

同济大学计算机系  
计算机系统实验报告



实验三题目：嵌入式 Linux 系统内核编译和改造

学 号： 2154312

姓 名： 郑博远

指导教师： 秦国锋

日期： 2024 年 5 月 22 日

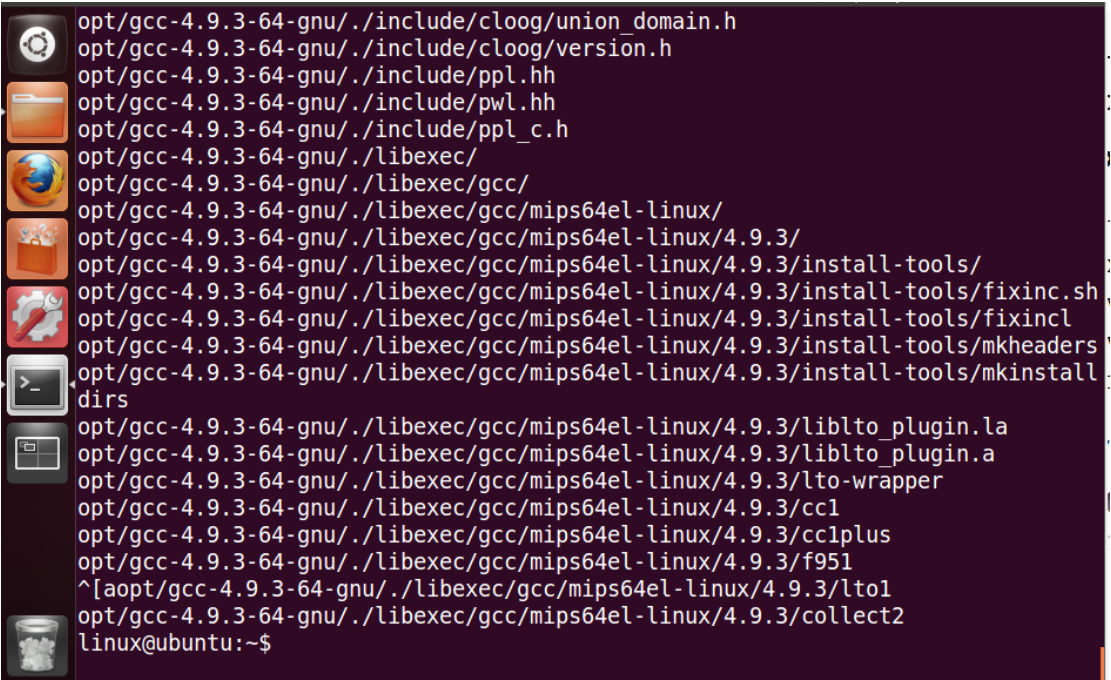
## 一、实验环境

虚拟机	VMware® Workstation 17
Linux 镜像	Ubuntu 12.04.1 LTS

## 二、实验目的

本次实验需利用第二次实验中配置的 pmon 引导程序完成 Linux-3.10.0 内核的编译，使之能够在 Nexys 4 DDR Artix-7 FPGA 开发板上运行。此外还需掌握嵌入式开发操作系统内核编译方法，从而为后续 RISC-V 指令集计算机系统做前置准备。

