**中国矿业大学计算机学院**

**2020 级本科生课程报告**

课程名称 操作系统课程设计

报告时间 2022年5月14日

学生姓名 杨再润

学 号 06203203

专 业 数据科学与大数据

任课教师 鲍 宇

**目 录**

**一 实验环境的使用**

**1.实验目的 3**

**2.实验内容 3**

**3.实验步骤 3**

**4.运行结果 13**

**5.实验体会 17**

# 实验一 实验环境的使用

1. **实验目的**

* 熟悉VSCode的基本使用方法。
* 练习编译、调试Linux内核及应用程序。
* 学习Linux中的基本命令和常用工具的使用方法。

1. **实验内容**
   1. **VSCode的基本使用方法**

**安装和启动VSCode**

本书充分利用了VSCode高度可定制的特性，为Linux 0.11内核专门制作了一个VSCode编程环境。该VSCode编程环境中已经集成了Make构建工具、GCC编译器、NASM汇编器、GDB调试器、Bochs虚拟机等必要工具，同时包含了一些有用的VSCode插件，目的就是免去读者手工构建实验环境所带来的学习成本，使读者可以将主要精力放在对操作系统原理和Linux 0.11源代码的分析与理解上。

读者首先需要下载与本书配套的VSCode压缩包文件，然后将其解压缩到64位Windows 7或Windows 10本地磁盘的一个目录中（例如D:\vscode-for-linux011，目录路径不要包含中文字符或者空格），无需修改环境变量，也无需进行任何安装操作。在启动VSCode之前，读者还需要单独安装Python语言解释器（3.8及以上版本）和Git客户端软件（2.18及以上版本）。

最后，读者就可以双击D:\vscode-for-linux011\Code.exe文件启动VSCode了。

**VSCode的窗口布局**

VSCode的窗口布局由下面的若干元素组成：

* 编辑器：这是主要的代码编辑区域，可以多列或者多行的打开多个编辑器。
* 侧边栏：位于左侧的侧边栏包含了文件资源管理器、文件搜索、源代码版本管理、调试与运行、插件等基本视图。
* 活动栏：位于侧边栏的左侧，可以方便的让用户在不同的视图之间进行切换。
* 状态栏：位于底部的状态栏用于显示当前打开文件的光标位置、编码格式等信息。
* 面板：编辑器的下方可以展示不同的面板，包括显示输出信息的面板、显示调试信息的面板、显示错误信息的面板和集成终端。面板也可以被移动到编辑器的右侧。
  1. **Linux 0.11内核项目的生成和调试**

在后续的实验过程中，读者主要会使用两个项目，一个是Linux 0.11内核项目，用于生成Linux 0.11操作系统，另外一个是Linux 0.11应用程序项目，用于生成Linux 0.11的应用程序。接下来，首先学习一下Linux 0.11内核项目。

**将Linux 0.11内核项目克隆到本地**

使用VSCode内置的Git功能可以将平台上的Linux 0.11内核项目克隆到本地。操作步骤如下：

1. 在VSCode的“View”菜单中选择“Command Palette...”，会在VSCode的顶部中间位置显示一个用来输入命令的面板。
2. 在VSCode的命令面板中输入“Git”后，会在列表中提示出所有与Git相关的命令。选择列表中的“Git: Clone”命令后按回车，会提示输入Git远程库的URL，将Linux 0.11内核项目Git远程库的URL地址<https://www.codecode.net/engintime/linux011/project-template/linux011kernel.git>填入命令面板中并按回车。注意，这里给出的是互联网平台的URL地址，如果读者使用的是校园网或局域网中的平台，通常需要将URL中的<https://www.codecode.net>进行替换。
3. Git远程库的URL输入成功后，会自动打开“选择文件夹”窗口，提示用户选择一个本地文件夹用来保存项目。此时，读者就可以在本地磁盘中选择一个合适的文件夹（注意，本地文件夹路径中不要包含中文字符和空格），然后点击“Select Repository Location”按钮。

|  |
| --- |
| **注意：**如果在选择文件夹后，弹出了Windows安全中心的凭据登录窗口，读者可以选择其中的“取消”按钮，跳过此步骤。这里不建议读者使用Windows凭据管理Git远程库的账号信息，一方面是由于一旦读者把用户名密码输入错误后，需要去Windows控制面板中的凭据管理器修改、或者删除凭据，比较麻烦；另外一方面，如果读者是在一台公共电脑上操作的话，使用Windows凭据管理器保存登录的用户名和密码是非常危险的。 |

1. 选择本地文件夹后，读者需要提供Git远程库的用户名和密码，远程服务器校验成功后，才允许克隆。首先，会在命令面板中提示输入“Username”，输入平台的用户名后按回车。接下来会提示输入“Password”，输入平台的密码后按回车。用户名和密码校验成功后就开始将Git远程库克隆到本地了。
2. 克隆成功后，会在VSCode的右下角弹出克隆完成提示框，点击其中的“open”按钮会使用VSCode打开克隆到本地的项目。
3. 为了确保在Git的提交记录中保存正确的签名（包括姓名和电子邮箱），在打开项目后，VSCode会在顶部的命令面板中提示输入“email”，输入平台注册时使用的邮箱后按回车。接下来会提示输入“name”，输入在平台注册时使用的真实姓名后按回车。设置完成后，会在VSCode右下角弹出提示框显示“Local git config successfully set.”。

|  |
| --- |
| **提示：**如果在设置Git签名的时候发现电子邮箱或姓名填写错误了，可以在“View”菜单中选择“Command Palette...”，会在VSCode的顶部中间位置显示一个用来输入命令的面板，输入“git-autoconfig: Get Config”命令，会在右下角弹出提示框显示当前设置的电子邮箱和姓名，如果确实填写错误，可以再次打开命令面板，并输入“git-autoconfig: Set Config”命令，然后选择列表中的“Custom”，重新填写电子邮箱和姓名。 |

此项目就是一个Linux 0.11操作系统内核项目，在左侧的“文件资源管理器”窗口中可以查看Linux 0.11内核项目包含的所有文件夹和源代码文件。也可以使用Windows资源管理器打开项目所在的文件夹，方法是在“文件资源管理器”窗口中的任意一个文件夹或文件节点上点击右键，然后在弹出的快捷菜单中选择“Reveal in File Explorer”。

