苏州大学实验报告

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 院系 | 计算机学院 | | 年级专业 | | 21计科 | | 姓名 | 方浩楠 | 学号 | 2127405048 |
| 课程名称 | | Linux操作系统 | | | | | | | 成绩 |  |
| 指导教师 | | 曹敏 | | 同组实验者 | | 无 | | 实验日期 | 2023.10.19 | |

|  |  |
| --- | --- |
| 实 验 名 称 | 实验4 |

1. 实验目的

熟悉并掌握Linux系统中用户组管理命令groupadd和groupdel的基本使用。

通过练习，理解并熟悉文本编辑器vim的不同模式及其操作命令。

学习并掌握Linux系统中常见的压缩工具gzip、bzip2、xz和zip的基本使用方法，包括压缩和解压缩操作。

对比分析gzip、bzip2、xz三种压缩工具的特点和区别。

掌握在Linux系统中利用tar命令进行打包+压缩和解压缩的操作。

理解并掌握Linux系统中包管理工具yum、apt和dnf的基本功能及其区别。

通过实际操作，学习并掌握如何使用yum工具搜索、安装和卸载软件包。

学习并掌握如何从源码编译和安装软件，理解编译和安装的基本流程。

1. 实验内容

groupadd练习:

使用-g选项创建一个新的用户组，并指定组ID。

不使用-g选项创建一个新的用户组，系统自动分配组ID。

groupdel练习:

删除上述创建的两个用户组。

vim使用练习:

设计一个场景，实现vim的一般模式、编辑模式和命令模式下的操作，按照表7-1至表7-5的命令进行操作练习。

压缩工具使用练习:

设计一个场景，实现gzip、bzip2和xz压缩工具的压缩及解压缩操作。

压缩工具特点和区别:

描述gzip、bzip2和xz压缩工具的特点，并分析它们之间的区别。

打包+压缩练习:

设计一个场景，实现使用tar命令进行打包+gzip压缩、打包+bzip2压缩、打包+zip压缩以及相应的解压缩操作。

zip压缩练习:

设计一个场景，实现zip压缩及解压缩操作。

包管理工具详述:

详细介绍yum、apt和dnf包管理工具的基本功能及它们之间的区别。

yum工具操作练习:

使用yum工具搜索包含关键词tidy的RPM包并安装，安装后再使用yum工具将其卸载。

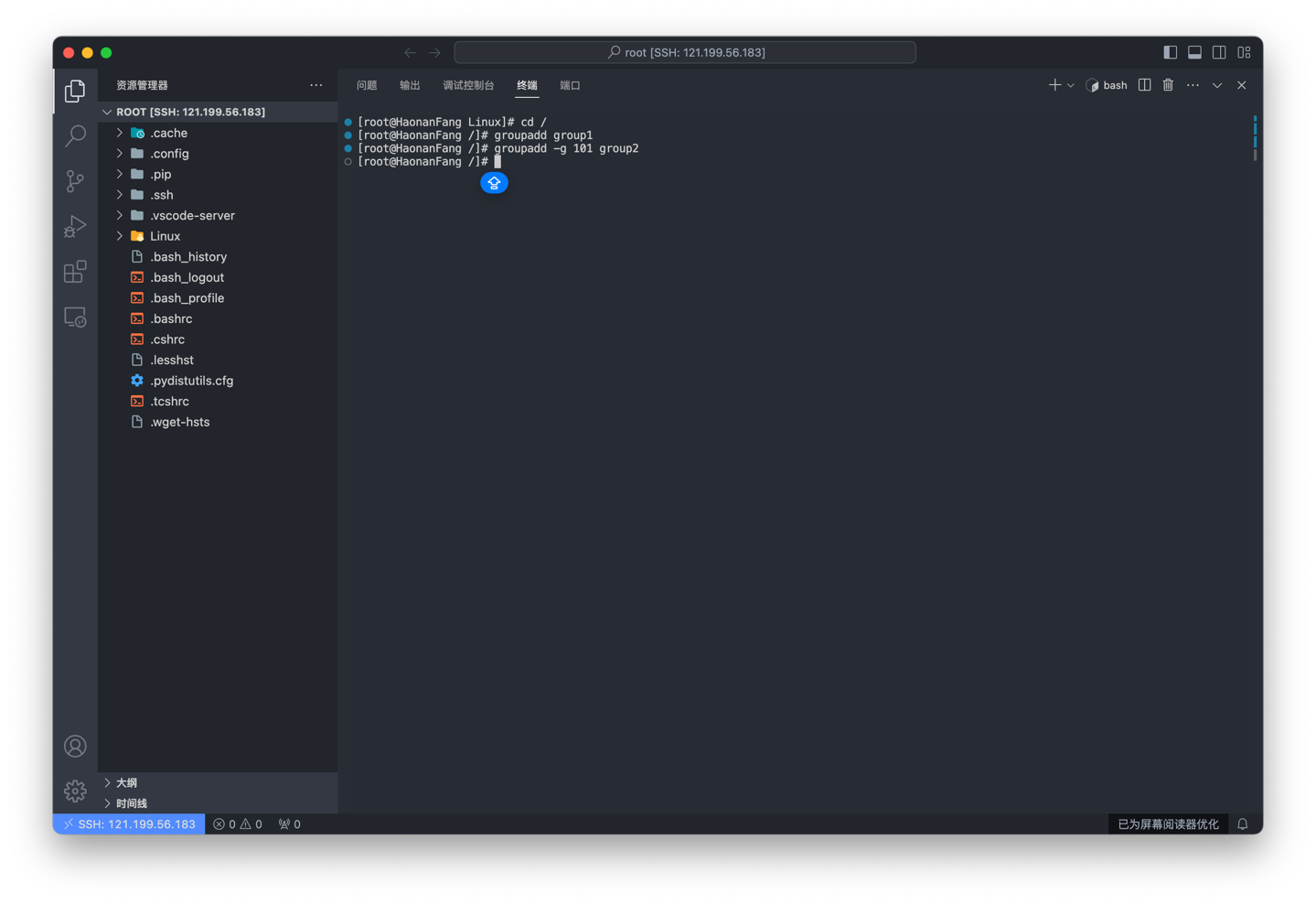
源码编译和安装练习:

