kernel/fs/ocfs2/cluster': Permission denied
rm: cannot remove '/mnt/ext4/lib/modules/4.19.127-v8+/kernel/fs/overlayfs': Permission denied
rm: cannot remove '/mnt/ext4/lib/modules/4.19.127-v8+/kernel/fs/ntfs': Permission denied
rm: cannot remove '/mnt/ext4/lib/modules/4.19.127-v8+/kernel/fs/fuse': Permission denied
rm: cannot remove '/mnt/ext4/lib/modules/4.19.127-v8+/kernel/fs/jffs2': Permission denied
rm: cannot remove '/mnt/ext4/lib/modules/4.19.127-v8+/kernel/fs/jfs': Permission denied
rm: cannot remove '/mnt/ext4/lib/modules/4.19.127-v8+/kernel/fs/nfs': Permission denied
rm: cannot remove '/mnt/ext4/lib/modules/4.19.127-v8+/kernel/fs/xfs': Permission denied
rm: cannot remove '/mnt/ext4/lib/modules/4.19.127-v8+/kernel/crypto/async_tx': Permission denied
make: *** [Makefile:1261: _modinst_] Error 1
bubblevan@Bubbles:~/lab2_zju/linux$ sudo ./make.sh 6
INSTALL arch/arm64/crypto/aes-arm64.ko
INSTALL arch/arm64/crypto/aes-neon-blk.ko
INSTALL arch/arm64/crypto/aes-neon-bs.ko
INSTALL crypto/af_alg.ko
INSTALL crypto/algif_skcipher.ko
INSTALL crypto/arc4.ko
INSTALL crypto/async_tx/async_memcpy.ko
INSTALL crypto/async_tx/async_pq.ko
INSTALL crypto/async_tx/async_raid6_recov.ko
INSTALL crypto/async_tx/async_tx.ko
INSTALL crypto/async_tx/async_xor.ko
INSTALL crypto/authenc.ko
INSTALL crypto/authencasn.ko
INSTALL crypto/cast5_generic.ko
```

直接运行果然出错了，加上 sudo 提升权限就好了。

3、挂载与卸载日志

```
cp: cannot create regular file '/mnt/fat32/overlays/README': Permission denied
bubblevan@Bubbles:~/lab2_zju/linux$ sudo ./make.sh 7
mkdir: missing operand
Try 'mkdir --help' for more information.
bubblevan@Bubbles:~/lab2_zju/linux$ sudo ./make.sh 7
[sudo] password for bubblevan:
bubblevan@Bubbles:~/lab2_zju/linux$ vim /mnt/fat32/config.txt
bubblevan@Bubbles:~/lab2_zju/linux$ sudo vim /mnt/fat32/config.txt
bubblevan@Bubbles:~/lab2_zju/linux$ ./umount.sh
umount: /mnt/fat32: must be superuser to unmount.
umount: /mnt/ext4: must be superuser to unmount.
bubblevan@Bubbles:~/lab2_zju/linux$ sudo ./mount.sh
mount: /mnt/fat32: /dev/sdd1 already mounted on /mnt/fat32.
dmesg(1) may have more information after failed mount system call.
mount: /mnt/ext4: /dev/sdd2 already mounted on /mnt/ext4.
dmesg(1) may have more information after failed mount system call.
bubblevan@Bubbles:~/lab2_zju/linux$ uname -a
Linux Bubbles 5.15.133.1-microsoft-standard-WSL2 #2 SMP Tue Oct 10 00:21:20 CST 2023 x86_64 x86_64 x86_64 GNU/Linux
bubblevan@Bubbles:~/lab2_zju/linux$

