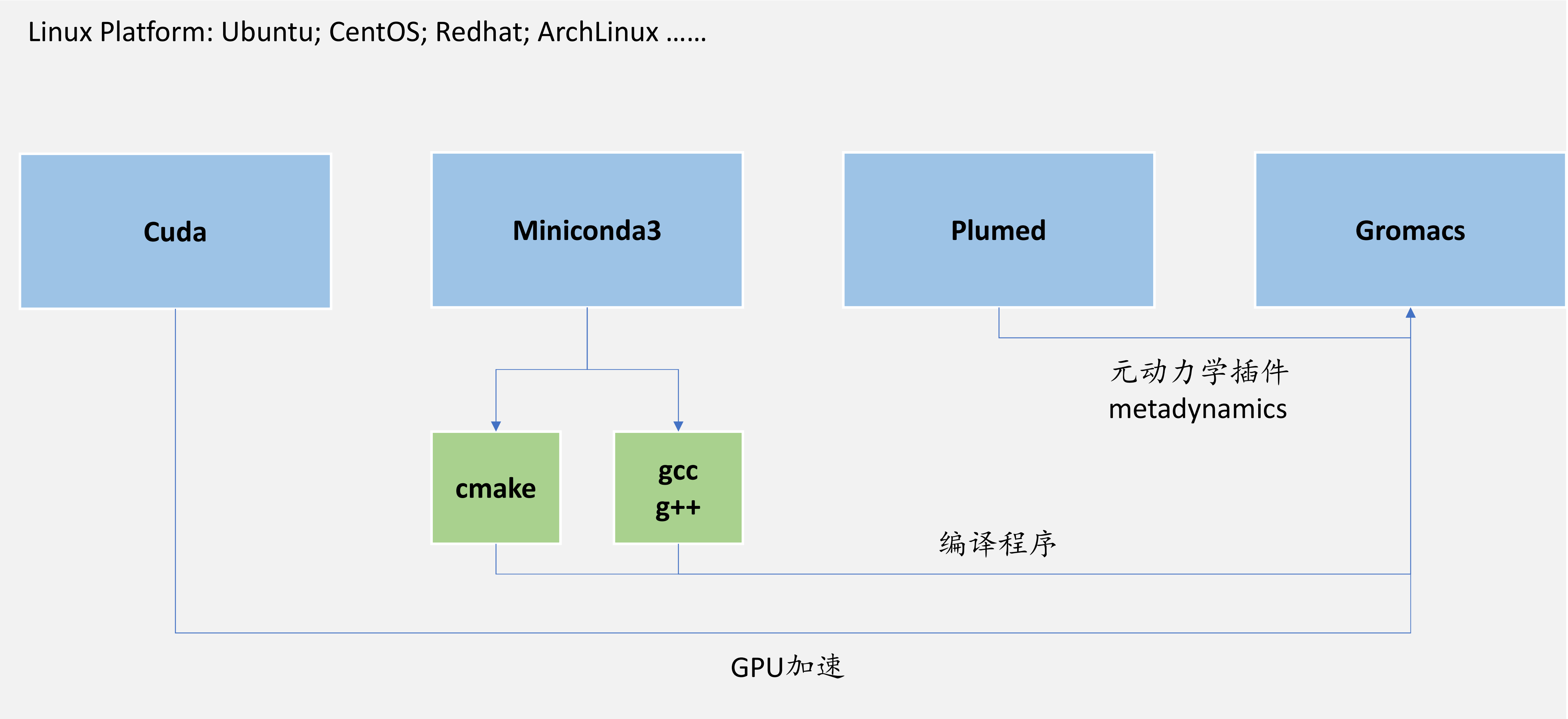
Gromacs Install

Naf Guo
2025

What we need to install gromacs



Linux Platform: Ubuntu; CentOS; Redhat; ArchLinux ……

Cuda

Miniconda3

Plumed

Gromacs

cmake

gcc
g++

元动力学插件
metadynamics

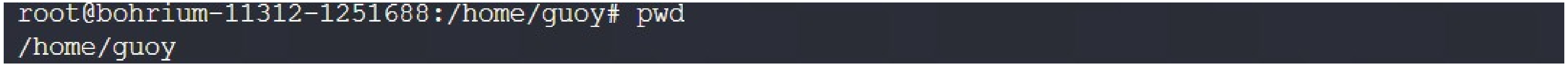
编译程序

GPU加速

Some simple Linux commands

pwd

pwd 命令意为打印当前所在路径，所谓路径(path)，是指当前所在目录的绝对位置



```
root@bohrium-11312-1251688:/home/guoy# pwd
/home/guoy
```

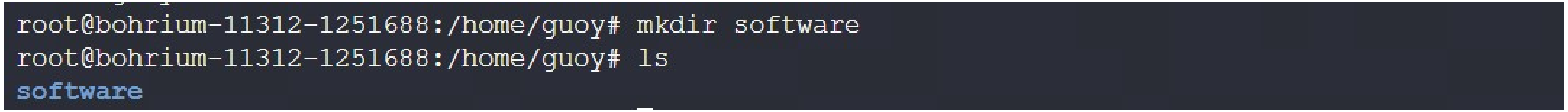
比如上面，我所在的地址就是/home文件夹下的/guoy文件夹下；一般这个文件夹是特殊的，称为用户的家目录。

ls

ls 命令会将当前路径下所有文件和文件夹的名字打印出来

mkdir dir_name

mkdir 命令意为make directory，创建一个文件夹，dir_name需要被替换为你希望的文件夹名字

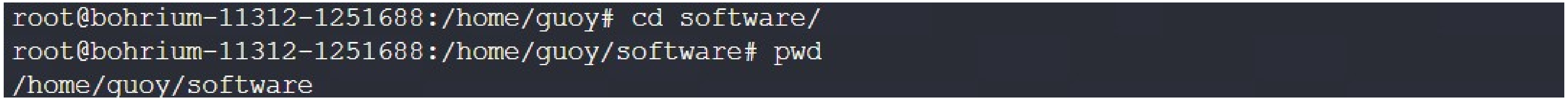


```
root@bohrium-11312-1251688:/home/guoy# mkdir software
root@bohrium-11312-1251688:/home/guoy# ls
software
```

如上，我先在/home/guoy下建立了一个叫software的文件夹，再使用ls命令把/home/guoy下的文件和文件夹都打印了出来，这里只有一个，就是我刚建立的文件夹

cd dir_name

cd 命令意为change directory，把路径改到某个文件夹下



```
root@bohrium-11312-1251688:/home/guoy# cd software/
root@bohrium-11312-1251688:/home/guoy/software# pwd
/home/guoy/software
```

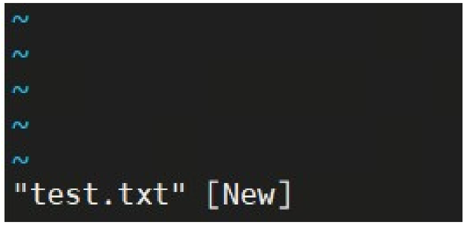
Some simple Linux commands

vim

vim 是linux下一个用于编辑文本的软件

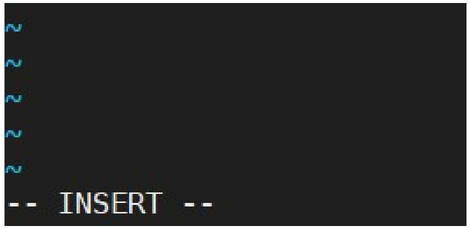
vim test.txt

此时进入是不能直接打字的，需要敲击i键进入insert模式；
当编辑结束以后，先敲击ESC键，再依次敲击:wq!就可以保存并退出了。



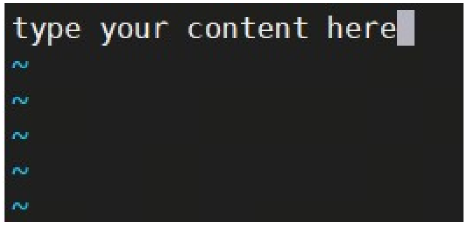
```
~
~
~
~
~
"test.txt" [New]
```

刚进入时的左下角



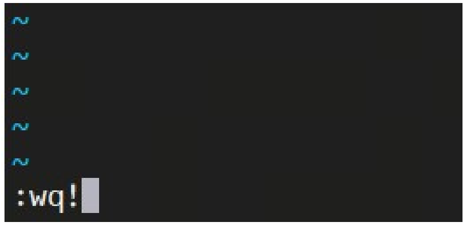
```
~
~
~
~
~
-- INSERT --
```

敲击i键进入insert模式，
此时可以进行编辑了



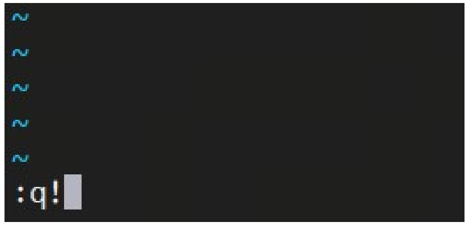
```
type your content here
~
~
~
~
~
```