选择一个开源项目，进行源码的编译和安装练习，记录操作过程和遇到的问题。

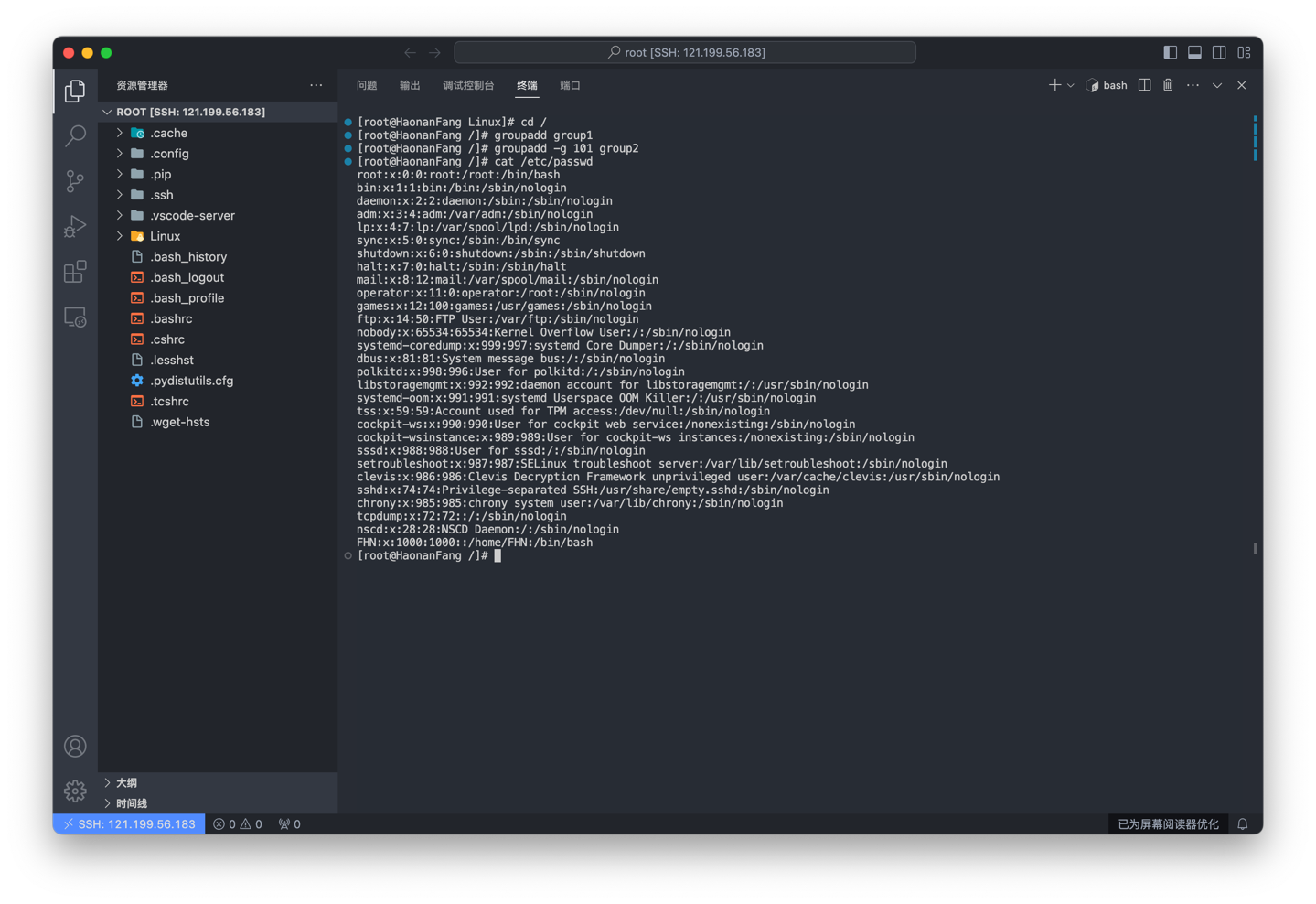
1. 实验步骤和结果

1.groupadd的练习

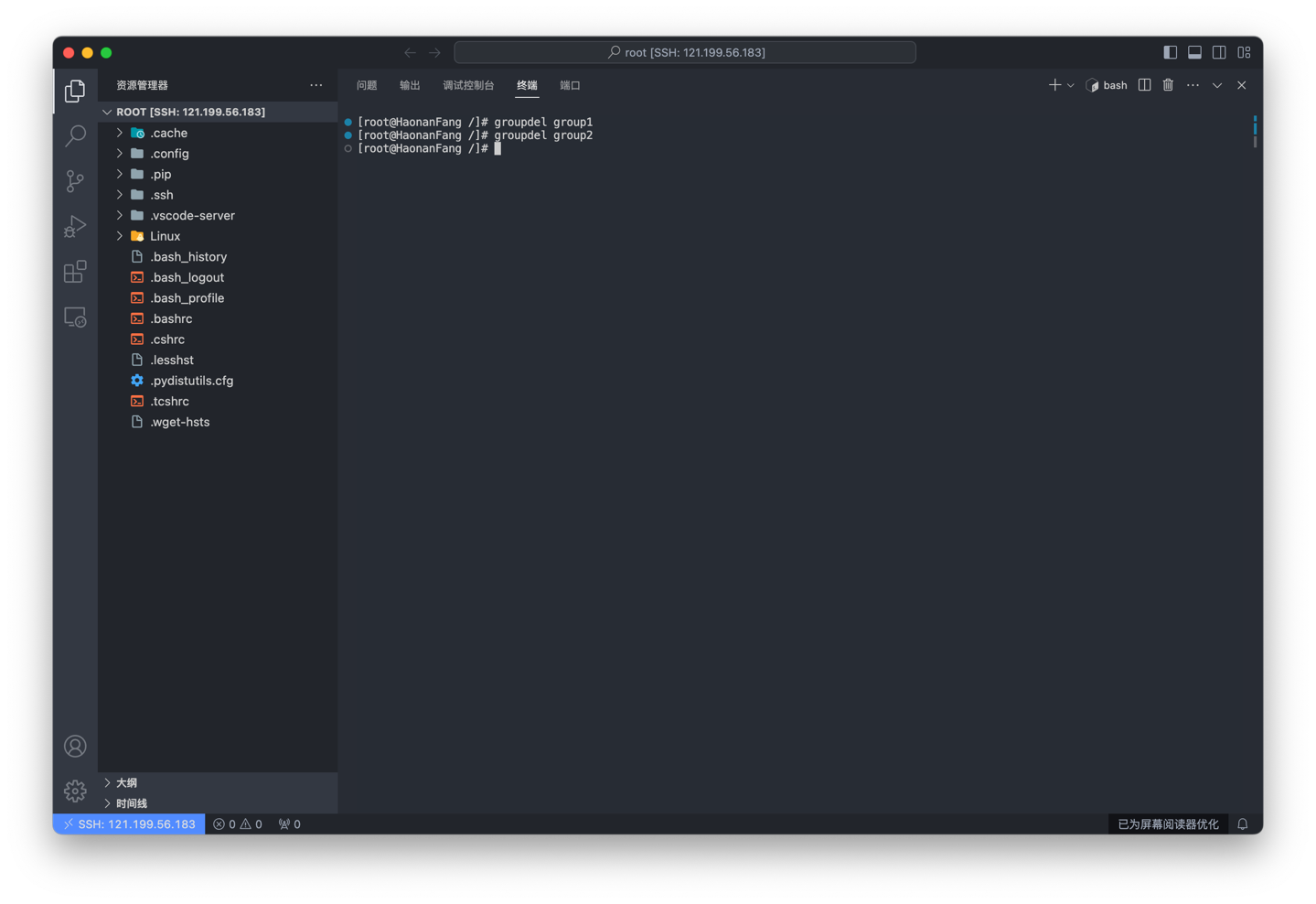
分别插入group1和group2,其中group2有-g参数, 用户组指定新的组标识号为101



