



中国科学技术大学

University of Science and Technology of China

超算中心用户入门指南

沈瑜

中国科学技术大学，超级计算中心

2023 年 5 月 30 日



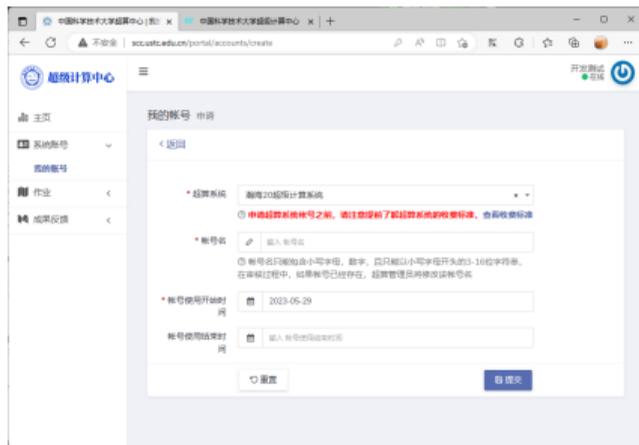
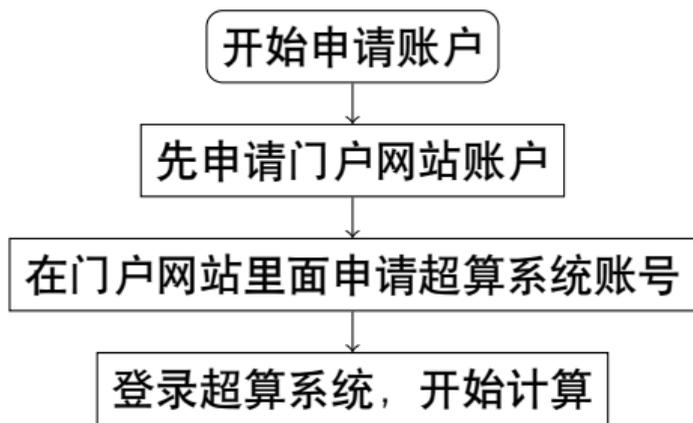
- 1 账户申请
- 2 系统登录与软件设置
- 3 Linux 基础
- 4 计算作业提交和管理
- 5 文件编辑和程序编译



- 1 账户申请
- 2 系统登录与软件设置
- 3 Linux 基础
- 4 计算作业提交和管理
- 5 文件编辑和程序编译

门户网站账户 统一管理超算系统账号、账单、作业等信息，在门户网站上直接注册。

超算系统账号 登录超算系统进行计算，登录门户网站后申请再开通。





- 1 账户申请
- 2 系统登录与软件设置
 - 账号设置 ■ 登录超算系统
- 3 Linux 基础
- 4 计算作业提交和管理
- 5 文件编辑和程序编译



- 1 账户申请
- 2 系统登录与软件设置
 - 账号设置 ■ 登录超算系统
- 3 Linux 基础
- 4 计算作业提交和管理
- 5 文件编辑和程序编译



您好，账户已开设，信息如下：

- 系统登录 IP: 211.86.151. [redacted], 211.86.151. [redacted]
- 用户手册: <http://scc.ustc.edu.cn/389/list.htm>
- 提问的智慧: <https://lug.ustc.edu.cn/wiki/doc/smart-questions/>。要学会问问题，问问题时需提供详细信息，如系统、作业号、目录、命令、出错信息等。
- 登录名: hu [redacted]
- 密码: \$75 [redacted]
- 为了安全，超算系统 SSH 及 SSH 相关协议增加验证码(Verification Code)二次登录验证。您账户 hujianyu22 在该系统上的信息如下：
二维码链接: <https://www.google.com/chart?chs=200x200&chld=M|0&cht=qr&chl=otpauth://totp/hujianyu22@hanhai22-01%3Fsecret%3DVZNOU3QGHWZRP4IWWRE2QCTLU%26issuer%3Dhanhai22-01>
- 安全密钥 (Secret Key): VZNO [redacted] LU (推荐使用此简单方法进行设置)
每 30 秒允许最多输入 3 次验证码
验证码窗口大小 (当前时间附近验证周期总数): 17
禁止验证码复用
紧急安全码 (请保存在秘密地方, 无法使用客户端获取验证码时使用, 各安全码均为一次性使用):
99 [redacted] 36 96 [redacted] 31 94 [redacted] 69 25 [redacted] 643 26 [redacted] 5

安全密钥一旦泄漏, 请立即作废以前的且重新生成 (浏览器、手机客户端等需用新的安全密钥注册): 运行 `opt/bin/ga.sh -recreate` 或 `google-authenticator`。

用法参见: <http://scc.ustc.edu.cn/2018/0926/c409a339006/page.htm>

联系方式:

- 单位: 中国科学技术大学 [网络信息中心](#) [超级计算中心](#)
- 地址: 安徽省合肥市金寨路 96 号科大东区新图书馆一楼东侧 126 室, 邮编: 230026
- 电话/传真: 0551-63602248、63601540、63600316
- 邮箱: sccadmin@ustc.edu.cn
- 主页: <http://scc.ustc.edu.cn>
- QQ 用户交流群: 8355136 (面向普通超算用户)、337561074 (面向系统管理运维)



超算系统最常见的入侵方式:

- ▶ 攻破弱密码
- ▶ 攻破可以免密码登录的跳板主机

可以采用随时间变化的二次验证码来阻挡此类入侵。超算中心目前采用的是最常见的 TOTP 二次验证码。

TOTP (Time-Based One-Time Password) 算法是基于时间的一次性密码算法, 根据预共享的密钥与当前时间计算一次性密码。它已被互联网工程任务组接纳为 RFC 6238 标准, 成为主动开放认证的基石, 并被用于众多多因子认证系统当中。¹

所有支持 TOTP 的程序都可以用来设置登录超算系统的二次验证码。我们以最常见的 **edge 浏览器扩展**和**手机应用**为例为大家介绍使用方式。

¹参考简介: <https://zhuanlan.zhihu.com/p/299135419>

使用 edge 浏览器的扩展设置二次验证码

打开管理扩展 ("edge://extensions/") 页面

点击“获取 Microsoft Edge 扩展”

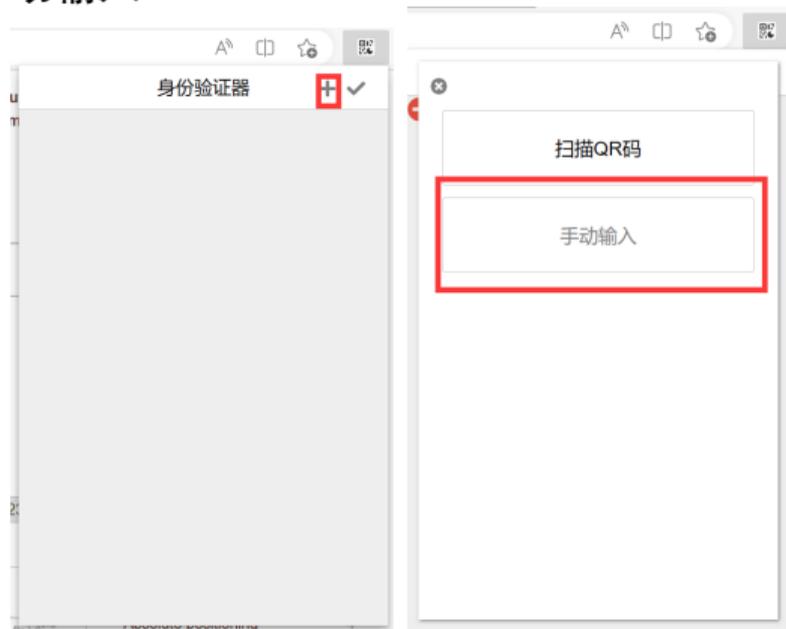
在打开的“Edge 外接程序”网页里面搜索“Authenticator”，
可以选择“Authenticator: 2FA Client”安装

在工具栏安装“身份验证器”按钮

点击该按钮可以设置和查看二次验证码



点击“+”增加新的验证码设置，选择“手动输入”



将开户邮件中的“安全密钥”填入“密钥”，签发方可以自己起名，建议使用系统名称，例如“hanhai22@scc”。

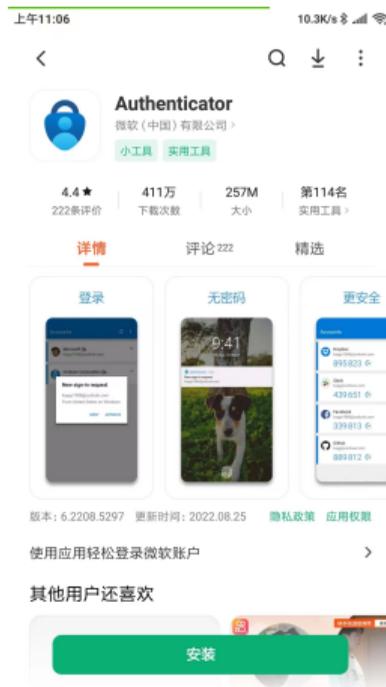


以微软提供的“Authenticator”应用为例

在手机应用市场中找到
Autheticator 应用并安装

在应用中增加“其他账户”的验证码

采用手动输入方式输入安全密钥
(ENTER CODE MANUALLY)



点击“添加账户”



