

Linux 系统入门

生物信息学

助教-刘柯

助教-方明昊

目录

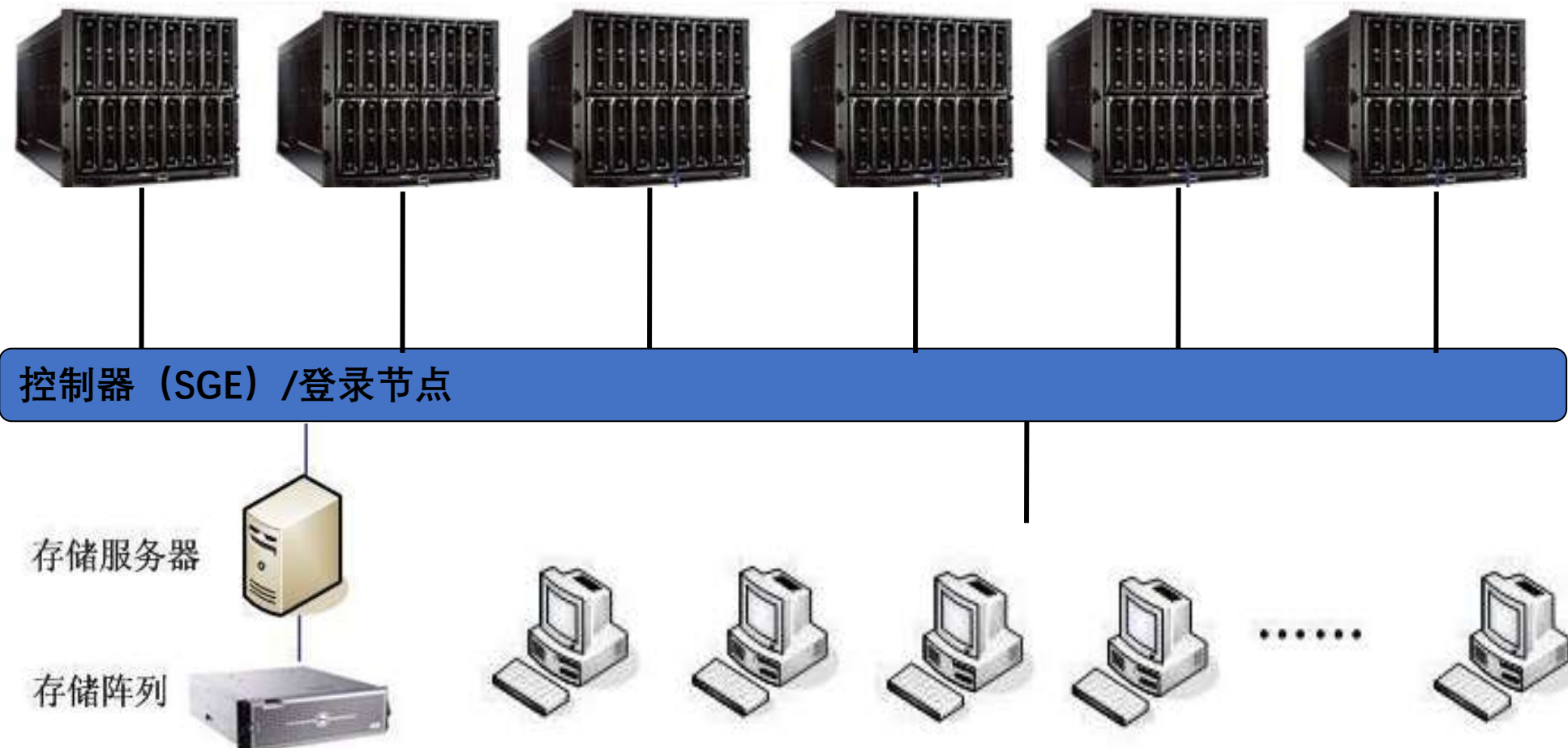
❖ Linux 基本操作

❖ Conda&Jupyter 安装(可视化分段编程)

❖ 简要环境安装

❖ 集群介绍

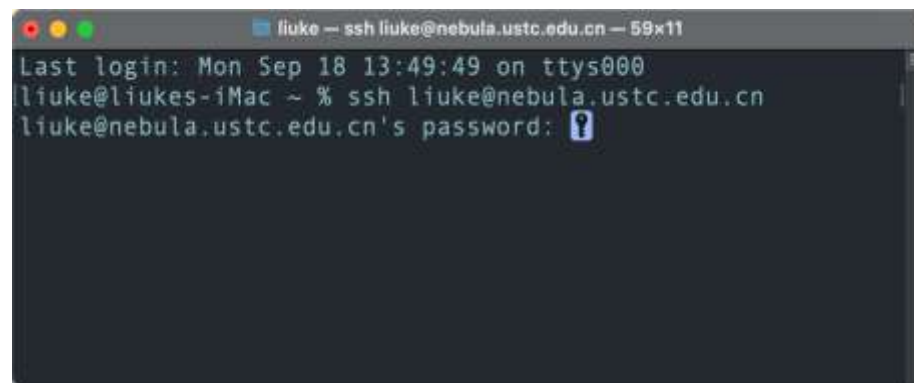
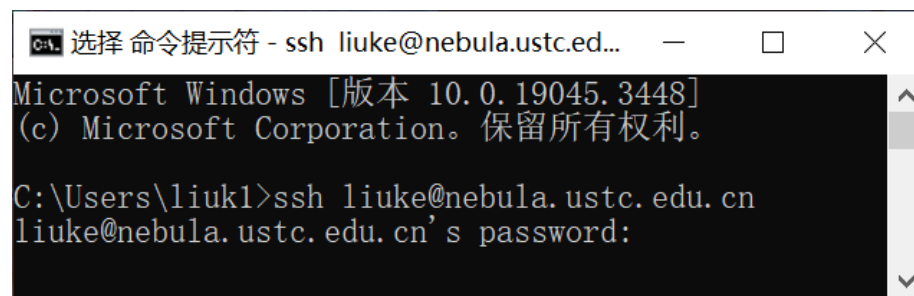
集群（cluster）就是一组计算机，它们作为一个整体向用户提供一组网络资源，这些单个的计算机系统就是集群的**节点**（node）。



❖ 集群登录

在MacOS系统或者Linux系统下打开终端(Terminal), Windows打开命令提示符

输入ssh <username>@nebula.ustc.edu.cn



重要的事情说三遍：
不要在登录节点跑程序！！
不要在登录节点跑程序！！
不要在登录节点跑程序！！

注意：用户登陆机群之后，实际上位于登陆节点（资源很少）

❖ 常用linux命令

➤ 文件及目录管理

• 列出目录 ls

```
liuke@mgt:~ — -ssh liuke@nebula.ustc.edu.cn — 3...  
(base)[liuke@mgt ~]$ls  
miniconda3  perl5      software  
ossutil     recordings  used_pbs  
ossutil64   reference   workspace  
(base)[liuke@mgt ~]$
```