插入照片后查询用户组状态:



2.groupdel



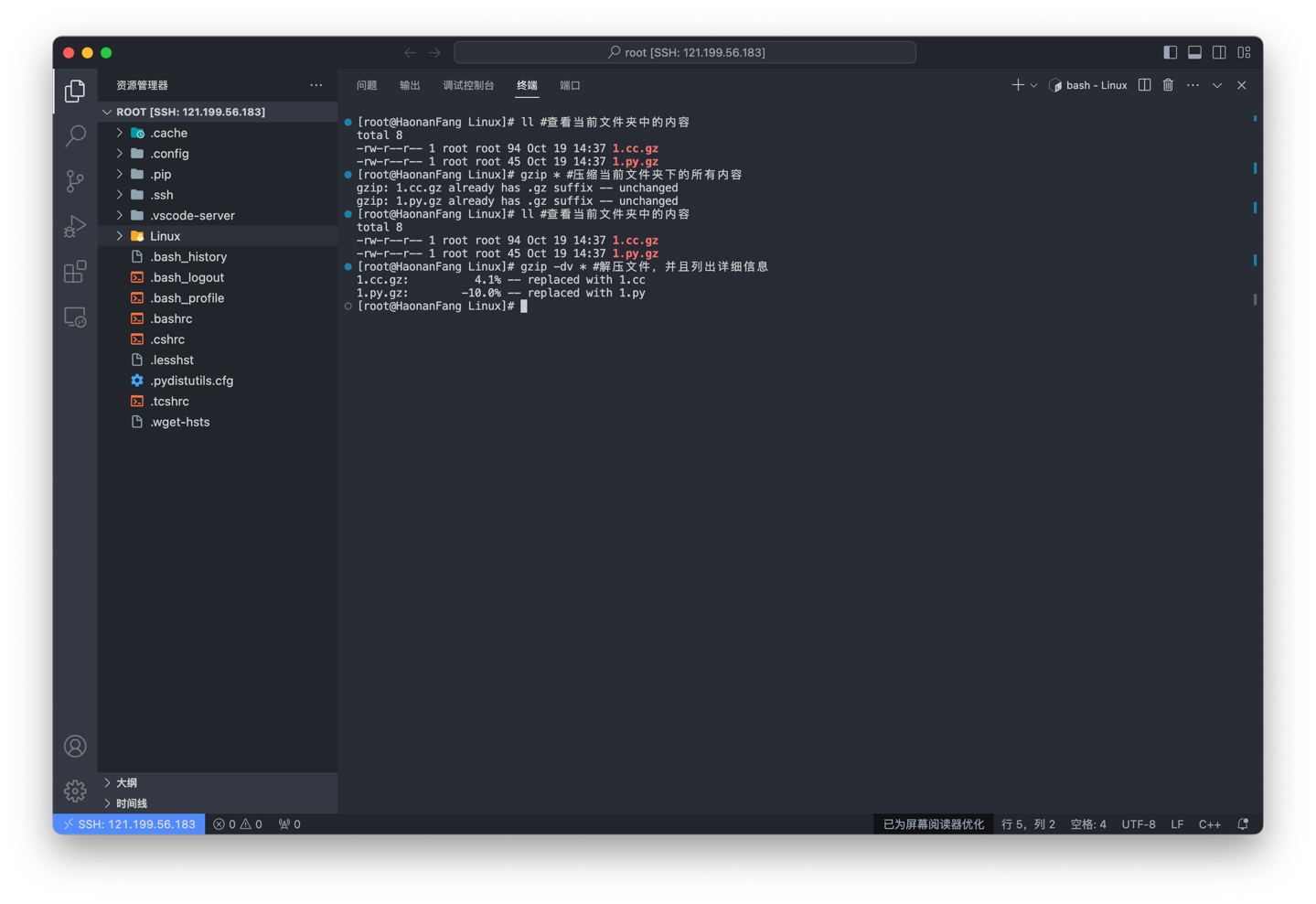
使用groupdel删除了刚刚创建的两个用户组

3.vim操作

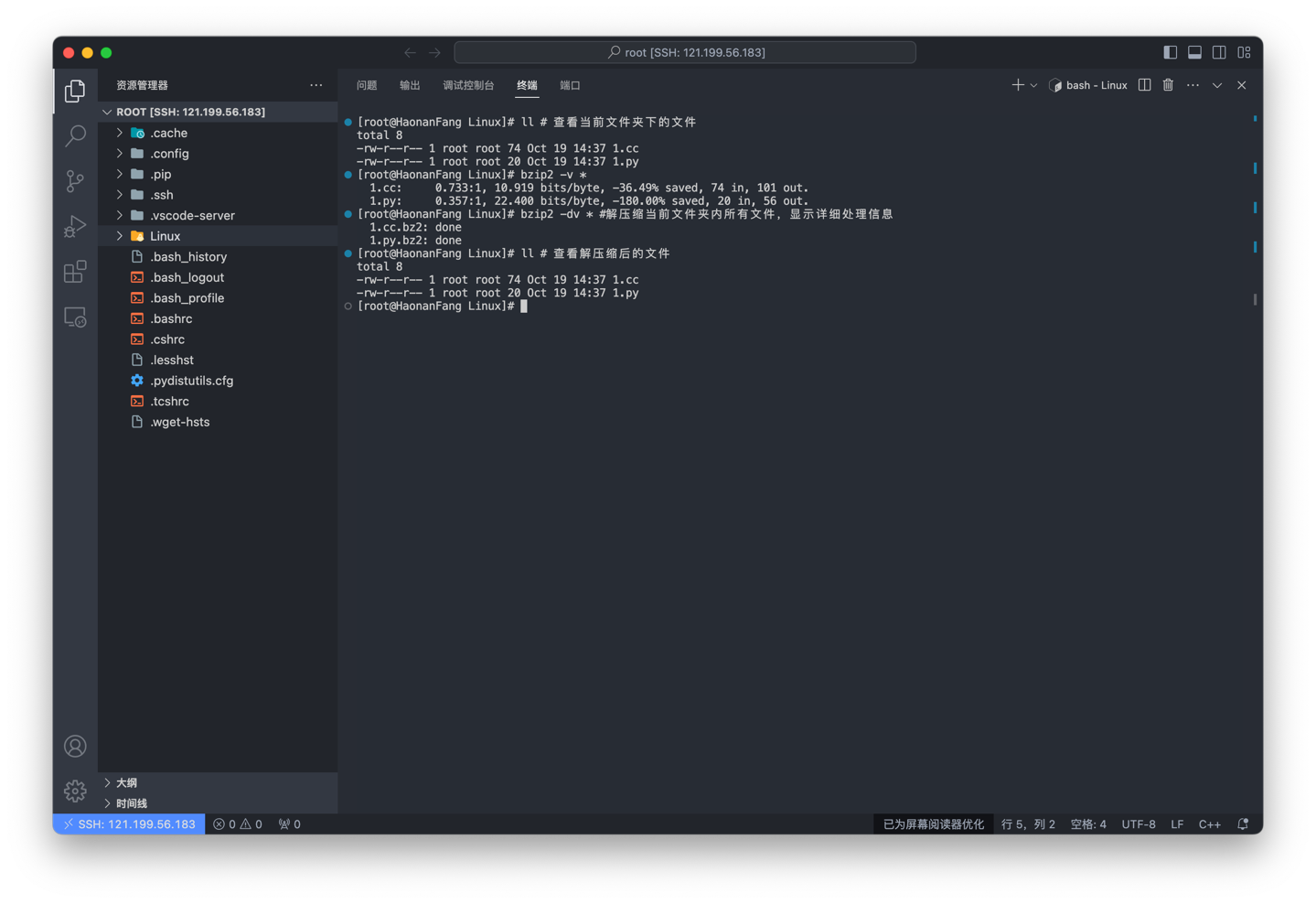