**使用VSCode登录平台**

使用VSCode打开Linux 0.11操作系统内核项目后，需要登录平台，才能继续使用为Linux 0.11内核定制的功能。在“View”菜单中选择“Command Palette...”，会在VSCode的顶部中间位置显示命令面板，输入“Lab: Login”命令后，VSCode会在其顶部弹出一个窗口让读者选择平台的URL，如果读者已经拥有了互联网平台的用户名和密码，可以选择默认的URL，如果读者需要登录校园网或者局域网中的平台，就需要手动输入一个URL。然后，按照提示依次输入对应平台的用户名和密码即可完成登录。

**生成Linux 0.11内核项目**

在Linux 0.11内核项目的文件夹中提供了一个makefile文件，在各个子文件夹中也提供了makefile文件。Make工具就是使用这些makefile文件中的脚本将Linux 0.11内核的源代码生成为可以运行的二进制文件的。由于篇幅的限制，本书没有详细说明每个makefile文件的内容，请读者参考《Linux内核完全注释》一书中的第3.6节学习Make工具的使用方法，并通过其它章节中对makefile文件内容的介绍自行学习相关的内容。虽然本书提供的Linux 0.11内核中的makefile文件与原版的文件有一些差异，但是实现的功能是完全一致的。

生成Linux 0.11内核项目的方法是，在VSCode的“Terminal”菜单中选择“Run Build Task...”，会在VSCode的顶部中间位置弹出一个可以执行的Task列表，选择其中的“生成项目”即可。在项目生成的过程中，位于编辑器下方的“TERMINAL”窗口会实时显示生成的进度和结果。如果源代码中不包含语法错误，会在最后提示“Build Linux 0.11 success!”，表示生成成功。

如果源代码中存在语法错误，“TERMINAL”窗口会输出相应的错误信息（包括错误所在文件的路径，错误在文件中的行号，以及错误原因），并在最后提示生成失败。此时在“TERMINAL”窗口中，在按下Ctrl键的同时，用鼠标左键点击错误信息所在的行开始部分的文件路径，VSCode会使用源代码编辑器打开错误所在的文件，并自动定位到错误对应的代码行。定位错误的另外一种方法是，在编辑器下方的“PROBLEMS”窗口的错误列表中选择对应的行。

读者可以尝试在某个C源代码文件中故意输入一些错误的代码（例如删除一个代码行结尾的分号），然后再次生成项目，尝试通过错误信息来完成定位，将代码修改正确后再生成项目。

生成项目成功后，可以在左侧的“文件资源管理器”窗口中打开boot文件夹，找到刚刚生成的bootsect.bin和setup.bin文件，还可以在根目录下找到linux011.bin文件，这三个二进制文件就是Linux 0.11操作系统需要运行的可执行文件。这三个二进制文件已经被写入大小为1.44MB的软盘镜像文件floppya.img中。在启动调试时，会将该软盘镜像文件插入虚拟机的软盘驱动器A中，然后让虚拟机从软盘A开始引导，并最终运行其中的Linux 0.11操作系统（相当于将写有三个二进制文件的软盘插入一台裸机的软盘驱动器A中，然后按下开机按钮）。

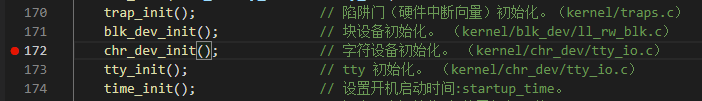
读者可能会注意到，生成项目成功后在左侧的“文件资源管理器”窗口中的根目录下，还新出现了几个文件，包括软盘镜像文件floppya.img和floppyb.img，硬盘镜像文件harddisk.img，以及与Bochs虚拟机相关的多个文件。这些文件是在生成项目的过程中安装到项目目录中的，并且没有被包含到Git库中（在“文件资源管理器”窗口中使用浅色显示文件名），原因是这些文件占用磁盘空间比较大，这样设计可以显著减小Git库的大小，大大加快克隆Git库和提交Git库的速度。

**启动调试Linux 0.11内核项目**

VSCode 提供的调试器（GDB）是一个功能强大的工具，使用此调试器可以观察程序的运行时行为并确定逻辑错误的位置，可以中断（或挂起）程序的执行以检查代码，计算和编辑程序中的变量，查看寄存器，以及查看从源代码创建的指令。为了顺利进行后续的各项实验，读者一定要学会灵活使用这些调试功能。

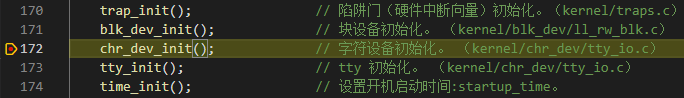
按照下面的步骤练习调试Linux 0.11源代码的过程：

1. 在“文件资源管理器”窗口的init文件夹中找到main.c文件，双击此文件节点使用源代码编辑器打开。
2. 在main.c文件中start函数的“chr\_dev\_init();”语句所在行（第172行）的行号左侧点击鼠标左键，添加一个断点，如图1-1所示。Linux 0.11启动时执行的第一个内核函数就是这个start函数。



**图1-1：在Linux 0.11内核项目的init/main.c文件中添加一个断点**

1. 启动调试的方法是选择“Run”菜单中的“Start Debugging”或者按F5快捷键。启动调试后，Bochs虚拟机开始运行软盘镜像A中的Linux 0.11操作系统。Bochs虚拟机启动后有两个窗口，一个是Console窗口，用来输入Bochs命令（目前不需要输入任何命令），另外一个是Display窗口，相当于计算机的显示器。随后，VSCode窗口会被自动激活，并且在刚刚添加断点的代码行左侧显示一个黄色箭头，表示程序已经在此行代码处中断执行（也就是说下一个要执行的就是此行代码），如图1-2所示。在Bochs的Display窗口中可以看到Linux 0.11的启动过程也中断了。



