

实验一：UNIX V6++ 运行调制环境的安装与配置

1. 实验目的

- (1) 安装配置 UNIX V6++ 的运行环境；
- (2) 安装配置 UNIX V6++ 的调试环境。

2. 实验设备及工具

已安装浏览器和 vscode 开发环境的 PC 机一台。

3. 预备知识

- (1) gcc 编译
- (2) gdb 调试

4. 实验内容

4.1. 安装后端服务根证书

考虑到后端服务器部署于内网，我们使用自签名证书实现安全连接。你只需要在系统上安装一个证书。打开下面这个链接，以下载证书文件：

<https://git.tongji.edu.cn/vesper-system/vesper-center/-/raw/main/src/main/resources/ssl/ca.crt?inline=false>

我们以 Windows 系统为例，说明证书的安装方法。双击打开刚下载的证书文件，将看到如下界面：

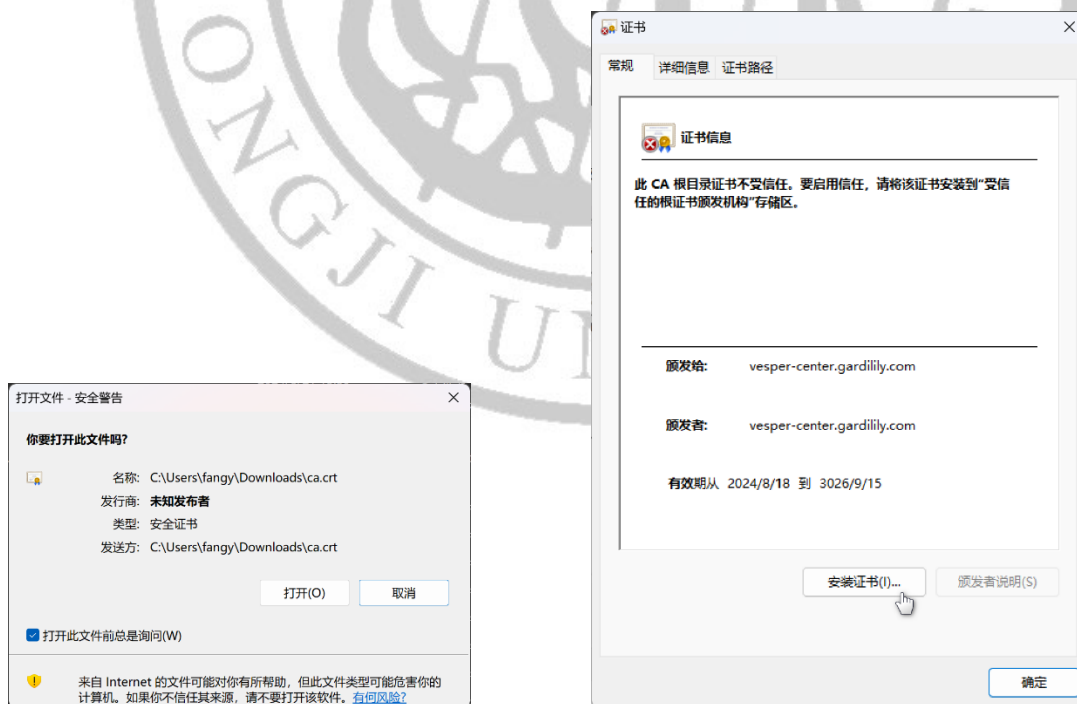


图 1

点击下方的“安装证书（Install Certificate）”，然后按图 2 中一组截图所示的步骤安装证书到你的机器上。

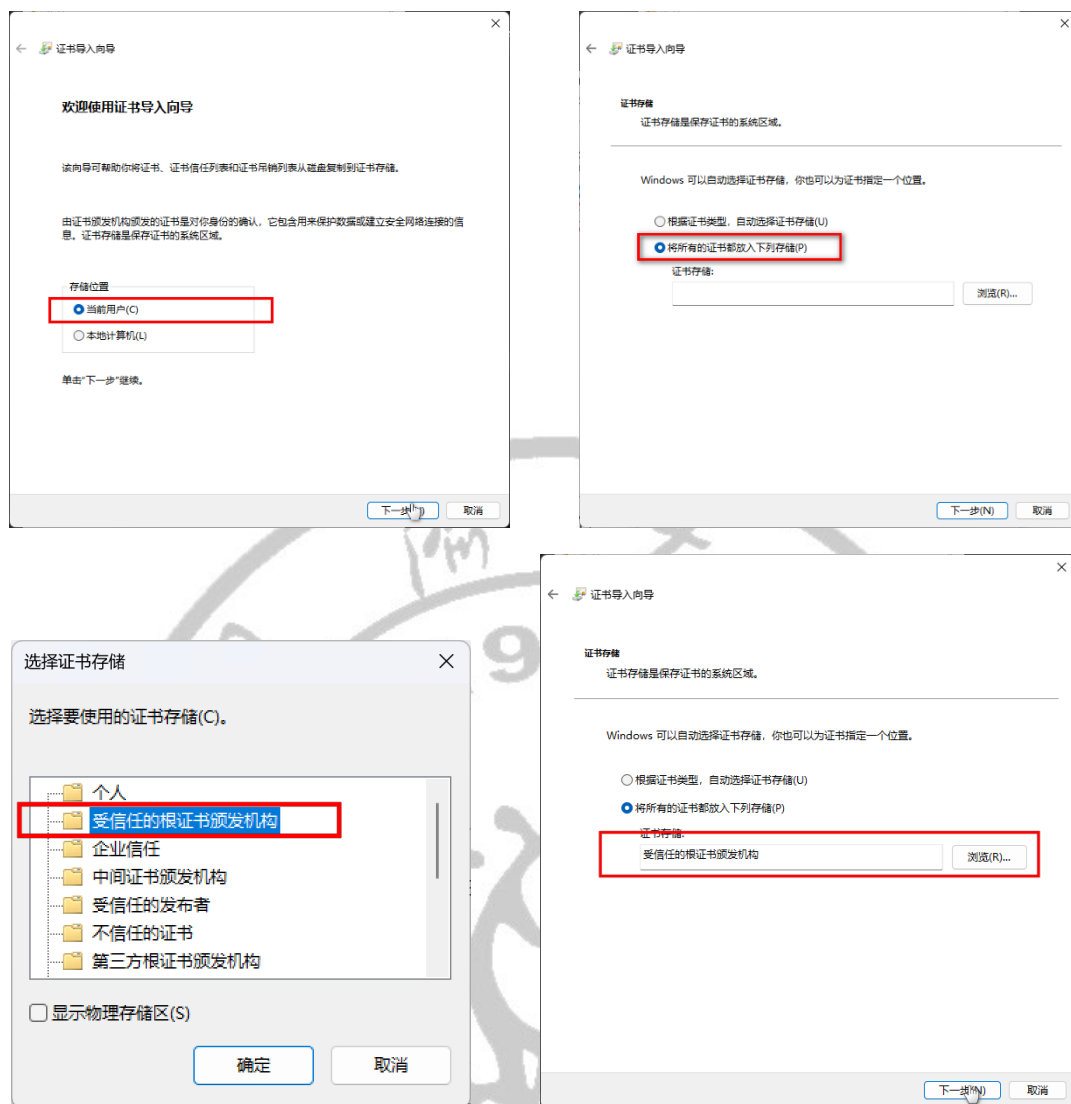


图 2：证书安装步骤

4.2. 登录实验平台

使用较为现代的浏览器（比如 Google Chrome 和 Microsoft Edge。不推荐使用 Safari），打开如下链接：

<http://vesper-system.pages.tongji.edu.cn/vesper-front/>

初次访问时，你会被引导到如图 3 所示的网络监测页面。请在后端地址处输入：

<https://vesper-center.gardilily.com:9000>,

然后点击右边的确认按钮。之后，请看后端可达指示。

- 指示灯变为**绿色**，表示后端配置正确，可以回到主页
- 指示灯为**红色**，可再点击几次确认按钮
- 指示灯仍为**红色**，请关闭电脑 VPN（注，如果 IP 属地不是 CN，表明你的梯子可能没关好），并保证自己连接的是校园网，然后重试。