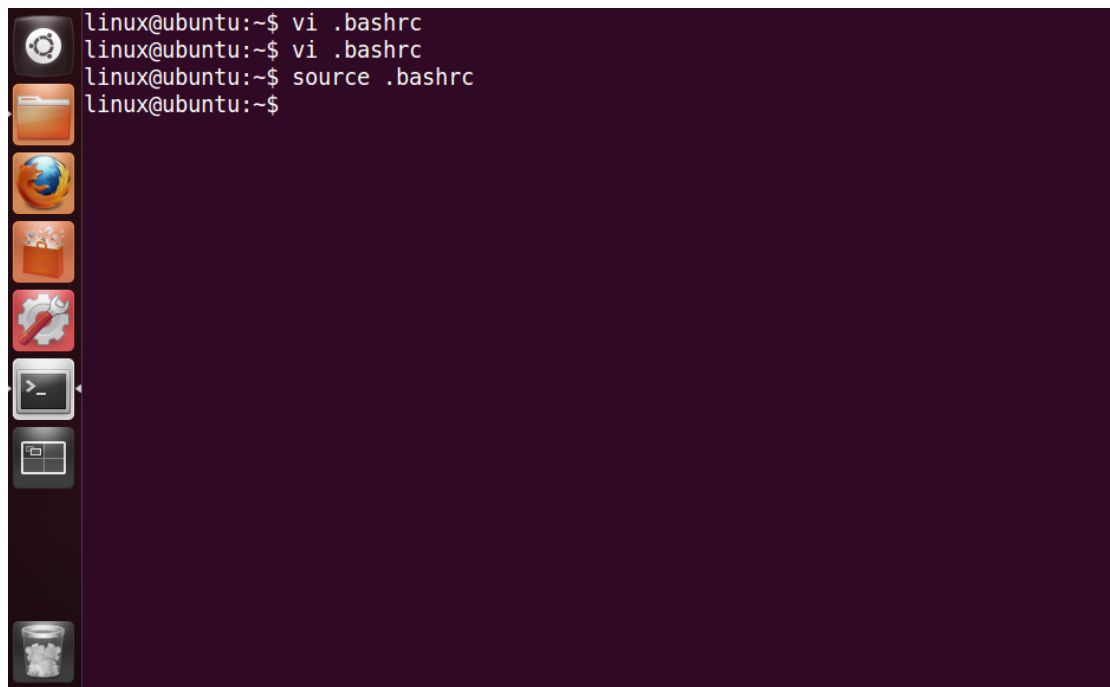
## 三、实验过程及内容

A terminal window with a dark purple background and white text. On the left side, there is a vertical toolbar with icons for various file managers and applications. The terminal text lists a series of file paths for the gcc-4.9.3-64-gnu package, including include files, libraries, and installation scripts. The paths are organized into a tree-like structure, starting from the root directory of the package and branching out to specific subdirectories and files. The paths end with a prompt 'linux@ubuntu:~\$'.

