

**操作系统课程设计**

|  |  |
| --- | --- |
| **实验一、编译Linux内核** | **Experiment 1, Compiling the Linux Kernel** |

学院：计算机学院

专业：计算机科学与技术

学生姓名：夏奇拉

学号：1820171025

班级：07111705

|  |  |
| --- | --- |
| **实验目的** | **Experiment Purpose** |
| 在内核中加入本人学号，具体做法：  修改makefile文件中有字段EXTRAVERSION， 通过修改它就可以使得自己编译的linux内核版本信息改变。使用文本编辑器即可进行修改。  样例：  EXTRAVERSION = -1120131860  注意：不要忘记学号前面的短横线(半角) | The experiment asks that one  1. Compiles their own Linux kernel  2. adds their student ID to the kernel by modifying the EXTRAVERSION field in the makefile.  Example:  EXTRAVERSION = -1120131860  Note: Don't forget the dash in front of the student number  \*Modifications can be made using a text editor. |

|  |  |
| --- | --- |
| **实验内容** | Experiment Content |
| 1. 从源码编译安装Linux内核  2. 修改内核，使其运行时版本字符串中包含字符“=1820171025”  术语定义  Linux：是一种操作系统，一种控制系统硬件组件的软件。 操作系统由引导加载程序、内核、网络、外壳等组成。  内核是系统的核心，管理着 CPU、内存和外围设备。 它是用 C 和汇编编程语言编写的。  Source 是指内核源代码，可以在 kernel.org 上找到，kernel.org 是由 Linux Kernel Organization 运营的网站。  自己编译 Linux Kernel 的原因：   * 了解它是如何完成的 * 启用标准内核中没有的硬件支持 * 启用在标准内核中找不到的选项 * 每次实现自己的系统调用或修改内核源代码时； 您将需要重新编译内核以实现更改。   从源代码编译和安装 Linux 内核涉及   * 从 kernel.org 下载最新的内核 * 使用 apt-get 安装所需的软件包 * 配置内核和模块 * 编译和构建内核 * 更新 grub 配置 * 重新启动系统 | 1. Compile and install Linux kernel from source  2. Modify the kernal so that running it has the characters “=1820171025” in the version string  Definition of terms  Linux: is an operating system, a piece of software that controls the hardware components of a system. An operating system consists of a bootloader, kernel, networking, shell, etc.  The Kernel is the core of the system and manages the CPU, memory and peripheral devices. It is written in C and Assembly programming languages.  Source refers to the kernel source code which can be found at kernel.org, a website run by the Linux Kernel Organization.  Reasons for compiling the Linux Kernel yourself:   * to learn how it's done * to enable hardware support not found in the standard kernel * to enable options not found in the standard kernel * each time you implement your own system call or modify kernel source code; you will need to recompile the kernel to implement the changes.   Compiling and installing the Linux kernel from source involves   * Downloading the latest kernel from kernel.org * Installing required packages using apt-get * Configuring the kernel and modules * Compiling and building the kernel * Updating the grub configuration * Rebooting the system |

|  |  |
| --- | --- |
| **实验环境** | Experiment **Environment** |
| 先决条件：  基于 Linux 的操作系统  12 GB 可用磁盘空间  用于下载源代码的 Internet 连接  还有很多时间，至少90分钟 | Prerequisites:   * A Linux based operating system * 12 GB of free disk space * An internet connection to download the source code * And a lot of time, at least 90 minutes |