指示灯变为绿色后，点击“返回主页”，将看到图 4 所示的登录页面。

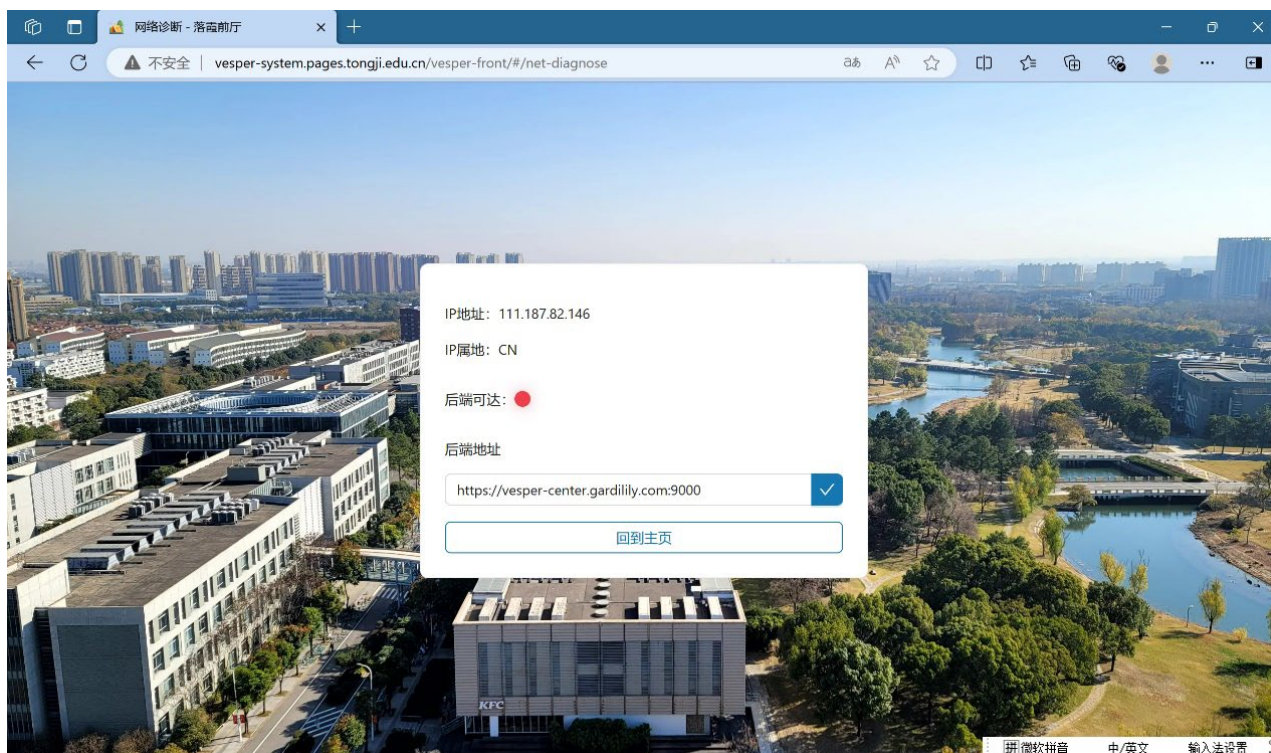


图 3

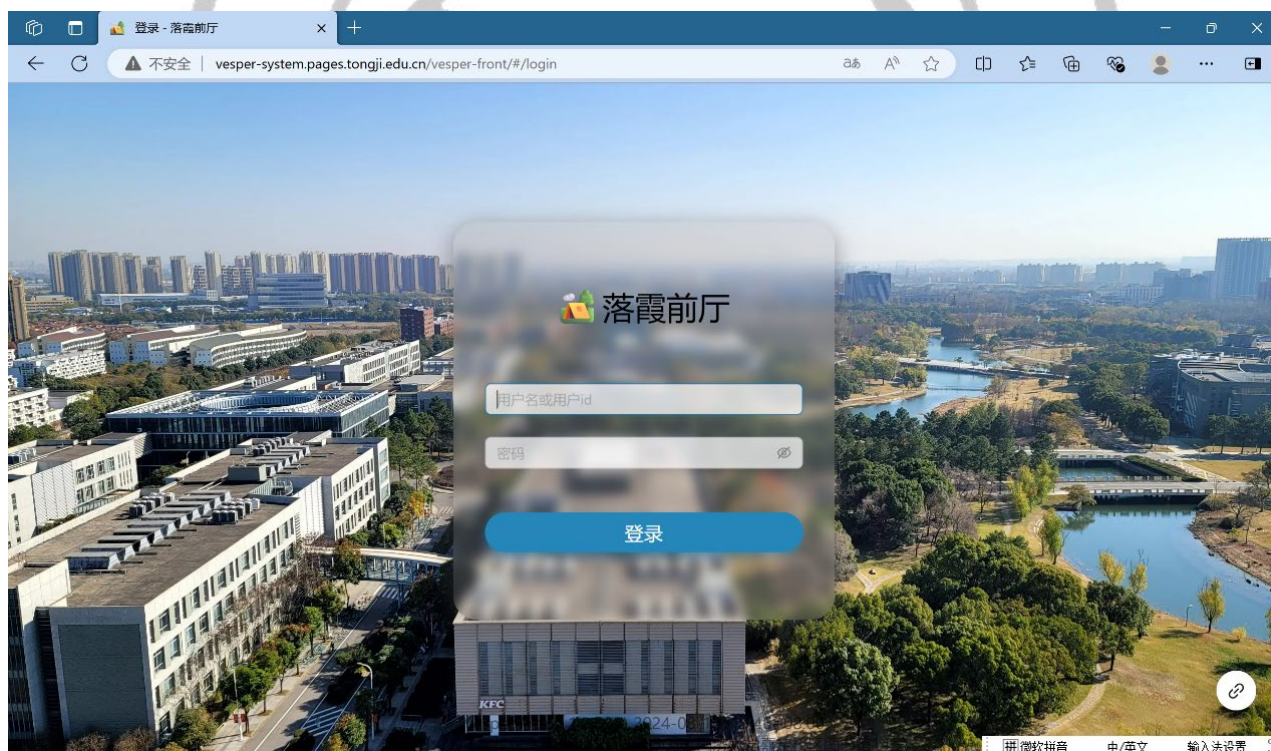


图 4：登录界面

请在此处输入教师下发的用户名和密码。登录成功后，你将看到如图 5 所示的系统主页。

首次登录成功后，请务必完成以下两项操作：

(1) 选择“我”，进行初始密码的修改（如图 6 所示）。

(2) 点击“桌面环境”，将鼠标停留在“主机名”上，完成主机名修改（如图 7 所示），请将主机名修改为你的真实姓名。教师将在确认主机名正确无误后，启用你的桌面环境。



图 5：系统主页

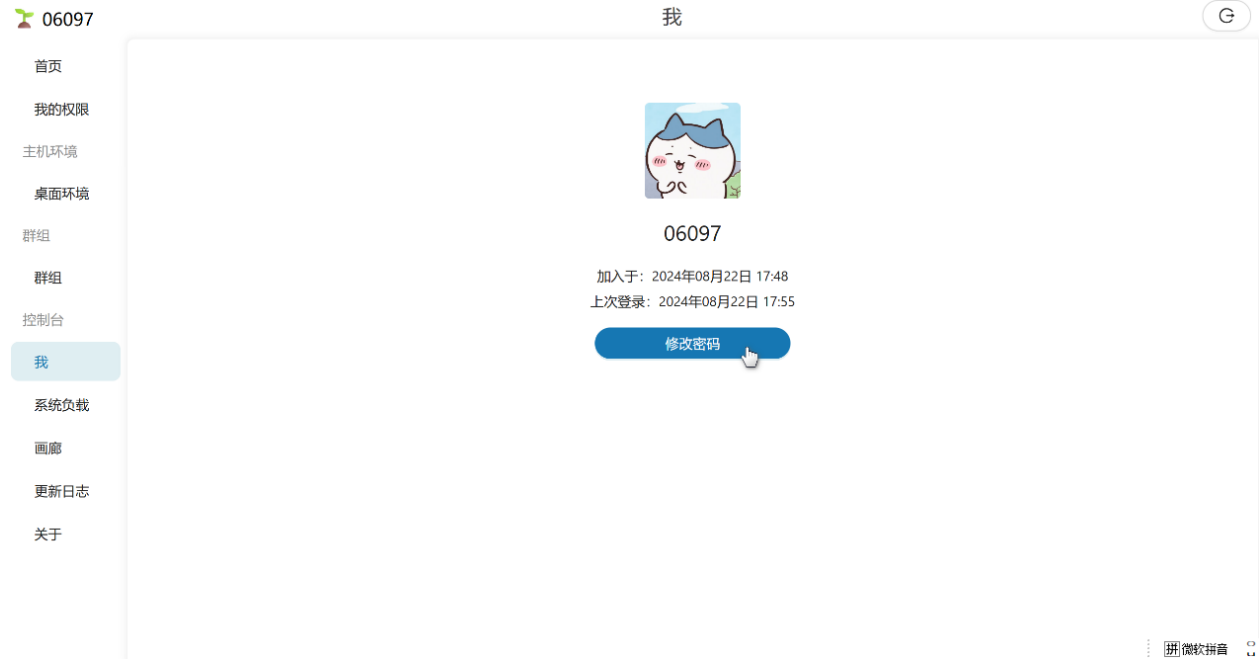


图 6：修改密码

所以，首次登录实验平台的体验只能到此结束了，等待一段时间，待桌面环境启用后，再按照上述过程重新登录，完成后续步骤。

4.3. 运行远程桌面环境

登录实验平台，进入你的桌面环境，如果显示状态为“启用”，则可以通过点击主机右面的“登录”，一键进入桌面环境（如图 8 所示）。注意，如果一键登录失败，可以打开“详情”，依次点击“关机”、“开机”、“启动 vesper”、“用内置 VNC Viewer 连接”。如果还是不行，请联系老师。