bubblevan@raspberrypi:~$ uname -a
Linux raspberrypi 4.19.127-v8+ #1 SMP PREEMPT Tue Nov 19 16:43:14 CST 2024 aarch64 GNU/Linux
```

可以发现内核版本号已经发生改变。

六、实验结果与分析

实验结果显示，经过交叉编译和镜像替换后，树莓派成功启动了新的内核版本。uname -a 输出验证了版本更新，同时系统运行稳定，说明内核镜像替换操作成功。

七、讨论、心得

1、Git 学习

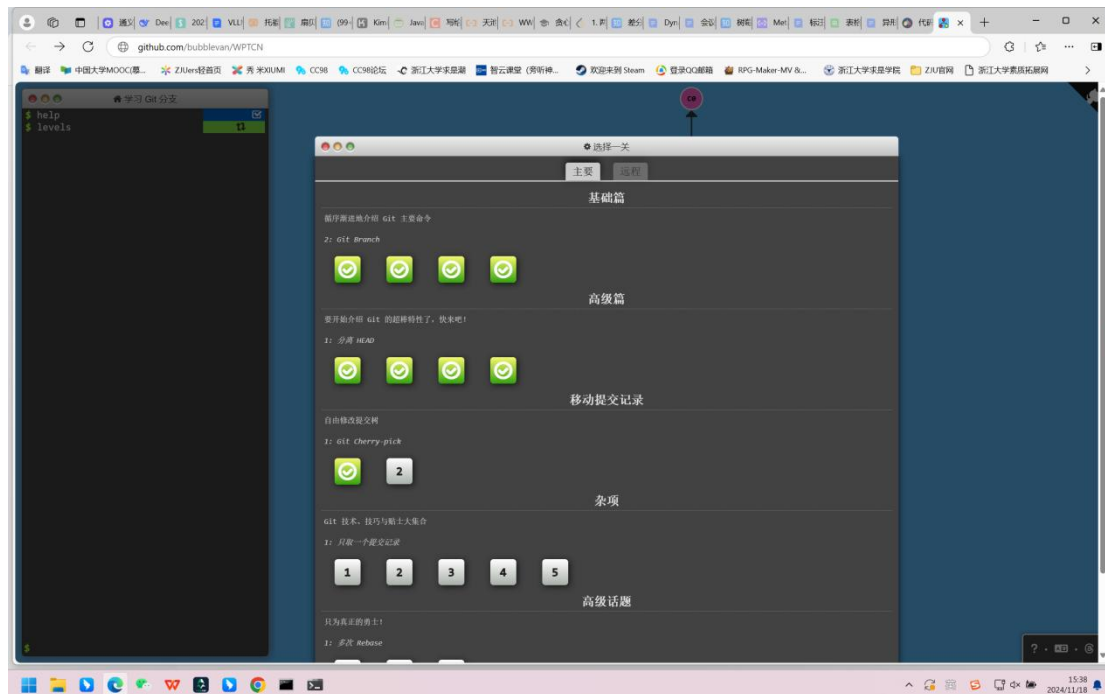
Git 是一个分布式版本控制系统，用于跟踪文件的变化，协调多人协作开发，并管理项目的版本历史。

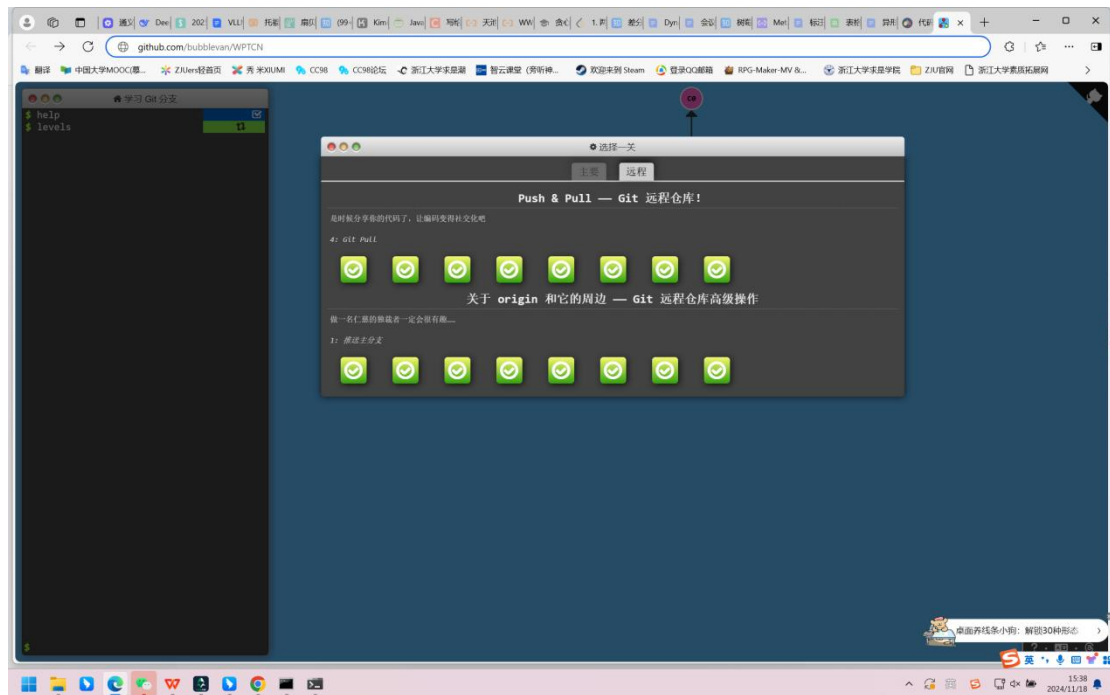
命令	作用
git init	初始化一个新的 Git 仓库，创建一个 .git 目录。
git clone <url>	从远程仓库克隆一个项目到本地。
git add <file>	将文件添加到暂存区，准备提交。

<code>git commit -m "message"</code>	提交暂存区的文件到本地仓库，并附上提交信息。
<code>git status</code>	查看工作区和暂存区的状态，显示哪些文件被修改、添加或删除。
<code>git log</code>	查看提交历史，显示提交的详细信息。
<code>git branch</code>	列出所有分支，当前分支前面会有 * 标记。
<code>git branch <branch></code>	创建一个新的分支。
<code>git checkout <branch></code>	切换到指定的分支。
<code>git merge <branch></code>	将指定分支的修改合并到当前分支。
<code>git pull</code>	从远程仓库拉取最新的修改并合并到当前分支。
<code>git push</code>	将本地仓库的修改推送到远程仓库。
<code>git remote -v</code>	查看远程仓库的详细信息，显示远程仓库的 URL。
<code>git fetch</code>	从远程仓库获取最新的修改，但不合并到当前分支。
<code>git diff</code>	显示工作区和暂存区之间的差异。
<code>git reset <file></code>	将文件从暂存区移除，但不改变工作区的内容。
<code>git reset --hard <commit></code>	回滚到指定的提交，丢弃之后的所有修改。
<code>git stash</code>	将当前工作区的修改暂存起来，以便切换分支或清理工作区。
<code>git stash pop</code>	将暂存的修改应用到当前工作区，并从暂存列表中移除。
<code>git tag <tagname></code>	创建一个标签，标记某个特定的提交。
<code>git show <commit></code>	显示指定提交的详细信息，包括修改内容。

<code>git rm <file></code>	从工作区和暂存区中删除文件。
<code>git mv <old> <new></code>	重命名文件或移动文件到新的路径。

下面是 Learn Git Branching 的通关截图，实际上我大一就开始使用 Git 来学习一些经典的开源项目了，两年学习下来给人的感觉是不需要把 Git 学的面面俱到，所以下面我第一面的关卡只做了基础部分，但是第二面团队协作的关卡全部完成。本机证据的话，一个是 edge 浏览器多到离谱的网站卡片，一个是搜索栏里自己的 github 仓库地址。





2、思考题

(1) 如果你发现解压

`gcc-linaro-7.5.0-2019.12-x86_64_aarch64-linux-gnu.tar.tar` 失败了，请思考一下如何解决？（提示：和文件所有权有关，`ls -l` 命令 和 `chown` 命令：
`sudo chown -R wendell:wendell ../lab2_zju`）