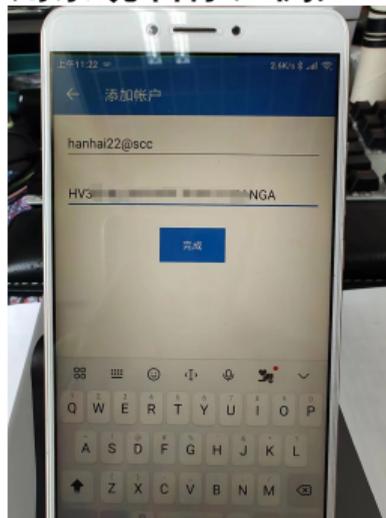
选择“其他账户”



在扫描二维码界面下方点击“OR ENTER CODE MANUALLY”来手动输入安全密钥

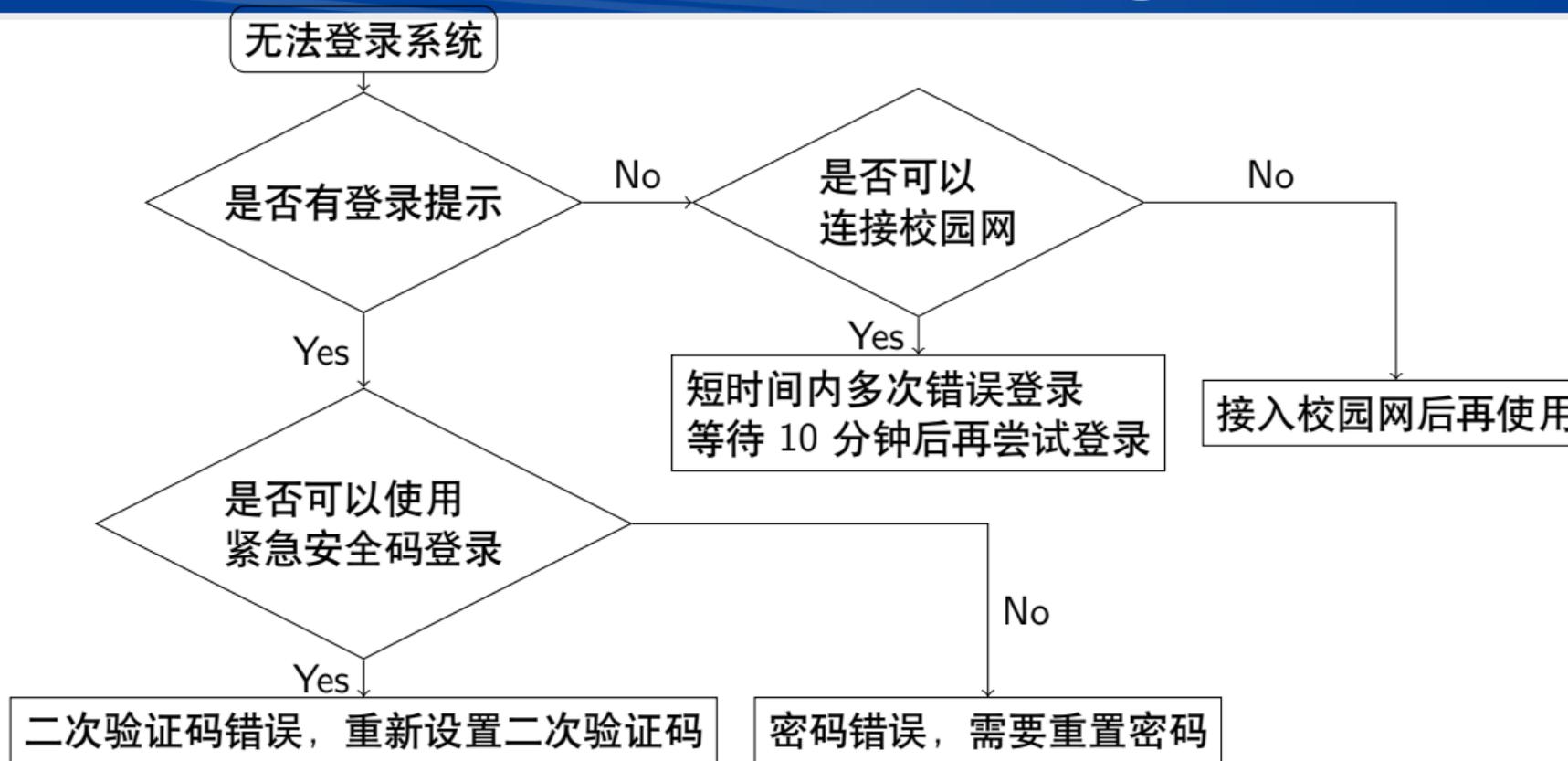


将开户邮件中的“安全密钥”填入“密钥”，“账户名称”可以自己起名，建议使用系统名称，例如“hanhai22@scc”。





- ▶ 可以登录系统时：
 - 修改密码 瀚海 20 和瀚海 22 登录到门户网站上修改
 - TC4600 系统 登录进系统后使用“passwd”命令修改
 - 修改二次验证码 登录到系统上运行“/opt/bin/ga.sh”命令
- ▶ 无法登录系统时：可以使用注册账号的 email 向超算管理员邮箱 (mailto:sccadmin@ustc.edu.cn) 发送邮件要求重置密码或者二次验证码。





教师 请直接使用网络通的 VPN 功能，使用说明：

<http://ustcnet.ustc.edu.cn/33591/list.htm>

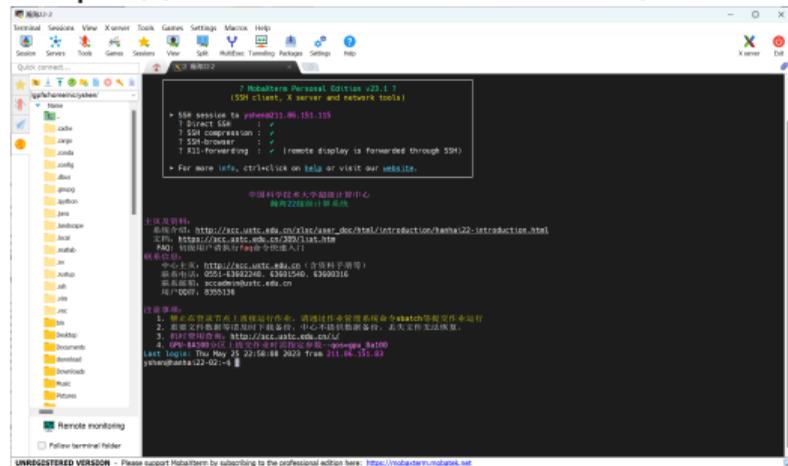
学生 超算中心 VPN，可在门户网站里面申请。注意：仅支持连接超算系统，不能用来连接校内其他网站，比如图书馆提供的电子期刊数据库等等。



- 1 账户申请
- 2 系统登录与软件设置
 - 账号设置 ■ 登录超算系统
- 3 Linux 基础
- 4 计算作业提交和管理
- 5 文件编辑和程序编译

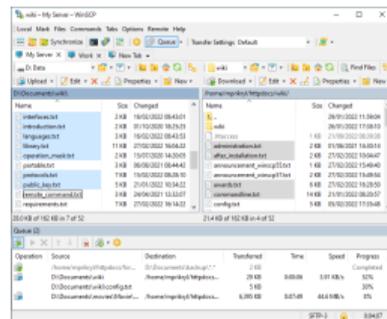
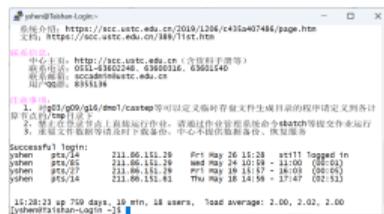
mobaxterm

<https://mobaxterm.mobatek.net/>



putty+winscp

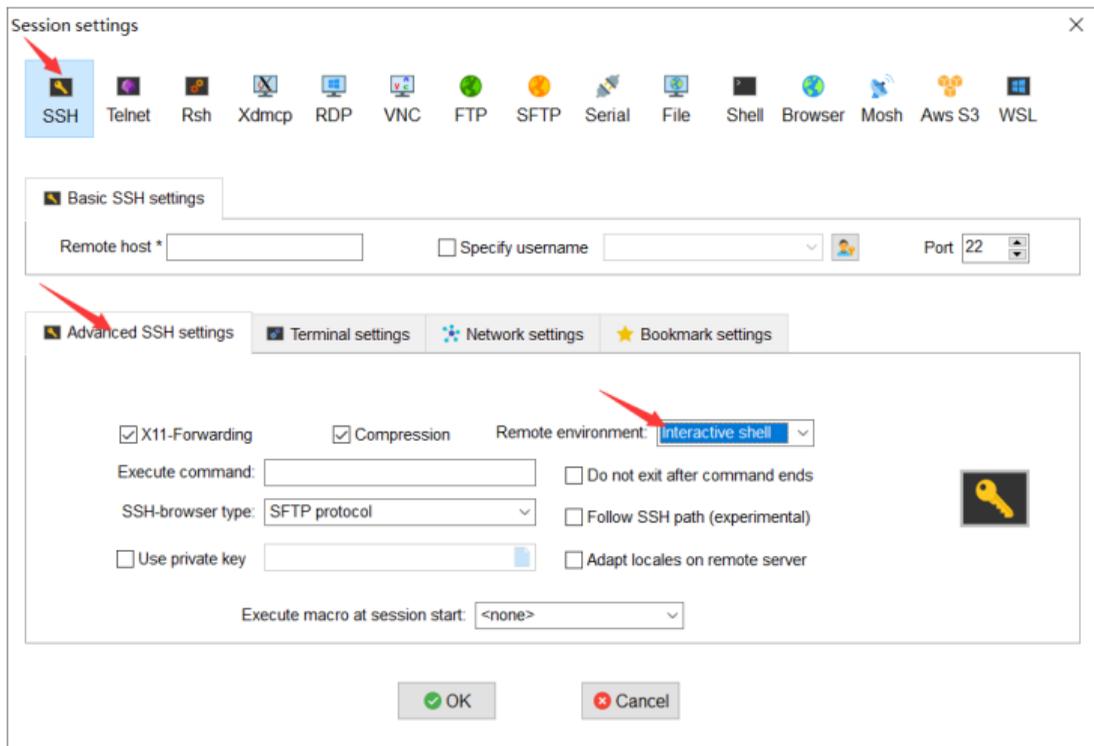
- ▶ putty: <https://www.chiark.greenend.org.uk/~sgtatham/putty/a>
- ▶ winscp: <https://winscp.net>