**图1-2：Linux 0.11内核启动调试后在断点处中断执行**

**单步调试**

单步调试功能可以让读者“逐过程”或者“逐语句”的调试源代码，进而跟踪程序执行的过程。按照下面的步骤练习使用单步调试功能：

1. 在VSCode的“Run”菜单中选择“Step Over”或者按F10快捷键，会执行黄色箭头当前指向的代码行，并将黄色箭头指向下一个要执行的代码行，即“tty\_init ();”语句所在行（第173行）。这就是“逐过程”调试功能，该功能不会调试进入chr\_dev\_init函数。
2. 在VSCode的“Run”菜单中选择“Step Into”或者按F11快捷键，可以发现黄色箭头指向了函数tty\_init中，说明“逐语句”功能可以进入函数，进而调试函数中的语句。
3. 在VSCode的“Run”菜单中选择“Step Out”或者按Shift+F11快捷键，会跳出tty\_init函数，返回到上级函数中继续调试（此时tty\_init函数已经执行完毕）。

**查看变量的值**

在调试的过程中，VSCode提供了多种查看变量值的方法，按照下面的步骤练习这些方法：

1. 将鼠标移动到源代码编辑器中变量的名称上，此时会弹出一个窗口显示出变量的值。例如将鼠标移动到变量main\_memory\_start或者memory\_end的上方，就可以查看这些变量的值。
2. 第二种方法是，在源代码编辑器中变量main\_memory\_start或者memory\_end的名称上双击鼠标左键，可以选中变量的名称，然后再点击鼠标右键，在弹出的快捷菜单中选择“Add to Watch”，可以将变量添加到左侧“调试和运行”窗口中的“WATCH”中，并显示变量的值。这种方式的好处是，可以在调试的过程中随时查看变量的值。

**调用堆栈（Call Stack）**

调用堆栈显示在左侧“调试和运行”窗口中的“CALL STACK”中，用来在调试的过程中查看当前堆栈上的函数，可以帮助理解函数的调用层次和调用过程。按照下面的步骤练习使用调用堆栈：

1. 按F11（“逐语句”功能的快捷键）调试进入time\_init函数，查看“CALL STACK”中的内容，可以发现在堆栈上有两个函数time\_init和start。其中当前正在调试的time\_init函数在栈顶位置，start函数在紧邻time\_init函数的下方，说明是在start函数中调用了time\_init函数。
2. 将time\_init函数开始位置定义的变量time添加到“WATCH”中，可以查看此局部变量的值。
3. 在“CALL STACK”中双击start函数所在的行，会激活start函数的堆栈帧。同时，编辑器会自动跳转到start函数的源代码，并有一个绿色箭头指向调用time\_init函数的代码行。在“WATCH”中的time变量会报告无法计算其值，原因是time变量是一个time\_init函数中的局部变量，在start函数中无法访问此变量。
4. 在“CALL STACK”中双击time\_init函数所在的行，可以重新激活此堆栈帧。

**继续运行和停止调试**

1. 如果要让程序从当前中断的位置继续运行，可以在VSCode的“Run”菜单中选择“Continue”或者按F5快捷键。此时查看Bochs虚拟机的Display窗口，会显示Linux 0.11操作系统已经启动完毕，并且终端已经打开，可以接受用户输入命令了。
2. 停止调试通常包括两部分，一个是停止VSCode的调试功能，另外一个是关闭Bochs虚拟机。在VSCode的“Run”菜单中选择“Stop Debugging”或者按Shift+F5快捷键可以停止VSCode的调试功能；点击Bochs虚拟机Console窗口右上角的关闭窗口按钮可以关闭Bochs虚拟机。通常这两个操作都需要进行，才能完全停止调试，但是先后顺序没有要求。

**学习Linux常用命令**

在附录2的表格中列出了Linux的常用命令。无论是最新版本的Linux，还是比较古老的Linux 0.11，都可以支持这些常用命令，由此可见这些命令旺盛的生命力和它们的重要性。

读者在前面的实验内容中使用了VSCode提供的调试功能，并搭配Bochs虚拟机的远程调试模式完成了Linux 0.11内核源代码的调试。在练习Linux的常用命令时，就不需要启动Bochs虚拟机的调试模式了，因为调试模式需要时刻检查调试状态和断点的位置，导致Bochs虚拟机在运行Linux 0.11的时候性能明显下降。

这里建议读者在VSCode的“Terminal”菜单中选择“Run Build Task...”，会在VSCode的顶部中间位置弹出一个可以执行的Task列表，选择其中的“Bochs 运行（不调试）”，即可让Bochs虚拟机在非调试的模式下用最快的速度来运行Linux 0.11操作系统。练习Linux常用命令后关闭Bochs虚拟机。

|  |
| --- |
| **提示：**在Linux的命令行中，可以使用TAB键让Linux提示用户可以使用的命令，例如，当用户输入了“ch”两个字符后按Tab键，Linux会将所有以“ch”开始的命令在下方提示出来，方便用户继续输入命令。如果用户在输入了“chm”三个字符后按Tab键，由于以这三个字符开始的命令只有“chmod”，Linux会直接帮助用户完成这个命令，这样可以大大加快用户输入命令的速度。使用“cd”命令进入某个目录时，同样可以使用此技巧，在输入目录名称前面的若干字符后按Tab键，尝试让Linux自动补全目录的名称。用户也可以使用键盘上的向上箭头和向下箭头按钮来显示之前输入过的历史命令，方便用户直接使用。 |

**学习代码导航功能**

Linux 0.11的源代码虽然不足两万行，仅为任何一个具有商业价值的软件代码量的十分之一，但是，读者很可能还是首次接触到具有如此规模的源代码，这就为读者系统、高效的阅读和理解Linux 0.11的源代码带来不小的挑战。即便是一位有着丰富经验的开发者，在没有适当工具的帮助下，阅读如此数量的源代码也会感到非常棘手。例如，当遇到一个变量时，想查看这个变量是在哪里定义的，其数据类型是什么；或者遇到一个函数时，想查看这个函数是如何实现的，在整个系统中都有哪些地方调用了此函数。