进入insert模式以后可
以进行正常的文档编辑



```
~
~
~
~
~
:wq!
```

编辑完以后保存并退出,:和!需要用
shift键哈，这里必须是英文输入法

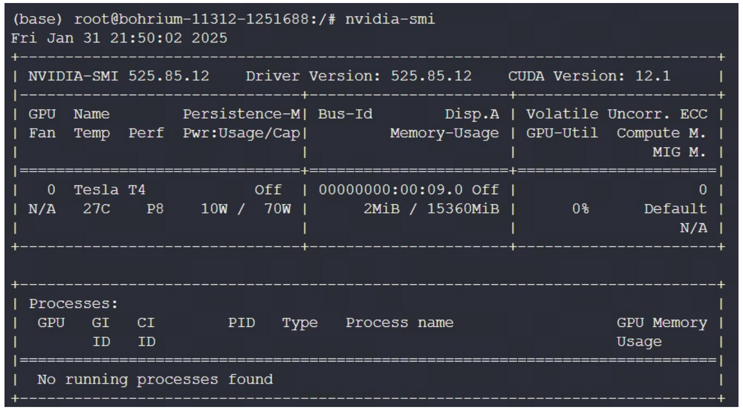


```
~
~
~
~
~
:q!
```

不保存直接退出

CUDA

nvidia-smi

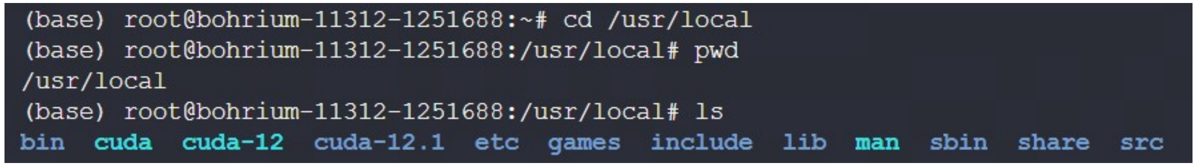


```
(base) root@bohrium-11312-1251688:/# nvidia-smi
Fri Jan 31 21:50:02 2025
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 525.85.12    Driver Version: 525.85.12    CUDA Version: 12.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:09.0 Off |                    0 |
| N/A   27C    P8    10W /  70W |      2MiB / 15360MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

用nvidia-smi命令可以检查本地有没有安装cuda，以及cuda版本；

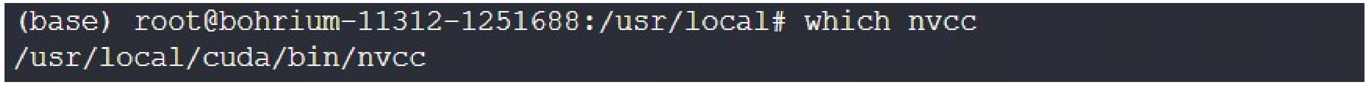
但如果你用的是计算集群，有登录节点和计算节点的区别，那登录节点可能是没有显卡的，因为计算都要提交到计算节点上完成

一般cuda都被安装在/usr/local里，可以检查一下看有没有cuda



```
(base) root@bohrium-11312-1251688:~# cd /usr/local
(base) root@bohrium-11312-1251688:/usr/local# pwd
/usr/local
(base) root@bohrium-11312-1251688:/usr/local# ls
bin  cuda  cuda-12  cuda-12.1  etc  games  include  lib  man  sbin  share  src
```

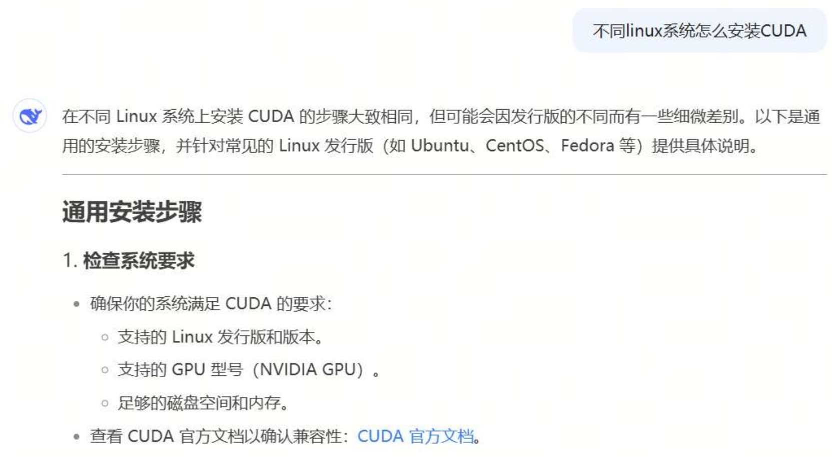
没找到的话可以用which命令查找一下nvcc的位置



```
(base) root@bohrium-11312-1251688:/usr/local# which nvcc
/usr/local/cuda/bin/nvcc
```

CUDA

CUDA的安装比较麻烦，一般还是推荐有经验的人来做，可以问大模型比如deepseek



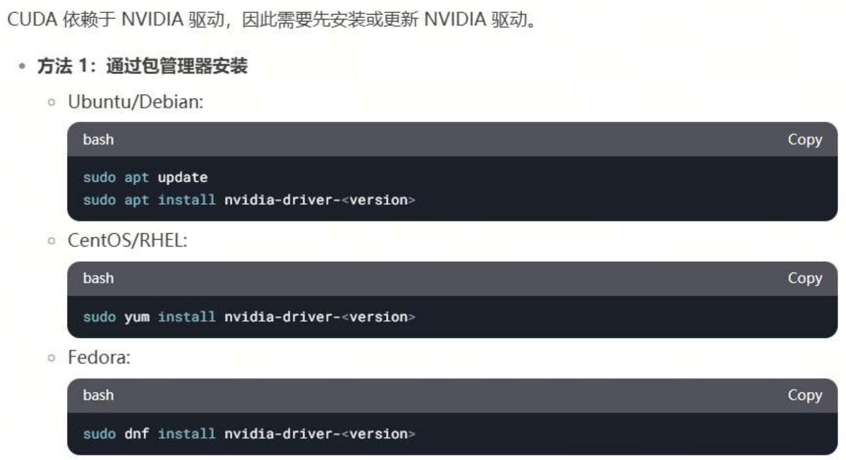
不同linux系统怎么安装CUDA

在不同 Linux 系统上安装 CUDA 的步骤大致相同，但可能会因发行版的不同而有一些细微差别。以下是通用的安装步骤，并针对常见的 Linux 发行版（如 Ubuntu、CentOS、Fedora 等）提供具体说明。

通用安装步骤

1. **检查系统要求**

- 确保你的系统满足 CUDA 的要求：
  - 支持的 Linux 发行版和版本。
  - 支持的 GPU 型号（NVIDIA GPU）。
  - 足够的磁盘空间和内存。
- 查看 CUDA 官方文档以确认兼容性：CUDA 官方文档。



CUDA 依赖于 NVIDIA 驱动，因此需要先安装或更新 NVIDIA 驱动。

- **方法 1：通过包管理器安装**
  - Ubuntu/Debian:

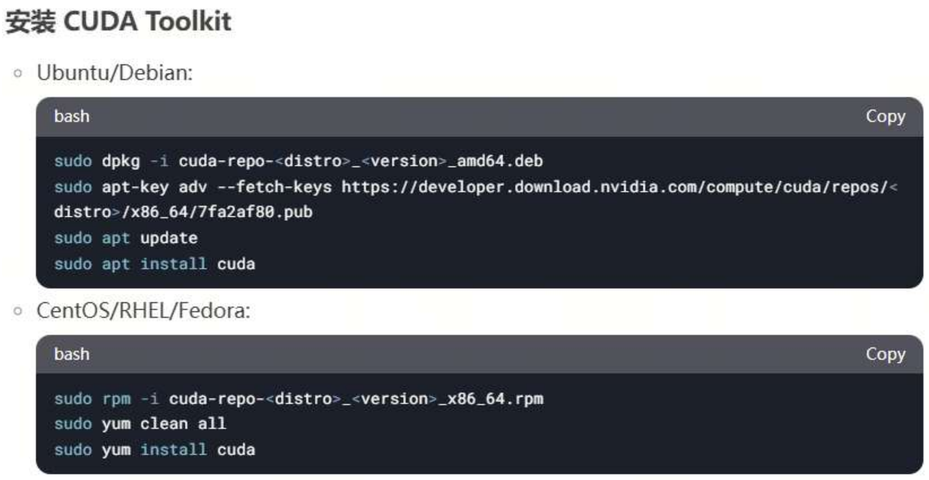
```bash
sudo apt update
sudo apt install nvidia-driver-<version>
```

  - CentOS/RHEL:

```bash
sudo yum install nvidia-driver-<version>
```

  - Fedora:

```bash
sudo dnf install nvidia-driver-<version>
```



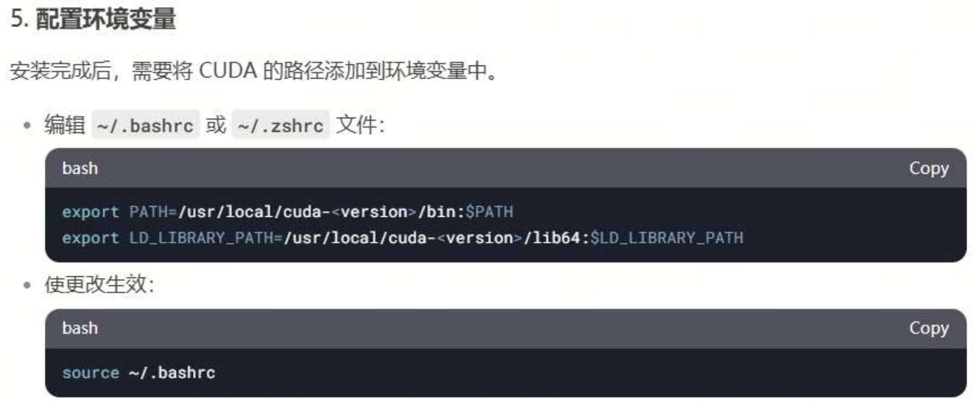
安装 CUDA Toolkit

- Ubuntu/Debian:

```bash
sudo dpkg -i cuda-repo-<distro>_<version>_amd64.deb
sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/compute/cuda/repos/<
distro>/x86_64/7fa2af80.pub
sudo apt update
sudo apt install cuda
```

- CentOS/RHEL/Fedora:

```bash
sudo rpm -i cuda-repo-<distro>_<version>_x86_64.rpm
sudo yum clean all
sudo yum install cuda
```



5. 配置环境变量

安装完成后，需要将 CUDA 的路径添加到环境变量中。

- 编辑 `~/.bashrc` 或 `~/.zshrc` 文件：

```bash
export PATH=/usr/local/cuda-<version>/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda-<version>/lib64:$LD_LIBRARY_PATH
```

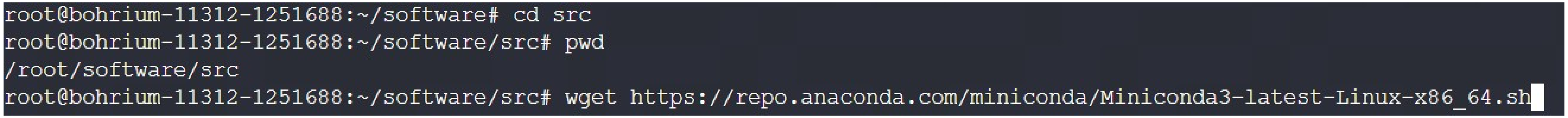
- 使更改生效：

```bash
source ~/.bashrc
```

Install Miniconda

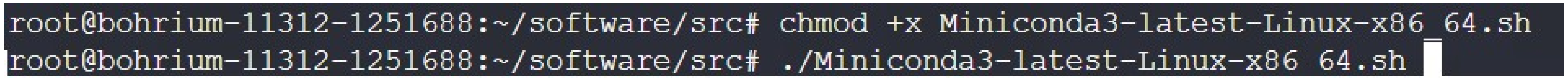
```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

再linux下，可以使用wget命令下载文件，需要对应文件的链接，很多软件或文件的供应商会提供

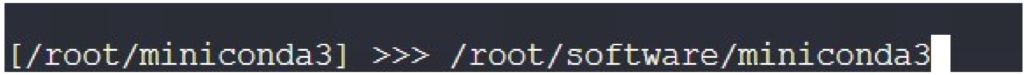


```
root@bohrium-11312-1251688:~/software# cd src
root@bohrium-11312-1251688:~/software/src# pwd
/root/software/src
root@bohrium-11312-1251688:~/software/src# wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

```
chmod +x Miniconda3-latest-Linux-x86_64.sh
 ./Miniconda3-latest-Linux-x86_64.sh
```

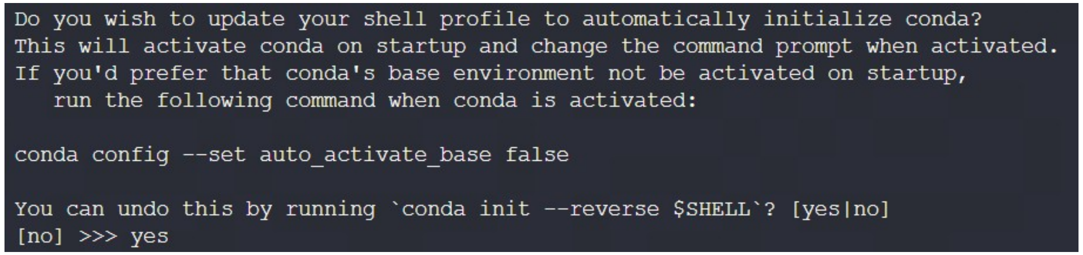


```
root@bohrium-11312-1251688:~/software/src# chmod +x Miniconda3-latest-Linux-x86_64.sh
root@bohrium-11312-1251688:~/software/src# ./Miniconda3-latest-Linux-x86_64.sh
```



```
[/root/miniconda3] >>> /root/software/miniconda3
```

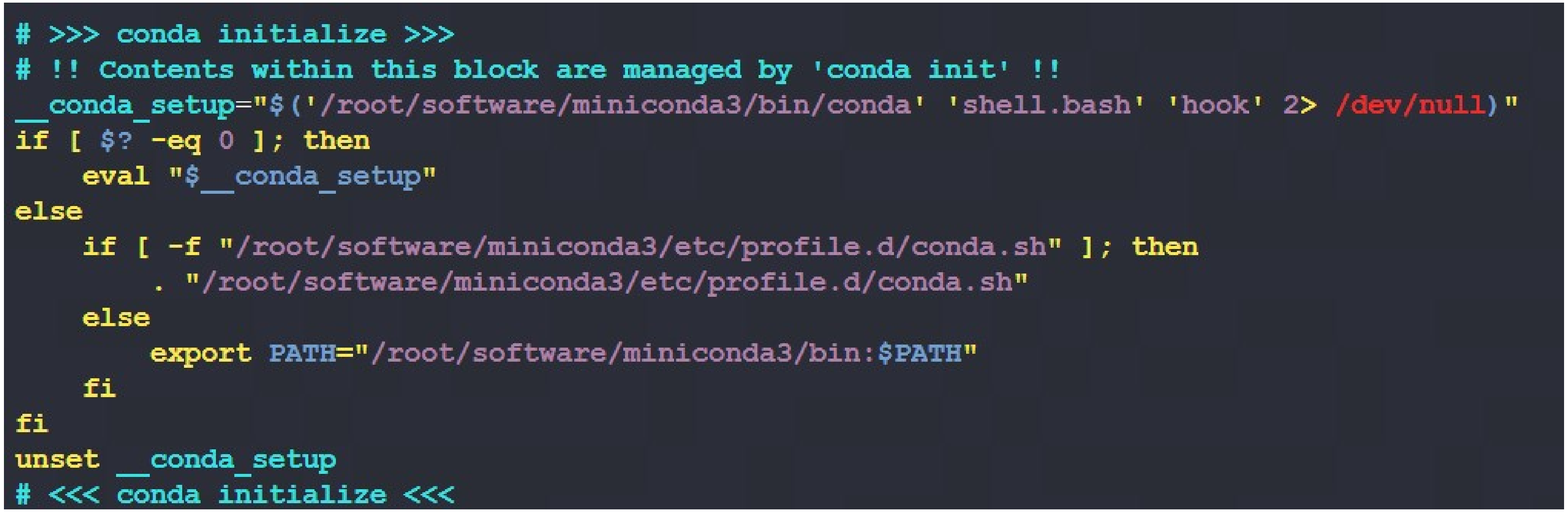
我一般会在家目录里建立一个software的文件夹，在software里建立一个src文件夹，将下载下来的安装文件放在src里，软件安装在software里；这里使用chmod命令添加权限，./Miniconda3-latest-Linux-x86_64.sh运行它以后，读完了用户手册，需要提供安装位置，这里提供了自己创建的software目录



```
Do you wish to update your shell profile to automatically initialize conda?
This will activate conda on startup and change the command prompt when activated.
If you'd prefer that conda's base environment not be activated on startup,
   run the following command when conda is activated:

conda config --set auto_activate_base false

You can undo this by running `conda init --reverse $SHELL`? [yes|no]
[no] >>> yes
```

安装的最后会问你要不要
进行conda init，记得yes

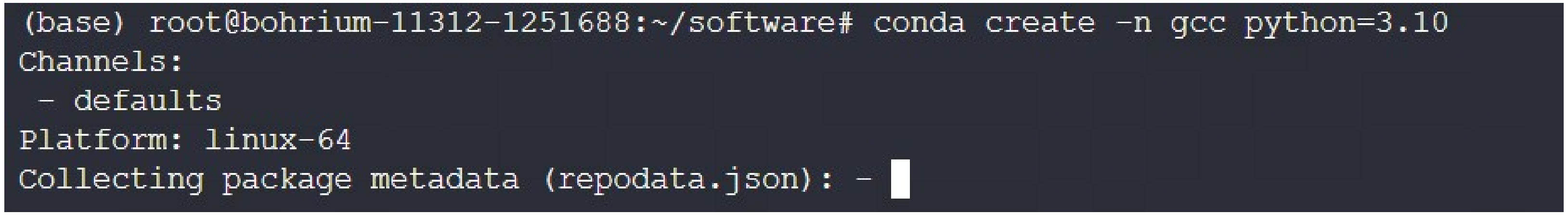
Install Miniconda



```
# >>> conda initialize >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$('/root/software/miniconda3/bin/conda' 'shell.bash' 'hook' 2> /dev/null)"
if [ $? -eq 0 ]; then
    eval "$__conda_setup"
else
    if [ -f "/root/software/miniconda3/etc/profile.d/conda.sh" ]; then
        . "/root/software/miniconda3/etc/profile.d/conda.sh"
    else
        export PATH="/root/software/miniconda3/bin:$PATH"
    fi
fi
unset __conda_setup
# <<< conda initialize <<<
```

conda init会修改你的一个配置文件，会在家目录下的.bashrc文件中添加以上内容

```
conda create –n env_name [package_name]
```

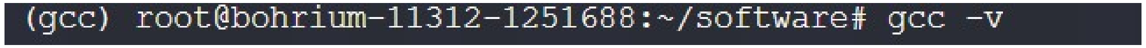