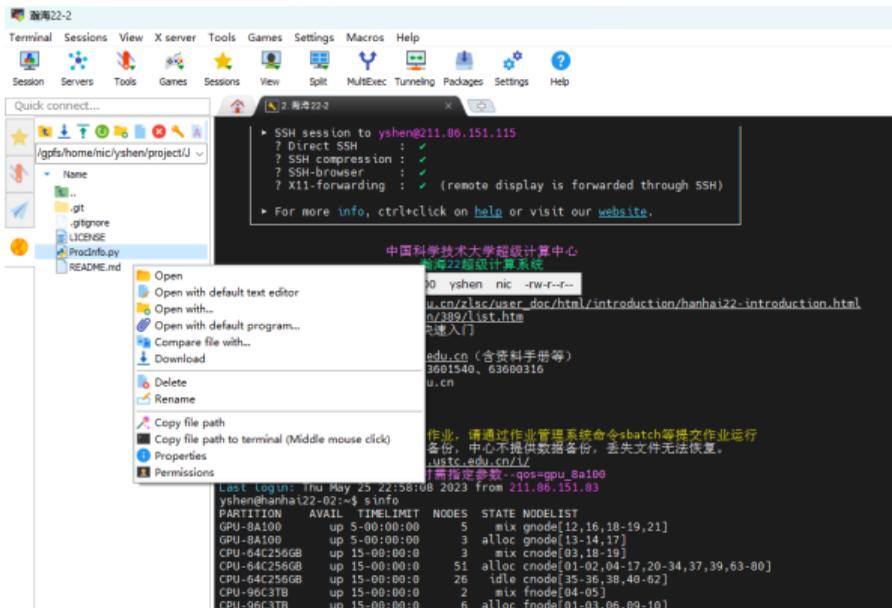
^a某搜索引擎 B 曾经将带有黑客代码的结果排在前面，请注意辨别正确网址

设置要点

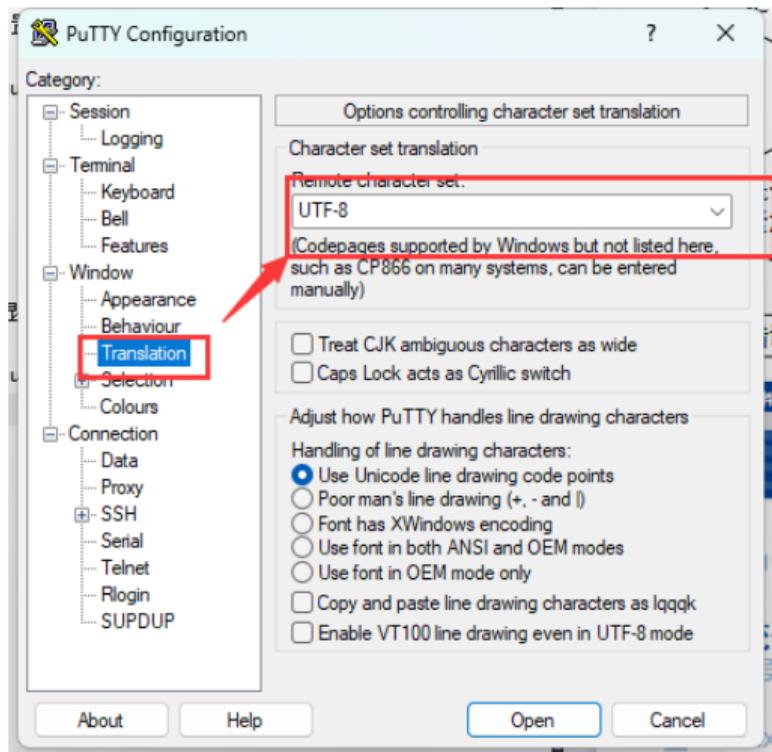
- ▶ 选择“SSH”协议
- ▶ “Remote host”里面填写服务器 IP 地址
- ▶ 选择使用“interactive shell”



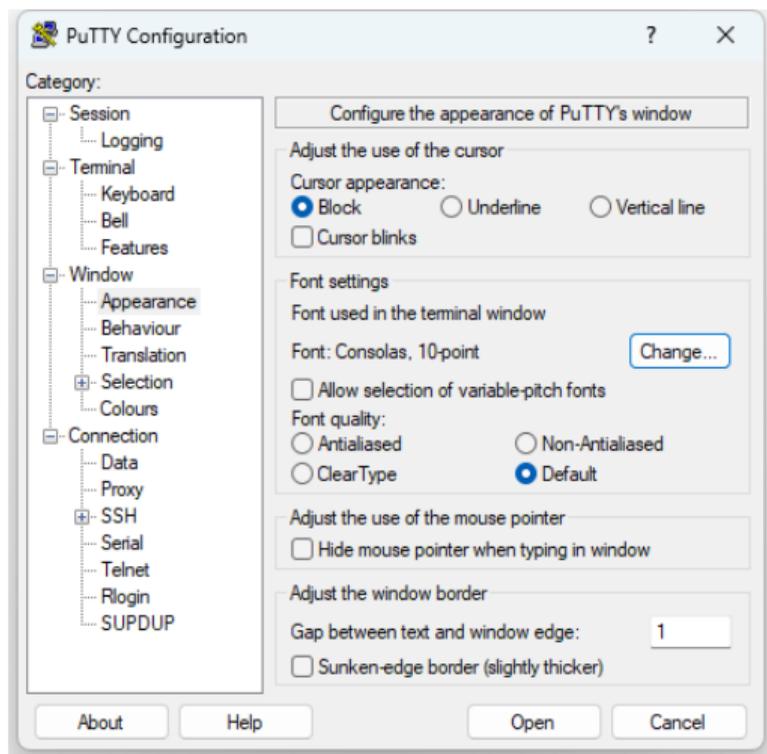
登录后在左侧会显示服务器上的文件，可以直接使用右键菜单打开编辑，保存文件时会自动同步到服务器上。