VSCode为了解决此类问题，提供了强大而灵活的代码导航命令。通过这些命令读者可以快速的在不同的代码之间进行跳转，对于提高读者阅读源代码的效率十分有帮助。请读者按照下面的步骤练习使用代码导航功能：

* 1. 打开init/main.c文件，在第183行调用了一个init函数。在init函数名称上点击鼠标右键，选择菜单中的“Go to Definition”，就会跳转到该函数的定义（第221行）。
  2. 在init函数名称上点击鼠标右键，选择菜单中的“Go to Declaration”，就会跳转到该函数的声明（第65行）。
  3. 在init函数名称上点击鼠标右键，选择菜单中的“Go to References”，就会使用内联编辑器显示所有引用此函数的位置，包括此函数的声明、定义和调用。

在本书第2.7节详细介绍了VSCode提供的与代码导航相关的功能，以及阅读源代码相关的方法与技巧，请读者自行练习。

* 1. **Linux 0.11应用程序项目的生成和运行**

之前是通过直接克隆Linux0.11内核项目的Git远程库到本地磁盘来新建一个Linux内核项目。接下来，会引导读者使用从平台领取任务的方式，将Linux 0.11应用程序项目克隆到本地磁盘的方法。在修改Linux应用程序项目的源代码后，还需要将作业提交平台。

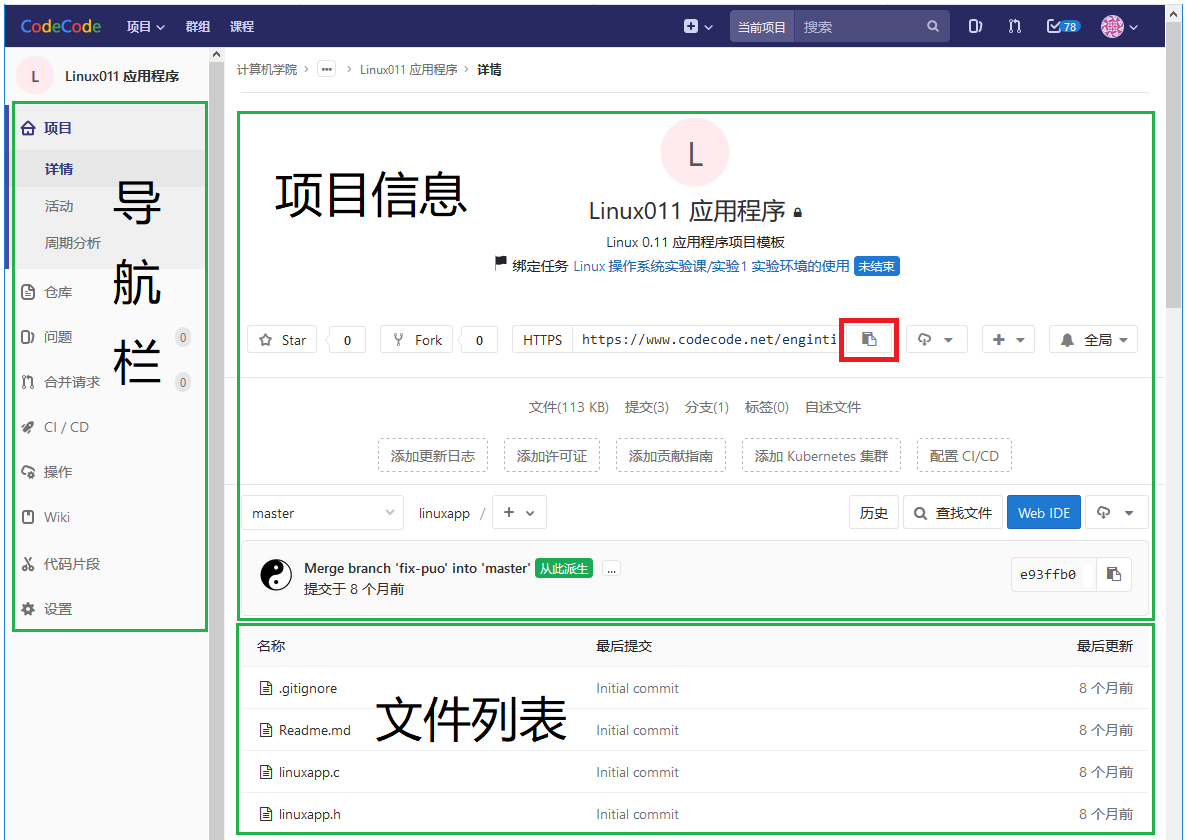
**从平台领取任务**

平台是用于教师在线布置实验任务，并统一管理学生提交的作业的平台。

1. 读者通过浏览器访问平台，可以打开平台的登录页面。注意，如果读者使用的是校园网或局域网中的平台，需要从平台管理员处获得URL。
2. 在登录页面中，读者输入用户名和密码，点击“登录”按钮，可以登录平台。
3. 登录成功后，在“课程”列表页面中，可以找到Linux操作系统对应的实验课程。点击此课程的链接，可以进入该课程的详细信息页面。
4. 在课程的详细信息页面中，可以查看课程描述信息，该信息对于完成实验十分重要，建议读者认真阅读。点击左侧导航中的“任务”链接，可以打开任务列表页面。
5. 在任务列表中找到本次实验对应的任务，点击右侧的“领取任务”按钮，可以进入“领取任务”页面。
6. 在“领取任务”页面填写“新建项目名称”和“新建项目路径”，然后选择“项目所在的群组”，点击“领取任务”按钮后，可以创建个人项目用于完成本次实验，并自动跳转到该项目所在的页面。

|  |
| --- |
| **提示：**在“领取任务”页面中，“新建项目名称”和“新建项目路径”这两项的内容可以使用默认值，如果选择的群组中已经包含了名称或者路径相同的项目，就会导致领取任务失败，此时需要修改新建项目名称和新建项目路径的内容。 |

1. 在新建的个人项目页面中，包括左侧的导航栏、项目信息、文件列表等，如图1-3所示。
2. 点击图1-3红色方框中的按钮，可以复制个人项目的URL，在本实验后面的练习中会使用这个URL将此项目克隆到读者计算机的本地磁盘中。