相对路径和绝对路径：

绝对路径 /home/qukun/liuke/workspace

相对路径 ./workspace ../liuke/workspace

```
liuke@mgt:~ — -ssh liuke@nebula.ustc.edu.cn — 67x5  
(base)[liuke@mgt ~]$ls -la  
total 10336  
drwxr-x--- 27 liuke qukun    8192 Sep 18 14:37 .  
drwxr-x--- 96 root  qukun    4096 Sep  4 10:21 ..  
drwxr-xr-x  3 liuke qukun    4096 Sep 27  2022 .aspera
```

❖ 常用linux命令

➤ 文件及目录管理

文档属性:

[-] [rwx] [r-x] [r--]

[文档类别] [拥有者权限] [同群组权限] [其他使用者权限]

- 更改文档权限 chmod

```
liuke — liuke@mgt:~/test — ssh liuke@nebula.ustc.edu.cn — 60x8
(base)[liuke@mgt test]$ls -l
total 0
-rw-r--r-- 1 liuke qukun 244 Sep 18 16:00 test.txt
(base)[liuke@mgt test]$chmod 777 test.txt
(base)[liuke@mgt test]$ls -l
total 0
-rwxrwxrwx 1 liuke qukun 244 Sep 18 16:00 test.txt
(base)[liuke@mgt test]$
```

可读取 r - 4
可写入 w - 2
可执行 x - 1

❖ 常用linux命令

➤ 文件及目录管理

- 创建目录 mkdir

```
liuke — liuke@mgt:~ — ssh liuke@nebula.ustc.edu.cn — 60x6
(base)[liuke@mgt ~]$ls
miniconda3  ossutil64  recordings  software  workspace
ossutil     perl5       reference   used_pbs
(base)[liuke@mgt ~]$mkdir test
(base)[liuke@mgt ~]$mkdir -p test1/test2/test3
```



- 变换目录 cd

```
liuke — liuke@mgt:~/test — ssh liuke@nebula.ustc.edu.cn — 60x5
(base)[liuke@mgt ~]$cd /home/qukun/liuke/test1/test2/test3/
(base)[liuke@mgt test3]$cd ../../../../test
```


❖ 常用linux命令

➤ 文件及目录管理

- 查看绝对路径 pwd

```
liuke — liuke@mgt:~/test1/test2 — ssh liuke@nebula.ustc.edu.cn — 60x6
(base)[liuke@mgt test]$cd /home/qukun/liuke/test1/test2/
(base)[liuke@mgt test2]$ls
test3
(base)[liuke@mgt test2]$pwd
/home/qukun/liuke/test1/test2
(base)[liuke@mgt test2]$
```

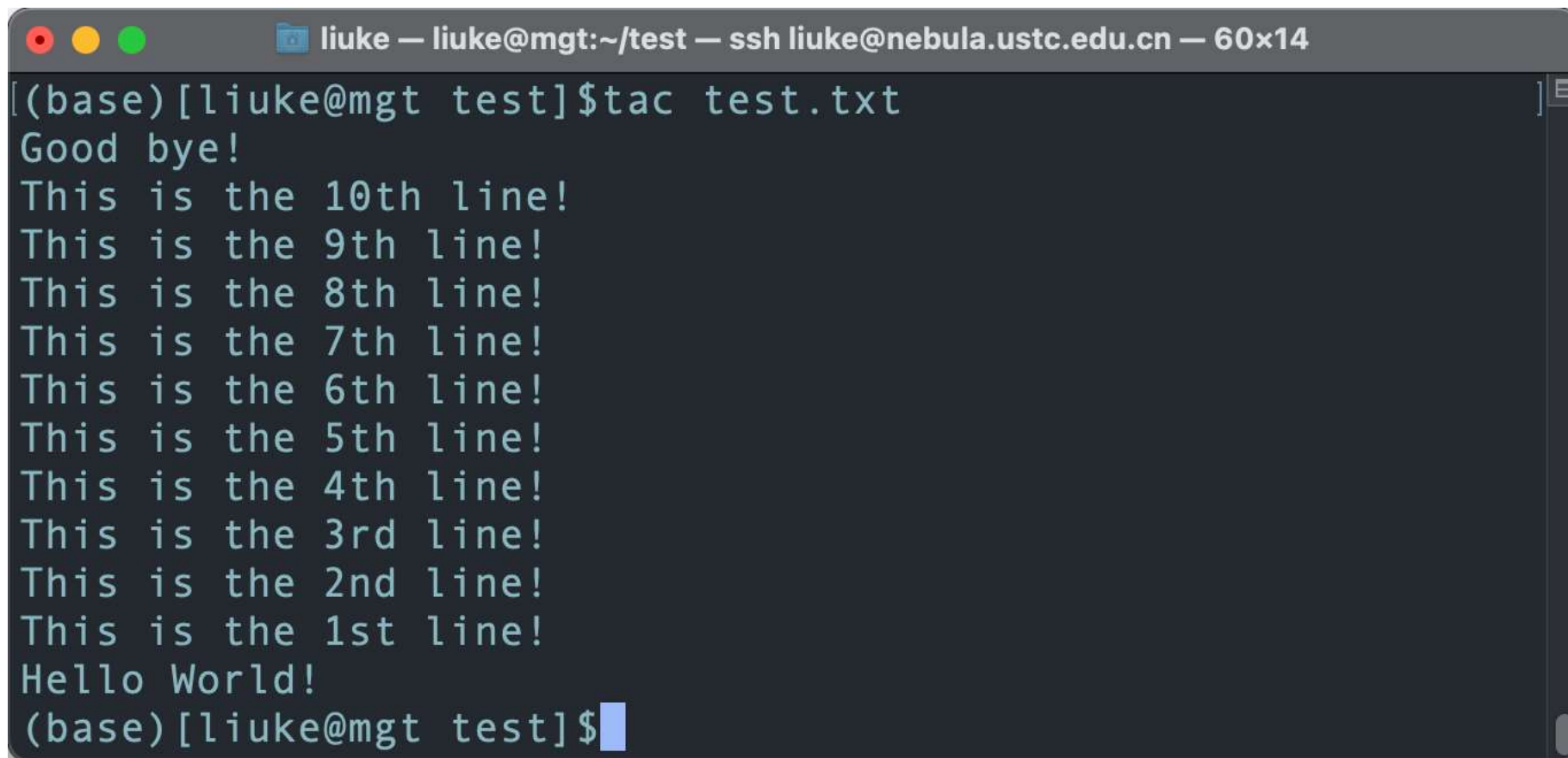
- 删除目录 rmdir

```
liuke — liuke@mgt:~ — ssh liuke@nebula.ustc.edu.cn — 60x6
(base)[liuke@mgt test]$cd ~
(base)[liuke@mgt ~]$rmdir test
(base)[liuke@mgt ~]$rmdir test1/
rmdir: failed to remove 'test1/' : Directory not empty
(base)[liuke@mgt ~]$rmdir -p test1/test2/test3/
```