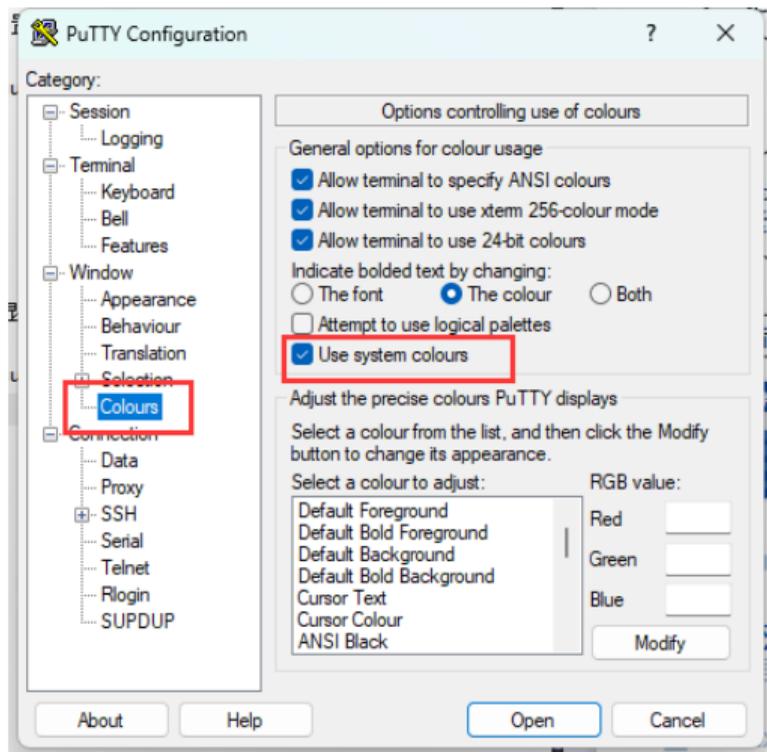
登录提示为中文，需要使用
UTF-8 编码才能正确显示



默认字体有时显示效果不好，可以尝试采用 consolas 等无衬线等宽字体

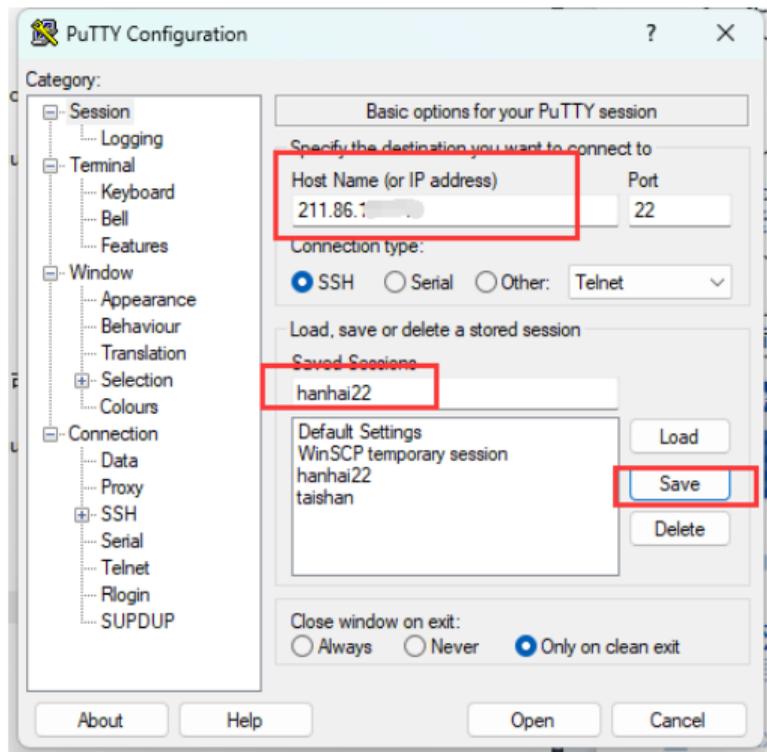


默认的黑底白字在某些情况下可能不容易看清，可以更改为白底黑字





填入服务器 ip、自行起名后可以保存设置，下次 load 后可以直接连接

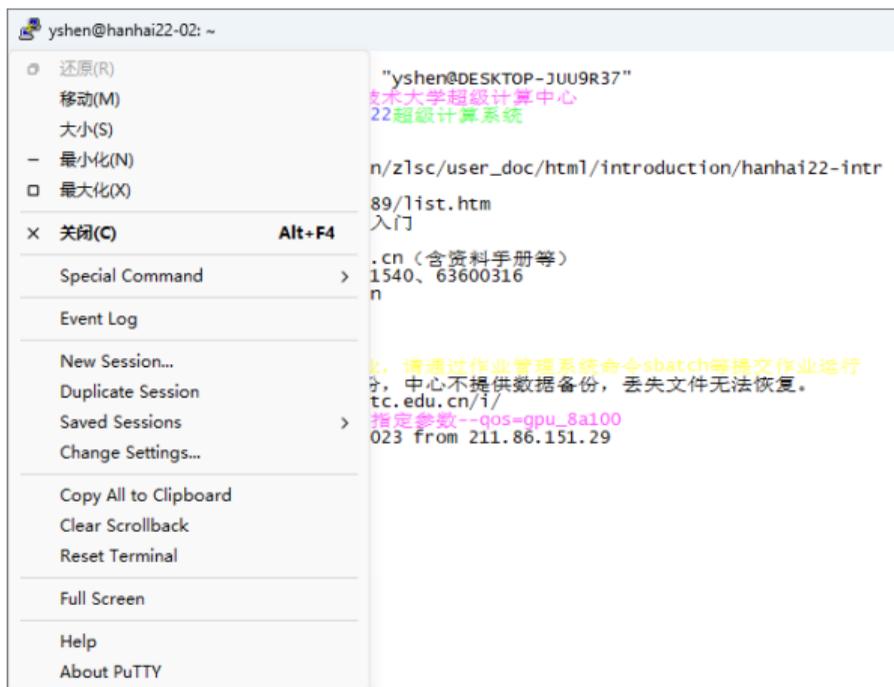


ssh 主机都是“指纹”作为安全验证，当连接到新的服务器或者原先的服务器发生变化（可能是重装系统，或者连接了被伪造的假服务器），会有如下提示，如果是首次登录可确认正常，如果不是要跟服务器管理员确认是不是服务器的相关设置发生了改变。

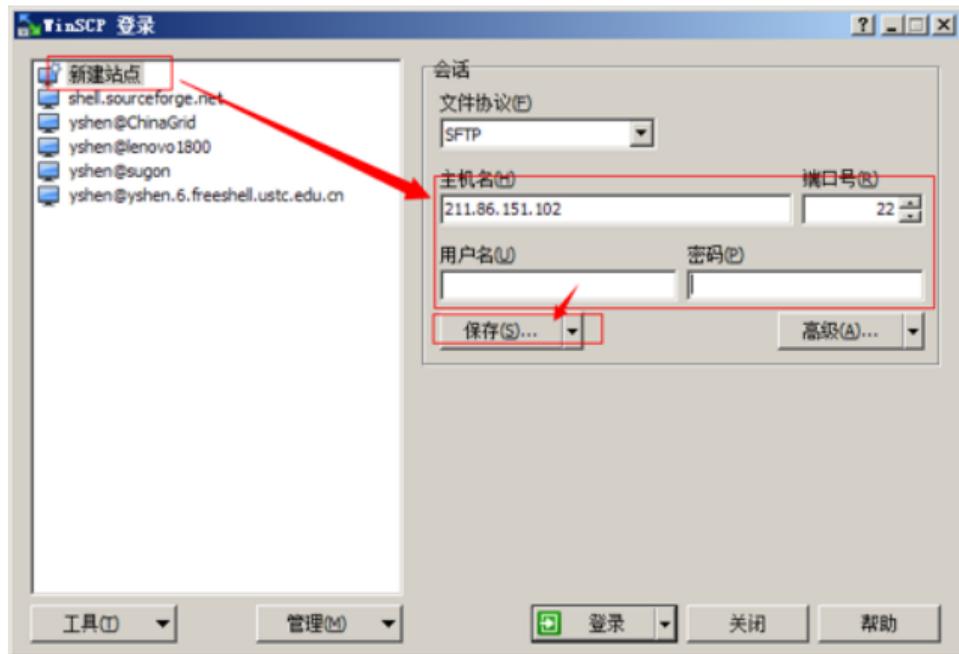




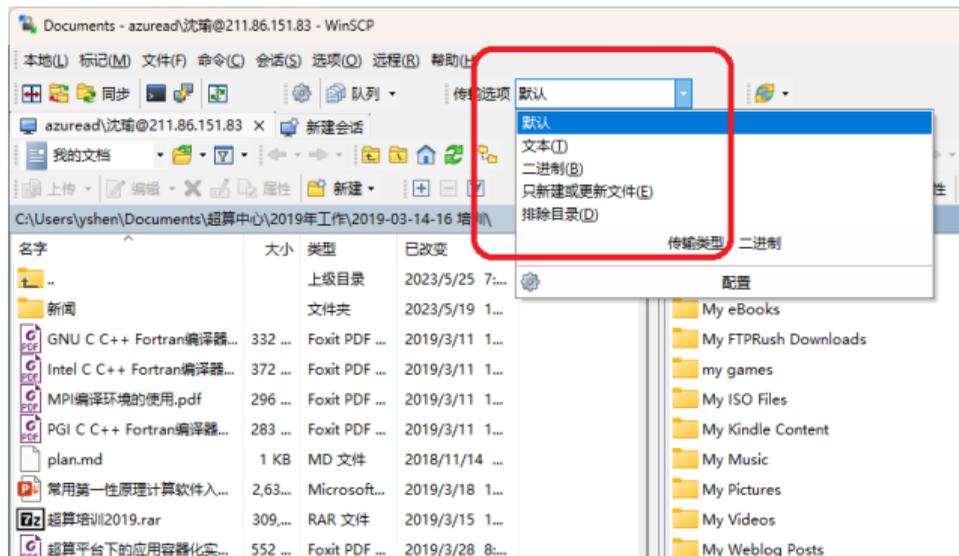
putty 没有菜单栏，设置菜单需要点击标题栏最左侧图标来打开



- ▶ 文件协议：sftp
- ▶ 主机名：服务器 IP 地址
- ▶ 端口号：22



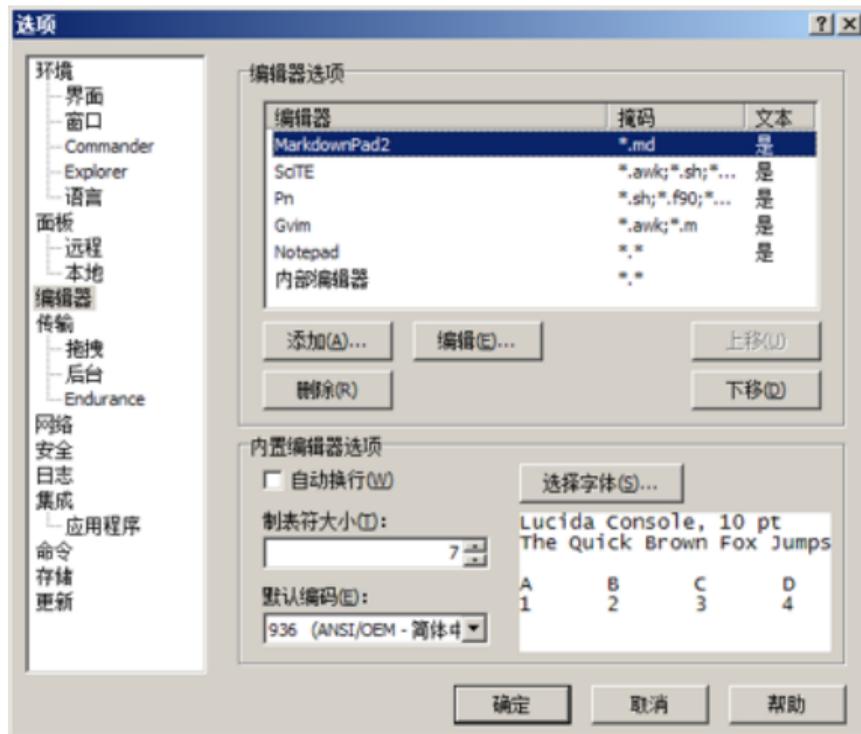
由于文本文件的换行符在 linux 和 windows 下有区别，有时候需要设置使用文本模式传输数据。