****

**图1-3：领取任务后得到的个人项目页面**

**将Linux 0.11应用程序项目克隆到本地**

如果读者从平台领取了任务，可以按照下面的步骤将个人项目克隆到本地磁盘中。

1. 在VSCode的“View”菜单中选择“Command Palette...”，会在VSCode的顶部中间位置显示一个用来输入命令的面板。
2. 在VSCode的命令面板中输入“Git”后，会在列表中提示出所有与Git相关的命令。选择列表中的“Git: Clone”命令后按回车，会提示输入Git远程库的URL，将之前复制的个人项目的URL粘贴到命令面板中并按回车。
3. Git远程库的URL输入成功后，会自动打开“选择文件夹”窗口，提示用户选择一个本地文件夹用来保存项目。此时，读者就可以在本地磁盘中选择一个合适的文件夹（注意，本地文件夹路径中不要包含中文字符和空格），然后点击“Select Repository Location”按钮。
4. 选择本地文件夹后，会在命令面板中提示输入“Username”，输入平台的用户名后按回车。接下来会提示输入“Password”，输入平台的密码后按回车。用户名和密码校验成功后就开始将Git远程库克隆到本地了。
5. 克隆成功后，会在VSCode的右下角弹出克隆完成提示框，点击其中的“open”按钮会使用VSCode打开克隆到本地的项目。
6. 为了确保在Git的提交记录中保存正确的签名（包括姓名和电子邮箱），在打开项目后，VSCode会在顶部的命令面板中提示输入“email”，输入在平台注册时使用的邮箱后按回车。接下来会提示输入“name”，输入在平台注册时使用的真实姓名后按回车。设置完成后，会在VSCode右下角弹出提示框显示“Local git config successfully set.”。

|  |
| --- |
| **提示：**如果在设置Git签名的时候发现电子邮箱或姓名填写错误了，可以在命令面板中输入“git-autoconfig: Get Config”命令，会在右下角弹出提示框显示当前设置的电子邮箱和姓名，如果确实填写错误，可以在命令面板中输入“git-autoconfig: Set Config”命令，然后选择列表中的“Custom”，重新填写电子邮箱和姓名。 |

**生成Linux 0.11应用程序项目**

在Linux 0.11应用程序项目的文件夹中提供了一个makefile文件。Make工具就是使用该makefile文件中的脚本将Linux 0.11应用程序的源代码文件linuxapp.c生成为可以运行的二进制文件linuxapp.exe的。请读者自行学习makefile文件中的内容。

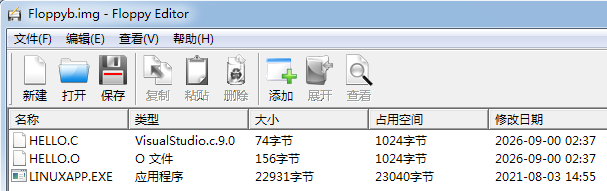
生成Linux 0.11应用程序项目的方法与之前介绍的生成Linux 0.11内核项目的方法完全一致，同样是在VSCode的“Terminal”菜单中选择“Run Build Task...”，会在VSCode的顶部中间位置弹出一个可以执行的Task列表，选择其中的“生成项目”即可。在项目生成的过程中，位于编辑器下方的“TERMINAL”窗口会实时显示生成的进度和结果。如果源代码中不包含语法错误，会在最后提示“Build Linux 0.11 app success!”，表示生成成功。

如果源代码中存在语法错误，“TERMINAL”窗口会输出相应的错误信息（包括错误所在文件的路径，错误在文件中的行号，以及错误原因），并在最后提示生成失败。此时在“TERMINAL”窗口中，在按下Ctrl键的同时，用鼠标左键点击错误信息所在的行开始部分的文件路径，VSCode会使用源代码编辑器打开错误所在的文件，并自动定位到错误对应的代码行。

生成项目成功后，可以在左侧的“文件资源管理器”窗口中找到刚刚生成的linuxapp.exe文件，该文件就是Linux 0.11应用程序的可执行文件。注意，该可执行文件使用的是aout格式，而不是Windows的PE格式，所以不能在Windows中运行，只能在Linux 0.11中运行。VSCode每次生成应用程序项目成功后，都会自动将应用程序的可执行文件写入软盘镜像文件floppyb.img中，以便在Linux 0.11中运行。

**查看软盘镜像文件floppyb.img中的内容**

在VSCode的“Terminal”菜单中选择“Run Build Task...”，会在VSCode的顶部中间位置弹出一个可以执行的Task列表，选择其中的“打开 floppyb.img”后会使用Floppy Editor工具打开该项目中的floppyb.img文件，用于查看软盘镜像中的文件。其中的linuxapp.exe文件就是刚刚生成的Linux 0.11应用程序，可以注意查看一下该文件的修改日期，如图1-4所示。查看完毕后关闭Floppy Editor。



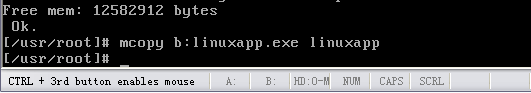
**图1-4：查看floppyb.img中Linux 0.11应用程序项目生成的可执行文件**

|  |
| --- |
| **注意：**Bochs虚拟机在调试内核项目的过程中，会独占访问软盘镜像文件，此时，使用Floppy Editor工具打开该项目中的floppyb.img文件会失败。如果需要修改软盘镜像中的文件，需先停止Bochs虚拟机，然后再使用Floppy Editor工具打开软盘镜像文件，修改软盘镜像中的文件后，点击“保存”按钮。 |

**运行Linux 0.11应用程序**

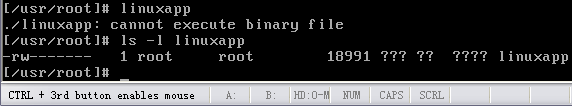
由于Linux 0.11应用程序可执行文件使用的是一种既古老，又相对简单的可执行文件格式——aout格式，而VSCode中的GDB已经不再支持调试此格式的可执行文件了，所以这里无法使用类似前面介绍的调试Linux 0.11内核那样的功能来调试Linux 0.11应用程序，也就是说不能在VSCode中按F5启动调试，无法添加断点，也无法提供单步调试、查看变量的值、查看调用堆栈等功能。但是读者仍然可以充分利用VSCode提供的强大的源代码编辑功能，提高开发Linux 0.11应用程序的效率，并按照下面的步骤查看Linux 0.11应用程序的运行结果：