登录成功后，你将看到一个桌面环境（如图 9 所示）。该桌面环境基于 GNU/Linux，运行于后台服务器。



图 7：修改主机名



图 8：登录远程桌面环境

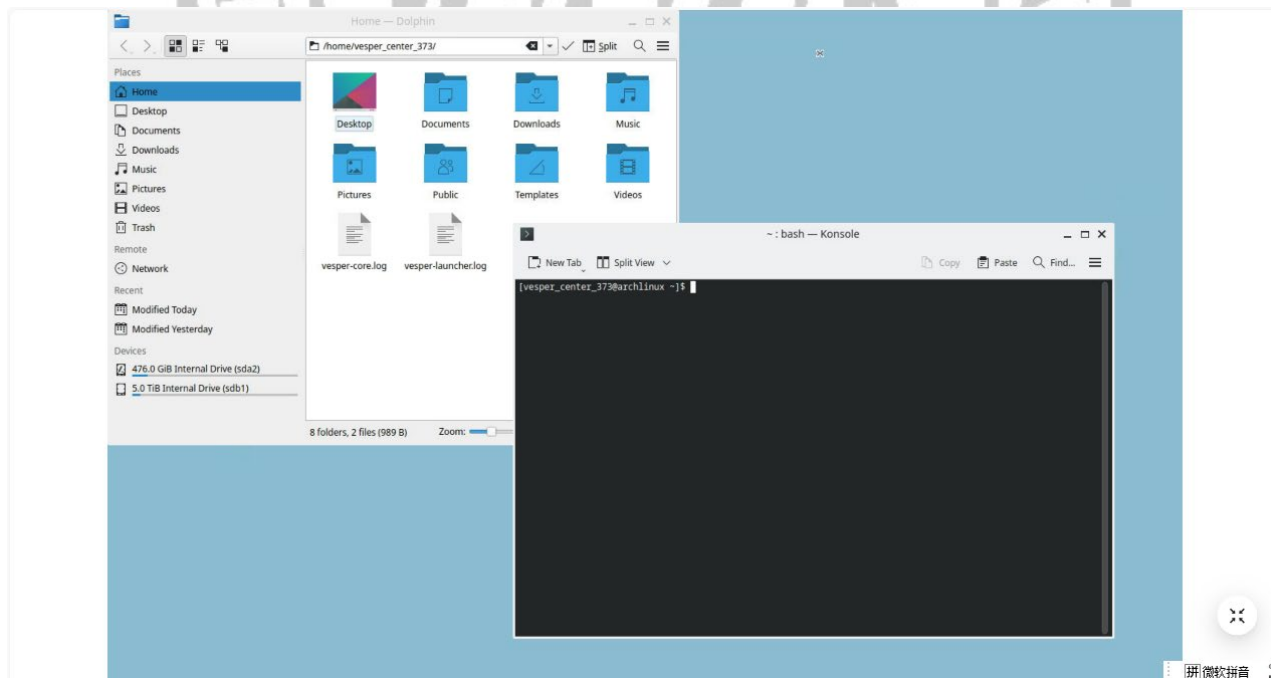


图 9：远程桌面环境启动成功

屏幕分辨率默认是 1440×900 。如果你需要使用其他分辨率，请打开“详情”页面将鼠标移动到“启动 vesper”，在弹出气泡内更改。**不建议使用低于 1280×720 的分辨率。也不建议使用更高的分辨率！高分辨率会导致系统卡顿。**

桌面默认打开一个文件浏览器和一个命令行。**不要关闭这个命令行！**截至撰写本文时，该桌面环境并未提供启动新命令行的快捷键。如果你不小心关了它，请退回到上一个网页，关闭主机，并重新开机。

4.4. 准备你的代码仓库

本节非必选。你可以直接跳过整个 4.4 节！

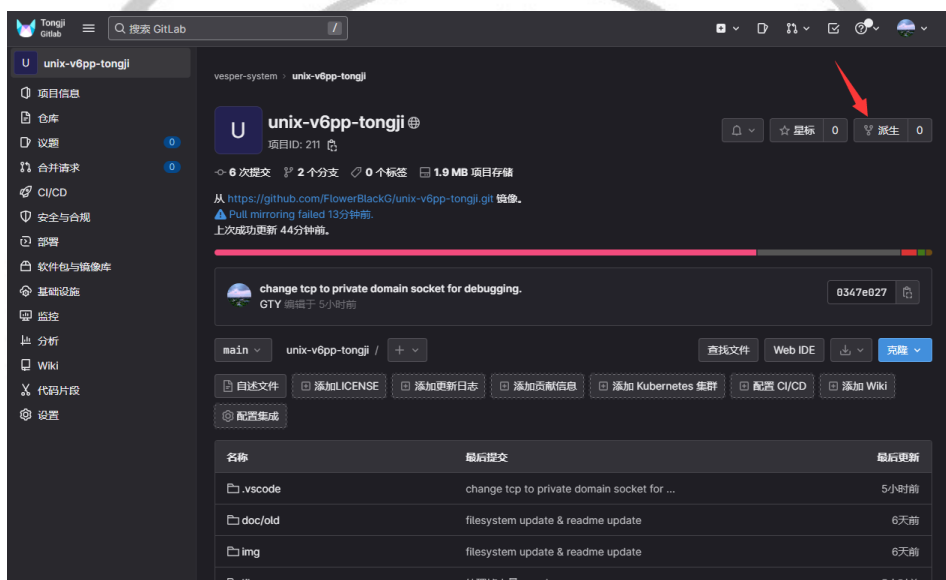
本节会带你熟悉 git 工具基础的操作。步骤完成后，你将可以使用自己的代码仓库完成实验，以记录成长，留下美好回忆。

4.4.1. 创建属于你自己的代码分支

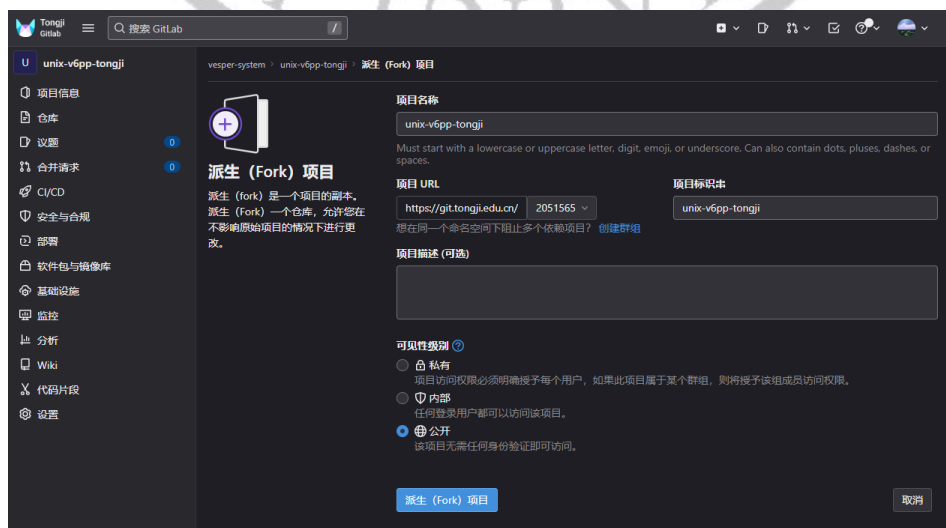
前往 <https://git.tongji.edu.cn/>，使用“统计统一身份认证登录”。登录后，前往 unix v6++实验系统的仓库：

<https://git.tongji.edu.cn/vesper-system/unix-v6pp-tongji>

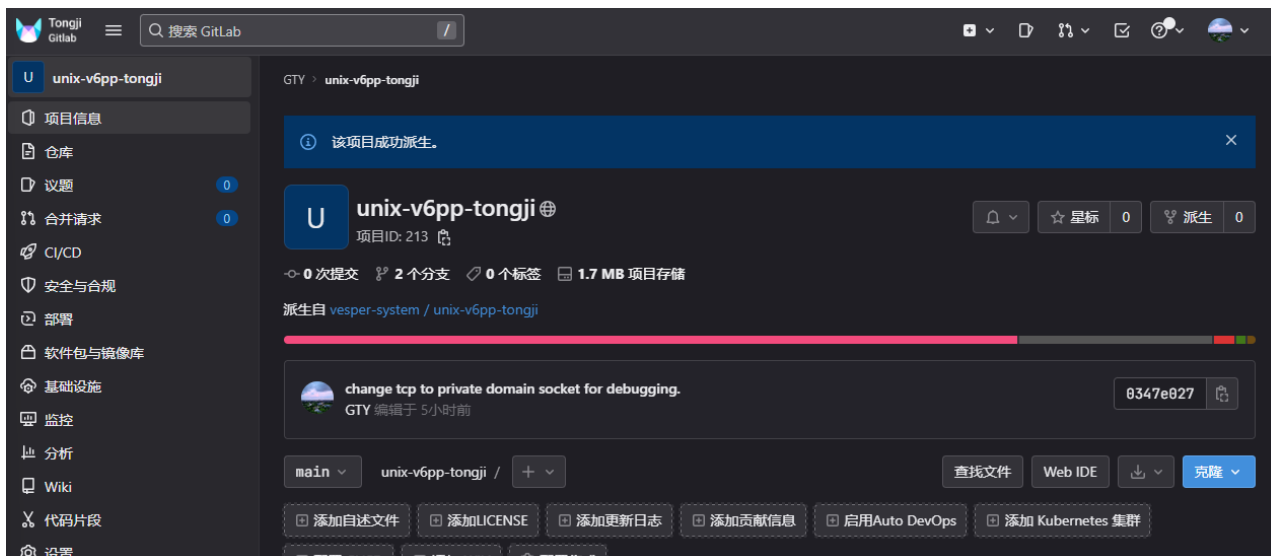
点击右上角的“派生（fork）”：



配置好基本信息，然后点击下方蓝色的“派生”按钮。建议不要改变项目名称。项目命名空间请选择你的学号。



操作完毕，将可以看到存放在你自己仓库里的 `unix v6++` 项目。此后，只要打开同济 GitLab 主页，你就能找到专属于你的那个版本。



4.4.2. 初始化 RSA 密钥对

为让你将实验环境内的代码上传到你的仓库，你需要配置一个 RSA 密钥对。注意，实验环境的超级管理员是可以直接访问你的环境的。所以，本学期结束后，请从同济 GitLab 移除密钥对（见 4.4.4 节）。

打开实验环境，在命令行输入“`ssh-keygen`”，一路回车直到密钥生成完毕（见下图）。输入“`cat ./ssh/id_rsa.pub`”，即可看到刚才创建的公钥。将它拷贝（选中，然后使用组合键 `Ctrl+Shift+C`）出来。