TODO

4. gzip压缩工具、bzip2压缩工具、xz压缩工具的使用

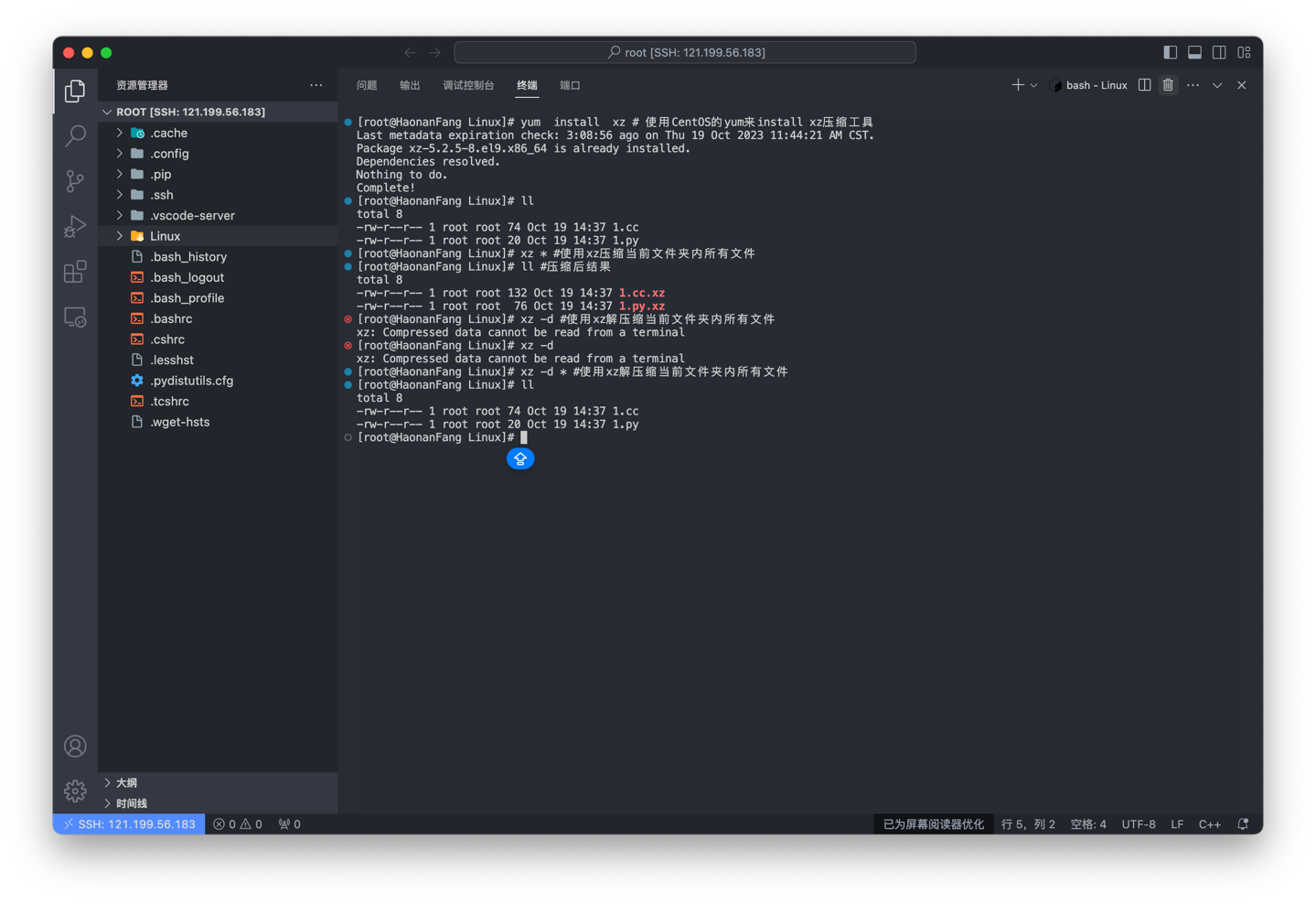
4.1 gzip



4.2 bzip2



4.3 xz



5.三种压缩工具的特点和区别

gzip:

压缩时间: gzip的压缩时间随着压缩级别的增加而稍微增加，但是相对较快。

压缩比: 相对于bzip2和xz，gzip的压缩比较低，这意味着它的压缩文件大小较大。

内存使用: gzip的内存使用量最低，使其成为内存受限环境中的理想选择。

通用性: gzip是非常普遍和广泛接受的，因此几乎所有的系统上都能很好地支持它​。

bzip2:

压缩时间: bzip2的压缩时间随着压缩级别的增加变化不大，但是通常比gzip长。

压缩比: bzip2的压缩比优于gzip，但略低于xz，这意味着它可以产生比gzip更小的压缩文件，但比xz稍大。

通用性: bzip2也是广泛接受和使用的，但可能不如gzip普遍​。

xz:

压缩时间: xz的压缩时间随着压缩级别的增加而显著增加，特别是在压缩级别3以上，它的压缩时间大大增加，但是可以提供最佳的压缩比。

压缩比: xz提供了最佳的压缩比，可以产生最小的压缩文件。

内存使用: xz在解压缩时的内存使用量低，但在压缩时的内存使用量高。这使得它在内存受限的环境中不是很理想的选择，尤其是在压缩时。

通用性: xz可能不如gzip和bzip2普遍，但它的压缩效率很高，使得它成为大文件压缩的理想选择​​

6. 打包+压缩的使用

7. zip压缩的使用

文本

描述已自动生成

8. 详述yum、apt、dnf，并阐述它们之间的区别

yum、apt和dnf是三种不同的软件包管理器，主要用于在Linux操作系统中安装、更新和删除软件。它们之间的区别主要在于支持的Linux发行版和包管理方式。

yum：

yum是Red Hat、CentOS和Fedora等发行版中的软件包管理器。它基于RPM包管理，可以从指定的服务器自动下载RPM包并安装，能自动处理依赖关系，并一次性安装所有依赖的软件包，无须繁琐地一次次下载、安装。yum会自动管理依赖关系，底层调用rpm管理软件包。

apt：

apt是Debian、Ubuntu等发行版中的软件包管理器。它是适用于deb包管理式的操作系统，主要用于自动从互联网的软件仓库中搜索、安装、升级、卸载软件或操作系统。

dnf：

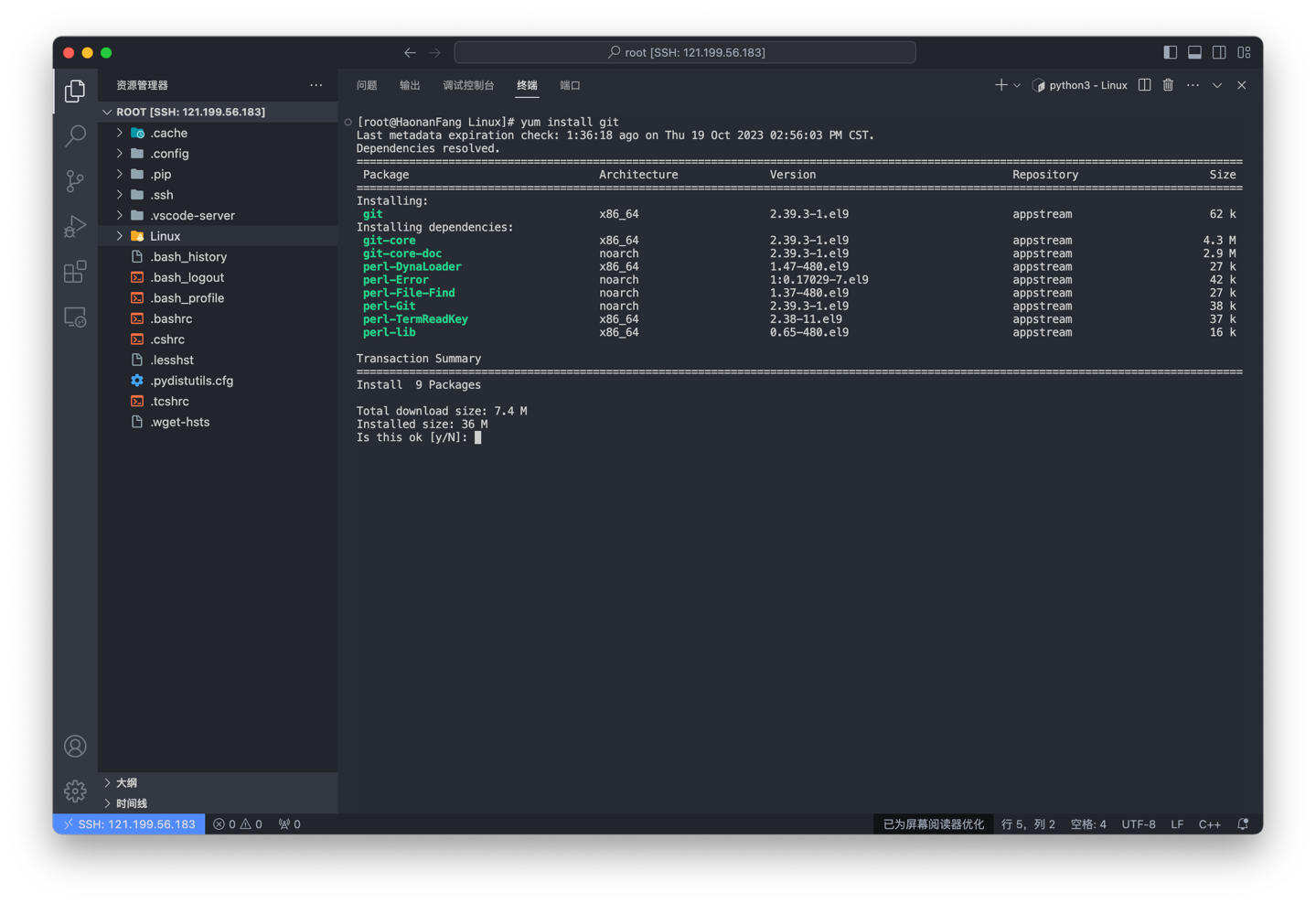
dnf是Fedora、CentOS等发行版的新一代RPM软件包管理器，全称为Dandified yum。它在Fedora 18版本中首次出现，目前在Fedora 22及之后的版本中取代了yum，成为默认的包管理器。dnf克服了yum的一些瓶颈，提升了包括用户体验、内存占用、依赖分析、运行速度等多方面的内容。dnf使用Hawkey库，该库解析RPM依赖性以在客户端计算机上运行查询。 它们基于libsolv构建, libsolv是一种使用可满足性算法的程序包相关性求解器。