❖ 常用linux命令

➤ 文档查看及管理

- cat和tac

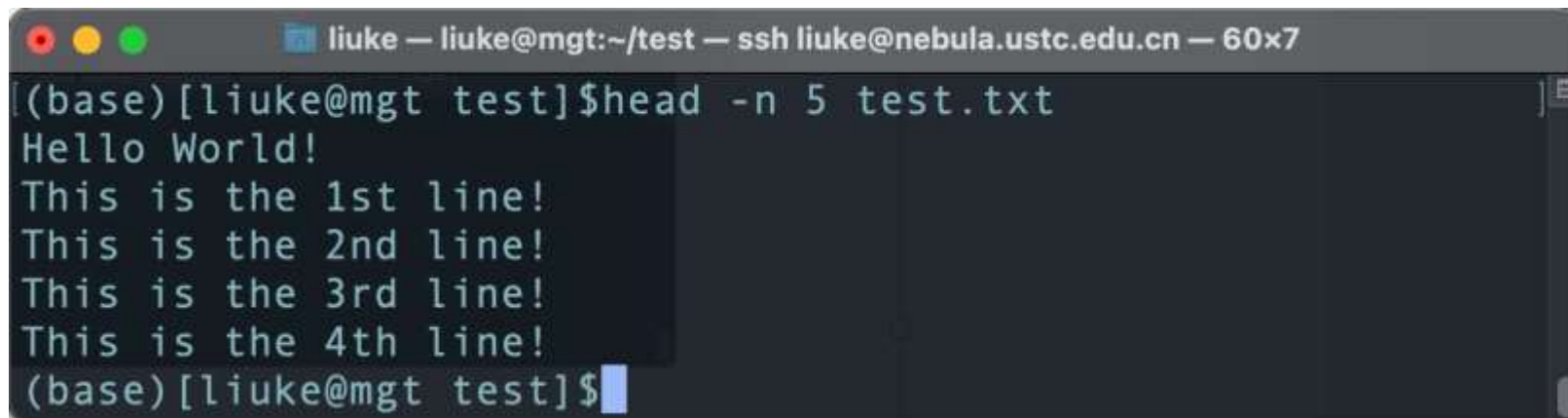


```
liuke — liuke@mgt:~/test — ssh liuke@nebula.ustc.edu.cn — 60x14
(base)[liuke@mgt test]$tac test.txt
Good bye!
This is the 10th line!
This is the 9th line!
This is the 8th line!
This is the 7th line!
This is the 6th line!
This is the 5th line!
This is the 4th line!
This is the 3rd line!
This is the 2nd line!
This is the 1st line!
Hello World!
(base)[liuke@mgt test]$
```

❖ 常用linux命令

➤ 文档查看及管理

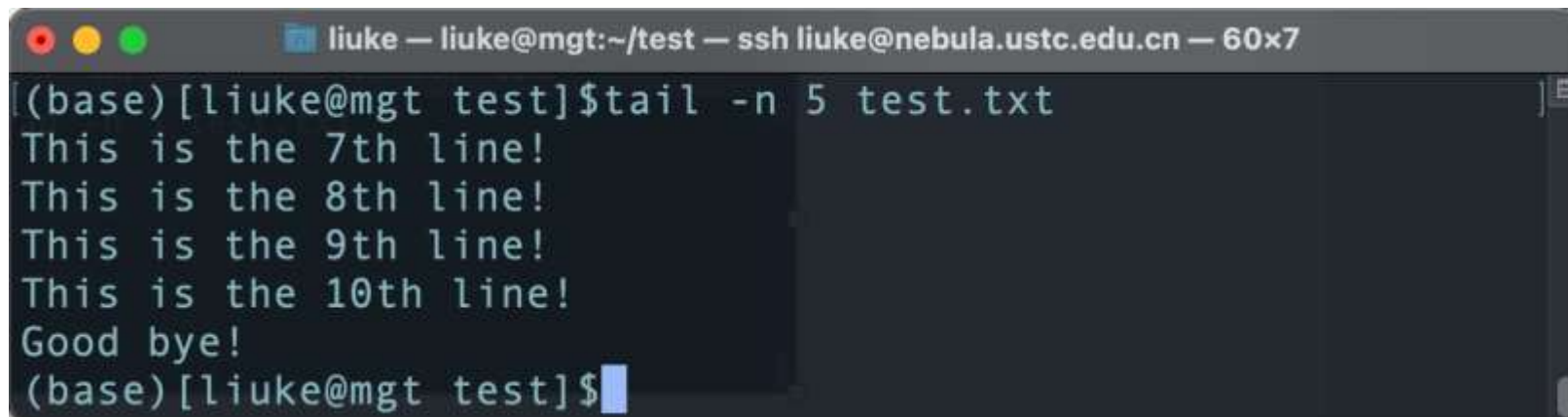
- head



```
liuke — liuke@mgt:~/test — ssh liuke@nebula.ustc.edu.cn — 60x7
(base)[liuke@mgt test]$head -n 5 test.txt
Hello World!
This is the 1st line!
This is the 2nd line!
This is the 3rd line!
This is the 4th line!
(base)[liuke@mgt test]$
```

A terminal window titled "liuke — liuke@mgt:~/test — ssh liuke@nebula.ustc.edu.cn — 60x7". The prompt is "(base)[liuke@mgt test]". The command "head -n 5 test.txt" has been executed, displaying the first five lines of the file "test.txt": "Hello World!", "This is the 1st line!", "This is the 2nd line!", "This is the 3rd line!", and "This is the 4th line!". The prompt is now "(base)[liuke@mgt test]\$".

- tail



```
liuke — liuke@mgt:~/test — ssh liuke@nebula.ustc.edu.cn — 60x7
(base)[liuke@mgt test]$tail -n 5 test.txt
This is the 7th line!
This is the 8th line!
This is the 9th line!
This is the 10th line!
Good bye!
(base)[liuke@mgt test]$
```

A terminal window titled "liuke — liuke@mgt:~/test — ssh liuke@nebula.ustc.edu.cn — 60x7". The prompt is "(base)[liuke@mgt test]". The command "tail -n 5 test.txt" has been executed, displaying the last five lines of the file "test.txt": "This is the 7th line!", "This is the 8th line!", "This is the 9th line!", "This is the 10th line!", and "Good bye!". The prompt is now "(base)[liuke@mgt test]\$".