可以在 winscp 里面设置好编辑器，以便直接编辑服务器上的文件。

注意

- ▶ 不推荐使用 windows 默认的记事本做编辑器，因为它不能正确识别 linux 下的换行符
- ▶ 推荐使用 notepad3 这样轻量级、支持多种编程语言高亮的编辑器





服务器几乎支持所有 ssh 和 sftp 客户端软件，值得推荐的还有以下一些选择

- ▶ windows terminal: ssh + scp
- ▶ xshell+xftp

注意：不推荐使用 vscode 自带的 ssh 客户端连接到服务器上编辑文件，因为它过于臃肿，会在服务器上启动比较占用资源的后台进程。



- 1 账户申请
- 2 系统登录与软件设置
- 3 Linux 基础
 - Linux 命令 ■ 环境变量
- 4 计算作业提交和管理
- 5 文件编辑和程序编译



- 1 账户申请
- 2 系统登录与软件设置
- 3 Linux 基础
 - Linux 命令 ■ 环境变量
- 4 计算作业提交和管理
- 5 文件编辑和程序编译



登录服务器后进入 linux 字符界面，当前目录为用户的主目录 ()

常用 linux 命令

- ▶ “ls”：列出当前目录下文件 (参数：“-h”，“-l”，“-a”，…)
- ▶ “cd”：改变目录
- ▶ “pwd”：查看当前工作目录
- ▶ “cat”：查看文件内容
- ▶ “more”/“less”：分页显示

常用 linux 程序

- ▶ 编辑器：nano, vim
- ▶ 压缩打包程序：tar, gzip



- ▶ 不明白的命令可以使用“man 命令名”查看使用手册，例如“man ls”
- ▶ 输入命令或者参数前几个字符后按 `tab` 键，如果只有一种可能则自动补全，否则再按一次 `tab` 会列出所有可能选择。用 `tab` 补全还可以避免偶尔打错字。
- ▶ 大多数命令加上“--help”可输出简要帮助信息，例如“ls --help”
- ▶ 屏幕输出太多一页看不完可以在命令后加上“| more”或者“| less”来分页显示，例如“ls --help | more”
- ▶ 推荐参考书《鸟哥的 linux 私房菜》
<https://wizardforcel.gitbooks.io/vbird-linux-basic-4e/content/>



- 1 账户申请
- 2 系统登录与软件设置
- 3 Linux 基础
 - Linux 命令 ■ 环境变量
- 4 计算作业提交和管理
- 5 文件编辑和程序编译



- ▶ 环境变量：运行环境中存在的通用变量
- ▶ 查看环境变量：“env”
- ▶ 设置环境变量：“export”
- ▶ 重要环境变量：
 - ▶ PATH: 运行程序路径
 - ▶ LD_LIBRARY_PATH: 动态链接库路径
 - ▶ OMP_NUM_THREADS: OpenMP 默认线程数
 - ▶ MKL_NUM_THREADS: Intel MKL 默认线程数
 - ▶ EDITOR: 默认编辑器



超算平台上安装众多软件，为了避免不同软件运行环境发生冲突，系统使用 module 来管理环境变量，module 会自动增加或者删除当前环境变量的内容。

常用命令

- ▶ “module avail”：列出可用 module 模块
- ▶ “module display 模块名”：列出模块说明
- ▶ “module load 模块名”：载入模块
- ▶ “module rm 模块名”：去除已载入的模块
- ▶ “module purge”：清除所有载入的模块

实验：可以先用“env”命令查看现有环境变量，然后使用“module load”命令载入一个模块，再用“env”命令查看载入后的环境变量，并和载入前的进行对比



超算系统上已经预安装各种软件并制作了相应的模块，大体上分类如下：

- ▶ /opt/MODULES/app：常用科研应用
- ▶ /opt/Modules/compiler：编译环境，各种版本的 Intel、PGI、GCC 编译器等
- ▶ /opt/Modules/lib：运行库支持
- ▶ /opt/Modules/python：python 环境
- ▶ /opt/Modules/tool：工具
- ▶ /opt/Modules/not_recommended：过于陈旧不推荐的模块

各个系统上的模块有所不同，具体分类以各个系统上实际情况为准，用户可以使用“`module avail`”查看具体包含的模块名。



- 1 账户申请
- 2 系统登录与软件设置
- 3 Linux 基础
- 4 计算作业提交和管理
 - Slurm 作业调度系统 ■ LSF 作业调度系统 ■ 系统状态查看
- 5 文件编辑和程序编译

超算系统有很多用户同时使用，为了避免计算冲突，采用非交互式提交计算任务。

用户提交作业到调度系统

作业参数正确，加入相应队列等待分配计算资源

作业调度系统根据一定的算法处理等待作业队列，给优先级最高的作业分配计算资源

作业分配到计算资源后，自动到计算节点上开始进行计算

作业运行完成后，释放计算资源

作业开始运行后，用户无法直接干预作业运行，因此用户需要先调试好程序，再提交作业。系统提供了“test”队列，供用户测试程序能否正常运行使用。“test”队列仅供测试使用，并限制了很短的运行时间，请不要用它长期运行计算任务。

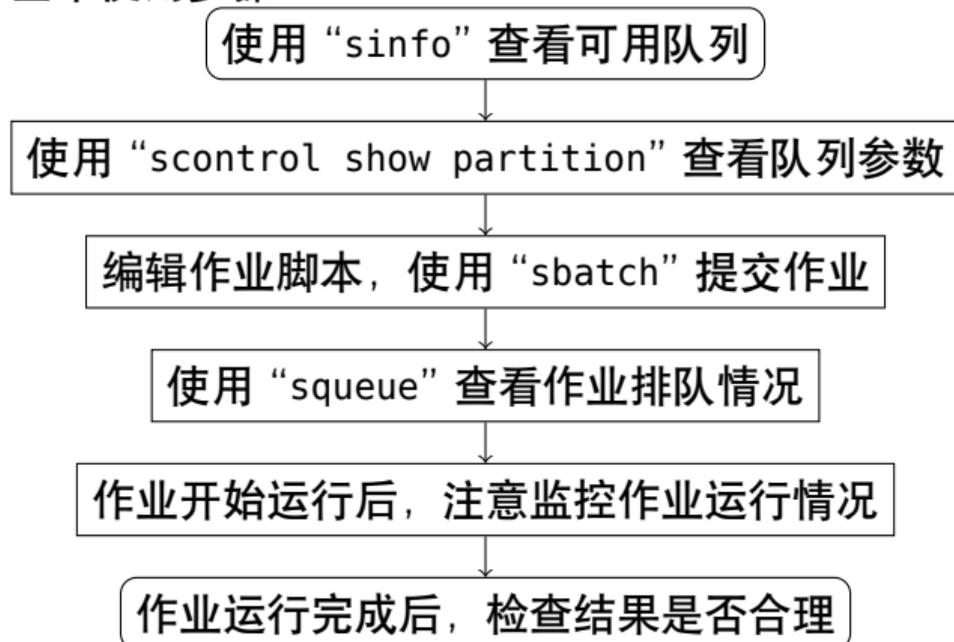


- 1 账户申请
- 2 系统登录与软件设置
- 3 Linux 基础
- 4 计算作业提交和管理
 - Slurm 作业调度系统 ■ LSF 作业调度系统 ■ 系统状态查看
- 5 文件编辑和程序编译



- ▶ 瀚海 20 和瀚海 22 系统使用 Slurm 作业调度系统。
- ▶ Slurm 是一个开源、容错、高度可扩展的集群管理和作业调度系统。
- ▶ Slurm Project:
<https://slurm.schedmd.com/overview.html>

基本使用步骤：





“sinfo” 用来查看 slurm 节点和队列 (partition) 状态。

一个典型的输出如下：

```
yshen@hanhai22-02:~$ sinfo
PARTITION    AVAIL    TIMELIMIT    NODES    STATE    NODELIST
GPU-8A100    up 5-00:00:00    6    mix    gnode[13-14,16-17,19,21]
GPU-8A100    up 5-00:00:00    2    alloc  gnode[12,18]
CPU-64C256GB up 15-00:00:00    1    drain  cnode40
CPU-64C256GB up 15-00:00:00    1    mix    cnode19
CPU-64C256GB up 15-00:00:00   77    alloc  cnode[01-18,21-39,41-80]
CPU-64C256GB up 15-00:00:00    1    idle   cnode20
CPU-96C3TB   up 15-00:00:00    2    mix    fnode[01,05]
CPU-96C3TB   up 15-00:00:00    6    alloc  fnode[03,06-10]
CPU-96C3TB   up 15-00:00:00    2    idle   fnode[02,04]
visual       up 1-00:00:00    1    idle   hanhai22-03
yshen@hanhai22-02:~$
```

其中，PARTITION 列是可用的队列，STATE 是状态，NODES 是各个分区不同状态的节点数目



常见的 STATE 有:

- idle** 节点完全空闲, 可以接受作业分配。
- alloc** 节点资源已经全部分配。
- mix** 节点部分资源已分配, 还有部分空闲。例如某节点有 8 块 GPU 卡, 已分配使用 1 块 GPU 卡, 还有 7 块 GPU 卡空闲, 那它就处于 mix 状态, 还可以接受不超过 7 个 GPU 卡的作业。
- drain** 节点有异常, 需要维护, 暂时不能分配作业。



可以用“scontrol show partition 队列名”查看各个队列的使用参数，例如：

```
yshen@hanhai22-02:~$ scontrol show partition GPU-8A100
PartitionName=GPU-8A100
  AllowGroups=ALL AllowAccounts=ALL AllowQos=gpu_8a100
  AllocNodes=ALL Default=NO QoS=gpu_8a100
  DefaultTime=NONE DisableRootJobs=YES ExclusiveUser=NO GraceTime=0 Hidden=NO
  MaxNodes=8 MaxTime=5-00:00:00 MinNodes=0 LLN=NO MaxCPUsPerNode=UNLIMITED
  Nodes=gnode[12-14,16-19,21]
  PriorityJobFactor=1 PriorityTier=1 RootOnly=NO ReqResv=NO OverSubscribe=NO
  OvertimeLimit=NONE PreemptMode=OFF
  State=UP TotalCPUs=512 TotalNodes=8 SelectTypeParameters=NONE
  JobDefaults=(null)
  DefMemPerNode=UNLIMITED MaxMemPerNode=UNLIMITED
  TRES=cpu=512,mem=7812.50G,node=8,billing=512,gres/gpu=64
```

其中需要关注的参数有：

QoS QoS 策略名称，这里是 `gpu_8a100`

MaxTime 最长作业时间，超时作业会被强行终止



用户作业要通过“sbatch”命令用脚本方式提交，典型的脚本模板在用户开户时会预先放在用户的主目录里面。

例如：

```
1 #!/bin/sh
2 #An example for GPU job.
3 #SBATCH -J job_name
4 #SBATCH -o job-%j.log
5 #SBATCH -e job-%j.err
6 #SBATCH -p GPU-8A100
7 #SBATCH -N 1 -n 2
8 #SBATCH --gres=gpu:2
9 echo Time is `date`
10 echo Directory is $PWD
11 echo This job runs on the following nodes:
12 echo $SLURM_JOB_NODELIST
13 echo This job has allocated $SLURM_JOB_CPUS_PER_NODE cpu core.
14 ./yourprog
```

其中第 3 行到第 8 行是 sbatch 参数，分别控制作业名，日志输出文件，日志错误文件，队列名，节点数和 CPU 核心数，GPU 卡数。具体参数可以使用“man sbatch”命令查看。第 9 行到第 13 行是显示一些作业运行时的信息，可以在日志输出文件中看到。第 14 行在实际运行时要改为正确的程序或者脚本命令。在一个作业里面可以运行多个计算，将运行命令依次写在脚本里面即可。

使用 `sbatch` 提交到系统上的作业不一定能立即运行，使用“`squeue`”命令可以看到系统上正在运行和排队的所有作业。使用“`squeue -u 用户名`”可以看到该用户的作业的排队情况，例如：

```
yshen@hanhai22-02:~$ squeue -u luomj0503
      JOBID PARTITION    NAME   USER  ST       TIME  NODES NODELIST(REASON)
      20215 GPU-8A100    test luomj050 PD        0:00      1 (QOSMaxJobsPerUserLimit)
      20240 GPU-8A100    test luomj050 PD        0:00      1 (QOSMaxJobsPerUserLimit)
      19234 GPU-8A100    test luomj050 R 3-14:00:50      1 gnode21
      19235 GPU-8A100    test luomj050 R 3-08:08:36      1 gnode16
      20212 GPU-8A100    test luomj050 R      21:04:46      1 gnode14
      20213 GPU-8A100    test luomj050 R      21:04:11      1 gnode19
      20214 GPU-8A100    test luomj050 R      21:02:09      1 gnode21
```

其中，`ST` 表示作业状态，`PD` 表示在等待，`R` 表示在运行。等待的作业在 `NODELIST(REASON)` 栏会给出原因，这里的原因“(QOSMaxJobsPerUserLimit)”表示超过了单用户最大作业限制从而等待。运行的作业会显示运行节点，用户可以在作业运行时通过“`ssh 节点名`”登录到运行作业的节点上通过“`top`”或者“`nvidia-smi`”等命令查看 CPU 或者 GPU 工作状态，确认程序正确运行。



作业在运行时，屏幕上的输出会被保存到“-o”选项指定的日志文件和“-e”选项指定的错误文件中，其中文件名中的“%j”会被替换为作业号。
如果作业有问题，可以查看这两个文件中的内容，来判断可能时什么问题。

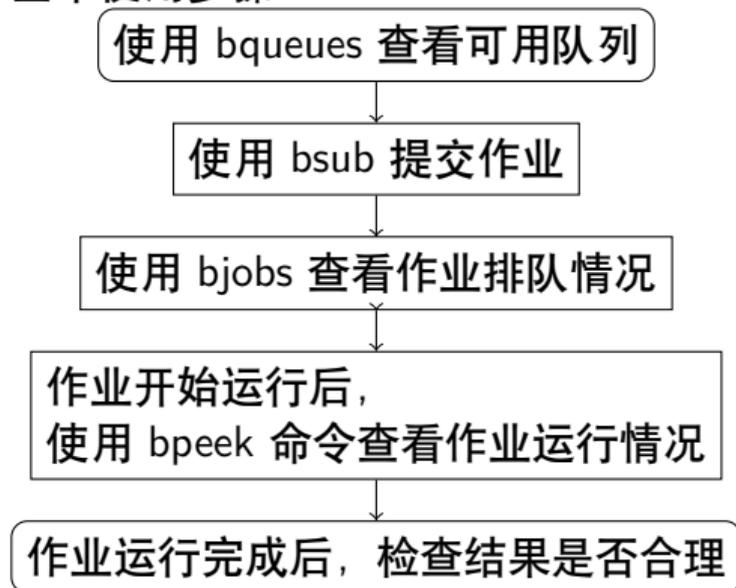


- 1 账户申请
- 2 系统登录与软件设置
- 3 Linux 基础
- 4 计算作业提交和管理
 - Slurm 作业调度系统 ■ LSF 作业调度系统 ■ 系统状态查看
- 5 文件编辑和程序编译



- ▶ TC4600 系统使用 IBM Platform LSF 作业调度系统
- ▶ LSF 作业调度系统采用命令行方式提交任务，更简单方便

基本使用步骤：





通过“bqueues -u \$USER”可以查看当前用户可用的队列，“\$USER”可以用自己的用户名代替。

“bqueues -l 队列名”可以查看队列的详细情况。右图为“longopa”队列的详细情况。其中，“RUNLIMIT”为最长作业时间限制，图中最长作业时间限制是 14400 分钟，即 10 天；

“TASKLIMIT”为任务并行规模限制，图中最小为 113 个进程，最多为 280 个进程。

```

yshen@tc4600v4:~
(base) yshen@tc4600v4:~ $ bqueues -l longopa

QUEUE: longopa
-- Maximum number of job slots that each user can use in this queue is 960.

PARAMETERS/STATISTICS
PRIO NICE STATUS          MAX JL/U JL/P JL/H NJOBS  PEND   RUN  SSUSP USUSP  RSV
59   20 Open:Active      1400 280   -   - 10938 10108  830   0     0   0
Interval for a host to accept two jobs is 0 seconds

RUNLIMIT
14400.0 min

TASKLIMIT
113 113 280

SCHEDULING PARAMETERS
loadSched r15s r1m r15m ut pg io ls it tmp swp mem
loadStop  -  -  -  -  -  -  -  -  -  -  -

          adapter_windows poe nrt_windows
loadSched -  -  -  -
loadStop  -  -  -  -

SCHEDULING POLICIES: EXCLUSIVE

USERS: scc_group/ paiduigroup/
HOSTS: node352 node353 node354 node340 node391 node392 node393 node394 node395
node396 node397 node398 node399 node400 node341 node408 node409 node410 node401
node402 node403 node404 node405 node406 node407 node356 node342 node419 node420
node411 node412 node413 node414 node415 node416 node417 node418 node343 node430
node421 node422 node423 node424 node425 node426 node427 node428 node429 node440
node431 node432 node433 node434 node435 node436 node437 node438 node439 node345
node331 node450 node441 node442 node443 node444 node445 node446 node447 node448
node449 node309 node310 node301 node302 node303 node304 node305 node306 node307
node308 node346 node332 node320 node311 node312 node313 node314 node315 node316
node317 node318 node319 node347 node333 node330 node321 node322 node324 node325
node326 node327 node328 node329 node348 node349 node335 node337 node370 node361
node362 node363 node364 node365 node366 node367 node368 node369 node380 node371
node372 node373 node374 node376 node377 node378 node379 node339 node381 node382
node383 node385 node386 node387 node388 node389 node390 node350

POST_EXEC: /home2/user_input/collectviaPOST_EXEC/script/collectAll.sh
RES_REQ: span[ptile=28]

(base) yshen@tc4600v4:~ $
    
```



用户可以使用 bsub 命令提交作业，下面为一些提交作业的例子

提交普通作业 “bsub -q test -o %J.log -e %J.err -n 1 ./a.out”

提交 mpi 作业 “bsub -q test -o %J.log -e %J.err -n 20 mpijob ./a.out”