文件的所有权和权限可能会影响解压操作的执行。所以假如我真的解压失败了，我就得用 `ls -l` 命令查看文件的详细信息（包括文件的所有者和权限），然后用 `chown` 命令将文件的所有权从 `wendell` 修改为 `bubblevan` 了。

(2) `tar` 命令是干什么的？如何使用？

`tar` 命令用于创建、查看和提取 `tar` 归档文件。`tar` 文件通常用于将多个文件和目录打包成一个单独的文件，便于传输和备份。

- c: 创建一个新的 `tar` 文件。
- x: 提取 `tar` 文件中的内容。
- v: 显示详细信息 (verbose)。
- f: 指定 `tar` 文件的名称。
- z: 使用 `gzip` 压缩或解压缩 `tar` 文件。

示例：

创建 tar 文件: `tar -cvf archive.tar file1 file2 dir1`

提取 tar 文件: `tar -xvf archive.tar`

创建并压缩 tar.gz 文件: `tar -czvf archive.tar.gz file1 file2 dir1`

解压缩 tar.gz 文件: `tar -xzvf archive.tar.gz`

(3) `chmod` 是干什么的? 为什么要执行这行命令?

`chmod` 命令用于修改文件或目录的权限。权限决定了哪些用户可以读取、写入或执行文件。

(4) 学习 shell 基础语法, 阅读 `make.sh`, 回答这三条命令实际上是执行了哪几条命令?

`./make.sh 3`: 这个命令会执行脚本中的 `make bcm2711_defconfig` 命令。这个命令通常用于设置交叉编译的默认配置, `bcm2711_defconfig` 可能是针对特定硬件平台 (如 Raspberry Pi 4) 的配置文件。

`./make.sh 4`: 这个命令会执行 `make menuconfig` 命令。这是一个交互式的配置工具, 允许用户通过文本界面选择和配置内核选项。当用户完成配置并退出 `menuconfig` 后, 脚本会因为 `exit` 命令而终止, 不会执行任何后续命令。

`./make.sh 0`: 这个命令会执行 `make Image modules dtbs` 命令。这通常意味着脚本会编译内核镜像 (Image)、模块 (modules) 以及设备树二进制文件 (dtbs): 这些是内核构建过程中的常见产物。

(5) `mount.sh` 执行了什么命令? `mount` 的作用是什么?

`mount` 命令用于将文件系统挂载到指定的目录。在实验中, `mount` 命令将 SD 卡的两个分区分别挂载到 `/mnt/fat32` 和 `/mnt/ext4` 目录, 以便后续操作。

(6) 本次实验是更换了树莓派上的 Linux 内核, 那么这两个内核除了版本还有什么区别呢? (提示: `armv7l` & `aarch64`)

`armv7l` 是 32 位 ARM 架构, 适用于较旧的 ARM 处理器; `aarch64` 则是 64 位 ARM 架构, 适用于较新的 ARM 处理器。主要区别包括:

- **性能**: `aarch64` 提供更高的性能和更大的内存寻址能力。
- **应用程序**: `aarch64` 支持更复杂的应用程序和更大的内存需求。
- **兼容性**: `armv7l` 可能与一些旧的硬件和软件兼容性更好, 而 `aarch64` 则更适合现代硬件和软件。

(8) 这个实验为什么要交叉编译? (提示: 可以从开发环境 (你的电脑) 和运

行环境（树莓派）的处理器架构差异的角度回答。）如果你用的是 M 系列处理器的苹果电脑，需要交叉编译吗？

交叉编译是因为开发环境和运行环境的处理器架构不同。开发环境我们的 PC，也即 x86 架构，而树莓派使用的是 ARM 架构。为了在 PC 上编译出适用于 ARM 架构的 Linux 内核，需要使用交叉编译工具链。而 M 系列处理器也是 ARM 架构，理论上不需要交叉编译，但没试过我也不太清楚。

3、实验感受

本实验展示了交叉编译在异构系统开发中的重要性。由于 PC 和树莓派的处理器架构不同，需使用工具链生成 ARM 架构可运行的内核镜像。

通过此次实验，我学会了 Linux 内核配置与编译的基本操作，为后续内核裁剪和优化打下基础。

八、代码附件