```
(base) root@bohrium-11312-1251688:~/software# conda create -n gcc python=3.10
Channels:
 - defaults
Platform: linux-64
Collecting package metadata (repodata.json): -
```

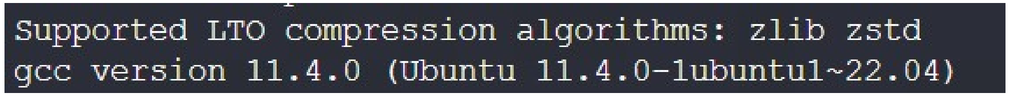
使用conda create创建一个虚拟环境，用于容纳以后需要的gcc和cmake，这里我把环境命名为gcc
之后可以使用conda activate gcc来激活这个环境

Install gcc & g++

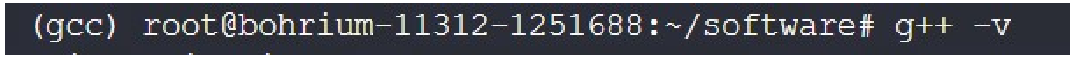
为了安装gromacs以及其它很多软件，我们需要C语言和C++语言的编译器gcc和g++，每台linux机器都会有一个，但版本可能不是我们想要的



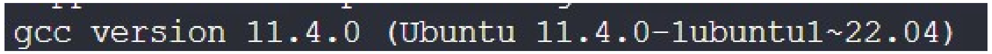
```
(gcc) root@bohrium-11312-1251688:~/software# gcc -v
```



```
Supported LTO compression algorithms: zlib zstd
gcc version 11.4.0 (Ubuntu 11.4.0-1ubuntu1~22.04)
```



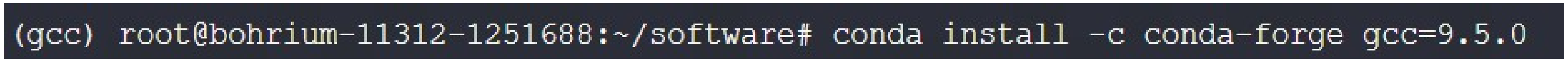
```
(gcc) root@bohrium-11312-1251688:~/software# g++ -v
```



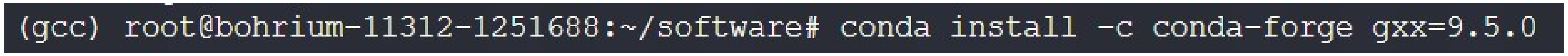
```
gcc version 11.4.0 (Ubuntu 11.4.0-1ubuntu1~22.04)
```

比如我这里的版本是11.4.0，但也许我想要一个稍微早一点的版本，因为很多软件安装需要的gcc版本不一样，过旧或者过新都可能导致安装失败

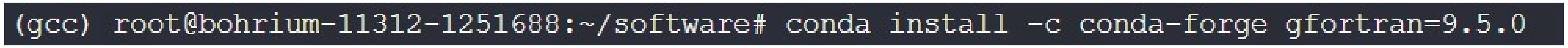
```
conda activate gcc
conda install -c conda-forge gcc=9.5.0 gxx=9.5.0 gfortran=9.5.0
```



```
(gcc) root@bohrium-11312-1251688:~/software# conda install -c conda-forge gcc=9.5.0
```

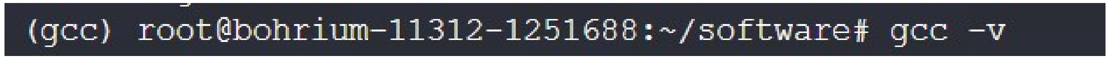


```
(gcc) root@bohrium-11312-1251688:~/software# conda install -c conda-forge gxx=9.5.0
```

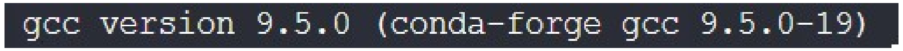


```
(gcc) root@bohrium-11312-1251688:~/software# conda install -c conda-forge gfortran=9.5.0
```

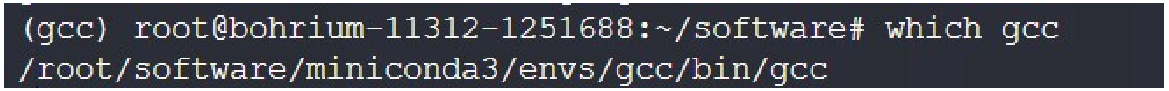
我这里安装了9.5.0，把C语言编译器gcc，C++编译器g++(conda安装的时候叫gxx)以及fortran语言编译器一起装了



```
(gcc) root@bohrium-11312-1251688:~/software# gcc -v
```



```
gcc version 9.5.0 (conda-forge gcc 9.5.0-19)
```



```
(gcc) root@bohrium-11312-1251688:~/software# which gcc
/root/software/miniconda3/envs/gcc/bin/gcc
```

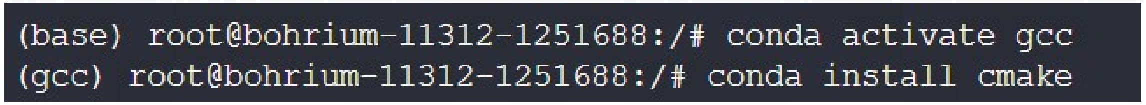
安装以后可以检查一下版本，用which 命令可以检查gcc的安装位置

Install cmake

cmake也是一个常用的安装软件需要的程序

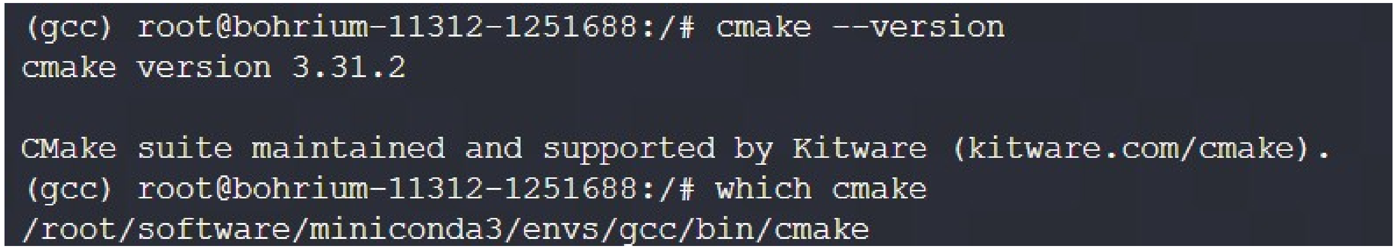
```
conda activate gcc
conda install –c conda-forge cmake
```

首先用conda activate gcc激活gcc环境，再使用conda install命令安装cmake



```
(base) root@bohrium-11312-1251688:/# conda activate gcc
(gcc) root@bohrium-11312-1251688:/# conda install cmake
```

检查版本以及安装位置



```
(gcc) root@bohrium-11312-1251688:/# cmake --version
cmake version 3.31.2

CMake suite maintained and supported by Kitware (kitware.com/cmake).
(gcc) root@bohrium-11312-1251688:/# which cmake
/root/software/miniconda3/envs/gcc/bin/cmake
```

```
conda install –c conda-forge cmake=3.20.1
```

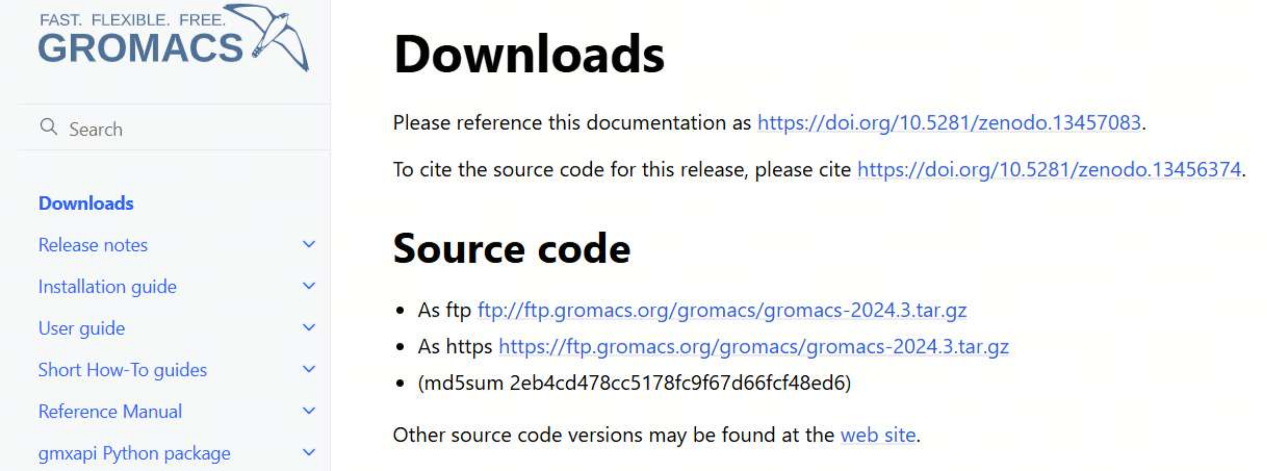
如果想安装别的什么版本的话，可以指定版本号

Install Gromacs

首先要下载gromacs源码，我这里为了和plumed插件匹配，下载的是gromacs2024.3版本

https://manual.gromacs.org/2024.3/download.html