提交 mpi+openmp 混合并行作业

“bsub -q test -o %J.log -e %J.err -n 20 mixjob -t 2 ./a.out”



bpeek 可以在作业运行的时候查看程序的屏幕输出，直接使用“bpeek 作业号”即可。还可以使用“bpeek -f 作业号”持续观察屏幕输出

```
yshen@tc4600v4:~/test/bpeek
(base) yshen@tc4600v4:~/test/bpeek $ bsub -q test -n 1 -o %J.log -e %J.err ./out
.sh
Job <5027504> is submitted to queue <test>.
(base) yshen@tc4600v4:~/test/bpeek $ bpeek 5027504
<< output from stdout >>
output 1
output 2
output 3
output 4
output 5

<< output from stderr >>
(base) yshen@tc4600v4:~/test/bpeek $ bpeek -f 5027504
<< output from stdout >>
output 1
output 2
output 3
output 4
output 5
output 6
output 7
output 8
output 9
█
```



- 1 账户申请
- 2 系统登录与软件设置
- 3 Linux 基础
- 4 计算作业提交和管理
 - Slurm 作业调度系统 ■ LSF 作业调度系统 ■ 系统状态查看
- 5 文件编辑和程序编译

在作业运行时可以直接 ssh 到计算节点上，使用“top”命令查看系统状态，“top”命令会不断刷新系统工作状态
 “top”命令界面如下：

```
top - 13:25:52 up 13 days, 1:52, 2 users, load average: 32.89, 32.81, 32.72
Tasks: 821 total, 4 running, 817 sleeping, 0 stopped, 0 zombie
%Cpu(s): 51.5 us, 0.2 sy, 0.0 ni, 48.4 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 1031729.+total, 1000289.+free, 8107.1 used, 23332.6 buff/cache
MiB Swap: 16384.0 total, 16362.8 free, 21.2 used, 1018445.+avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2582703	luomj05+	20	0	7165924	235616	127236	R	2265	0.0	1426:32	gmx
2589884	luomj05+	20	0	6071636	200756	126884	R	752.9	0.0	0:13.91	gmx
2585851	xhy1369+	20	0	11.7g	2.5g	533664	R	100.0	0.2	33:47.55	python
2589903	yshen	20	0	23808	5656	4044	R	5.9	0.0	0:00.02	top
1	root	20	0	168100	11900	6548	S	0.0	0.0	12:02.81	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:02.48	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq

在作业运行时可以直接 ssh 到计算节点上，使用“nvidia-smi”命令查看系统 GPU 状态

右图是“nvidia-smi”的输出示例，其中最上面显示了“nvidia-smi”程序、驱动程序和 CUDA 的版本号；主要部分显示共有 8 块 GPU 卡，其中前三块负载分别是 85%、37% 和 84%，后面 5 块空闲；最下面显示了使用显卡计算的进程信息。可见有两个进程是 gmx，使用了较小的显存，一个 python 进程使用了大一些的显存。

```
yshen@gnode21:~$ nvidia-smi
Sun May 28 13:24:51 2023
```

NVIDIA-SMI 525.60.13 Driver Version: 525.60.13 CUDA Version: 12.0							
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	NVIDIA A100-SXM...	On	00000000:10:00:0	off	85%	0	
N/A	42C	P0 189w / 400w	451MiB / 81920MiB			Default	Disabled
1	NVIDIA A100-SXM...	On	00000000:16:00:0	off	37%	0	
N/A	29C	P0 73w / 400w	1847MiB / 81920MiB			Default	Disabled
2	NVIDIA A100-SXM...	On	00000000:49:00:0	off	84%	0	
N/A	41C	P0 187w / 400w	451MiB / 81920MiB			Default	Disabled
3	NVIDIA A100-SXM...	On	00000000:4d:00:0	off	0%	0	
N/A	27C	P0 59w / 400w	0MiB / 81920MiB			Default	Disabled
4	NVIDIA A100-SXM...	On	00000000:89:00:0	off	0%	0	
N/A	27C	P0 59w / 400w	0MiB / 81920MiB			Default	Disabled
5	NVIDIA A100-SXM...	On	00000000:8e:00:0	off	0%	0	
N/A	27C	P0 61w / 400w	0MiB / 81920MiB			Default	Disabled
6	NVIDIA A100-SXM...	On	00000000:c5:00:0	off	0%	0	
N/A	27C	P0 62w / 400w	0MiB / 81920MiB			Default	Disabled
7	NVIDIA A100-SXM...	On	00000000:c9:00:0	off	0%	0	
N/A	28C	P0 60w / 400w	0MiB / 81920MiB			Default	Disabled

Processes:							
GPU	GI	CI	PID	Type	Process name	GPU Memory	
ID	ID	ID				Usage	
0	N/A	N/A	2582703	C	gmx	448MiB	
1	N/A	N/A	2585851	C	python	1844MiB	
2	N/A	N/A	2584464	C	gmx	448MiB	

```
yshen@gnode21:~$
```



- ▶ 注意软件版权。超算平台上预安装的软件很多，有的是开源免费的，也有的是商业软件。目前安装的商业软件中只有 MATLAB 和 Gaussian 已经购买了全校授权，其它商业软件在使用前请确保已经**自行购买**了使用权；
- ▶ 用户自己需要使用的软件可以自行安装在自己的目录里面，一般编译的时候需要单独指定安装目录；
- ▶ 如果自行安装软件时有困难，可以请超算老师协助安装；
- ▶ 如果认为某软件可能有多个用户都会使用，可以推荐安装到系统的公共软件目录里面。



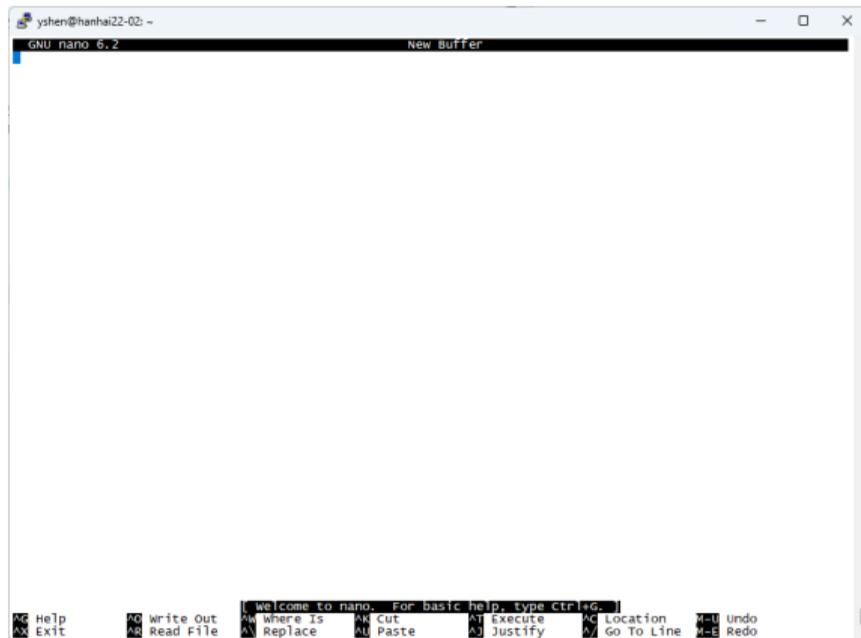
- 1 账户申请
- 2 系统登录与软件设置
- 3 Linux 基础
- 4 计算作业提交和管理
- 5 文件编辑和程序编译
 - 文本编辑器 ■ 编译器使用



- 1 账户申请
- 2 系统登录与软件设置
- 3 Linux 基础
- 4 计算作业提交和管理
- 5 文件编辑和程序编译
 - 文本编辑器 ■ 编译器使用



- ▶ 单模式编辑器
- ▶ 大部分指令都是 ctrl/alt/快捷键
- ▶ Ctrl+G: 获取帮助 (Get help)





Vim 是功能最强大、使用最广泛的文本编辑器之一，尤其适合在远程终端上使用。

Vim 是多模式编辑器，即文本输入和编辑等是在不同模式下操作的。常用模式有

普通模式 启动后默认进入普通模式。该模式下可执行一般的编辑命令，比如复制、剪切、粘贴移动光标等。编辑命令为普通按键，一般无需 ctrl 或者 alt；任何模式下按 Esc 键都会返回普通模式。

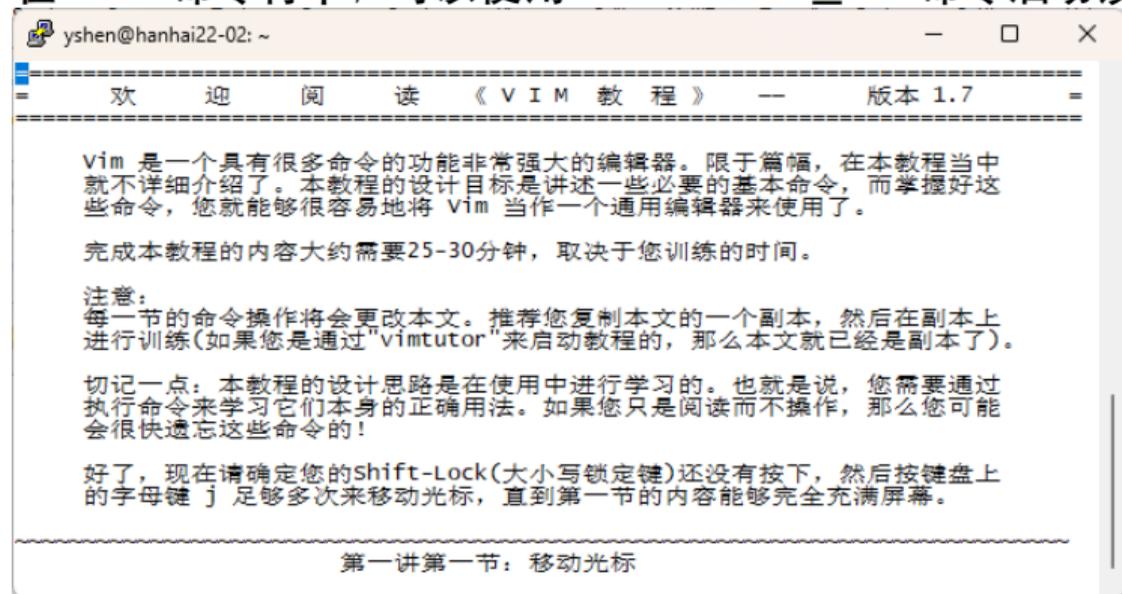
插入模式 在普通模式下按 i（在光标当前位置插入文本）或者 a（在光标后面一个字符的位置插入）等命令可进入插入模式。该模式下可输入文件内容。

可视模式 在普通模式下按 v 进入可视模式（按 ctrl+v 进入列选择模式）。在可视模式下移动光标会选择一段文本，这时执行其他命令将作用到选中的文本上。

命令模式 在普通模式下按: 进入。可进行打开/保存文件 (:o :e)，退出 vim(:q :q!)，搜索/替换 (/ ?s) 等操作

Vim 的多模式使得它与常见的文本编辑相比入门较难，最好的入门教程是 vim 自带的 vimtutor 命令，它会在启动时自动生成一个教程的临时副本，初学者可以在读教程的同时进行练习，一般 25 30 分钟即可熟悉 vim 最重要的基本功能。

在 linux 命令行下，可以使用“vimtutor zh_cn”命令启动该教程的中文版，如下图