1. 在VSCode的“Terminal”菜单中选择“Run Build Task...”，会在VSCode的顶部中间位置弹出一个可以执行的Task列表，选择其中的“Bochs 运行(不调试)”后，会使用Bochs虚拟机运行软盘镜像A中的Linux 0.11操作系统，并将文件floppyb.img加载到软盘驱动器B中。读者可以在Bochs虚拟机Display窗口的工具栏上依次点击软件驱动器A和B的按钮，确认加载了对应的软盘镜像文件。
2. 待Linux 0.11启动后，在Bochs的Display窗口的终端中使用“mcopy b:linuxapp.exe linuxapp”命令将软盘B中的可执行文件linuxapp.exe拷贝到硬盘的当前目录中,并命名为linuxapp，如图1-5所示。



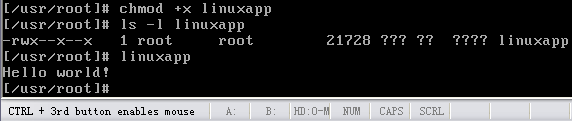
**图1-5：将可执行文件linuxapp.exe从软件B拷贝至硬盘**

1. 在终端输入“linuxapp”命令运行上一步拷贝的文件linuxapp,此时会提示“无法运行二进制文件”信息，原因是linuxapp文件没有可执行权限。使用“ls -l linuxapp”命令查看该文件的权限，可以看到其只有r（read）和w（write）权限，而没有x（execute）权限，如图1-6所示。



**图1-6：使用ls命令查看文件linuxapp的权限**

1. 使用“chmod +x linuxapp”命令使之具有可执行权限。再次使用ls命令确认文件权限修改后，就可以运行该文件了，如图1-7所示。



**图1-7：为文件linuxapp增加可执行权限并运行**

* 1. **在Linux 0.11操作系统中编写应用程序**

之前的内容详细介绍了使用VSCode编写并运行Linux 0.11应用程序的方法。经常在Windows操作系统中编写程序的读者会对这种操作方式十分熟悉，并觉得很方便。但是，在Linux 0.11操作系统中，由于没有可视化的人机交互界面，在其中编写应用程序的过程就会比较复杂，但是也有必要深入学习一下这种操作方式，为今后在Linux操作系统中编程打下基础。

接下来，会向读者详细介绍在Linux 0.11操作系统中如何使用vi编辑器编写源代码文件，如何使用GCC工具将源代码编译为可执行文件，以及如何编写makefile文件将多个源代码文件作为一个项目来进行管理的方法。

**使用vi编辑器编写Linux 0.11应用程序并编译运行**

vi（visual interpreter）是Linux操作系统使用的最基本的文本编辑器，每个Linux发行版都会自带vi编辑器，不需要用户单独安装。注意，vi和vim是不同的编辑器，vim可以认为是vi的增强版本，而且通常需要用户单独安装。

接下来请读者按照下面的步骤，练习在Linux 0.11中使用vi编辑器编写源代码文件，然后使用GCC将源代码文件编译为可执行文件。读者应该会感受到这种方法比较繁琐，没有使用VSCode直接编写Linux 0.11的应用程序来的方便，不过学习这种方法还是很有必要的，当读者今后在一个只有命令终端的Linux系统上工作时，就不得不使用这种方法来编写程序了。

1. 继续使用VSCode打开在本实验3.3节获取到的Linux 0.11应用程序项目。
2. 在VSCode的“Terminal”菜单中选择“Run Build Task...”，会在VSCode的顶部中间位置弹出一个可以执行的Task列表，选择其中的“Bochs 运行(不调试)”后，会使用Bochs虚拟机运行软盘镜像A中的Linux 0.11操作系统。Linux 0.11启动完毕后，在其终端中使用ls命令查看当前目录，并用rm命令将hello.c文件删除。
3. 在Linux 0.11终端中使用命令“vi hello.c”新建并打开hello.c文件，在此文件中使用C语言编写一个可以输出“hello world”的程序，保存源代码文件并退出vi。
4. 在Linux 0.11终端中使用命令“gcc hello.c -o hello”让GCC工具编译hello.c文件，得到可执行文件hello。
5. 运行可执行文件hello，查看输出的内容。

|  |
| --- |
| **注意：**由于Linux 0.11应用程序使用比较古老的、版本号为1.40的GCC进行编译，所以只支持标准的C语言语法，例如，仅支持使用“/\* \*/”进行块注释，不支持使用“//”进行行注释，另外必须在函数的开始位置定义局部变量。 |

**编写makefile文件管理项目**

Make工具可以用来管理一个项目中多个源代码文件的编译和链接过程，也可以用来管理多个模块间的依赖关系，甚至是软件的安装过程。练习编写简单的makefile文件，对于读者今后在Linux下开发应用程序还是非常必要的。

下面是一个可以完成编译链接hello.c任务的makefile文件的内容：

hello: hello.c

gcc hello.c -o hello

第一行描述依赖关系，指出目标文件hello依赖于源代码文件hello.c，第二行描述操作命令，指出要用GCC编译器的可执行文件gcc将源代码文件hello.c编译输出为可执行文件hello。当修改hello.c文件后，需要编译时，只需在该makefile文件所在目录中输入make命令即可。

|  |
| --- |
| **注意**，在makefile文件中跟在依赖关系命令行之后的操作命令行必须用制表符(Tab键)进行缩进，不能使用空格进行缩进，否则make命令会报告“missing separator”错误。 |

也可以在makefile文件分步骤完成对hello.c文件的编译和链接，内容如下：

hello:hello.o

gcc hello.o -o hello

hello.o:hello.c

gcc -c hello.c -o hello.o

clean:

rm -f \*.o

rm -f hello

可执行文件hello依赖于对象文件hello.o，而hello.o又依赖于源代码文件hello.c（选项 -c 指示GCC仅完成编译操作，而不进行链接操作）。另外又增加了一个clean目标，用于清理操作。可在任何时刻使用clean作为目标，删掉makefile生成的文件。