9. 采用yum工具搜索包含关键词tidy的RPM包并安装，安装后再使用yum工具将其卸载

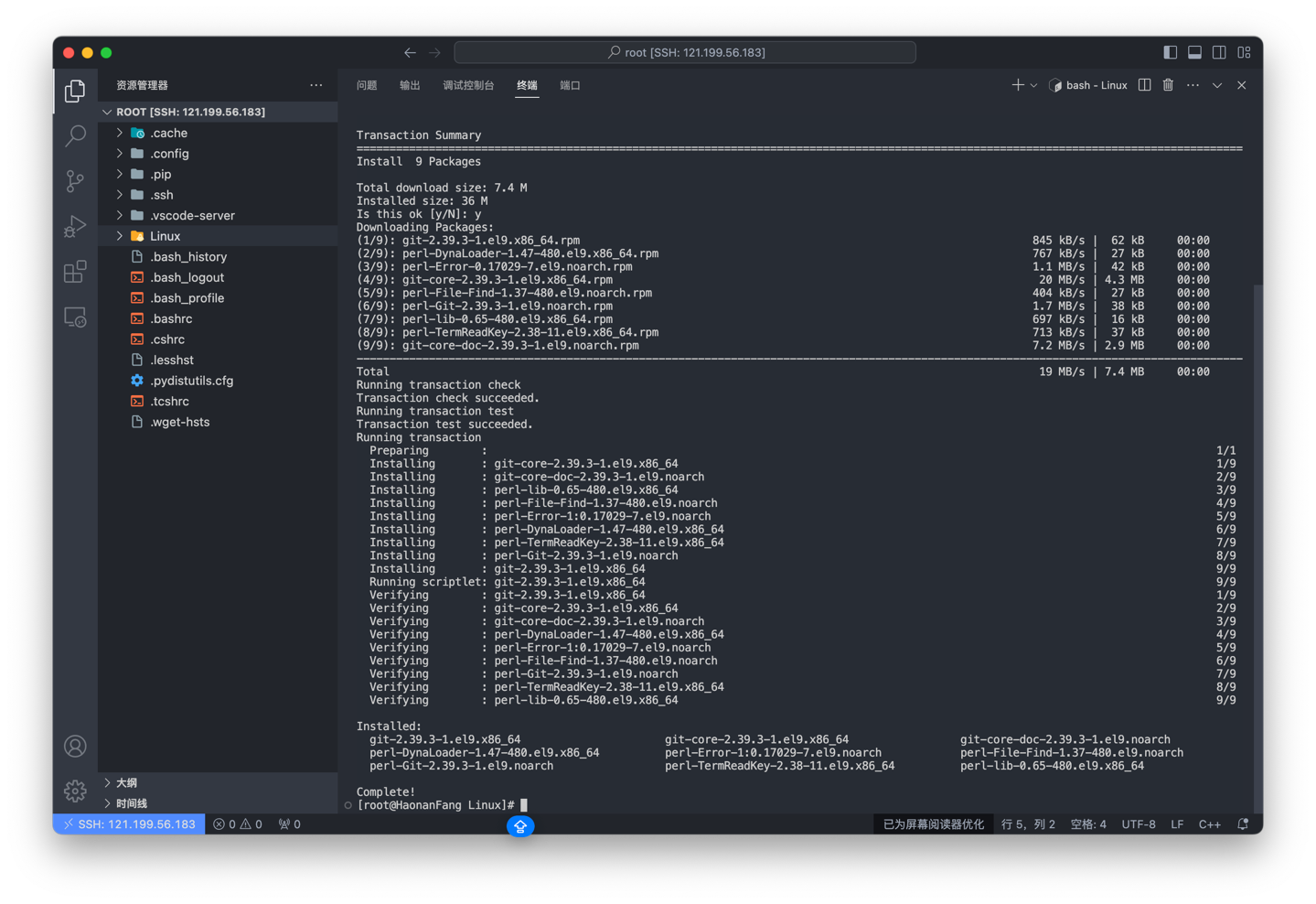
使用yum搜索git



使用yum下载git