```
wget ftp://ftp.gromacs.org/gromacs/gromacs-2024.3.tar.gz
```



Downloads

Please reference this documentation as https://doi.org/10.5281/zenodo.13457083.

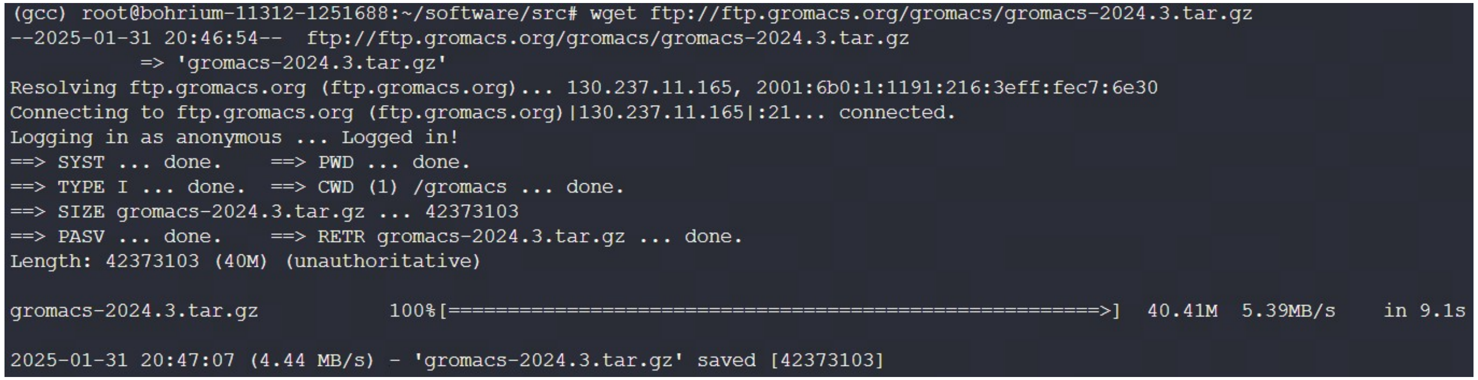
To cite the source code for this release, please cite https://doi.org/10.5281/zenodo.13456374.

Source code

- As ftp ftp://ftp.gromacs.org/gromacs/gromacs-2024.3.tar.gz
- As https https://ftp.gromacs.org/gromacs/gromacs-2024.3.tar.gz
- (md5sum 2eb4cd478cc5178fc9f67d66fcf48ed6)

Other source code versions may be found at the web site.

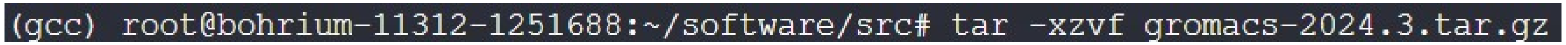
官网提供了ftp链接，所以我直接复制了ftp路径然后在linux系统里用wget命令下载了，也可以自己手动下载



```
(gcc) root@bohrium-11312-1251688:~/software/src# wget ftp://ftp.gromacs.org/gromacs/gromacs-2024.3.tar.gz
--2025-01-31 20:46:54--  ftp://ftp.gromacs.org/gromacs/gromacs-2024.3.tar.gz
           => 'gromacs-2024.3.tar.gz'
Resolving ftp.gromacs.org (ftp.gromacs.org)... 130.237.11.165, 2001:6b0:1:1191:216:3eff:fec7:6e30
Connecting to ftp.gromacs.org (ftp.gromacs.org)|130.237.11.165|:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done.    ==> PWD ... done.
==> TYPE I ... done.  ==> CWD (1) /gromacs ... done.
==> SIZE gromacs-2024.3.tar.gz ... 42373103
==> PASV ... done.    ==> RETR gromacs-2024.3.tar.gz ... done.
Length: 42373103 (40M) (unauthoritative)

gromacs-2024.3.tar.gz           100%[=======================================================>]  40.41M  5.39MB/s    in 9.1s

2025-01-31 20:47:07 (4.44 MB/s) - 'gromacs-2024.3.tar.gz' saved [42373103]
```

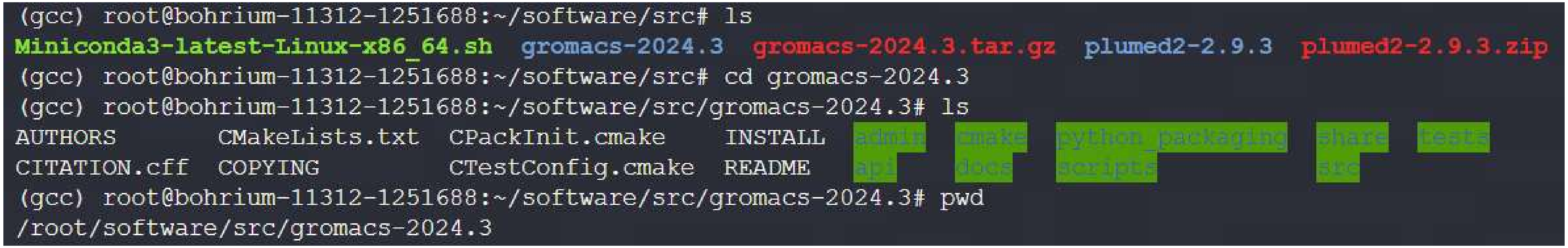
Install Gromacs

```
tar -xzvf gromacs-2024.3.tar.gz
```



```
(gcc) root@bohrium-11312-1251688:~/software/src# tar -xzvf gromacs-2024.3.tar.gz
```

解压缩，得到gromacs-2024.3文件夹，这就是我们的gromacs源码了



```
(gcc) root@bohrium-11312-1251688:~/software/src# ls
Miniconda3-latest-Linux-x86_64.sh  gromacs-2024.3  gromacs-2024.3.tar.gz  plumed2-2.9.3  plumed2-2.9.3.zip
(gcc) root@bohrium-11312-1251688:~/software/src# cd gromacs-2024.3
(gcc) root@bohrium-11312-1251688:~/software/src/gromacs-2024.3# ls
AUTHORS       CMakeLists.txt  CPackInit.cmake    INSTALL  admin  cmake  python_packaging  share  tests
CITATION.cff  COPYING         CTestConfig.cmake  README   api    docs   scripts           src
(gcc) root@bohrium-11312-1251688:~/software/src/gromacs-2024.3# pwd
/root/software/src/gromacs-2024.3
```

简单的说，安装需要执行以下命令，我们接下来一步步执行

```
cd gromacs-2024.3
mkdir build
cd build
cmake .. -DGMX_BUILD_OWN_FFTW=ON -DREGRESSIONTEST_DOWNLOAD=ON -DGMX_GPU=CUDA -
DCUDA_TOOLKIT_ROOT_DIR=/path-to-cuda -DCMAKE_INSTALL_PREFIX=INSTALL_DIR
make
#make check
sudo make install
```

Install Gromacs

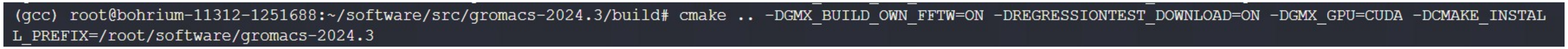
http://sobereva.com/457

```
cd gromacs-2024.3
mkdir build
cd build
conda activate gcc
conda install -c conda-forge lapack blas
```

进入gromacs源码所在文件夹，创建一个build文件夹，进入build文件夹，这里顺便在gcc环境里安装了两个线性代数库

```
cmake .. -DGMX_BUILD_OWN_FFTW=ON -DREGRESSIONTEST_DOWNLOAD=ON -DGMX_GPU=CUDA -
DCMAKE_INSTALL_PREFIX=/root/software/gromacs-2024.3
```

执行cmake命令，需要提供安装位置，我继续按习惯放到家目录下我自己创建的software下了，注意上面的cmake命令只有一行，这里是太长了换行了



```
(gcc) root@bohrium-11312-1251688:~/software/src/gromacs-2024.3/build# cmake .. -DGMX_BUILD_OWN_FFTW=ON -DREGRESSIONTEST_DOWNLOAD=ON -DGMX_GPU=CUDA -DCMAKE_INSTAL
L_PREFIX=/root/software/gromacs-2024.3
```

```
make -j 8
```

执行make命令，这一步时间很久，可能会报错，出现报错记得问deepseek

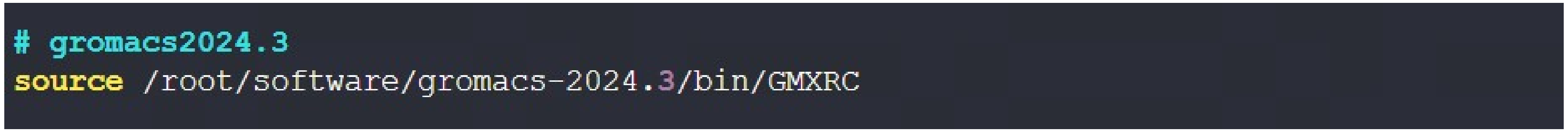
```
make install
```

舒一口气，进入make install环节，可能会碰到权限问题，就得提供root权限，sudo make install
根据我的经验，大部分问题都是gcc版本，cmake版本和gromacs版本不匹配；其次出现的问题是找不到CUDA的库文件；再次的话就是缺少一些库文件；余下的就千奇百怪，不一而足，我衷心希望你安装顺利

Install Gromacs

make install成功以后，要修改.bashrc，并source .bashrc文件使改动生效

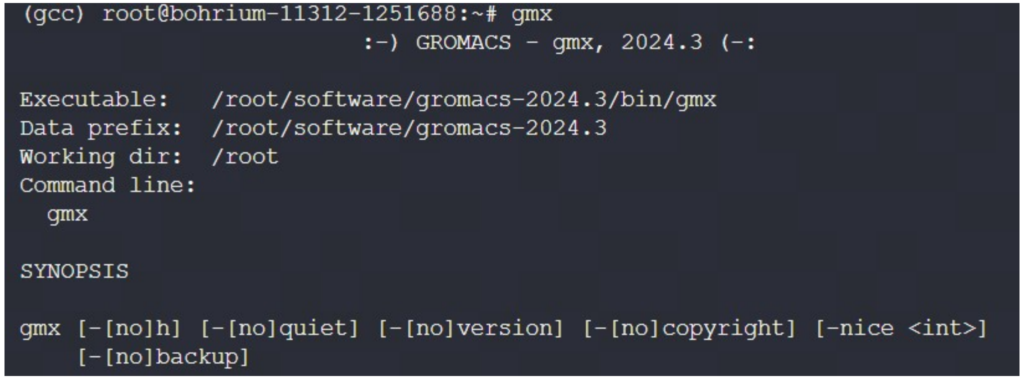
```
vim ~/.bashrc
```