将 gcc-4.9.3-64-gnu 放在 /opt 目录下并解压。



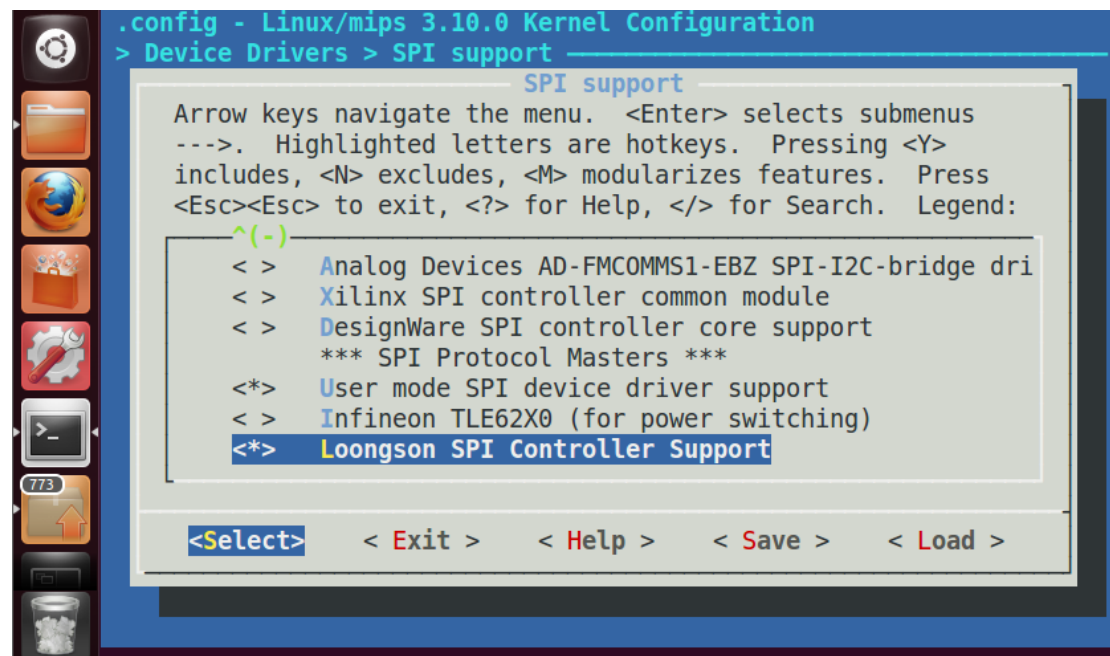
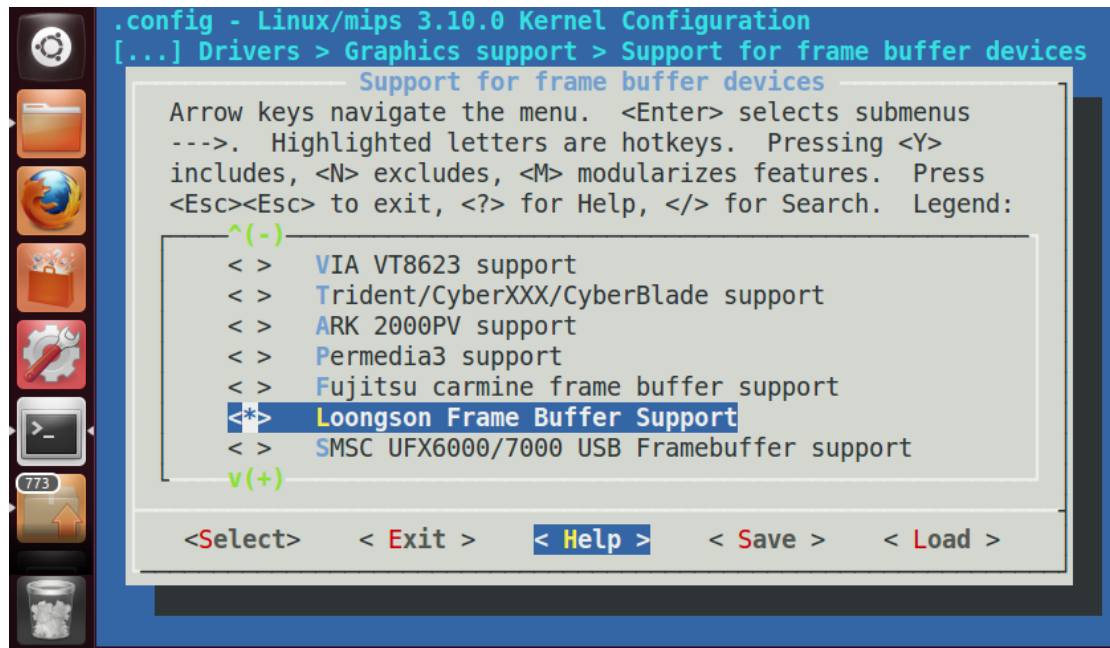
```
97
98 if [ -f ~/.bash_aliases ]; then
99     . ~/.bash_aliases
100 fi
101
102 # enable programmable completion features (you don't need to enable
103 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
104 # sources /etc/bash.bashrc).
105 if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
106     . /etc/bash_completion
107 fi
108
109 export PATH=/opt/gcc-4.9.3-64-gnu/bin:$PATH
```