|  |  |
| --- | --- |
| **操作方法和实验步骤**  **(程序设计与实现)** | **Operating Methods and Experimental Steps**  **(Program design and Implementation)** |
| **第一步：下载源代码**  Linux 内核源代码可以在 kernel.org 上找到。 这里我们可以通过点击“Latest Release”来获取内核的URL。    网址：https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.19.8.tar.xz  然后我们打开终端，使用 wget 命令下载 Linux 内核源代码：  wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.19.8.tar.xz  该命令的输出显示“下载完成时已保存的消息。  注意：Wget 是一个免费的 GNU 命令行实用工具，用于从 Internet 下载文件。  **第二步：提取源代码**  文件准备好后，运行 tar 命令提取源代码。  tar xvf linux-5.19.8.tar.xz  输出显示提取的内核源代码。    注意：Tar 命令是磁带存档的缩写，用于创建、提取和维护存档（压缩）文件。语法采用三个参数：  $ tar [选项] [归档文件] [要归档的目录/文件]  Xvf 是 Unix 风格的简短方法，用于实现 –extract –verbose –file。  X = 提取档案，  V = 显示详细信息  F = 指定文件名  详细信息提供有关正在执行的任务的其他详细信息。  **第 3 步：安装所需的软件包**  为了成功编译内核，需要特定的软件包。要安装这些软件包，我们运行以下命令：  sudo apt-get install git fakeroot build-essential ncurses-dev-xz-utils libssl-dev bc flex libelf-dev bison    注意：APT（Advanced Package Tool）是用于查找和安装新包、升级包、清理包等的命令行工具...  apt-get 用于安装、更新软件包。 IT 处理可用软件包的数据库。  我们使用的命令安装以下软件包：  混帐  跟踪并记录源代码开发过程中的所有更改。 它还允许还原更改。  假根  制作假根环境的打包工具。  构建必备  安装开发工具，例如 C、C++、gcc 和 g++。  ncurses-开发  为基于文本的终端提供 API 的编程库。  xz-utils  提供快速的文件压缩和解压缩。  libssl-开发  支持加密数据并确保互联网连接安全的 SSL 和 TSL。  bc（基本计算器）  一种支持交互式执行语句的数学脚本语言。  flex（快速词法分析器生成器）  生成将字符转换为标记的词法分析器。  自由开发  发布用于管理 ELF 文件（可执行文件、核心转储和目标代码）的共享库  野牛  将语法描述转换为 C 程序的 GNU 解析器生成器。  **第 4 步：配置和编译**  在编译内核之前，我们需要配置哪些模块要包含，哪些模块要省略。  有很多方法可以做到这一点。  一种简单直接的方法是首先复制现有的内核配置文件，然后使用“menuconfig”进行更改（如有必要）。 这是最快的方法，也可能是最安全的方法。  导航到 linux-5.9.6。 使用 cd 命令的目录：  cd linux-5.9.6  cp /boot/config-$(uname -r) .config  Make menuconfig  该命令启动几个脚本，然后打开配置菜单：    在这种情况下，我们不会对配置菜单进行任何更改，因此我们退出菜单。  注意：使用 cp 命令可以在不离开 Linux 终端的情况下执行复制和粘贴操作。该命令的语法是：    cp [...文件/目录源] [目标]    [file/directory-sources] 指定要复制的文件或目录的来源  [destination] 参数指定要将文件复制到的位置。  $(uname –r) 用于命令替换。它运行 uname –r 命令，该命令返回当前内核版本，然后将返回的内容放入 cp 命令中。  命令替换允许命令的输出替换命令本身。当命令包含如下时，会发生命令替换：  $（命令）  或者  `命令`  “make”命令可帮助系统管理员编译和安装开源实用程序。它执行或使用 makefile 中指定的参数来识别处理目标项目的不同操作。  **第 5 步：更新 makefile 中的 EXTRAVERSION**  我们通过使用文本编辑器打开位于源代码根目录中的内核的 makefile 来做到这一点。  现在我们将 EXTRAVERSION 的值更改为 –1820171025。  -外向 =  +EXTRAVERSION = -1820171025  现在我们可以重新编译我们的内核    **第 6 步：构建内核**  通过运行以下命令开始构建内核    构建和编译 Linux 内核的过程需要一些时间才能完成。  回答完一连串问题后，我们可以安装我们使用以下命令启用的模块：  sudo make modules\_install  （这需要很多时间）    最后，我们通过键入来安装内核  > sudo make install  （这需要很多时间）    完成后输出显示完成：    第 7 步：重新启动并验证内核版本  现在我们重新启动机器。  当系统启动时，使用 uname 命令验证内核版本：  无名——夫人  终端打印出当前的 Linux 内核版本。 | **Step 1: Download the source code**  The Linux kernel source code can be found at kernel.org. Here we can get the URL of the kernel by clicking “Latest Release”.  URL: <https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.19.8.tar.xz>  We then open the terminal and use the wget command to download the Linux kernel source code:  wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.19.8.tar.xz  The output of the command shows the “saved message when the download is complete.    Note: Wget is a free GNU command-line utility tool used to download files from the internet.  **Step 2: Extract the source code**  When the file is ready, run the **tar** command to extract the source code.  tar xvf [linux-5.19.8.tar.xz](https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.19.8.tar.xz)  The output displays the extracted kernel source code.    Note: Tar commands, short for tape archive, are used to create, extract and maintain archived (compressed) files. The syntax takes three arguments:  $ tar [options] [archive-file] [directory/file to be archived]  Xvf is the Unix-style, short method to implement –extract –verbose –file.  X = extracting the archive,  V = displaying verbose information  F = specifying a filename  Verbose information provides additional details about the task being carried out.  **Step 3: Install Required Packages**  In order to compile the kernel successfully, specific packages are required. To install these packages, we run the command:  Sudo apt-get install git fakeroot build-essential ncurses-dev-xz-utils libssl-dev bc flex libelf-dev bison    Note: APT (Advanced Package Tool) is the command line tool used to find and install new packages, upgrade packages, clean packages etc...  Apt-get is for installing, updating packages. IT works on a database of available packages.  The command we used installs the following packages:   |  |  | | --- | --- | | **Package** | **Package description** | | **git** | Tracks and makes a record of all changes during development in the source code. It also allows reverting the changes. | | **fakeroot** | Packaging tool that makes the fake root environment. | | **build-essential** | Installs development tools such as C, C++, gcc, and g++. | | **ncurses-dev** | Programming library that provides API for the text-based terminals. | | **xz-utils** | Provides fast file compression and decompression. | | **libssl-dev** | Supports SSL and TSL that encrypt data and make the internet connection secure. | | **bc** (Basic Calculator) | A mathematical scripting language that supports the interactive execution of statements. | | **flex** (Fast Lexical Analyzer Generator) | Generates lexical analyzers that convert characters into tokens. | | **libelf-dev** | Issues a shared library for managing ELF files (executable files, core dumps and object code) | | **bison** | GNU parser generator that converts grammar description to a C program. |   **Step 4: Configuring and Compiling**  Before compiling the kernel, we need to configure which modules are to be included and which ones are to be left out.  There are many ways to go about doing this.  An easy and straightforward way to do this is to first copy the existing kernel config file and then use ‘menuconfig’ to make changes (if necessary). This is the fastest way to do it and probably, the safest.  Navigate to the linux-5.9.6. directory using the cd command:  cd linux-5.9.6  cp /boot/config-$(uname -r) .config  make menuconfig  The command launches several scripts, which then opens the configuration menus:    In this case we won’t be making any changes in the configuration menu so we exit the menu.  Note: One can perform copy and paste operations without leaving the Linux terminal using the cp command. The syntax of this command is:  cp [...file/directory-sources] [destination]  [file/directory-sources] specifies the sources of the files or directories you want to copy  The [destination] argument specifies the location you want to copy the file to.  $(uname –r) is used for command substitution. It runs the uname –r command which returns the current kernel version and then puts what is returned into the cp command.  Command substitution allows the output of a command to replace the command itself. Command substitution occurs when a command is enclosed as follows:  $(command)  or  `command`  The ‘make’ command helps system administrators compile and install open-source utilities. It executes or uses the arguments specified in the makefile identifying different actions to handle the target project.  **Step 5: update EXTRAVERSION in makefile**  We do this by opening the kernel’s makefile, located in the root directory in the source code using the text editor.  Now we change the value of EXTRAVERSION to –1820171025.  -EXTRAVERSION =  +EXTRAVERSION = -1820171025  Now we can re-compile our kernel    **Step 6: Building the kernel**  Start building the kernel by running the following command    The process of building and compiling the Linux kernel takes some time to complete.  After answering the litany of questions, we can then install the modules we’ve enabled with the command:  sudo make modules\_install  (this takes a lot of time)  Finally, we install the kernel by typing  > sudo make install  (this takes a lot of time)  The output shows done when finished:  **Step 7: Reboot and verify kernel version**  Now we reboot the machine.  When the system boots up, verify the kernel version using the uname command:  Uname –mrs  The terminal prints out the current Linux kernel version. |