```
#!/bin/bash
# make.sh

export ARCH=arm64
#cross compile path
export CROSS_COMPILE=../gcc-linaro-7.5.0-2019.12-x86_64_aarch64-linux-gnu/bin/aarch64-linux-gnu-
if [ $# -lt 1 ]; then
    echo "usage $0 op (0:image 1:clean 2:distclean 3:defconfig 4:menuconfig 5:tags 6:install modules 7:install
kernel)"
    exit
fi
if [ $1 -eq 0 ]; then
    make -j8 Image modules dtbs
elif [ $1 -eq 1 ]; then
    make clean
elif [ $1 -eq 2 ]; then
    make distclean
elif [ $1 -eq 3 ]; then
    make bcm2711_defconfig
elif [ $1 -eq 4 ]; then
    make menuconfig
elif [ $1 -eq 5 ]; then
    make tags
elif [ $1 -eq 6 ]; then
    #modules path[../mnt/ext4] needs to be changed by your own path
    make INSTALL_MOD_PATH=/mnt/ext4/ modules_install
```

```
elif [ $1 -eq 7 ]; then
    #boot path[../mnt/fat32] needs to be changed by your own path
    #backup kernel
    cp /mnt/fat32/kernel8.img /mnt/fat32/kernel8.img.bak
    #update kernel
    cp arch/arm64/boot/Image /mnt/fat32/kernel8.img
    cp arch/arm64/boot/dts/broadcom/*.dtb /mnt/fat32/
    cp arch/arm64/boot/dts/overlays/*.dtb* /mnt/fat32/overlays/
    cp arch/arm/boot/dts/overlays/README /mnt/fat32/overlays
fi
```

```
#!/
# mount.sh

# mount path need to be changed by your own path
# executed by sudo cmd

mount /dev/sdd1 /mnt/fat32
mount /dev/sdd2 /mnt/ext4
```