```
# gromacs2024.3
source /root/software/gromacs-2024.3/bin/GMXRC
```

```
source ~/.bashrc
```

```
conda activate gcc
gmx
```

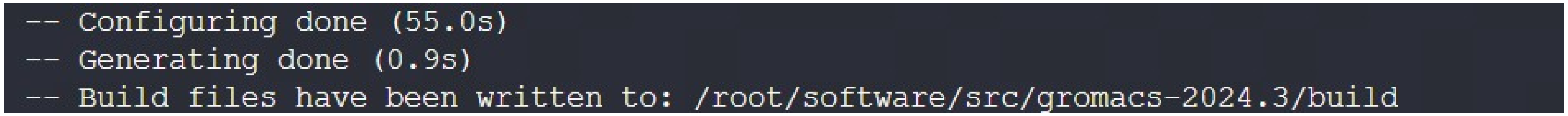


```
(gcc) root@bohrium-11312-1251688:~# gmx
                        :-) GROMACS - gmx, 2024.3 (-:

Executable:   /root/software/gromacs-2024.3/bin/gmx
Data prefix:  /root/software/gromacs-2024.3
Working dir:  /root
Command line:
  gmx

SYNOPSIS

gmx [-[no]h] [-[no]quiet] [-[no]version] [-[no]copyright] [-nice <int>]
    [-[no]backup]
```

检查一下安装情况吧，看起来安装成功了

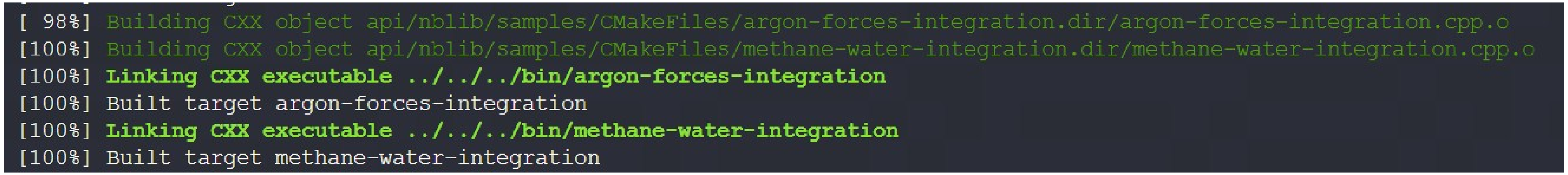
Install Gromacs

configure正常结束



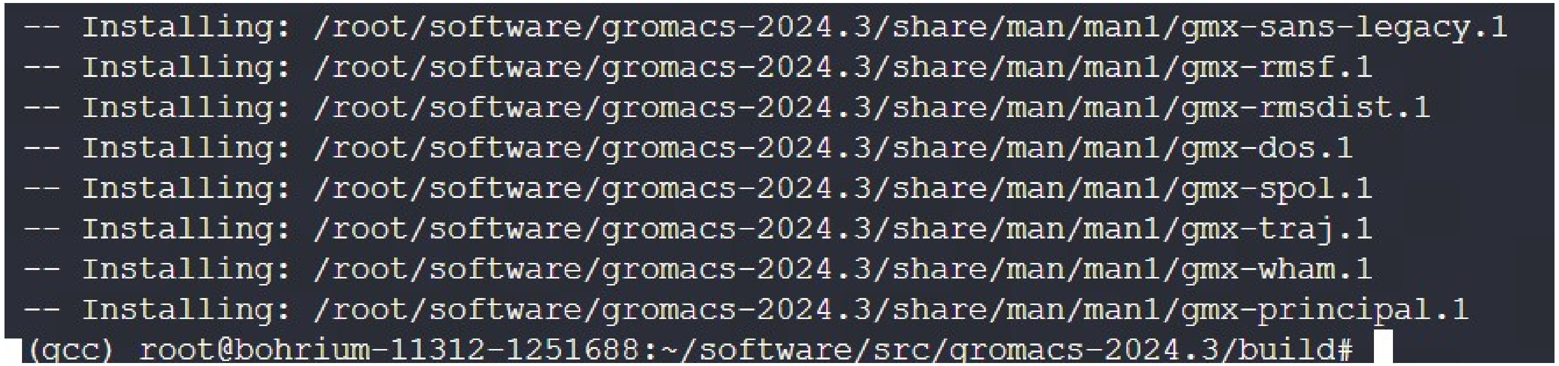
```
-- Configuring done (55.0s)
-- Generating done (0.9s)
-- Build files have been written to: /root/software/src/gromacs-2024.3/build
```

make正常结束



```
[ 98%] Building CXX object api/nblib/samples/CMakeFiles/argon-forces-integration.dir/argon-forces-integration.cpp.o
[100%] Building CXX object api/nblib/samples/CMakeFiles/methane-water-integration.dir/methane-water-integration.cpp.o
[100%] Linking CXX executable ../../../bin/argon-forces-integration
[100%] Built target argon-forces-integration
[100%] Linking CXX executable ../../../bin/methane-water-integration
[100%] Built target methane-water-integration
```

make install正常结束

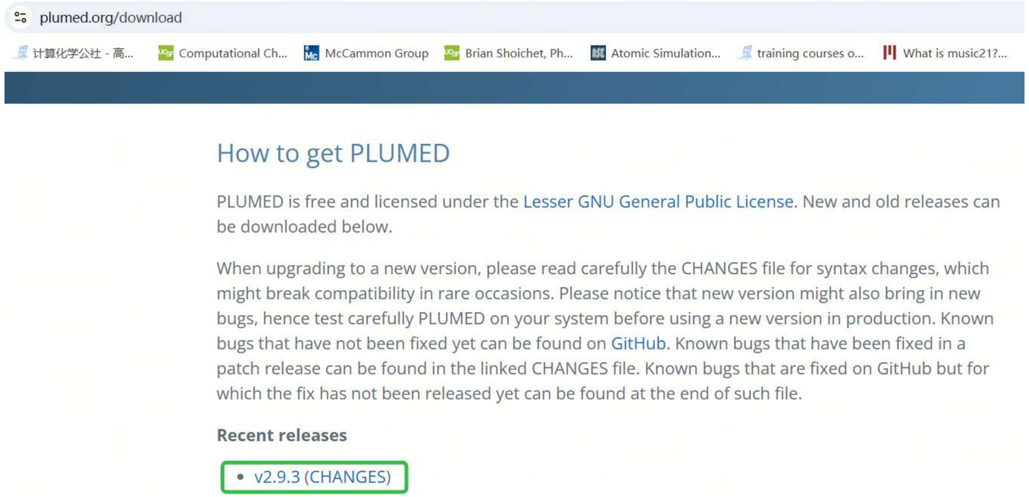


```
-- Installing: /root/software/gromacs-2024.3/share/man/man1/gmx-sans-legacy.1
-- Installing: /root/software/gromacs-2024.3/share/man/man1/gmx-rmsf.1
-- Installing: /root/software/gromacs-2024.3/share/man/man1/gmx-rmsdist.1
-- Installing: /root/software/gromacs-2024.3/share/man/man1/gmx-dos.1
-- Installing: /root/software/gromacs-2024.3/share/man/man1/gmx-spol.1
-- Installing: /root/software/gromacs-2024.3/share/man/man1/gmx-traj.1
-- Installing: /root/software/gromacs-2024.3/share/man/man1/gmx-wham.1
-- Installing: /root/software/gromacs-2024.3/share/man/man1/gmx-principal.1
(gcc) root@bohrium-11312-1251688:~/software/src/gromacs-2024.3/build#
```

“成功的安装总是相似，不成功的安装各有各的问题”

Install Plumed

plumed是进行元动力学模拟需要安装的插件



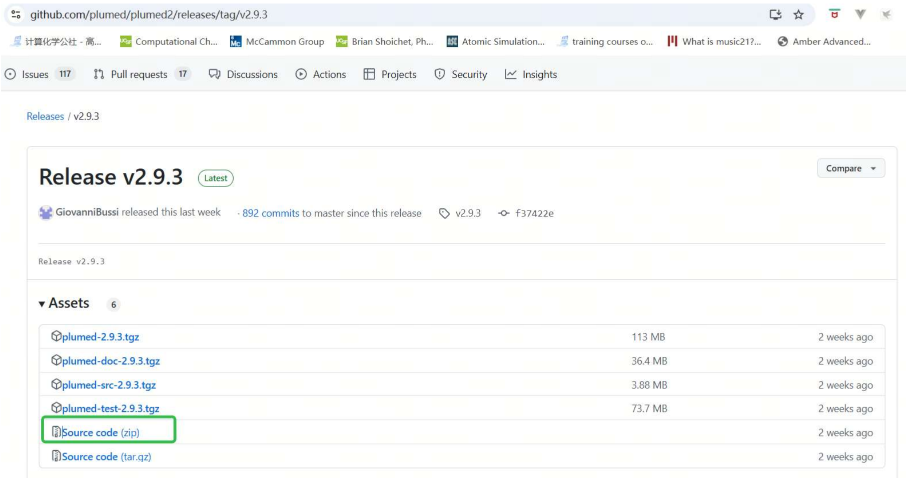
How to get PLUMED

PLUMED is free and licensed under the Lesser GNU General Public License. New and old releases can be downloaded below.

When upgrading to a new version, please read carefully the CHANGES file for syntax changes, which might break compatibility in rare occasions. Please notice that new version might also bring in new bugs, hence test carefully PLUMED on your system before using a new version in production. Known bugs that have not been fixed yet can be found on GitHub. Known bugs that have been fixed in a patch release can be found in the linked CHANGES file. Known bugs that are fixed on GitHub but for which the fix has not been released yet can be found at the end of such file.

Recent releases

- v2.9.3 (CHANGES)



Release v2.9.3

Latest

GiovanniBussi released this last week · 892 commits to master since this release v2.9.3 f37422e