❖ 常用linux命令

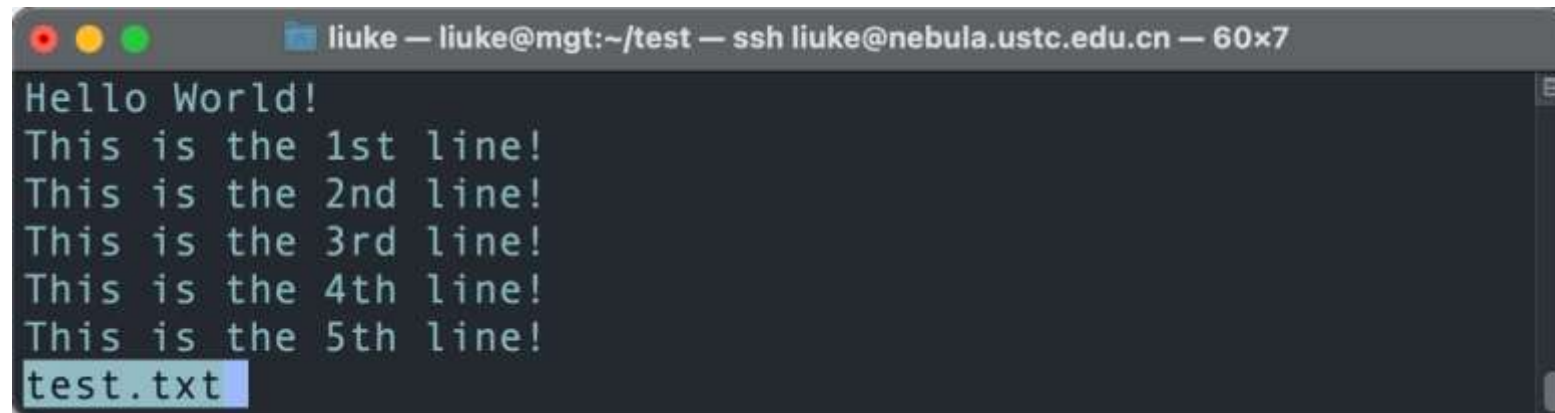
➤ 文档查看及管理

- more



A terminal window titled "liuke — liuke@mgt:~/test — ssh liuke@nebula.ustc.edu.cn — 60x7". The output of the 'more' command is displayed, showing five lines of text: "Hello World!", "This is the 1st line!", "This is the 2nd line!", "This is the 3rd line!", and "This is the 4th line!". The prompt "--More-- (50%)" is visible at the bottom, indicating that the output is paginated.

- less



A terminal window titled "liuke — liuke@mgt:~/test — ssh liuke@nebula.ustc.edu.cn — 60x7". The output of the 'less' command is displayed, showing five lines of text: "Hello World!", "This is the 1st line!", "This is the 2nd line!", "This is the 3rd line!", and "This is the 4th line!". The prompt "test.txt" is visible at the bottom, indicating that the output is paginated.

❖ 常用linux命令

➤ 文档查看及管理

- 移动mv

```
liuke — liuke@mgt:~/test1/test2/test3 — ssh liuke@nebula.ustc.edu.cn — 60x6
(base)[liuke@mgt test]$mv test.txt ../test1/test2/test3/
(base)[liuke@mgt test]$ls
(base)[liuke@mgt test]$cd ../test1/test2/test3/
(base)[liuke@mgt test3]$ls
test.txt
(base)[liuke@mgt test3]$
```

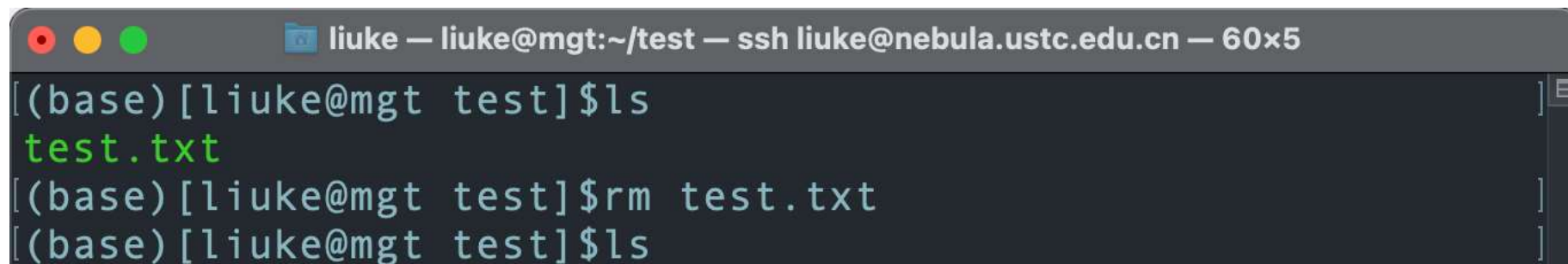
- 复制 cp

```
liuke — liuke@mgt:~/test1/test2/test3 — ssh liuke@nebula.ustc.edu.cn — 60x7
(base)[liuke@mgt test]$cp test.txt ../test1/test2/test3/
(base)[liuke@mgt test]$ls
test.txt
(base)[liuke@mgt test]$cd ../test1/test2/test3/
(base)[liuke@mgt test3]$ls
test.txt
(base)[liuke@mgt test3]$
```

❖ 常用linux命令

➤ 文档查看及管理

- 删除 rm



A terminal window titled "liuke — liuke@mgt:~/test — ssh liuke@nebula.ustc.edu.cn — 60x5". The terminal shows the following commands and output:

```
(base)[liuke@mgt test]$ls
test.txt
(base)[liuke@mgt test]$rm test.txt
(base)[liuke@mgt test]$ls
```

重要的事情说三遍：

慎用rm! ! ! ! !

慎用rm! ! ! ! !

慎用rm! ! ! ! !

❖ 常用linux命令

➤ 文档查看及管理

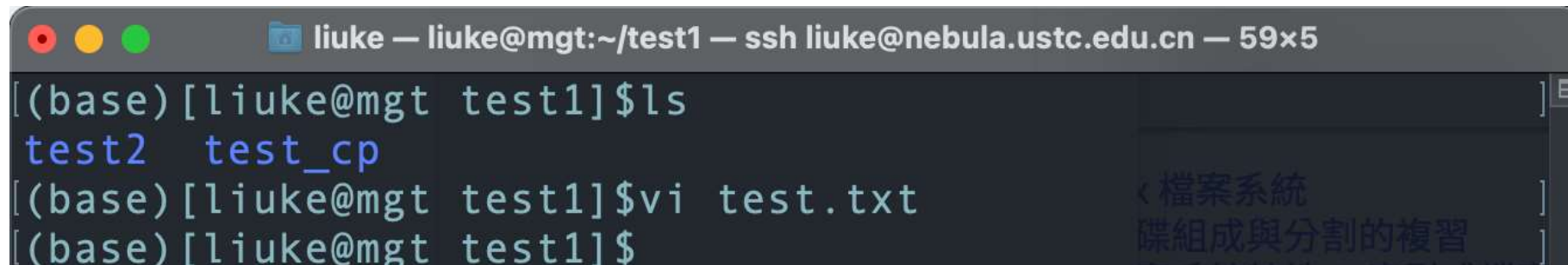
mv可以直接处理文件夹
cp和rm需要使用-r

```
liuke — liuke@mgt:~ — ssh liuke@nebula.ustc.edu.cn — 60x6
(base)[liuke@mgt ~]$rm test
rm: cannot remove 'test': Is a directory
(base)[liuke@mgt ~]$cp test test1/
cp: omitting directory 'test'
(base)[liuke@mgt ~]$mv test test1/
```

```
liuke — liuke@mgt:~/test1 — ssh liuke@nebula.ustc.edu.cn — 60x7
(base)[liuke@mgt test1]$ls
test  test2
(base)[liuke@mgt test1]$cp -r test test_cp
(base)[liuke@mgt test1]$rm -r test
(base)[liuke@mgt test1]$ls
test2  test_cp
```