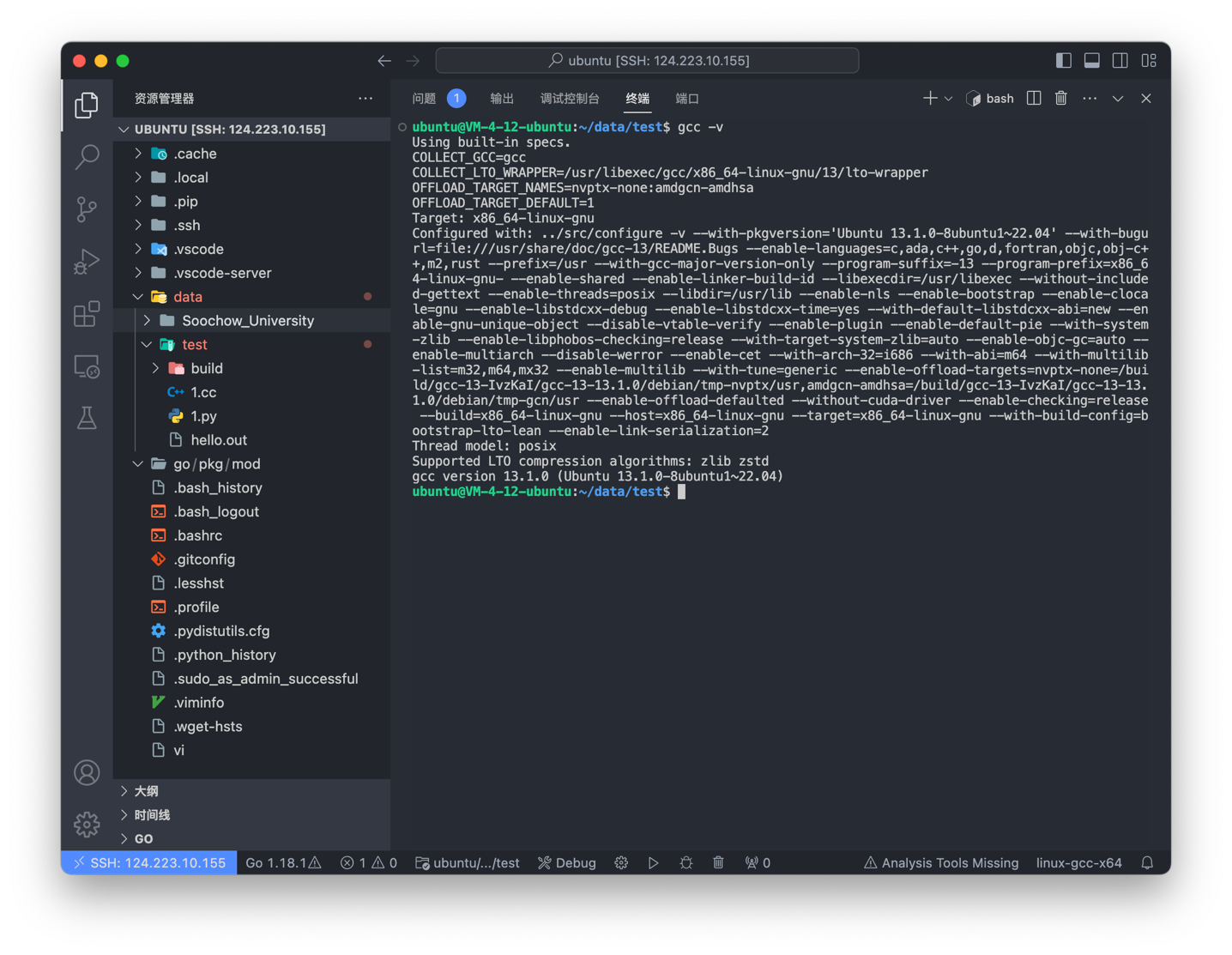


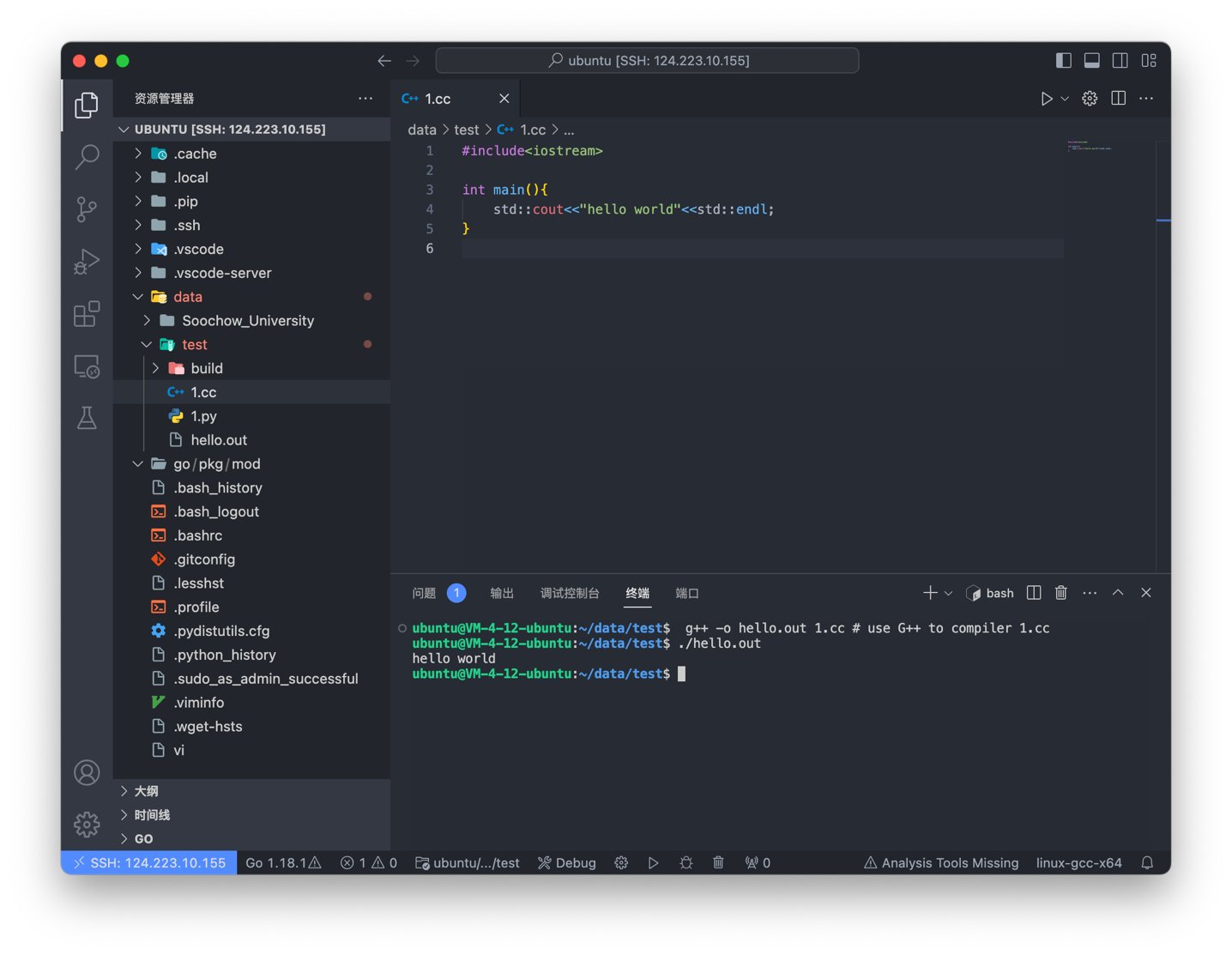
此时输入y



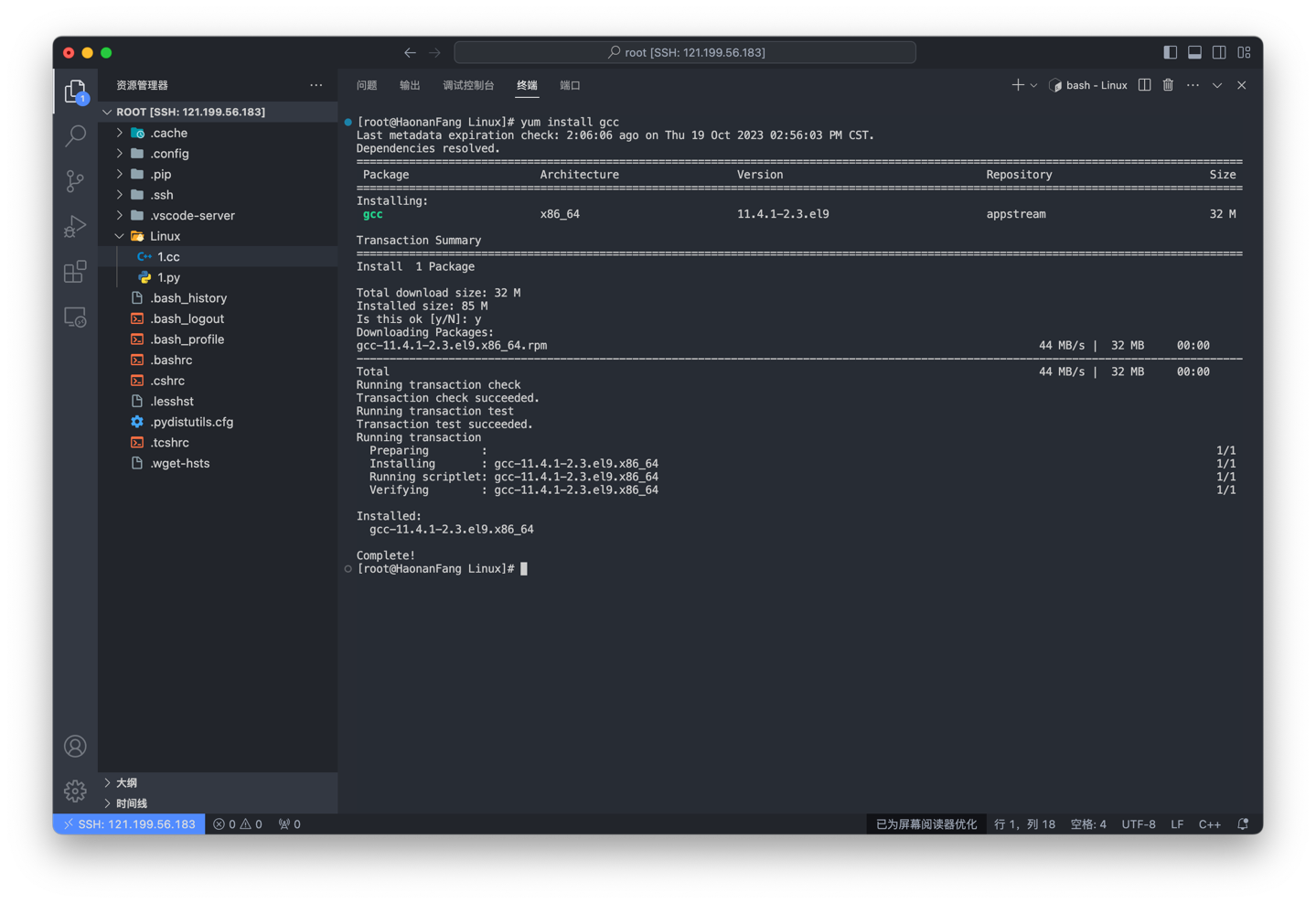
10. diy一个源码的编译和安装