Make工具默认会构造makefile文件中的第一个目标，但如果在命令行中指定目标，就可以构建任何一个目标，例如指定clean目标的命令如下：

make clean

* 1. **提交作业**

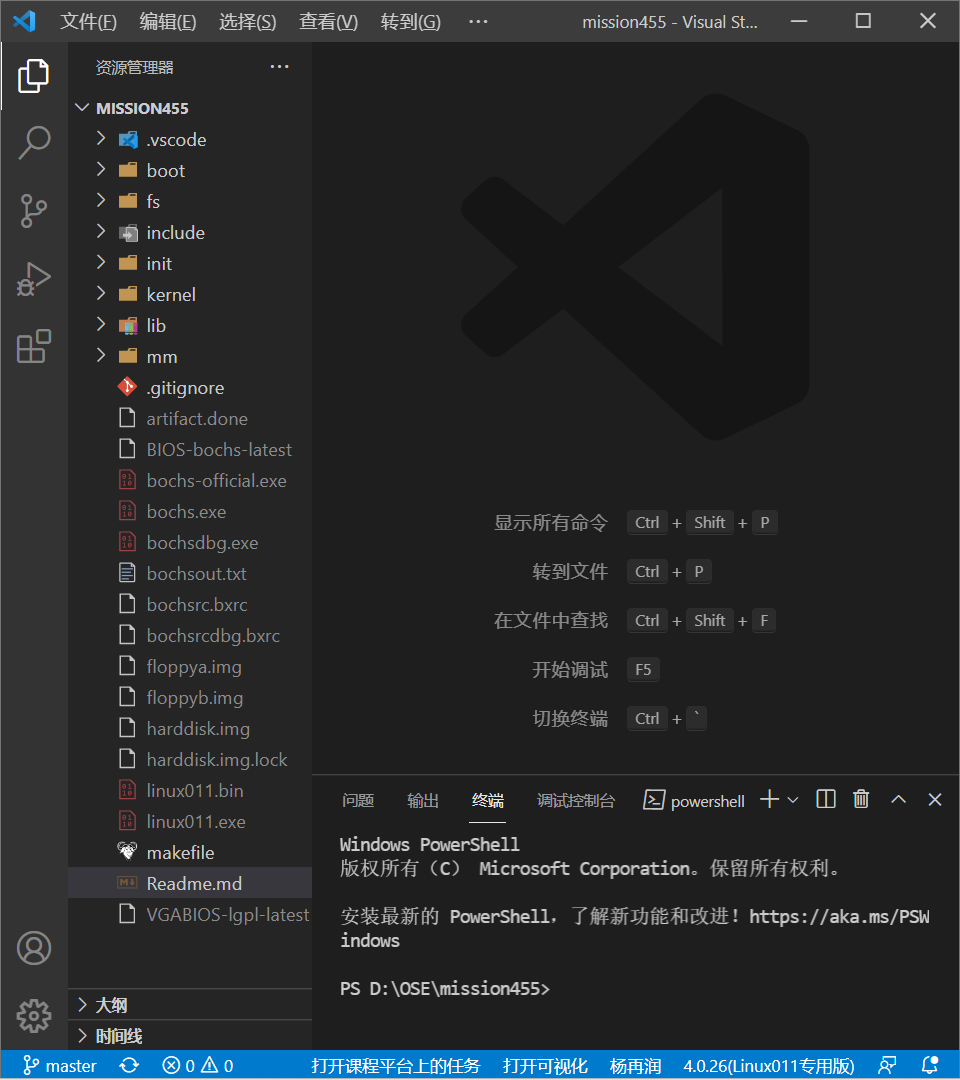
在本实验中，读者首先练习了使用VSCode生成和调试Linux 0.11内核项目，但是在这部分实验内容中，并没有要求读者对Linux 0.11内核项目中的任何文件进行修改，所以这个项目不需要提交作业。接下来，读者在练习使用VSCode生成和调试Linux 0.11应用程序项目时，按照本实验3.5节的内容在项目中添加了新的文件，读者就需要将这些文件作为本次实验的成果，提交到平台供教师审阅。

请读者按照下面的步骤提交作业：

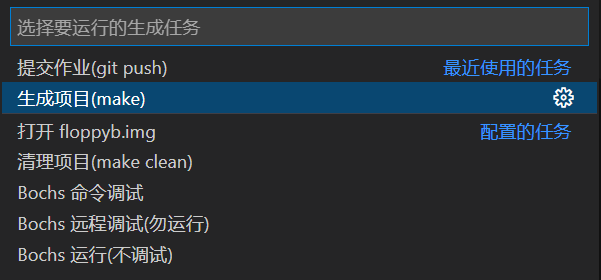
1. 使用VSCode查看文件变更详情，具体方法是，在VSCode的“View”菜单中选择“SCM”，打开左侧的“Git源代码版本控制”窗口，在变更列表中会显示最近新建、删除或内容被修改的文件。使用鼠标左键双击变更列表中文件，会使用编辑器显示当前的文件内容与上一个版本的文件内容相比较有哪些修改。如果想放弃对文件的修改，可以点击文件所在行右侧的“Discard Changes”按钮。
2. 在确认文件变更详情没有错误的情况下，读者就可以开始提交作业了。最快捷的方法是，在VSCode的“Terminal”菜单中选择“Run Build Task...”，会在VSCode的顶部中间位置弹出一个可以执行的Task列表，选择其中的“提交作业”，会自动完成Git库的提交（commit）和推送（push）操作，读者只需要在VSCode顶部中间位置弹出的命令窗口中，按照提示依次输入平台的用户名和密码即可。
3. 提交作业成功后，会自动使用浏览器打开平台中个人项目的页面，读者可以在这个页面中浏览项目中的文件和变更记录。
4. 如果在提交作业后发现仍然有文件需要修改，可以在完成修改后重复上面的步骤，再次提交作业。