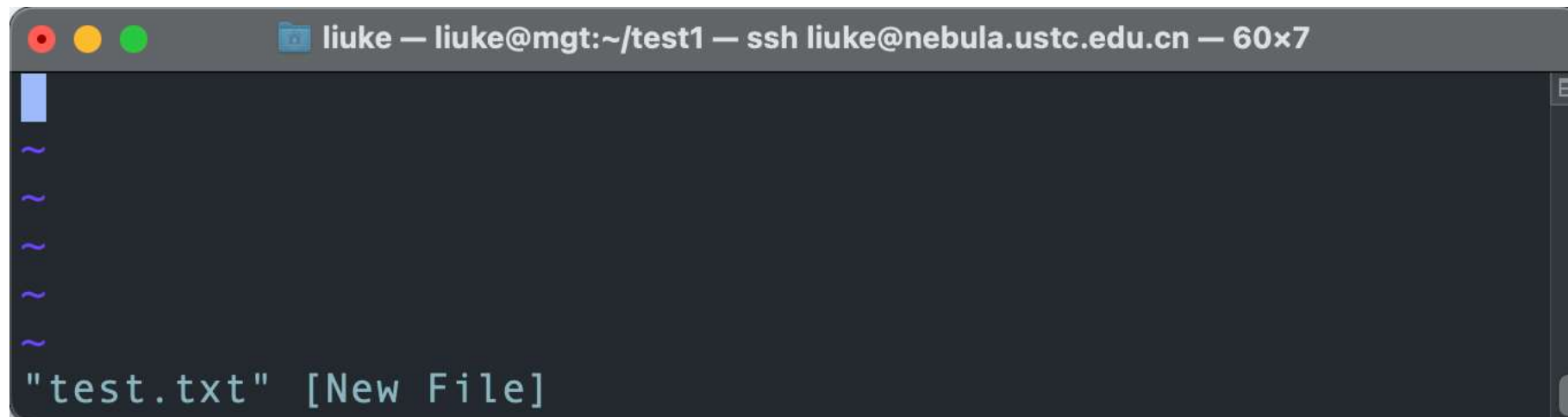

❖ 常用linux命令

➤ vim编辑器



```
liuke — liuke@mgt:~/test1 — ssh liuke@nebula.ustc.edu.cn — 59x5
(base)[liuke@mgt test1]$ls
test2  test_cp
(base)[liuke@mgt test1]$vi test.txt
(base)[liuke@mgt test1]$
```

The terminal window shows the user 'liuke' at the 'mgt' host in the directory '~/test1'. They run 'ls' and see 'test2' and 'test_cp'. Then they run 'vi test.txt' and the prompt changes to '(base)[liuke@mgt test1]\$'.

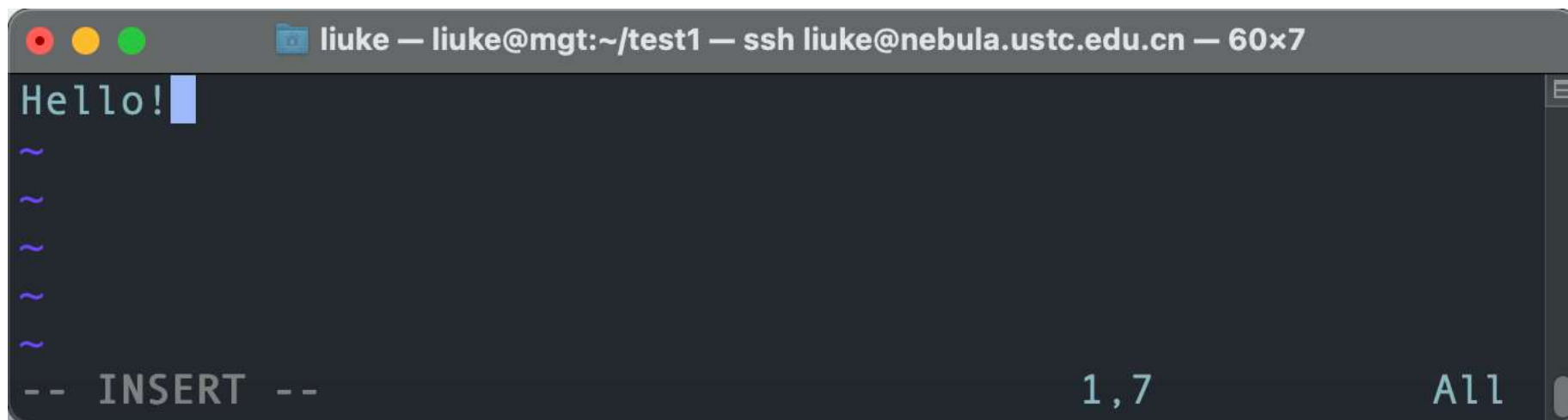


```
liuke — liuke@mgt:~/test1 — ssh liuke@nebula.ustc.edu.cn — 60x7
~
~
~
~
~
"test.txt" [New File]
```

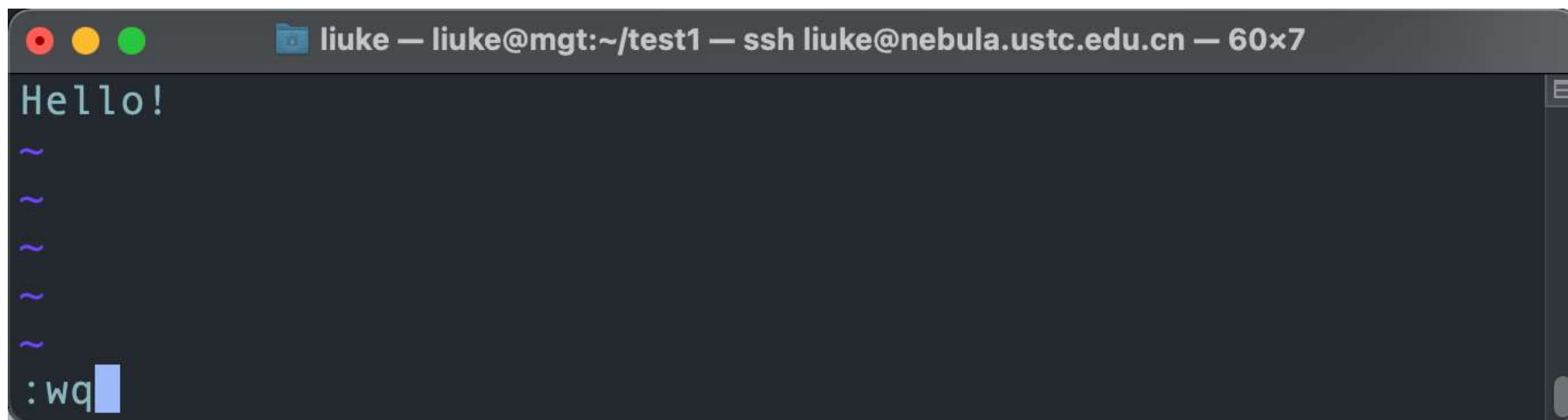
The terminal window shows the vim editor interface. The title bar indicates the window size is 60x7. The editor shows a new file 'test.txt' with a blue cursor at the top left. The status line at the bottom says '"test.txt" [New File]'. There are five tilde '~' characters in the first column, likely representing the first five lines of the file.