使用GCC 13.1 编译运行一个名为1.cc的C++文件

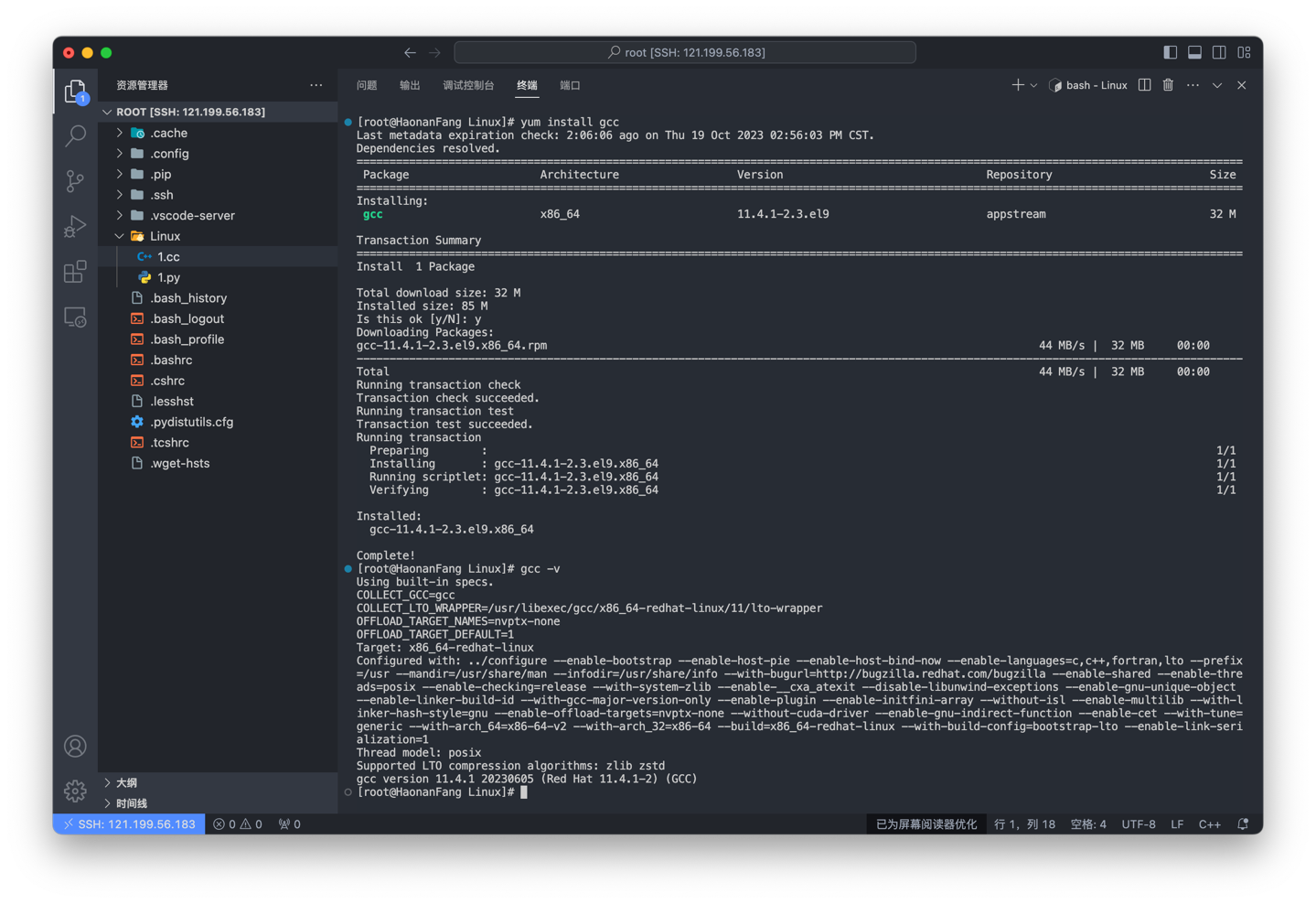




安装GCC过程:



可以看到此时gcc安装完成.



输入gcc -v后显示出当前的gcc版本

1. 实验总结