注意：截至 2024 年 8 月，Vesper VNC 尚未支持将文本内容复制到你的电脑。请通过实验环境内置的 Google Chrome 浏览器完成 4.4.3 节内容。Google Chrome 启动命令：

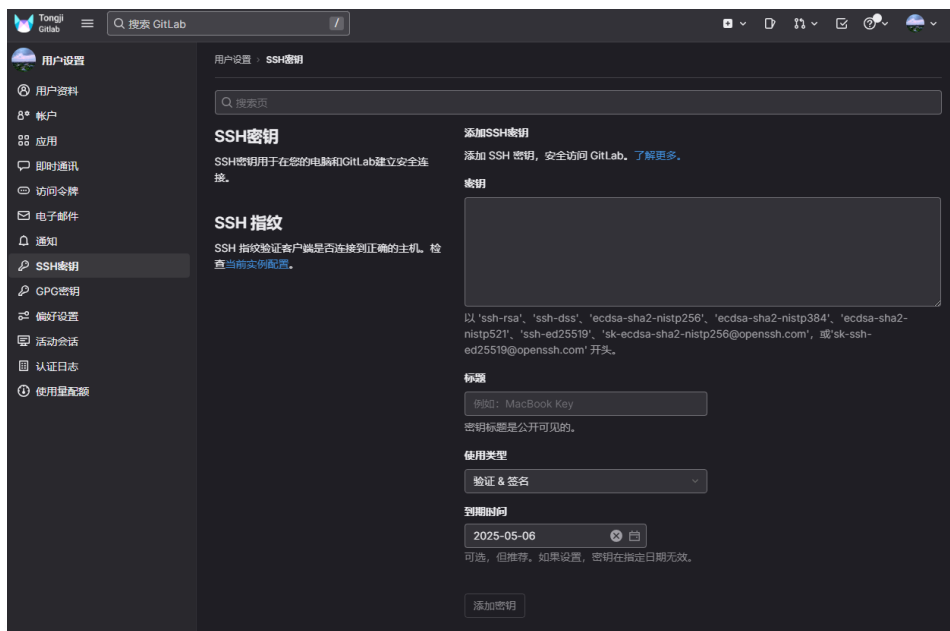
`google-chrome-stable --ozone-platform=wayland`

or: `chromium`

```
vesper_center_10@user-virtual-machine:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vesper_center_10/.ssh/id_rsa):
Created directory '/home/vesper_center_10/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vesper_center_10/.ssh/id_rsa
Your public key has been saved in /home/vesper_center_10/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:BF9fp+6j/QrX7cQ+JvtuZrwP0wIh0hzoJ/RSz61xr4Y vesper_center_10@user-virtual-machine
The key's randomart image is:
+---[RSA 3072]---+
  . . . .
  + . . . O
  o = . o .
  o = = o o
  + S + + .
  + . + + +
  . . . X *
  E . B /
  . o . % * 0
+----[SHA256]-----+
vesper_center_10@user-virtual-machine:~$
```

4.4.3. 上传公钥到同济 GitLAB

访问 <https://git.tongji.edu.cn/-/profile/keys>，将刚才得到的公钥全文粘贴到“密钥”输入框里，并随便填写一个标题。“使用类型”保持为“验证 & 签名”即可。你可以将到期时间设为一年后，以让其自动失效。



4.4.4. 从同济 GitLAB 删除公钥

虽然课程实验全部结束后，你的桌面环境会被删除。但如果不放心的话，你可以手动取消上传到同济 GitLab 的公钥。访问 <https://git.tongji.edu.cn/-/profile/keys>，找到你的公钥，删除即可。

4.5. 启动 UNIX V6++ 运行环境

4.5.1. 下载实验工具包

在 4.3 节中打开的桌面运行环境中，点击命令窗口，确保当前工作路径位于你的家目录内（即命令行里面，“\$”符号前面的是一个波浪号“~”）。

```
[vesper_center_373@archlinux ~]$
```

输入以下命令以下载 unix v6++ 实验包和辅助工具：

```
git clone https://git.tongji.edu.cn/{?}/unix-v6pp-tongji.git
```

如果你完成了 4.4 节的内容，请在 {?} 处填写你的学号。否则，请填写“vesper-system”。

请注意：截至 2024 年 8 月，Vesper VNC 尚未实现复制粘贴功能。这意味着你需要手写上面的命令。抱歉...)

小技巧：执行完第一条命令后，按键盘上的“↑”键，即可让命令行里重新出现上一条指令。

执行完毕，你可以在文件浏览器里看到刚下载下来的项目，如图 10 所示（注：有 unix-v6pp-tongji 这个文件夹就行。不需要有 unix-v6pp-filesystem-editor）。

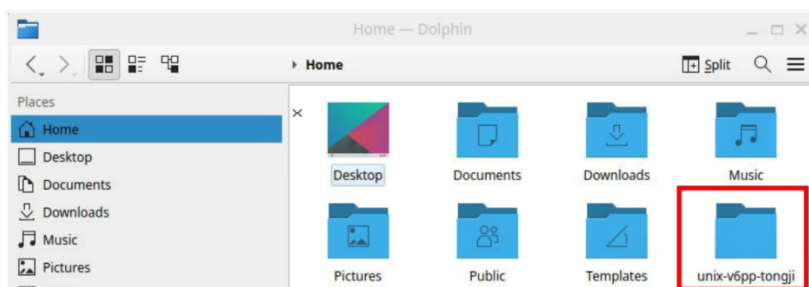


图 10：成功下载实验工具包

4.5.2. 运行 UNIX V6++

在 4.3 节中打开的桌面运行环境中，切换工作路径到 `unix-v6pp-tongji` 文件夹内。即输入如下命令：

```
cd unix-v6pp-tongji
```

执行 `make qemu` 命令，即可启动 UNIX V6++，如下图所示。

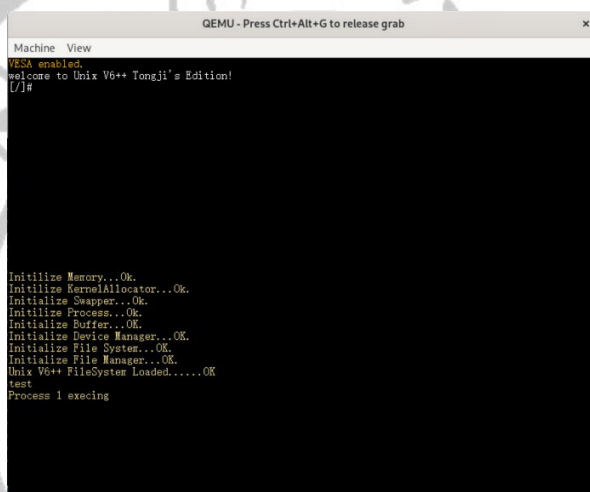


图 11：UNIX V6++启动界面

在启动的 UNIX V6++终端上可以尝试输入简单的 shell 命令，UNIX V6++中只实现了很少的几个 shell 命令。

如果 `make qemu` 命令运行后，控制台输出：

```
*** "filescanner not found. please run 'bash init.sh' first". Stop.
```

则需要补充执行命令：`bash init.sh`

一般来说，第一次下载项目后需要执行一次该指令。后续实验可直接运行 `make qemu`。

4.6. UNIX V6++调试环境的配置

为了更好的完成对 UNIX V6++代码的修改和重新编译，我们需要一个 `gdb` 的调试环境。这里我们详细介绍这个 `gdb` 调试环境的建立过程。

4.6.1. 启动 UNIX V6++的调试模式

4.5.2 节中，在桌面环境中执行命令 `make qemu` 启动的是 UNIX V6++的运行模式，如果需要调试，请在桌面环境中更换执行 `make qemug` 命令以重新启动 UNIX V6++的调试模式。之后，你会看到如图 12 的