❖ 常用linux命令

➤ vim编辑器



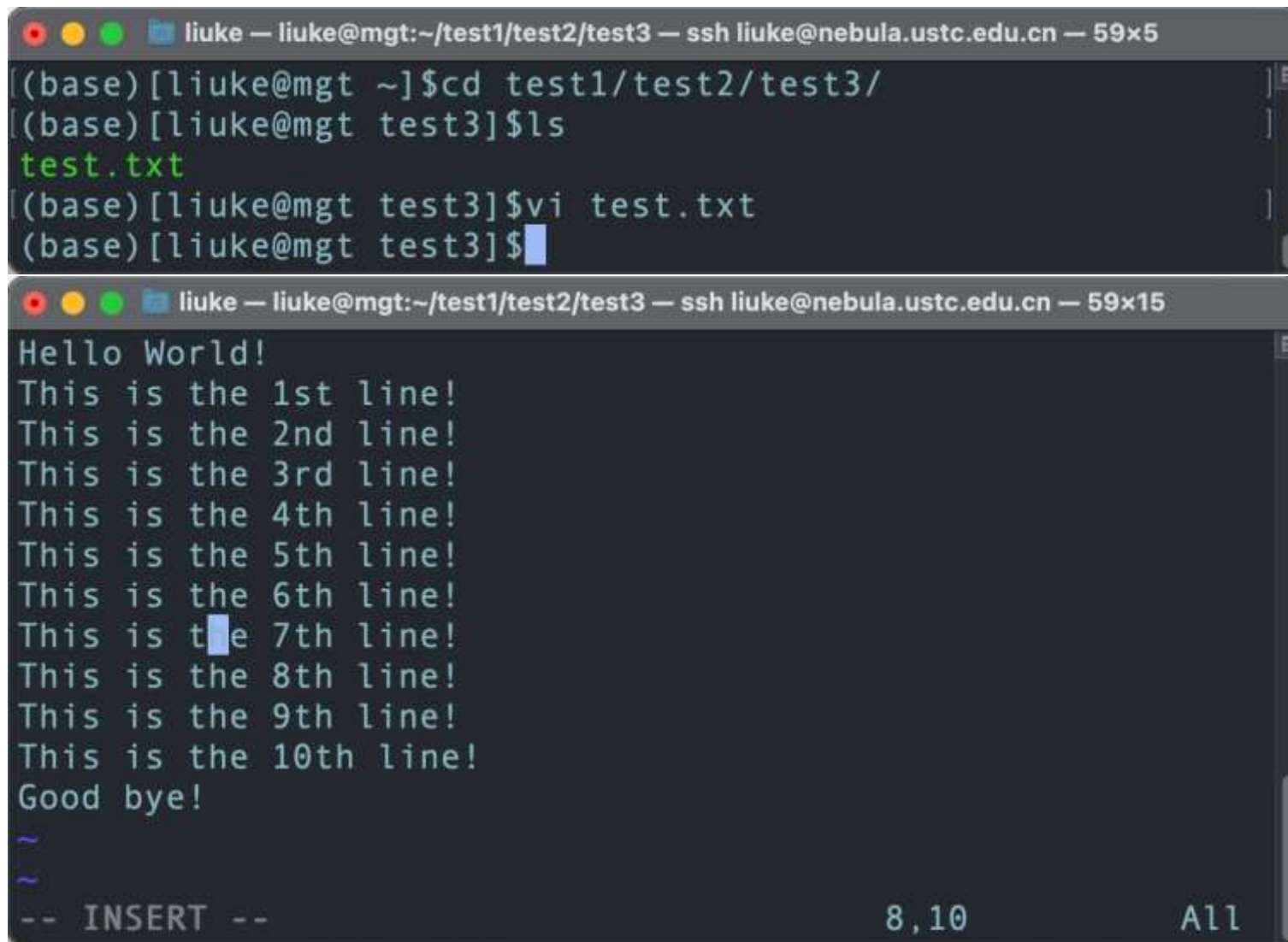
```
liuke — liuke@mgt:~/test1 — ssh liuke@nebula.ustc.edu.cn — 60x7
Hello!
~
~
~
~
~
-- INSERT -- 1,7 All
```



```
liuke — liuke@mgt:~/test1 — ssh liuke@nebula.ustc.edu.cn — 60x7
Hello!
~
~
~
~
~
:wq
```

❖ 常用linux命令

➤ vim编辑器



```
liuke — liuke@mgt:~/test1/test2/test3 — ssh liuke@nebula.ustc.edu.cn — 59x5
(base)[liuke@mgt ~]$cd test1/test2/test3/
(base)[liuke@mgt test3]$ls
test.txt
(base)[liuke@mgt test3]$vi test.txt
(base)[liuke@mgt test3]$

liuke — liuke@mgt:~/test1/test2/test3 — ssh liuke@nebula.ustc.edu.cn — 59x15
Hello World!
This is the 1st line!
This is the 2nd line!
This is the 3rd line!
This is the 4th line!
This is the 5th line!
This is the 6th line!
This is the 7th line!
This is the 8th line!
This is the 9th line!
This is the 10th line!
Good bye!
~
~
-- INSERT --                               8,10                               All
```

➤ PBS脚本的基本介绍

```
liuke — liuke@mgt:~/used_pbs/01_shell/00_jupyter — ssh liuke@nebula.ustc.edu.cn — 63x15
#!/bin/sh
#An example for serial job.
#DO NOT RUN THIS SCRIPT DIRECTLY,
#PLEASE RUN THIS SCRIPT WITH qsub: qsub serial_job.pbs
#
#PBS -N test                任务名称
#PBS -o test.log            任务输出文件
#PBS -e test.err            任务错误文件
#PBS -q batch                节点类型: fat, gpu, batch
#PBS -l walltime=360:00:00   任务时间: 时: 分: 秒
#PBS -l nodes=comput10:ppn=20 节点及线程

echo Start time: `date`
cd /home/qukun/liuke
```

1,1 Top

❖ PBS脚本

➤ PBS脚本的基本介绍

- **pestat查看节点信息**

```
liuke -- liuke@mgt:~/test1 -- ssh liuke@nebula.ustc.edu.cn -- 126x11
(base)[liuke@mgt test1]$pestat
node      state  load    phymem  ncpus   allmem resi   usrs    tasks  jobidlist
comput10  excl  8.42*   257679  28      321679 61617  37/7    28      12103675 lshouzhe2 12103424 liuke 12104753
richardo 12105087 migene 12105155 yanpingsun5
comput11  free  24.08*  257679  28      321679 29214  7/5     8       12105072 cuiting 12105147 yanpingsun5 12105
178 ws174012
comput12  free  0.02*   257679  28      321679 27442  15/4    26      12104899 lly090609 12105090 pyh234 12105101
hai_li
comput13  busy* 41.61*  257679  28      321679 56166  11/5    23      12104823 migene 12104952 yexin0616 12104953
yexin0616 12105076 ws174012
comput14  free  10.27*  257679  28      321679 78662  24/8    25      12104264 z10517 12104636 zfh 12104989 gchua
```