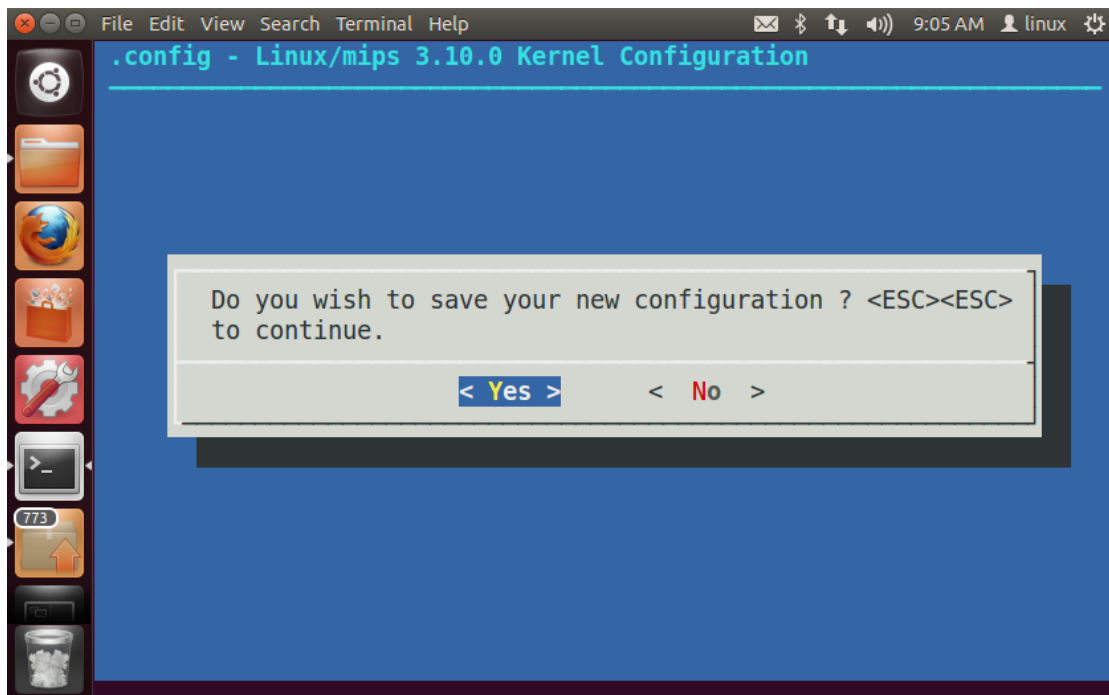
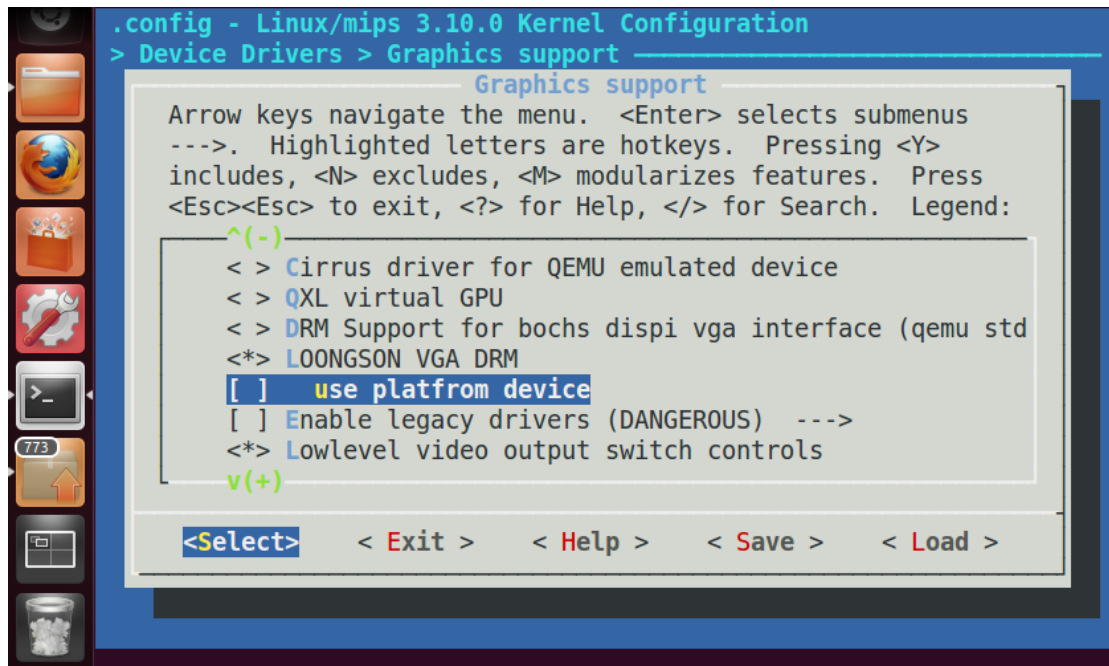


```
linux@ubuntu:~$ vi .bashrc
linux@ubuntu:~$ vi .bashrc
linux@ubuntu:~$ source .bashrc
linux@ubuntu:~$
```