窗口。虚拟机没有再继续运行下去，而是暂停运行后，等待来自 gdb 的调试指令。UNIX V6++的运行也暂停了。

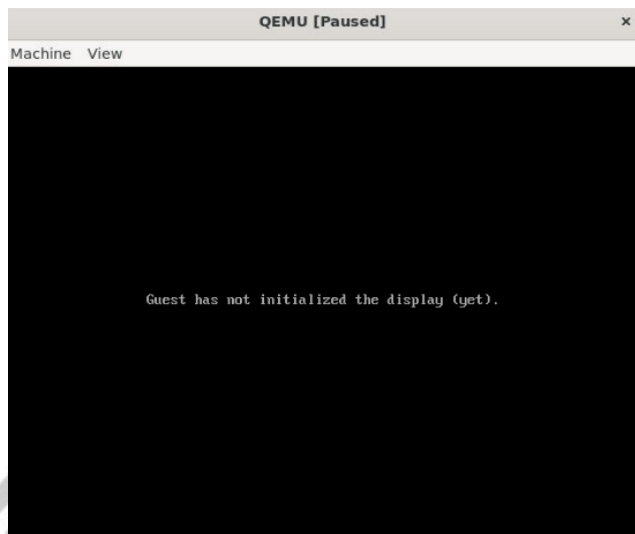


图 12: UNIX V6++启动调试模式

4.6.2. 在远程桌面环境中调试

在远程桌面环境的命令行窗口中，确保当前工作路径位于家目录（“\$”前有一个波浪号“~”）的前提下，手工输入如下命令并回车。

`code --ozone-platform=wayland unix-v6pp-tongji`

（小技巧：code --ozone-platform=wayland u<TAB>）

如果命令正确，等待 2 秒，将看到如图 13 所示的 VSCode 窗口。记得选择“Yes, I trust the authors”。

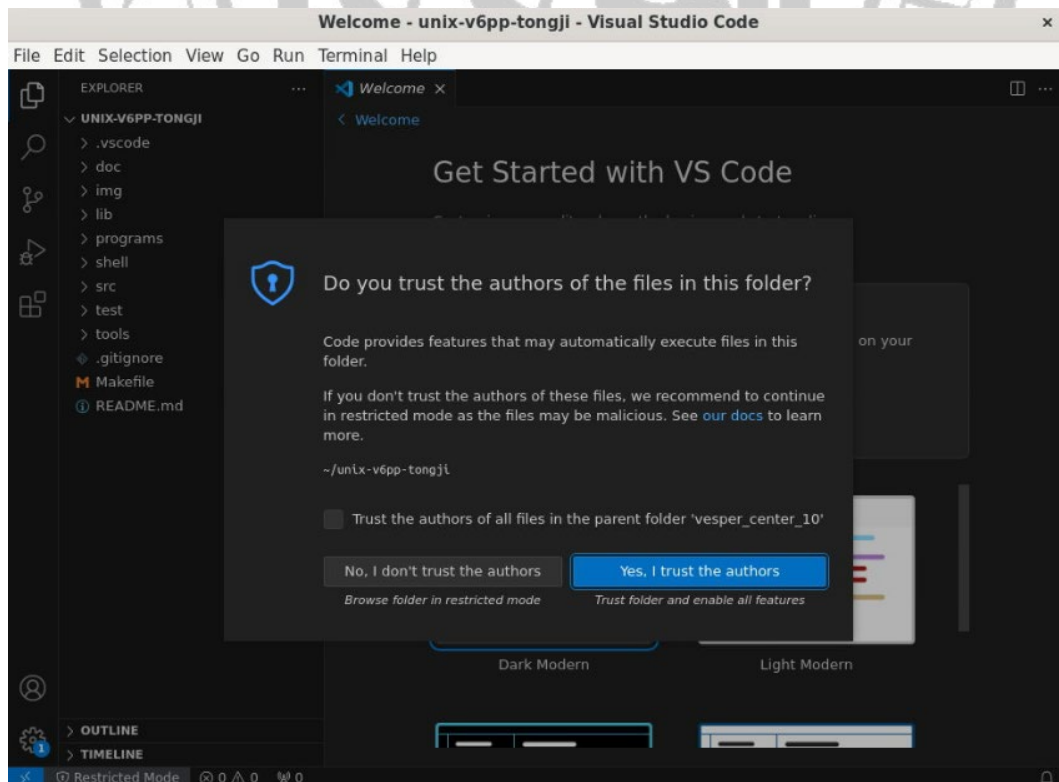



图 13: 在远程桌面环境中启动 vscode

接下来, 请安装 C++ 插件: 点击 VSCode 左侧的 , 搜索 “c++”。你将看到一个由 Microsoft 发布的 C/C++ Extension Pack。安装它即可。

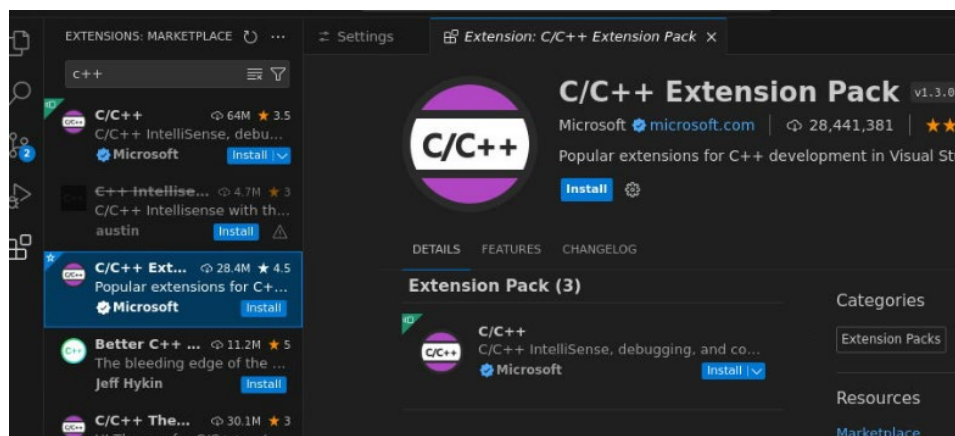


图 14: 安装 C/C++ 插件

请注意: 只安装这一个就够了, 勤安装其他插件, 服务器性能恰不消! 如果 VSCode 提示你安装 CMake、Gradle 等插件, 选择“不再提醒”即可。

4.6.3. 利用本地 VSCODE 进行远程调试

前述直接在远程桌面环境中的调试如果在配置和使用过程中出现下面的问题, 可以尝试利用本地 VSCODE 进行远程调试: (**强烈推荐!!!**)

(1) 运行 `code --ozone-platform=wayland unix-v6pp-tongji` 命令后没有反应, VSCODE 没有正常启动;

(2) 当你使用远程桌面进行调试时, 总感觉卡卡的。这是因为, 在这个过程中, 你的电脑不断将鼠标动作和键盘操作发送到实验主机, 实验主机不断将其内部画面发送给你的电脑。VSCode 等开发软件皆运行于实验主机。不难想象, 这样做会产生较大的延迟。对于键盘快速键入的操作, 这些延迟会令人很不爽。

因此, 本节将告诉你如何使用 VSCode 自带的远程连接功能开发调试。该模式下, 实验主机仅负责存储代码、编译代码、运行代码, 不再肩负编辑代码的任务。由于 VSCode 运行在你的电脑上, 你的打字体验会非常流畅舒适; 当你按下保存, VSCode 会将代码同步到服务器。该过程即使延迟高一点也不会觉得难受; 当你启动调试, 实验主机会运行 V6++, 并向你展示画面。调试控制台位于你本地的 VSCode, 依然是丝滑流畅的体验。

获取连接用户名和密码: 打开实验平台, 点开“桌面环境”, 点击你的实验环境右侧的“详细”。在这里, 你将看到 linux 用户名和 linux 登录密码 (见图 15)。

连接: 在你自己的电脑上打开 VSCode。之后, 按图 16~19 的步骤操作。键入“你的 linux 登录名@vesper-center.gardilily.com”之后按下回车。然后按照图 20~21 的步骤输入密码。