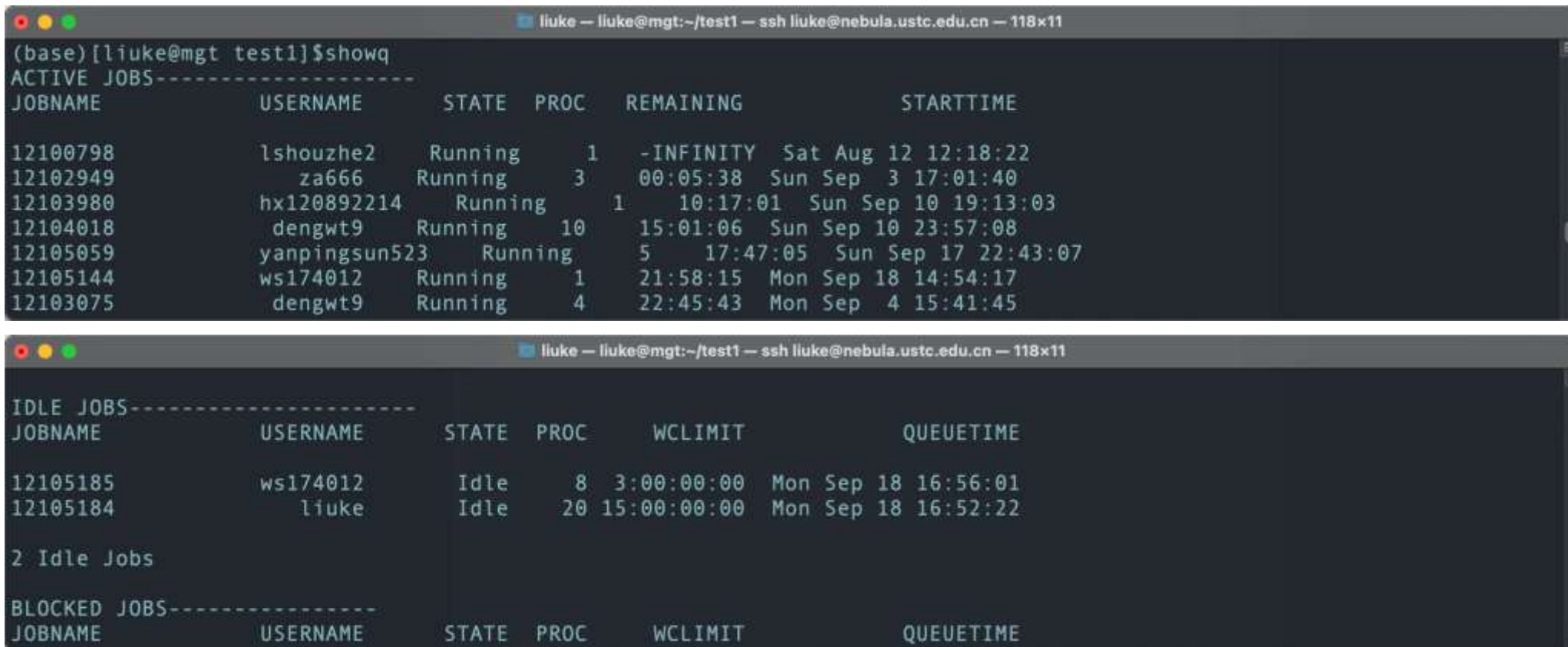
- **qstat -u <username>查看自己的任务**

```
liuke -- liuke@mgt:~/test1 -- ssh liuke@nebula.ustc.edu.cn -- 126x11
(base)[liuke@mgt test1]$qsub test.pbs
12105184.mgt
(base)[liuke@mgt test1]$qstat -u liuke

mgt:
Job_ID      Username  Queue    Jobname     SessID  NDS   TSK    Req'd  Req'd   S   Elap
          -----
12103424.mgt liuke     batch    lab1        1237    1    20     --    360:00:00 R 211:39:14
12105184.mgt liuke     batch    test        --      1    20     --    360:00:00 Q  --
(base)[liuke@mgt test1]$
```

➤ PBS脚本的基本介绍

- showq查看排队信息



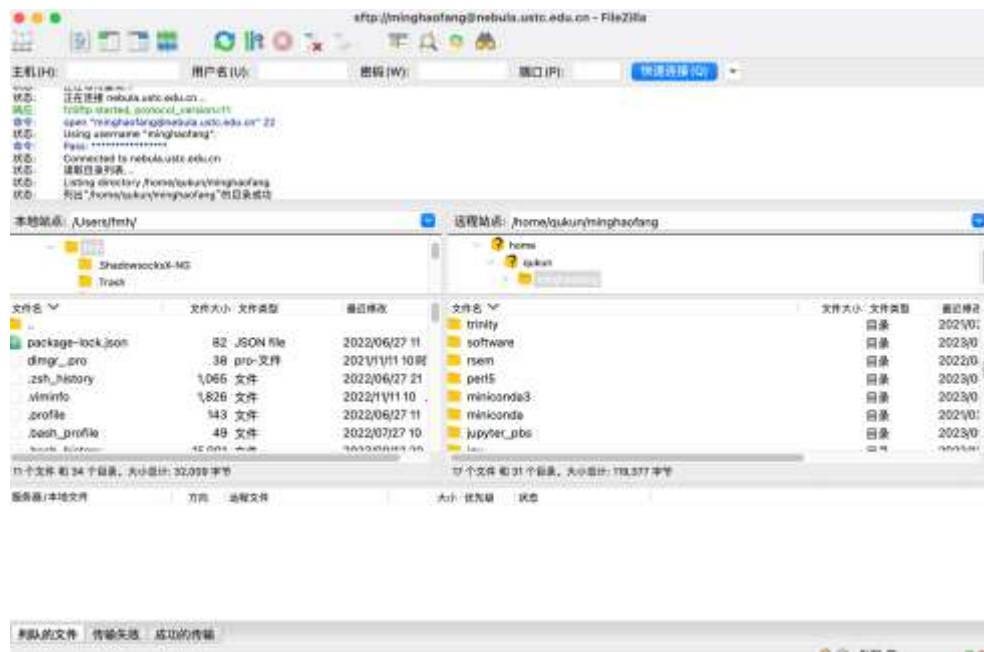
```
(base)[liuke@mgt test1]$showq
ACTIVE JOBS-----
JOBNAME            USERNAME      STATE  PROC   REMAINING      STARTTIME
12100798            lshouzhe2    Running  1    -INFINITY  Sat Aug 12 12:18:22
12102949              za666      Running  3    00:05:38  Sun Sep  3 17:01:40
12103980            hx120892214  Running  1    10:17:01  Sun Sep 10 19:13:03
12104018              dengwt9    Running 10    15:01:06  Sun Sep 10 23:57:08
12105059            yanpingsun523 Running  5    17:47:05  Sun Sep 17 22:43:07
12105144              ws174012   Running  1    21:58:15  Mon Sep 18 14:54:17
12103075              dengwt9    Running  4    22:45:43  Mon Sep  4 15:41:45

IDLE JOBS-----
JOBNAME            USERNAME      STATE  PROC   WCLIMIT      QUEUE TIME
12105185              ws174012   Idle    8    3:00:00:00  Mon Sep 18 16:56:01
12105184              liuke      Idle   20   15:00:00:00  Mon Sep 18 16:52:22

2 Idle Jobs

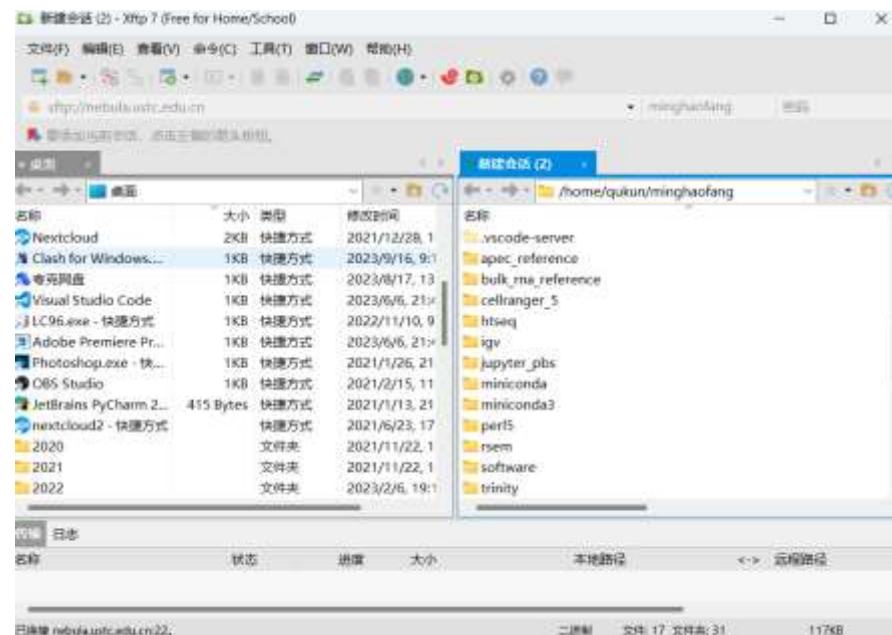
BLOCKED JOBS-----
JOBNAME            USERNAME      STATE  PROC   WCLIMIT      QUEUE TIME
```