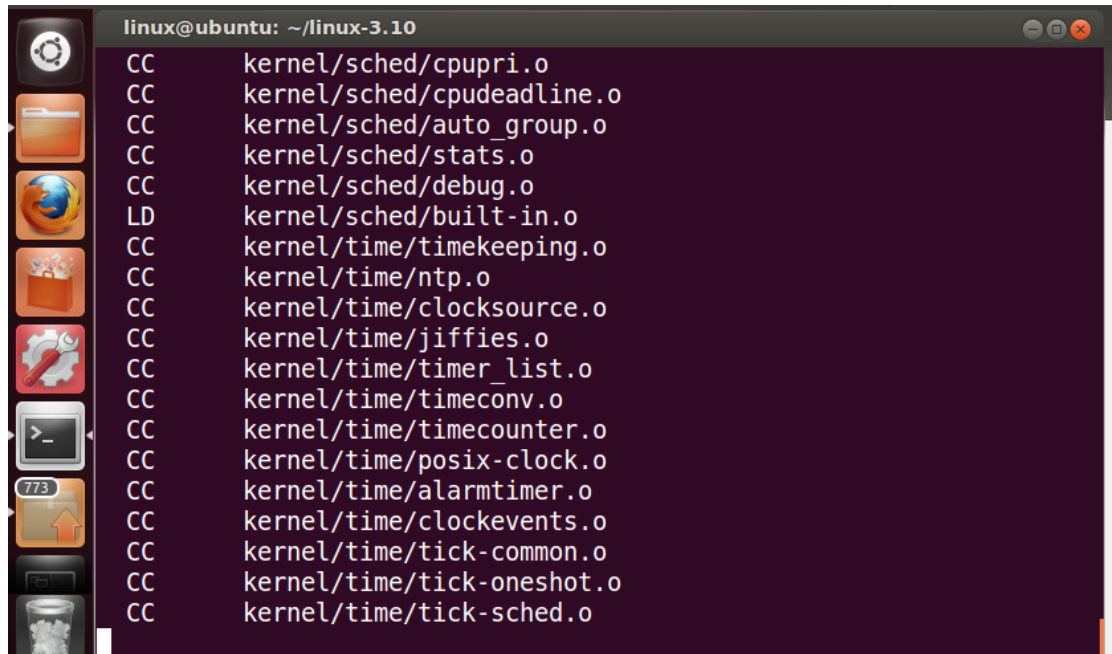
vi 进入.bashrc 文件中，在文件末尾添加“export PATH=/opt/gcc-4.9.3-64-gnu/bin:\$PATH”，将 opt 目录下的 gcc-4.9.3-63-gnu 添加到路径中。此后进行 source .bashrc。打开终端会默认执行该命令，使得 gcc-4.9.3-64-gnu 为路径最前的编译器。

在 linux-3.10 的目录下执行：make ARCH=mips CROS  
S\_COMPILE=mips64el-linux- menuconfig，进行编译配置。  
退出时保存对应配置。





执行 `make ARCH=mips CROSS_COMPILE=mips64el-linux-`，用于生成 `vmlinuz` 内核镜像（指令执行时间较长）：



```
linux@ubuntu: ~/linux-3.10
CC      kernel/sched/cpupri.o
CC      kernel/sched/cpudeadline.o
CC      kernel/sched/auto_group.o
CC      kernel/sched/stats.o
CC      kernel/sched/debug.o
LD      kernel/sched/built-in.o
CC      kernel/time/timekeeping.o
CC      kernel/time/ntp.o
CC      kernel/time/clocksource.o
CC      kernel/time/jiffies.o
CC      kernel/time/timer_list.o
CC      kernel/time/timeconv.o
CC      kernel/time/timecounter.o
CC      kernel/time/posix-clock.o
CC      kernel/time/alarmtimer.o
CC      kernel/time/clockevents.o
CC      kernel/time/tick-common.o
CC      kernel/time/tick-oneshot.o
CC      kernel/time/tick-sched.o
```



```
linux@ubuntu: ~/linux-3.10
LD [M]  sound/pci/hda/snd-hda-codec-conexant.ko
CC      sound/pci/hda/snd-hda-codec-generic.mod.o
LD [M]  sound/pci/hda/snd-hda-codec-generic.ko
CC      sound/pci/hda/snd-hda-codec-hdmi.mod.o
LD [M]  sound/pci/hda/snd-hda-codec-hdmi.ko
CC      sound/pci/hda/snd-hda-codec-idt.mod.o
LD [M]  sound/pci/hda/snd-hda-codec-idt.ko
CC      sound/pci/hda/snd-hda-codec-realtek.mod.o
LD [M]  sound/pci/hda/snd-hda-codec-realtek.ko
CC      sound/pci/hda/snd-hda-codec-si3054.mod.o
LD [M]  sound/pci/hda/snd-hda-codec-si3054.ko
CC      sound/pci/hda/snd-hda-codec-via.mod.o
LD [M]  sound/pci/hda/snd-hda-codec-via.ko
CC      sound/pci/hda/snd-hda-codec.mod.o
LD [M]  sound/pci/hda/snd-hda-codec.ko
CC      sound/pci/hda/snd-hda-intel.mod.o
LD [M]  sound/pci/hda/snd-hda-intel.ko
CC      sound/pci/hda/snd-hda-loongson.mod.o
LD [M]  sound/pci/hda/snd-hda-loongson.ko
linux@ubuntu:~/linux-3.10$
```