|  |  |
| --- | --- |
| **实验结果和分析** | **Experimental Results and Analysis** |
| 自定义内核编译成功，学号成功添加到makefile中的EXTRAVERSION。  重启后的 uname -mrs 命令显示结果 | The custom kernel was successfully compiled and student number was successfully added to EXTRAVERSION in the makefile.  The uname -mrs command after rebooting shows us the results |

|  |  |
| --- | --- |
| **讨论、心得** | **Discussion and experience** |
| 确定执行此实验的最佳步骤需要一些时间。 即便如此，最初的方法还是遗漏了一些东西。  我需要安装 gcc、flex、bison 才能使 make menuconfig 命令正常工作。    运行“make”命令时。 为了解决这个问题，我必须安装 libssl-dev 和 libelf-dev。    我遇到了这些错误，因为我的第一个 sudo apt-get install 命令有一个语法错误，省略了某些软件包的安装。    在运行 make 命令时，我遇到了认证错误。 要修复它，我必须更改我的配置文件 .config  CONFIG\_SYSTEM\_TRUSTED\_KEYS="debian/canonical-certs.pem"  至  CONFIG\_SYSTEM\_TRUSTED\_KEYS=""    这一切都需要几个小时才能完成 | It took some time to identify the best steps to execute this experiment. Even then there were some things that were missed in the initial approach.  I needed to install gcc, flex, bison to make the make menuconfig command work.    When running the ‘make’ command. To fix this I had to install libssl-dev and libelf-dev.    I ran into these errors because my first sudo apt-get install command had a syntax error that omitted installation of some packages.    While running the make command I ran into a certification error. To fix it I had to change my config file .config  CONFIG\_SYSTEM\_TRUSTED\_KEYS="debian/canonical-certs.pem"  to  CONFIG\_SYSTEM\_TRUSTED\_KEYS=""    This all took several hours to complete |