连接成功后, 选择“打开文件夹”, 打开“~/unix-v6pp-tongji”, 即可看到代码。

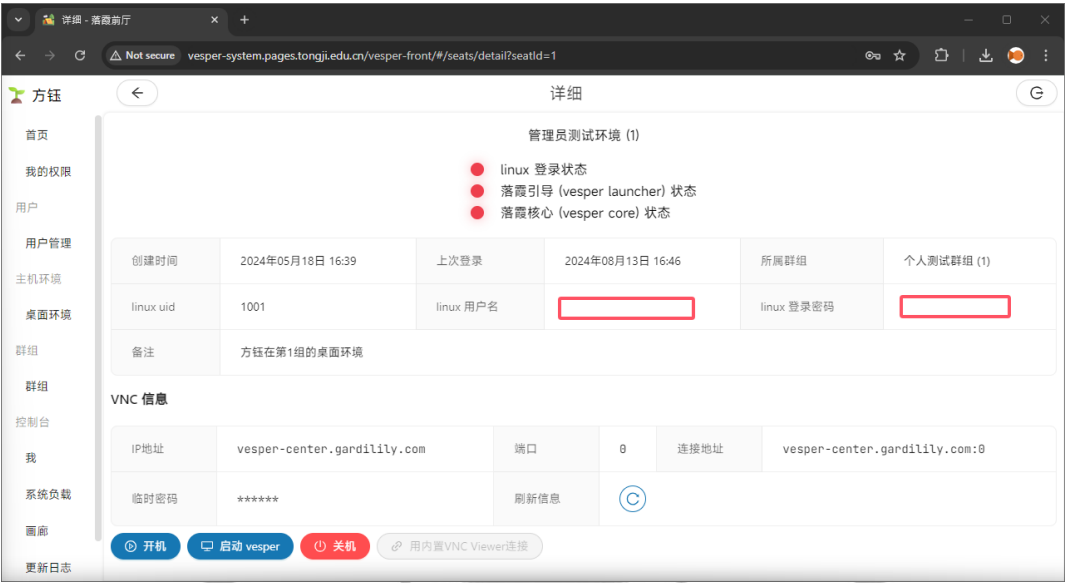


图 15: 桌面环境详情页

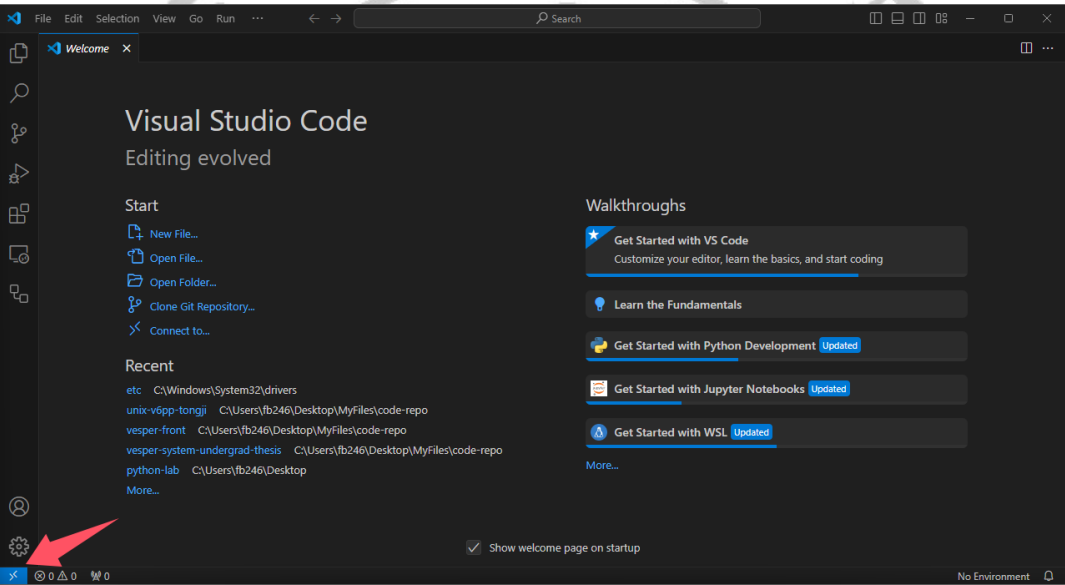


图 16

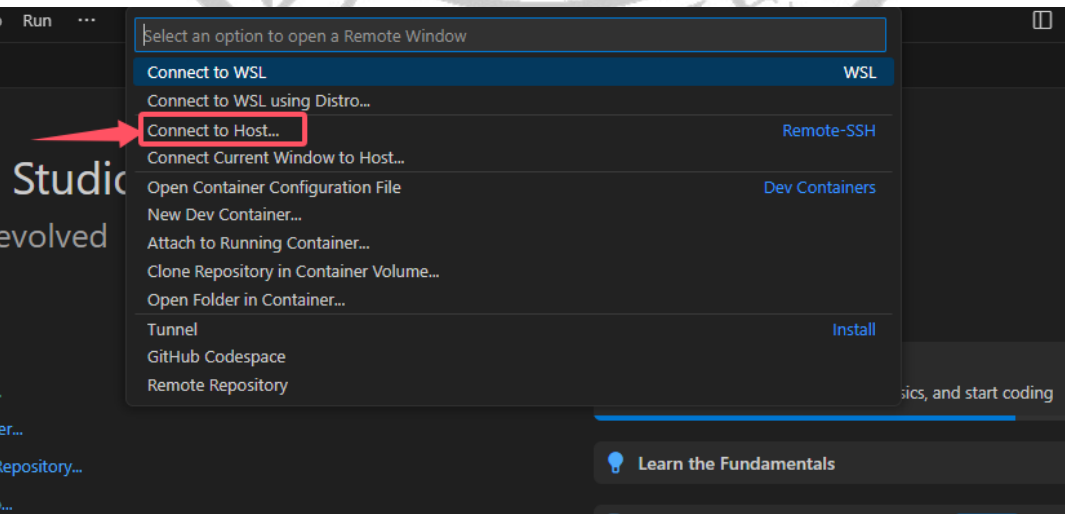


图 17

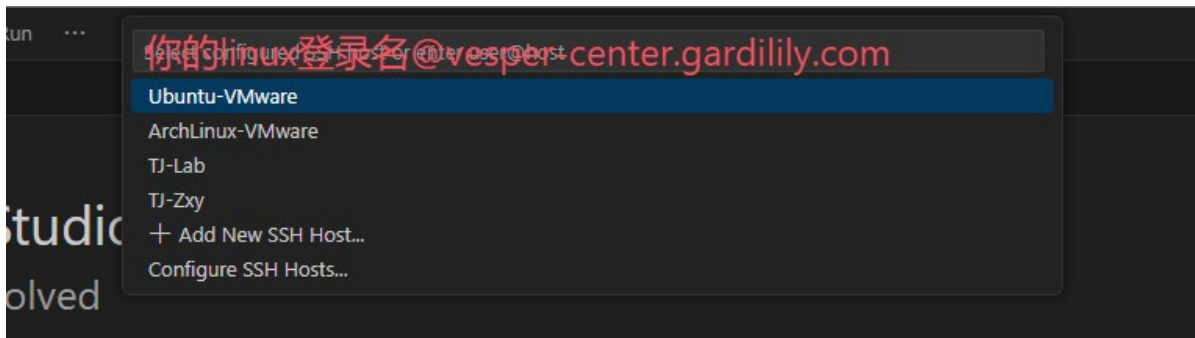


图 18

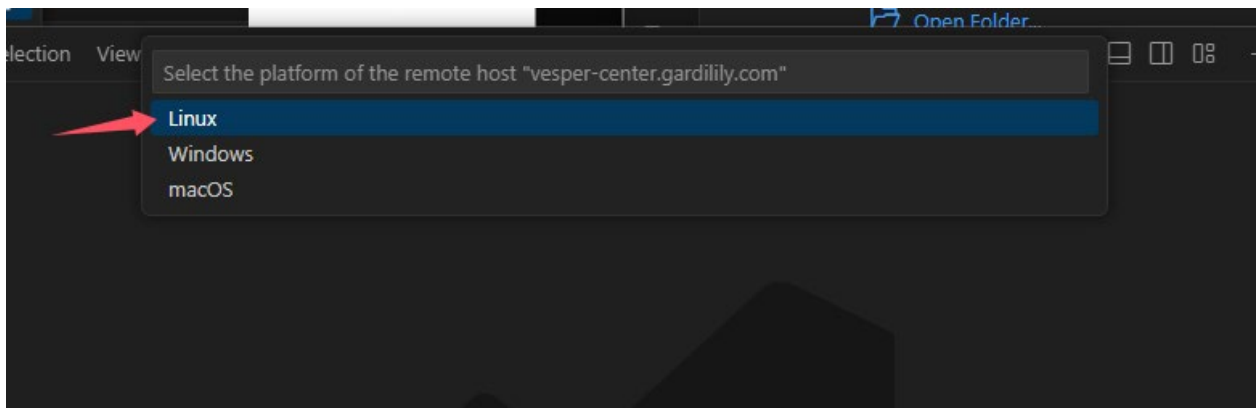


图 19

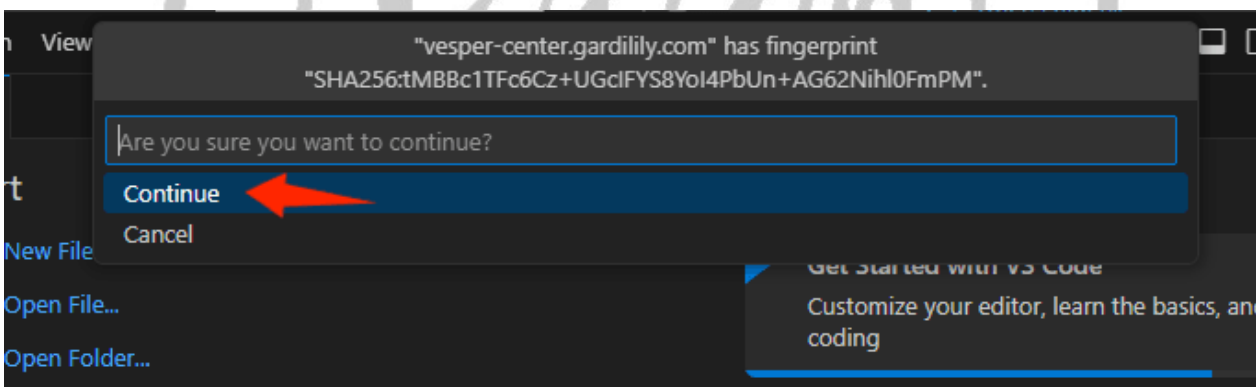


图 20

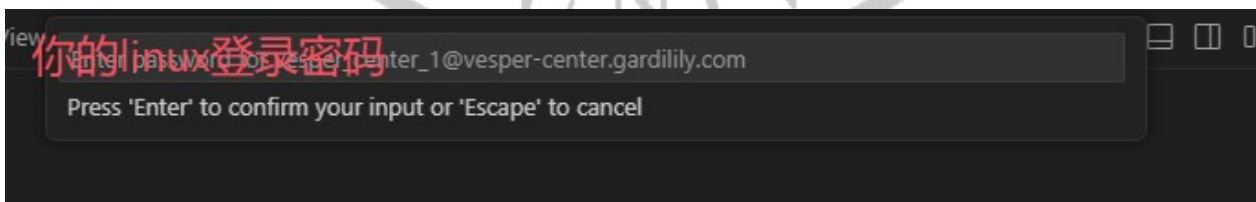


图 21

4.6.4. 开始调试

不论你采用上述两种方法的哪一种，至此，你已建立好了在基于 vscode 的调试环境。可以在 vscode 的窗口中编译或调试代码，在远程桌面环境中查看运行结果（如图 22 所示）。

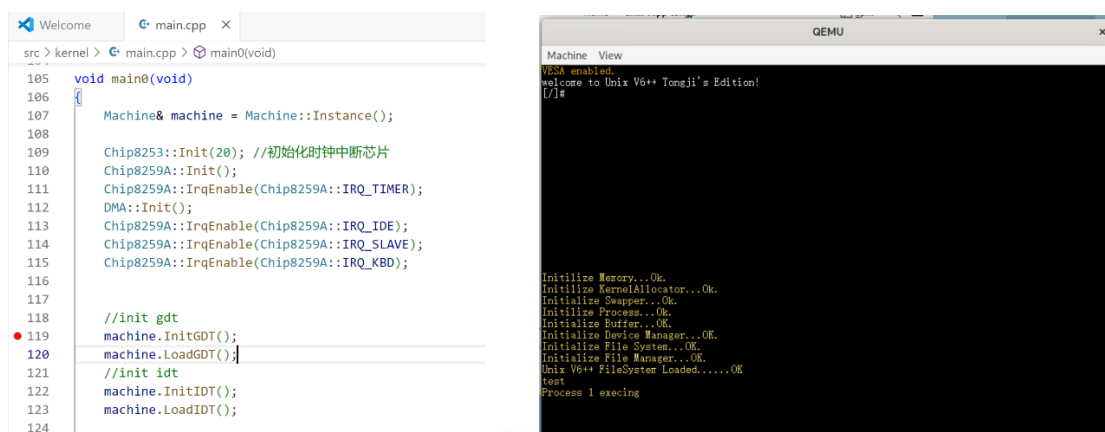
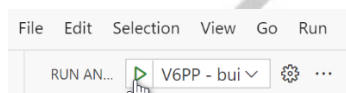