```
Release v2.9.3
```

▾ Assets 6

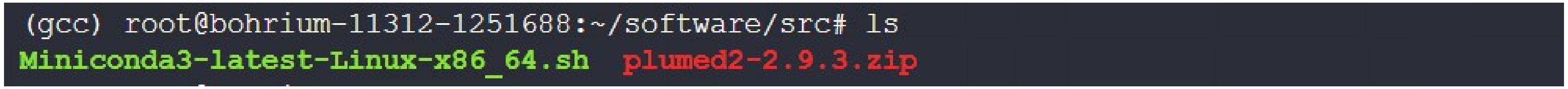
plumed-2.9.3.tgz	113 MB	2 weeks ago
plumed-doc-2.9.3.tgz	36.4 MB	2 weeks ago
plumed-src-2.9.3.tgz	3.88 MB	2 weeks ago
plumed-test-2.9.3.tgz	73.7 MB	2 weeks ago
Source code (zip)		2 weeks ago
Source code (tar.gz)		2 weeks ago

https://www.plumed.org/download

进入其官网找到其最新的版本，点击后会到GitHub，下载plumed的源码，上传到linux系统下之前创建的software/src文件夹里（放到哪个文件夹里是任意的，但我个人习惯这样组织起来）

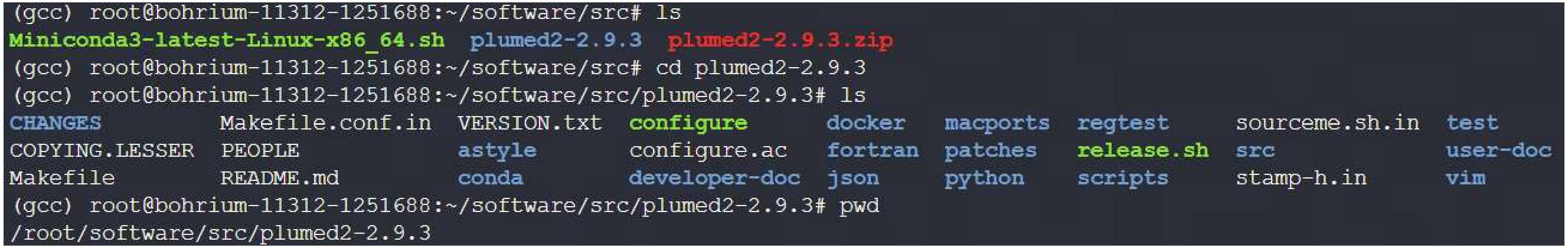
Install Plumed

刚才下载的源代码现在放在一个文件夹里，是一个zip压缩文件



```
(gcc) root@bohrium-11312-1251688:~/software/src# ls
Miniconda3-latest-Linux-x86_64.sh  plumed2-2.9.3.zip
```

```
unzip plumed2-2.9.3.zip
```



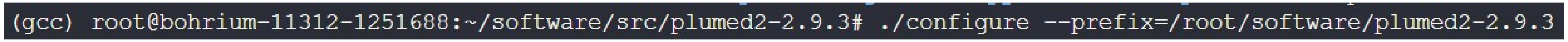
```
(gcc) root@bohrium-11312-1251688:~/software/src# ls
Miniconda3-latest-Linux-x86_64.sh  plumed2-2.9.3  plumed2-2.9.3.zip
(gcc) root@bohrium-11312-1251688:~/software/src# cd plumed2-2.9.3
(gcc) root@bohrium-11312-1251688:~/software/src/plumed2-2.9.3# ls
CHANGES         Makefile.conf.in  VERSION.txt  configure      docker   macports  regtest     sourceme.sh.in  test
COPYING.LESSER  PEOPLE            astyle       configure.ac   fortran  patches   release.sh  src             user-doc
Makefile        README.md         conda        developer-doc  json     python    scripts     stamp-h.in      vim
(gcc) root@bohrium-11312-1251688:~/software/src/plumed2-2.9.3# pwd
/root/software/src/plumed2-2.9.3
```

解压缩完毕以后，会得到一个plumed2-2.9.3文件夹，我们就要通过它来安装plumed

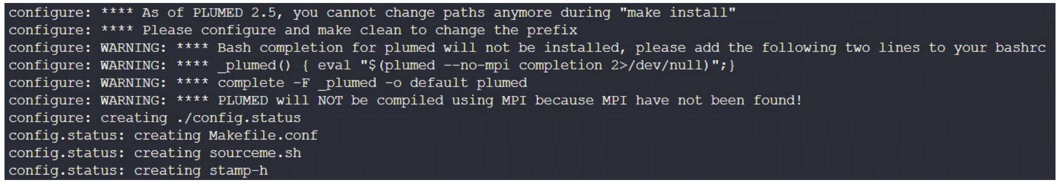
Install Plumed

进入plumed2-2.9.3文件夹，并进行安装，--prefix后要更新的是plumed最后的安装位置，--enable-modules代表激活的插件的某些额外功能，这里顺手激活了funnel metaD，但不是必须的。

```
cd plumed2-2.9.3
./configure --prefix=change_to_install_dir --enable-modules=funnel
make -j 4
make install
```

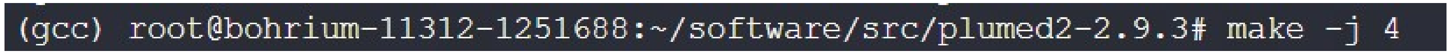


```
(gcc) root@bohrium-11312-1251688:~/software/src/plumed2-2.9.3# ./configure --prefix=/root/software/plumed2-2.9.3
```

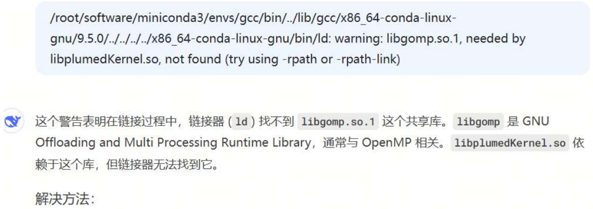


```
configure: **** As of PLUMED 2.5, you cannot change paths anymore during "make install"
configure: **** Please configure and make clean to change the prefix
configure: WARNING: **** Bash completion for plumed will not be installed, please add the following two lines to your bashrc
configure: WARNING: **** _plumed() { eval "$(plumed --no-mpi completion 2>/dev/null)";}
configure: WARNING: **** complete -F _plumed -o default plumed
configure: WARNING: **** PLUMED will NOT be compiled using MPI because MPI have not been found!
configure: creating ./config.status
config.status: creating Makefile.conf
config.status: creating sourceme.sh
config.status: creating stamp-h
```

进入源码文件夹后，执行第一步configure，我把安装路径设置为/root/software/plumed2-2.9.3，这个路径的设置也是任意的，重点是你自己之后能找的到。
看起来我第一步正常结束了。



```
(gcc) root@bohrium-11312-1251688:~/software/src/plumed2-2.9.3# make -j 4
```

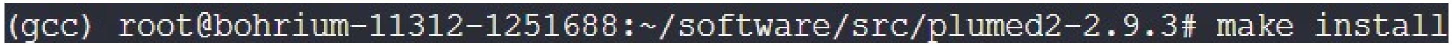


/root/software/miniconda3/envs/gcc/bin/../lib/gcc/x86_64-conda-linux-gnu/9.5.0/../../../../x86_64-conda-linux-gnu/bin/ld: warning: libgomp.so.1, needed by libplumedKernel.so, not found (try using -rpath or -rpath-link)

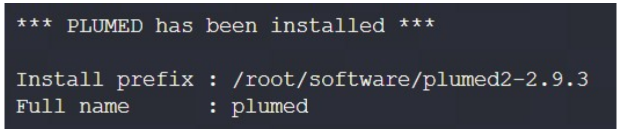
这个警告表明在链接过程中，链接器（`ld`）找不到 `libgomp.so.1` 这个共享库。`libgomp` 是 GNU Offloading and Multi Processing Runtime Library，通常与 OpenMP 相关。`libplumedKernel.so` 依赖于这个库，但链接器无法找到它。

解决方法：

我安装时出现了报错，我把报错信息输给了现在很火的语言模型deepseek，然后根据它的回复进行了调整，再进行make命令



```
(gcc) root@bohrium-11312-1251688:~/software/src/plumed2-2.9.3# make install
```

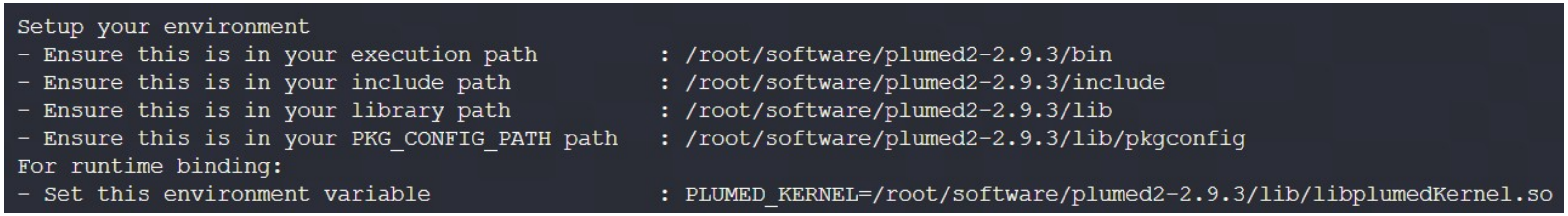


