

《嵌入式系统》实验报告 3

学号：21013134 姓名：徐昊博 班级：计 213 日期：2024 年 5 月 24 日

成绩：_____

指导教师：罗飞

实验名称：Linux 内核编译实验	实验地点：
实验仪器：ECS 云计算环境（提供 Linux 实验环境）	
<p>一、实验目的：</p> <ul style="list-style-type: none">1、了解 Linux 内核源代码的目录结构以及各目录的相关内容2、了解 Linux 内核各配置选项内容和作用3、掌握 Linux 内核配置文件 config.in 的作用4、掌握 Linux 内核的编译过程5、掌握将新增内核代码加入到 Linux 内核结构中的方法	
<p>二、实验内容：</p> <p>完成实验指导手册实验，进而完成如下内容：</p> <p>1.简述基于内核源码编译嵌入式 Linux 内核的步骤</p> <p>（1）安装交叉编译环境，即安装 toolchain</p> <p>建立工作目录：</p> <pre>ecust@Ubuntu mkdir /home/ecust/workplace</pre> <pre>ecust@Ubuntu cd /home/ecust/workplace</pre> <p>ToolChain 安装：</p> <pre>ecust@Ubuntu ~/workplace\$ tar jxvf /home/ecust/samba_share/embed/ToolChain/toolchain-4.5.1-farsight.tar.bz2</pre> <pre>ecust@Ubuntu ~/workplace\$ ln -s toolchain-4.5.1-farsight toolchain</pre> <p>修改 bashrc 文件：</p> <pre>ecust@Ubuntu ~/workplace\$ vim ~/.bashrc</pre> <p>在该文件的末尾输入：</p> <pre>export PATH=/home/ecust/workplace/toolchain/bin:\$PATH</pre> <p>保存退出后运行如下指令：</p> <pre>source ~/.bashrc</pre> <p>（2）获取及解压内核源码</p> <pre>ecust@Ubuntu:cd workplace</pre> <p>使用 tar 命令将内核源码解压到当前目录，并通过 cd 命令进入内核源码目录</p> <pre>ecust@Ubuntu ~/workplace\$ tar -xjf /home/ecust/samba_share/embed/Linux/linux-3.2.tar.bz2 -C ~/workplace</pre> <p>（3）修改内核编译参数，并编译内核</p> <p>修改内核顶层目录下的 Makefile</p> <pre>ecust@Ubuntu :~/workplace/linux-3.2\$ vim Makefile</pre> <p>修改：</p> <pre>ARCH?= \$(SUBARCH)</pre> <pre>CROSS_COMPILE?= \$(CONFIG_CROSS_COMPILE:"%"=%)</pre> <p>其中，目标板结构为 arm，所使用的交叉编译器为 arm-none-linux-gnueabi-gcc;</p>	

则将上述配置改为：

`ARCH ?= arm`

`CROSS_COMPILE ?= arm-none-linux-gnueabi-`

拷贝标准板配置文件

`ecust@Ubuntu :~/workplace/linux-3.2$ cp arch/arm/configs/s5pv210_defconfig .config`

配置内核

`ecust@Ubuntu :~/workplace/linux-3.2$ make menuconfig`

当前的 CPU 类型：

Intel Core Processor (BoardWell)

内核编译得到的内核文件有哪些？分别在什么目录？

内核映像文件：

文件名：通常为 `zImage` 或 `uImage`（取决于你使用的编译选项）。

位置：位于内核源码目录的 `arch/arm/boot` 目录下。

`arch/arm/boot/zImage`

设备树文件：

文件名：以 `.dtb` 为后缀。

位置：位于内核源码目录的 `arch/arm/boot/dts` 目录下。

`arch/arm/boot/dts/*.dtb`

内核模块文件：

文件名：以 `.ko` 为后缀。

位置：位于内核源码目录的 `modules` 目录下（通过 `make modules_install` 安装到 `/lib/modules/`）。

`/lib/modules/<kernel-version>/`

`System.map` 文件：

文件名：`System.map`

位置：位于内核源码目录。

`System.map`

内核配置文件：

文件名：`.config`

位置：位于内核源码目录。

`.config`

三、完成思考题

1、分析 `make config`、`make menuconfig`、`make xconfig` 三个 linux 内核配置界面的区别

在 Linux 内核的配置过程中，`make config`、`make menuconfig`、和 `make xconfig` 提供了三种不同的配置界面，它们的主要区别如下：

1. make config

`make config` 是文本模式配置工具。用户会被依次询问每一个配置选项，回答“是/否/模块”(Y/N/M)。适合快速、简单的配置，但如果内核选项很多，会显得比较繁琐和耗时。界面非常简单，仅在终端显示问题并接受输入。

2. make menuconfig

`make menuconfig` 基于文本的菜单配置工具。提供一个文本界面的菜单，通过箭头键导航，空格键选择。允许用户方便地浏览和修改配置选项。需要 `ncurses` 库支持，提供较为直观的菜单界面。

3. make xconfig

`make xconfig` 是基于图形的配置工具。提供图形用户界面（GUI），通过窗口和按钮进行配置。需要 `Qt` 或 `GTK` 库支持，提供最直观和易用的配置界面。

2、指出 linux 内核编译命令 `make`, `make zImage`, `make bzImage` 的区别

1、make

是通用的内核编译命令。用于编译整个内核及其模块且生成默认的内核镜像和模块。

2、make zImage

是生成压缩的内核镜像文件，适用于较小内存空间的系统。内核启动时先解压，然后加载到内存中运行。

3、make bzImage

生成大内核的压缩镜像文件。`bzImage` 适用于较大内存空间和复杂内核配置的系统。内核启动时解压并加载到高内存区域运行。