图 22: 在 vscode 环境中调试程序, 远程桌面 UNIX V6++环境中查看运行结果

在 VSCode 窗口, 点击左侧的  可以进入调试控制台。点击上方的绿色按钮以启动调试。



例如, 如果你在如图 22 所示的代码位置添加了断点, 启动调试运行后, 可以看到程序运行在这里停止, 4.6.1 节中, 远程桌面环境中等待调试指令的 UNIX V6++界面也发生了变化(如图 22 所示)。

此时, 如图 23 所示, 可以在 vscode 中可以查看各个变量的值, 或者通过“Registers”标签查看各个寄存器的值。

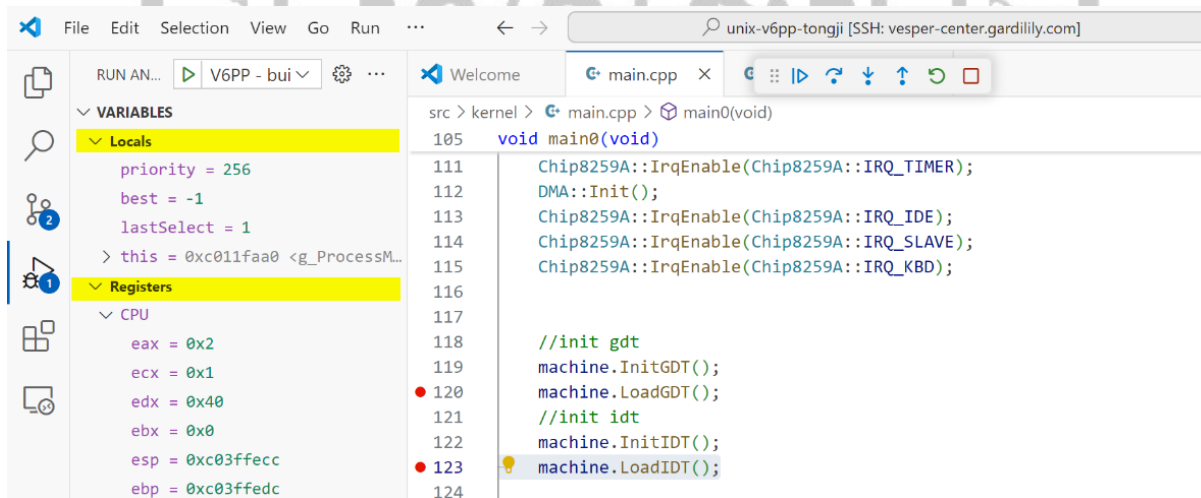


图 23

默认情况下, 调试只能调试内核。如果需要调试你自己的程序, 或者 Shell, 请修改.vscodc/launch.json 文件。如图 24 所示, 如需调试内核, 则将第 9 行语句注释去掉, 并将第 10 行注释; 反之, 如需调试自己的应用程序, 则将第 9 行语句注释, 而将第 10 行语句的注释去掉, 并修改为自己的程序即可。例如, 你要调试 fork 程序, 则将 program 设置成 “\${workspaceFolder}/target/objs/apps-elf/fork”。

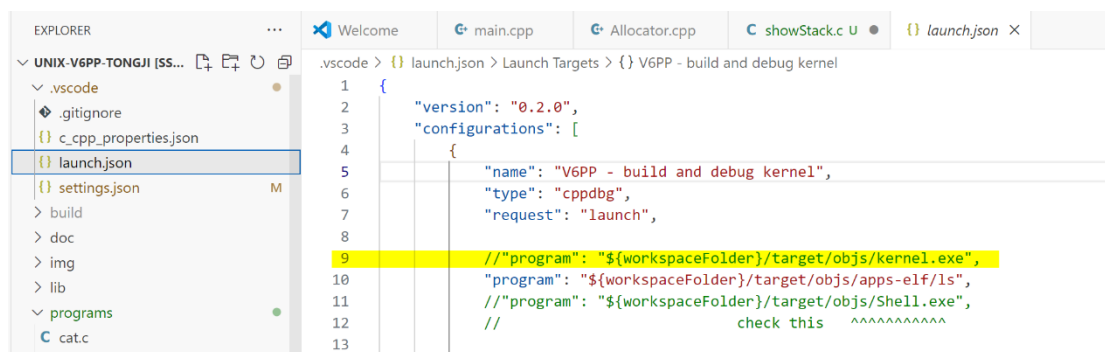


图 24: 修改调试目标

4.6.5. 关于 VSCode 远程调试的若干小技巧

本节我们给出几个读者可能马上会用到的 `vscode` 运行调试过程中的一些实用小技巧, 其他更多的秘籍请大家持续发现、学习和共享, 我们也会在后续实验需要用的时候陆续介绍给大家。

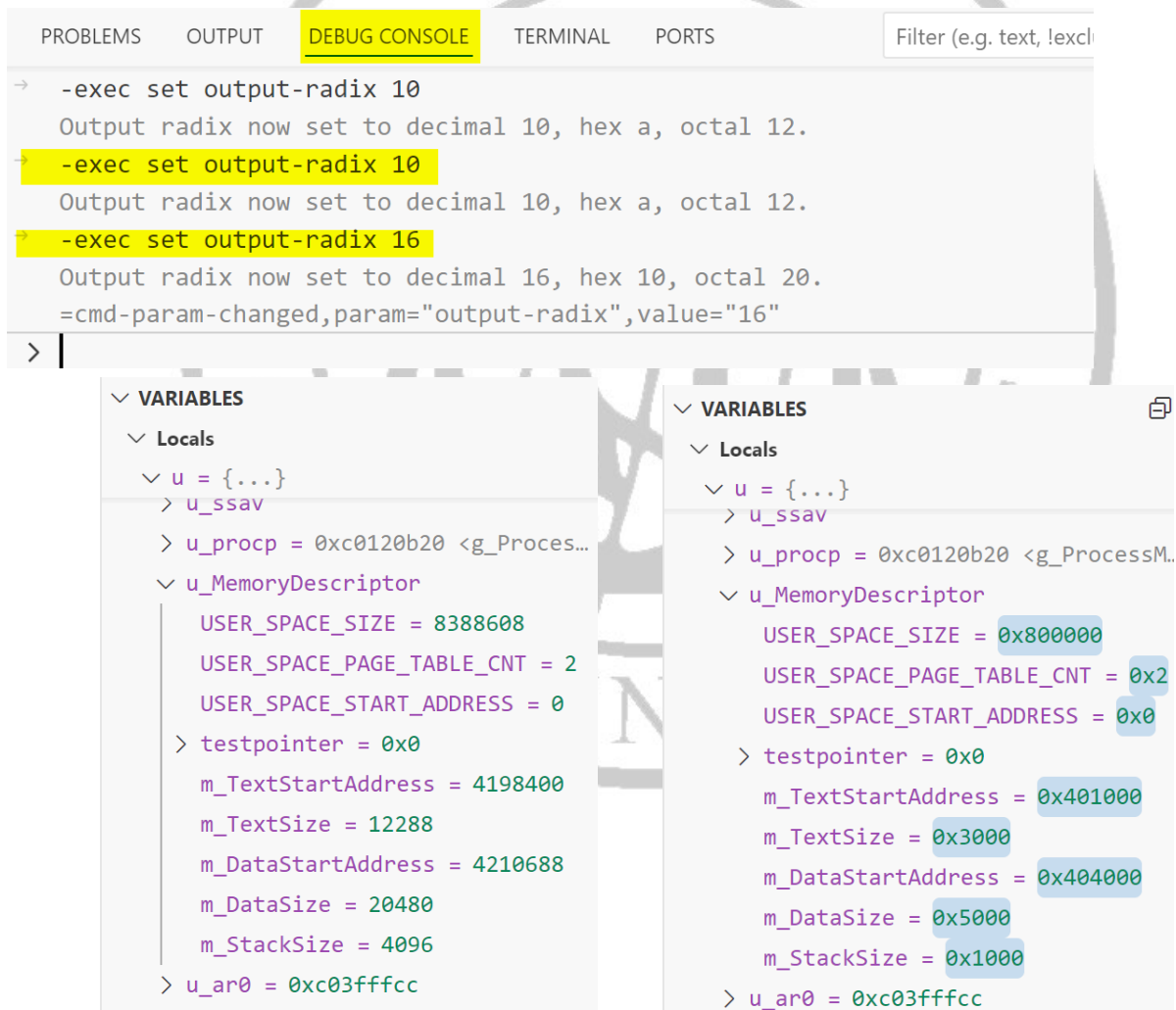


图 25: 设置变量显示的进制

(1) 调试过程中查看变量的值

在 `vscode` 上启动调试过程。你可以在合适的位置打一个断点，让程序暂停一下。此时，在变量监视器内可以看到局部变量、寄存器等的值。根据需要，我们可能会希望以 10 进制或 16 进制显示变量。这时，打开调试命令行。通过输入命令：