```
yshen@hanhai22-02: ~  
-----  
= 欢 迎 阅 读 《 V I M 教 程 》 一 版 本 1.7 =  
-----  
Vim 是一个具有很多命令的功能非常强大的编辑器。限于篇幅，在本教程当中  
就不详细介绍了。本教程的设计目标是讲述一些必要的基本命令，而掌握好这  
些命令，您就能够很容易地将 Vim 当作一个通用编辑器来使用了。  
  
完成本教程的内容大约需要25-30分钟，取决于您训练的时间。  
  
注意：  
每一节的命令操作将会更改本文。推荐您复制本文的一个副本，然后在副本上  
进行训练(如果您是通过"vimtutor"来启动教程的，那么本文就已经是副本了)。  
  
切记一点：本教程的设计思路是在使用中进行学习。也就是说，您需要通过  
执行命令来学习它们本身的正确用法。如果您只是阅读而不操作，那么您可能  
会很快遗忘这些命令的！  
  
好了，现在请确定您的Shift-Lock(大小写锁定键)还没有按下，然后按键盘上  
的字母键 ] 足够多次来移动光标，直到第一节的内容能够完全充满屏幕。  
  
-----  
第一讲第一节：移动光标
```



基本指令是在普通模式下直接在键盘上按下的按键，可以在已有文字中方便的移动，编辑……

i/a/o/O 在当前位置/下个字符/下一行/上一行插入文字

h/j/k/l 向左/上/下/右移动光标

gg/G/0/\$ 移动光标到文件开始/文件结束/当前行开始/当前行结束

w/W/b/B 按单词/词组移动，小写向下一个移动，大写往回移动

x/X 剪切当前/前一个字符

p/P 下个字符位置/当前位置上粘贴

u 撤销上一个操作，此命令在尝试 vim 复杂功能时特别有用

快捷操作 yy (复制当前行)，dd (删除当前行) ……



一些基本指令的组合可以实现复杂的功能，组合的基本原则是：“谓语 + 宾语”，还可以配合数字修饰和其他修饰一起使用。

谓语 y (复制), d (删除), c (更改: 删除已有内容并进入插入模式), ……

宾语 gg (文件头), G (文件尾), 0 (行首), \$ (行尾), w (单词), b/[/{ (小/中/大括号), ' / “(单/双引号), ……

数字修饰 命令前加数字表示重复该命令几次

其他修饰 i (括号/引号内部), a (包括括号/引号所有内容)

一些例子如下:

- ▶ 删除 2 个单词: 2dw
- ▶ 改变单引号内容: 光标移动到括号部分后顺序按下 c i ‘
- ▶ 删除包括小括号的所有内容: 光标移动到括号部分后顺序按下 d a b

复合指令可以在使用过程中不断摸索尝试，请牢记使用“u”命令可以方便的撤销上一步错误的操作。



Vim 入门第一个难题是如何关闭当前的文件退出 vim，常见的文件操作如下（在普通模式下操作，注意前面的冒号不可少，它用来进入命令模式）

`:q` 退出 vim，如果文件已修改但未保存，会提示未保存而不是直接退出

`:q!` 放弃未保存的修改强制退出 vim

`:e+ 文件名` 打开文件

`:e+ 目录名` 打开目录文件列表

`:e!` 将文件恢复到上一次保存时状态

`:w` 保存文件

`:w+ 文件名` 将编辑内容保存到文件



查找替换 使用 `s` 命令可以实现查找替换功能，再加上 `%` 可以进行全文替换，命令后面加上 `g` 可以替换每一行的所有匹配项。
常见例子如下（在普通模式下操作，`s` 前面的冒号不可少，它用来进入命令模式）：

`:s/xxx/yyy` 当前行替换首次出现的匹配项

`:%s/xxx/yyy` 全文替换

`:%s/xxx/yyy/g` 全文替换每一行中的多个匹配项

自动补全 插入模式下按 `ctrl+p`，默认匹配当前打开文件中已经出现的单词，如果有多个匹配项，在不松开 `ctrl` 键的时候按 `p` 可以循环显示多个匹配项。



- 1 账户申请
- 2 系统登录与软件设置
- 3 Linux 基础
- 4 计算作业提交和管理
- 5 文件编辑和程序编译
 - 文本编辑器 ■ 编译器使用



Intel 编译器是超算系统上最常用的编译器之一，它包括 C/C++,FORTRAN,MPI 编译器和高性能数学库 MKL，很多常用的科学计算软件都可以使用该编译器编译。最新版本的 Intel 编译器称为 oneapi，目前为免费使用。使用编译器前需要用“module load”命令载入相关的环境变量。oneapi 之前版本的 intel 编译器的模块文件（modulefile）一般放在各个系统的/opt/MODULES/compiler 里面，oneapi 版本自行提供了很多模块文件，一般单独放在/opt/MODULES/oneapi 里面。

载入模块示例

- ▶ TC4600 上 2018_update3 版：

```
“module load intel/2018.update3 intelmpi/2018.update3 mkl/2018.update3”
```

- ▶ 瀚海 22 上 oneapi 版：

```
“module load compiler/latest mpi/latest mkl/latest”
```



- ▶ 编译串行程序：使用“icc”或者“ifort”即可，可以搭配“-xHost -Ofast”参数优化性能。
- ▶ 编译 MPI 并行程序：使用“mpiicc”或者“mpiifort”
- ▶ 使用 MKL，使用“-mkl”参数。复杂情况可以使用 MKL Link Line advisor 来配置编译参数。<https://www.intel.com/content/www/us/en/developer/tools/oneapi/onemkl-link-line-advisor.html>



- ▶ OpenMPI 为开源 MPI 并行环境
- ▶ HPCX 为超算系统使用的 IB 网络优化的 MPI，以 OpenMPI 为基础。
- ▶ 它们可以搭配不同的编译器使用，例如
 - ▶ 瀚海 22 上 “hpcx/2.12/hpcx-mpi-intel-cuda” 是 HPCX 和 intel 编译器，cuda 运行库一起搭配使用的模块文件。
 - ▶ 瀚海 20 上 “openmpi/4.1.2/intel/2019.update5” 是 openmpi 和 intel 编译器搭配使用的模块文件
- ▶ 有的时候使用这两个 mpi 会有更好的兼容性或者更好的并行效率



- ▶ 超算系统上安装了多个版本的 python 环境和 conda 环境
- ▶ 由于 python 众多的第三方包很容易产生冲突，强烈推荐用户使用“conda create”来创建自己的工作环境
- ▶ 为了尽可能的保证系统安全，系统的防火墙也会经常变化，有时会限制登录节点直接连接到校外的服务器。如果在安装 python 环境时遇到网络问题，可以随时咨询超算老师。



- ▶ 超算中心主页，资料手册栏目：<http://scc.ustc.edu.cn/389/list.htm>，用户使用手册：http://scc.ustc.edu.cn/zlsc/user_doc/html/
- ▶ 登录系统后查看登录提示
- ▶ 在系统里面输入 `faq` 命令查看常见问题帮助
- ▶ 通过邮件、QQ 群等向超算老师们咨询或者和同学们一起讨论，注意提供关键信息，例如
 - ▶ 所在系统，是在瀚海 22、瀚海 20 还是 TC4600 上遇到的问题
 - ▶ 相关的作业号、运行作业的工作目录等
 - ▶ 如果有错误信息截图的话，最好截全屏，而不是一行出错提示。出错提示信息量太少，无法判断出错原因。



谢谢!