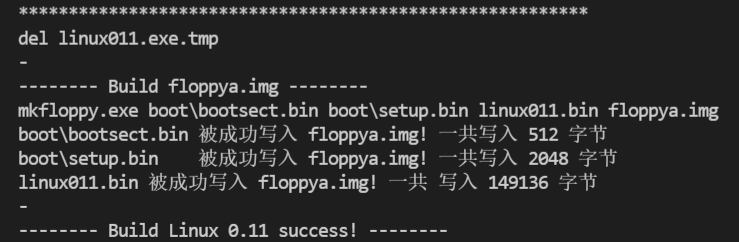
除了上面步骤中介绍的使用Task完成提交作业的方法，读者也可以使用VSCode提供的“Git源代码版本控制”窗口中提供的提交和推送功能完成作业的提交，具体方法请读者自行学习。

**三、实验结果**

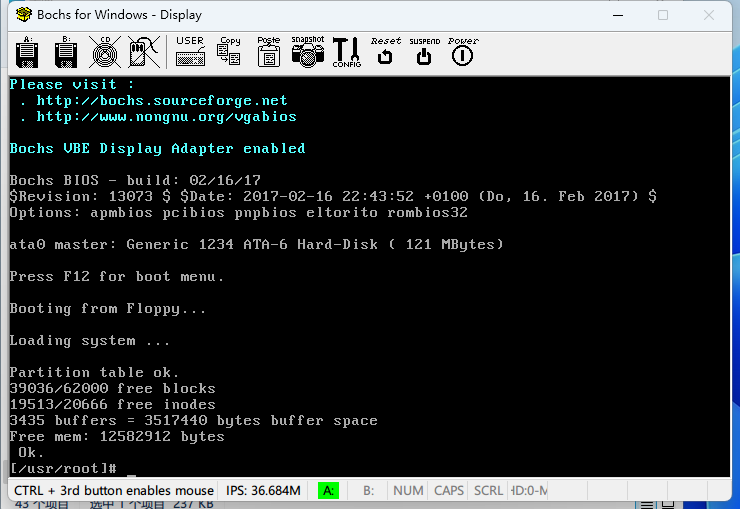


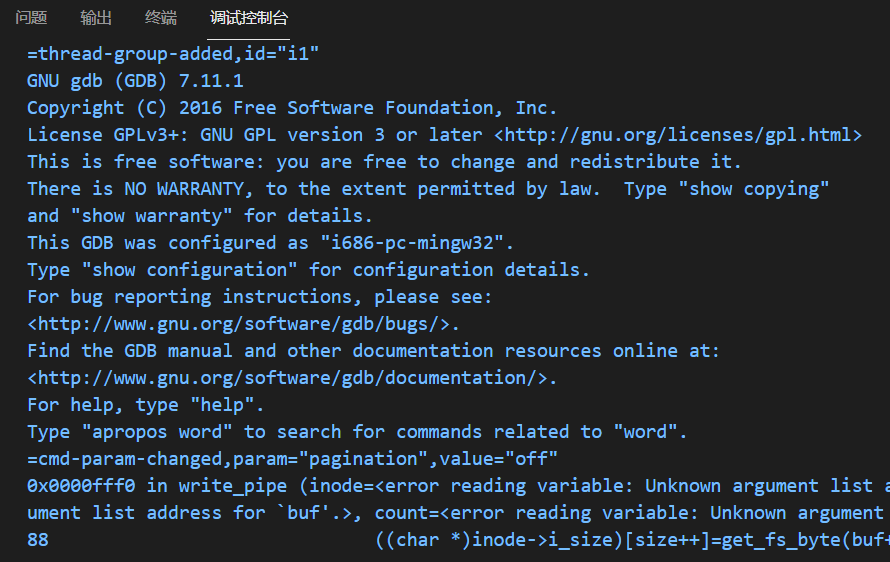
**启动平台并登录**



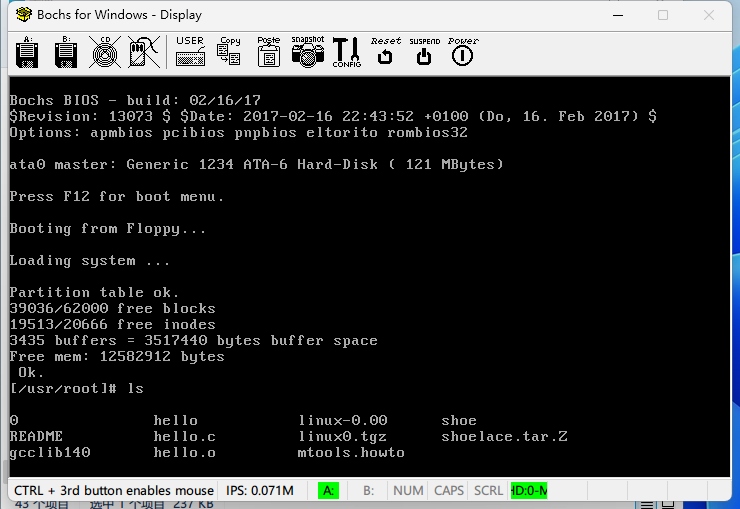


**生成项目**

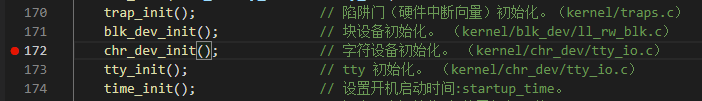


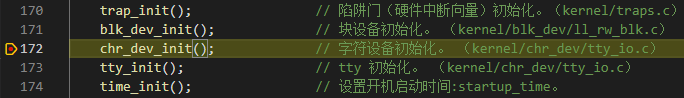


**打开虚拟机**



**测试命令**





**添加断点与调试**

1. **实验体会**
2. “细节决定成败”对于操作系统来说实在是再恰当不过了，即便是操作系统内核中某一个位的值0或者1发生了错误，都有可能会导致操作系统停机。
3. 只有通过亲身实践学习到的知识才能够真正被掌握，而那些仅仅从书本上得到的知识更容易被忘记。