```
*** PLUMED has been installed ***

Install prefix : /root/software/plumed2-2.9.3
Full name      : plumed
```

最后一步是make install，看到左边的信息就代表安装成功了

Install Plumed

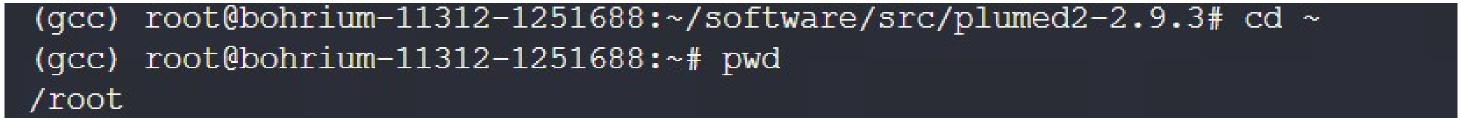
上面安装成功以后，会看到以下指示，需要你去编辑.bashrc配置文件来使得安装生效



```
Setup your environment
- Ensure this is in your execution path         : /root/software/plumed2-2.9.3/bin
- Ensure this is in your include path           : /root/software/plumed2-2.9.3/include
- Ensure this is in your library path           : /root/software/plumed2-2.9.3/lib
- Ensure this is in your PKG_CONFIG_PATH path   : /root/software/plumed2-2.9.3/lib/pkgconfig
For runtime binding:
- Set this environment variable                 : PLUMED_KERNEL=/root/software/plumed2-2.9.3/lib/libplumedKernel.so
```

.bashrc文件在家目录下，一般linux系统里~就代表家目录

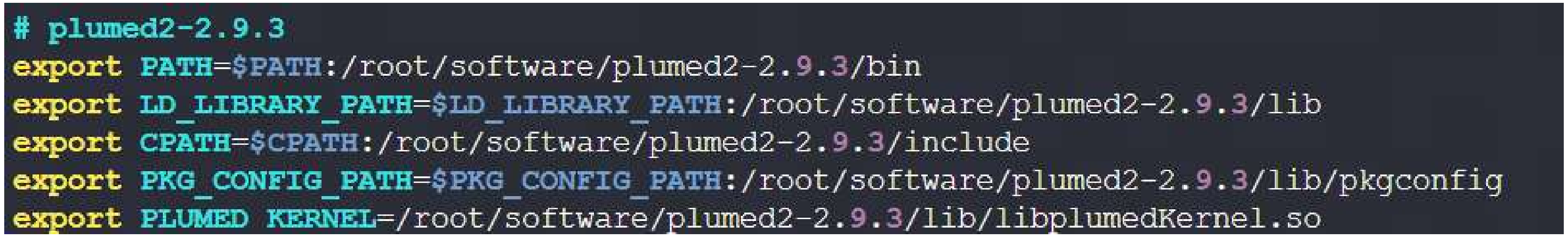
```
cd ~
```



```
(gcc) root@bohrium-11312-1251688:~/software/src/plumed2-2.9.3# cd ~
(gcc) root@bohrium-11312-1251688:~# pwd
/root
```

你的家目录几乎肯定是和我不一样的，很可能是一个类似/home/guoy的位置，这里的guoy在你那里应该是你的用户名，在.bashrc里添加对应内容，然后保存。记得这里plumed的安装路径要换成你自己的。如果你的系统是zsh，那你就得去.zshrf里修改，可以问大模型比如deepseek

```
vim ~/.bashrc
```

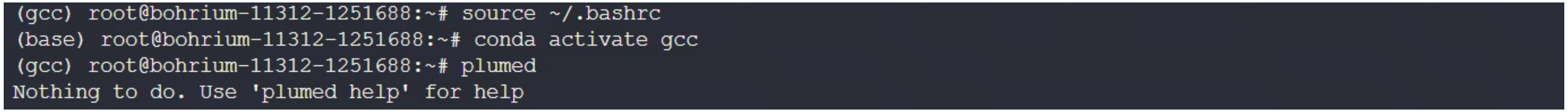


```
# plumed2-2.9.3
export PATH=$PATH:/root/software/plumed2-2.9.3/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/root/software/plumed2-2.9.3/lib
export CPATH=$CPATH:/root/software/plumed2-2.9.3/include
export PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/root/software/plumed2-2.9.3/lib/pkgconfig
export PLUMED_KERNEL=/root/software/plumed2-2.9.3/lib/libplumedKernel.so
```

```
source ~/.bashrc
```

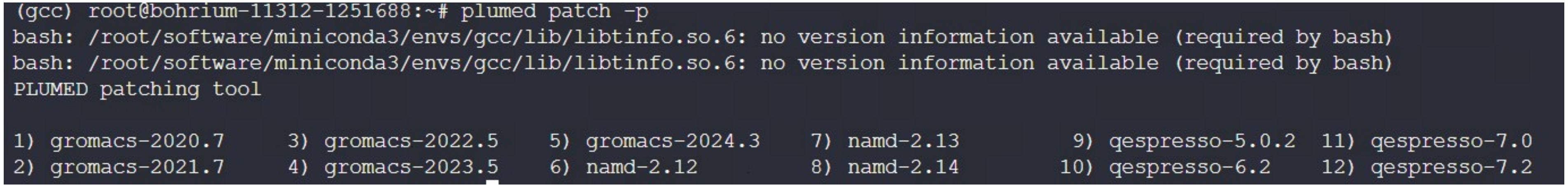
Install Plumed

source完以后看看能不能用，记得激活gcc虚拟环境



```
(gcc) root@bohrium-11312-1251688:~# source ~/.bashrc
(base) root@bohrium-11312-1251688:~# conda activate gcc
(gcc) root@bohrium-11312-1251688:~# plumed
Nothing to do. Use 'plumed help' for help
```

看起来没问题
检查以下支持的gromacs版本
看起来plumed2-2.9.3支持到gromacs-2024.3

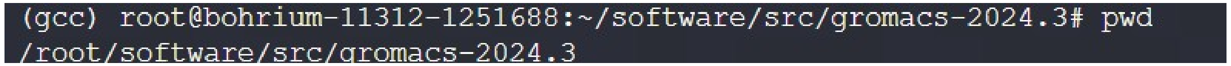


```
(gcc) root@bohrium-11312-1251688:~# plumed patch -p
bash: /root/software/miniconda3/envs/gcc/lib/libtinfo.so.6: no version information available (required by bash)
bash: /root/software/miniconda3/envs/gcc/lib/libtinfo.so.6: no version information available (required by bash)
PLUMED patching tool

1) gromacs-2020.7     3) gromacs-2022.5     5) gromacs-2024.3     7) namd-2.13          9) qespresso-5.0.2   11) qespresso-7.0
2) gromacs-2021.7     4) gromacs-2023.5     6) namd-2.12          8) namd-2.14         10) qespresso-6.2     12) qespresso-7.2
```

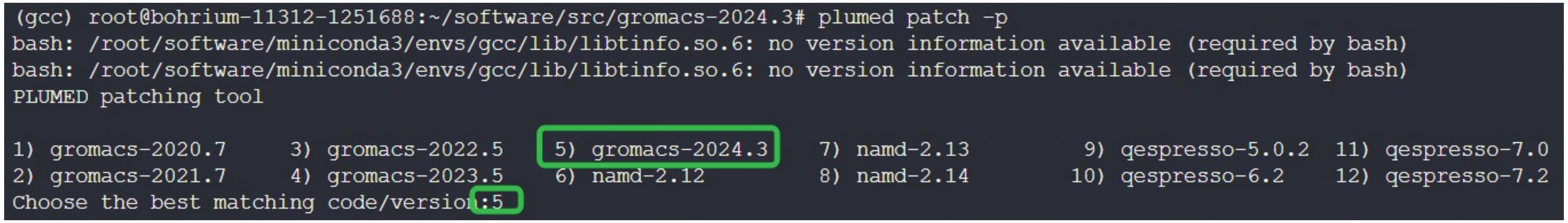
Install Gromacs-Plumed

安装能跑metadynamics的gromacs，步骤与普通版大致相同，就是需要用plumed patch命令修改一下gromacs的源代码，记得把之前安装时候的build文件删除，以及安装路径记得换一个

```
cd gromacs-2024.3
plumed patch –p
rm –rf build
mkdir build
cd build
cmake .. -DGMX_BUILD_OWN_FFTW=ON -DREGRESSIONTEST_DOWNLOAD=ON -DGMX_GPU=CUDA -
DCMAKE_INSTALL_PREFIX==/root/software/gromacs-plumed
make
sudo make install
```



```
(gcc) root@bohrium-11312-1251688:~/software/src/gromacs-2024.3# pwd
/root/software/src/gromacs-2024.3
```



```
(gcc) root@bohrium-11312-1251688:~/software/src/gromacs-2024.3# plumed patch -p
bash: /root/software/miniconda3/envs/gcc/lib/libtinfo.so.6: no version information available (required by bash)
bash: /root/software/miniconda3/envs/gcc/lib/libtinfo.so.6: no version information available (required by bash)
PLUMED patching tool

1) gromacs-2020.7     3) gromacs-2022.5     5) gromacs-2024.3     7) namd-2.13          9) qespresso-5.0.2   11) qespresso-7.0
2) gromacs-2021.7     4) gromacs-2023.5     6) namd-2.12          8) namd-2.14         10) qespresso-6.2     12) qespresso-7.2
Choose the best matching code/version:5
```