执行 `make ARCH=mips CROSS_COMPILE=mips64el-linux-modules_install`，在虚拟机 `/lib/modules/` 文件夹生成 2k 内核模块。

```
linux@ubuntu: ~/linux-3.10
INSTALL /lib/firmware/ti_5052.fw
INSTALL /lib/firmware/mts_cdma.fw
INSTALL /lib/firmware/mts_gsm.fw
INSTALL /lib/firmware/mts_edge.fw
INSTALL /lib/firmware/edgeport/boot.fw
INSTALL /lib/firmware/edgeport/boot2.fw
INSTALL /lib/firmware/edgeport/down.fw
INSTALL /lib/firmware/edgeport/down2.fw
INSTALL /lib/firmware/edgeport/down3.bin
INSTALL /lib/firmware/whiteheat_loader.fw
INSTALL /lib/firmware/whiteheat.fw
INSTALL /lib/firmware/keyspan_pda/keyspan_pda.fw
INSTALL /lib/firmware/keyspan_pda/xircom_pgs.fw
INSTALL /lib/firmware/bnx2/bnx2-mips-09-6.2.1a.fw
INSTALL /lib/firmware/bnx2/bnx2-rv2p-09-6.0.17.fw
INSTALL /lib/firmware/bnx2/bnx2-rv2p-09ax-6.0.17.fw
INSTALL /lib/firmware/bnx2/bnx2-mips-06-6.2.1.fw
INSTALL /lib/firmware/bnx2/bnx2-rv2p-06-6.0.15.fw
DEPMOD 3.10.0
linux@ubuntu:~/linux-3.10$
```

观察到对应文件夹已经生成模块，`vmlinuz` 打包即完成实验。

```
linux@ubuntu: /lib/modules/3.10.0
linux@ubuntu:/lib/modules$ cd 3.10.0
linux@ubuntu:/lib/modules/3.10.0$ ls
build          modules.dep.bin      modules.seriomap
kernel         modules.devname      modules.softdep
modules.alias  modules.ieee1394map  modules.symbols
modules.alias.bin  modules.inputmap    modules.symbols.bin
modules.builtin  modules.isapnpmap   modules.usbmap
modules.builtin.bin  modules.ofmap       source
modules.ccwmap    modules.order
modules.dep       modules.pcimap
linux@ubuntu:/lib/modules/3.10.0$
```



## 四、实验小结

本次实验的核心任务是编译嵌入式开发操作系统。实验首先需要准备操作系统源代码和相应的编译工具，其次需要设置编译环境，以确保编译过程能够顺利在正确的工具链中进行。此后还要进行执行编译操作，生成内核代码。最后还需要对生成的内核代码进行必要的处理，为后续的系统移植工作做好准备。通过这次实验，我对嵌入式系统的编译过程有了更深入的理解。每个步骤都需要精确执行，任何小的疏忽都可能导致整个编译过程失败。这个过程虽然复杂，但也让我对嵌入式系统开发有了更深的认识，希望在不断的实验过程中能学习知识、继续进步！