`-exec set output-radix n`

可以将调试器显示模式设置到 `n` 进制。如 25 图所示，我们可以看到调试命令行中输入的命令以及不同进制下变量显示的不同。

(2) 调试过程中查看汇编代码

调试过程中，当程序停在某一处断点时，用图 26 的方法，选中某一个函数，点击右键后，可以打开该函数的汇编指令。

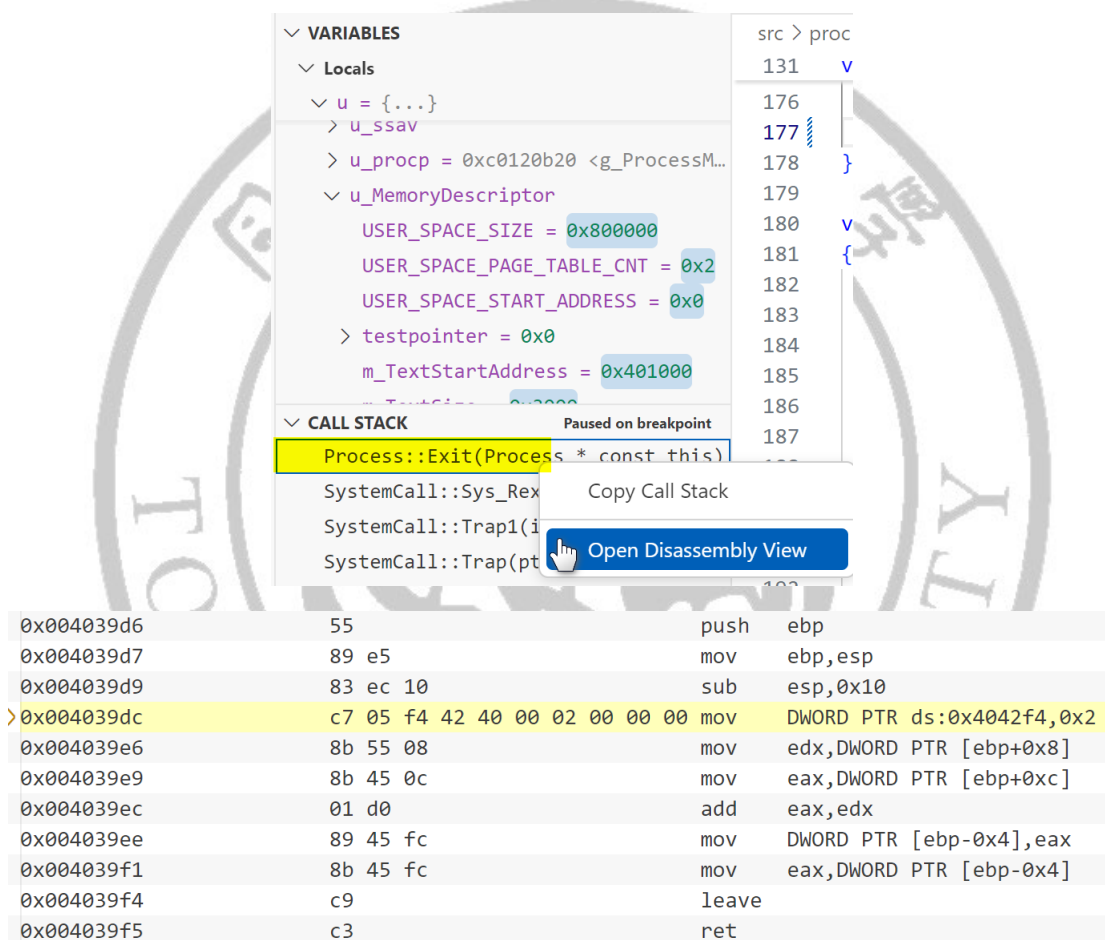


图 26：调试过程中查看汇编代码

(3) 调试过程中查看内存单元

调试过程中，有时候我们希望能查看某一个内存单元的值，这一点可以通过在调试命令行中输入下面这个命令实现对内存单元的查看：

`-exec x /nfu addr`

其中，`x` 表示要查看内存，`nfu` 表示含义为以 `f` 格式打印从 `addr` 开始的 `n` 个长度单位为 `u` 的内存值。图 27 中，我们给出了上述在不同参数设置下对同一个起始位置的内存单元的查看结果，读者可以自行体会不同参数设置的含义，并在后续实验中灵活使用。


```

→ -exec x /4xb 0xC0203004
0xc0203004: 0x65 0xe0 0x40 0x00
→ -exec x /4db 0xC0203004
0xc0203004: 101 -32 64 0
→ -exec x /4ob 0xC0203004
0xc0203004: 0145 0340 0100 0
→ -exec x /4tb 0xC0203004
0xc0203004: 01100101 11100000 01000000 00000000
→ -exec x /4xw 0xC0203004
0xc0203004: 0x0040e065 0x0040f065 0x00410065 0x00412067
→ -exec x /4dw 0xC0203004
0xc0203004: 4251749 4255845 4259941 4268135
→ -exec x /4ow 0xC0203004
0xc0203004: 020160145 020170145 020200145 020220147
→ -exec x /4tw 0xC0203004
0xc0203004: 0000000010000001110000001100101 0000000010000001111000001100101
001100101 0000000010000010010000001100111

```

图 27: 以不同的参数设置查看内存单元

(4) 中文字符显示乱码

如果 vscode 代码显示中出现中文注释字符乱码的情况, 如图 28 所示。可按途中的标号顺序操作, 即可正常显示。

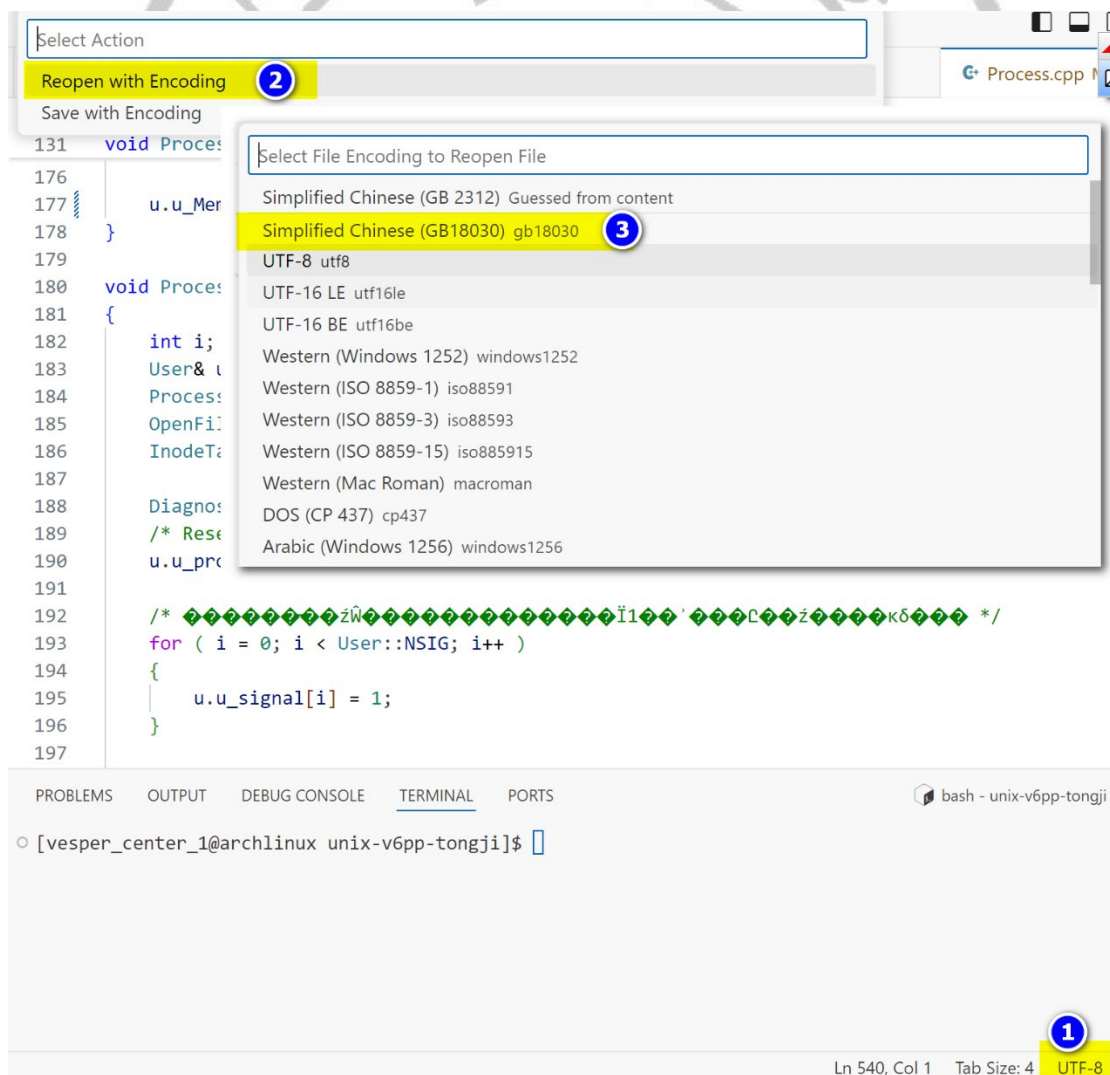


图 28

5. 实验报告要求

- (1) (1 分) 依照实验指导，在教师规定时间内完成首次登录，密码修改和主机名修改；
- (2) (1 分) 完成 UNIX V6++ 的运行和调式环境的启动与配置，关键步骤截图说明。