❖ Linux 系统文件传输



MAC

1. `Top -u $user` 命令查看进程和内存占用
2. **禁止**在登录节点直接运行程序(需通过pbs运行)
3. 登录节点可以运行基本装包和访问文件等命令



Windows

目录

❖ Linux 基本操作

❖ Conda&Jupyter 安装(linux)

❖ 简要环境安装

❖ Conda&Jupyter 安装(linux)



环境配置

JupyterLab 配置远程python、R环境(与Jupyter兼容)

python pip修改安装镜像源

linux 环境下配置ftp服务器

linux 环境下安装和配置mysql数据库以及远程登录

linux 环境下配置python虚拟环境

windows环境下python 虚拟环境的创建和使用 (virtualenvwrapper)

Linux环境下 安装python3



<https://blog.csdn.net/jeffery0207/article/details/103440598>

❖ Conda&Jupyter 安装(linux)



Anaconda 个人版

Anaconda 个人版是一个免费、易于安装的包管理器、环境管理器和 Python 发行版，包含 1,500 多个开源包，并提供 [免费社区支持](#)。Anaconda 与平台无关，因此无论您在 Windows、macOS 还是 Linux 上都可以使用它。

查看[Anaconda 个人版文档](#)。



一、什么是Jupyter Notebook?

1. 简介

Jupyter Notebook是基于网页的用于交互计算的应用程序。其可被应用于全过程计算：开发、文档编写、运行代码和展示结果。——[Jupyter Notebook官方介绍](#)

简而言之，Jupyter Notebook是以网页的形式打开，可以在网页页面中**直接编写代码和运行代码**，代码的**运行结果**也会直接在代码块下显示的程序。如在编程过程中需要编写说明文档，可在同一个页面中直接编写，便于作及时的说明和解释。

❖ Conda&Jupyter 安装(linux)

安装Anaconda

删繁就简就是我的个人原则，所以安装Anaconda, 我选择Miniconda, 它是Anaconda的最小化版本，只包含conda、python以及一些必须的包。进入Miniconda选择适合你的版本即可下载安装：

```
1 # 以下所有安装演示均在terminal中进行
2 $ wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
3 $ /bin/bash Miniconda3-latest-Linux-x86_64.sh # 启动安装，接下来按照提示完成即可
4
5 Do you wish the installer to initialize Miniconda3
6 by running conda init? [yes|no]
7 [no] >>> no # 建议不init，其他yes就好
```

接下来在 `.bashrc` 中添加环境变量，并加载环境变量：

```
1 echo 'export PATH="/miniconda3/bin:$PATH"' >> ~/.bashrc
2 source ~/.bashrc
3
4 conda info --envs # 测试以下conda
```

❖ Conda&Jupyter 安装(linux)

安装配置jupyter lab

它的安装很简单啦，[官网](#)列出了很多安装方式，但是以我的经验和踩坑经历，大多数的时候，尽量别在base环境下安装包，更加明智的选择是按照你的工作类型、项目创建不同的虚拟环境，以免一个环境被破坏，全盘奔溃。所以这里我们先创建一个虚拟环境，再进行各种包的安装以及环境配置。

```
1 | conda create --name common python=3.6.8
2 | source activate common
3 | pip install jupyterlab # 推荐在conda中使用pip安装各种包
```

conda list

查看对应环境所安装的包

conda env list

查看已创建虚拟环境

conda activate \$env_name

激活特定虚拟环境

❖ Conda&Jupyter 安装(linux)

下面先生成密钥，然后再在 `~/.jupyter/jupyter_notebook_config.py` 中修改配置文件即可：

```
1 jupyter lab --generate-config # 生成默认配置文件 ~/.jupyter/jupyter_notebook_c
2 ipython
3 # -----
4 In [1]: from notebook.auth import passwd
5 In [2]: passwd()
6 Enter password:
7 Verify password:
8 Out[2]: 'sha1:c2d56265c773:e06ef2b97c94bbb014f65bd2975d43c7da12520e'
```

update logs (2021.01.07) ==> jupyterlab 3.0近期发布， 配置文件为: `jupyter_lab_config.py`

修改配置文件中的以下项：

```
1 c.NotebookApp.ip='*'
2 # c.NotebookApp.allow_remote_access = True
3 c.NotebookApp.password = u'sha1:c2d56265c773:e06ef2b97c94bbb014f65bd2975d43c
4 c.NotebookApp.open_browser = False
5 c.NotebookApp.port =8989 #可自行指定一个端口， 访问时使用该端口
```

update logs (2021.01.07) ==> jupyterlab 3.0近期发布， 配置文件中的信息有所改变， `c.NotebookApp.*` 变成了 `c.ServerApp.*`，另外，如安装jupyter插件可以设置 `c.ExtensionApp.open_browser = False`

❖ Conda&Jupyter 安装(linux)

远程连接jupyter lab

首先我们在远程服务器开启jupyter lab, 下面代码中, `nohup` 可以使代码进入后台运行, 关于更多linux命令后台运行, 请参考[Linux 后台运行程序方法总结](#)

```
1 | source activate common
2 | nohup jupyter lab &
```

然后在本地使用ssh和服务器建立连接, 其中 `-N` : SSH没有命令要被远程执行; `-f` : SSH在后台执行; `-L` : 指定port forwarding的配置:

服务器

```
1 | ssh -p port -N -f -L localhost:8989:localhost:8989 username@ip
```

最后在本地图浏览器中访问: `http://localhost:8989/lab` 就可以啦

```
ssh -p 22 -N -f -L localhost:8989:wks1:8989 minghaofang@nebula.ustc.edu.cn
```

❖ Conda&Jupyter 安装(linux)

R语言

默认jupyter lab使用的应该是系统的R，我们一般使用自己虚拟环境中创建的R环境，这样不需要管理员权限就可以方便地管理自己的packages:

```
1 conda install r==3.6
2 which R
```

在R语言中安装相关package:

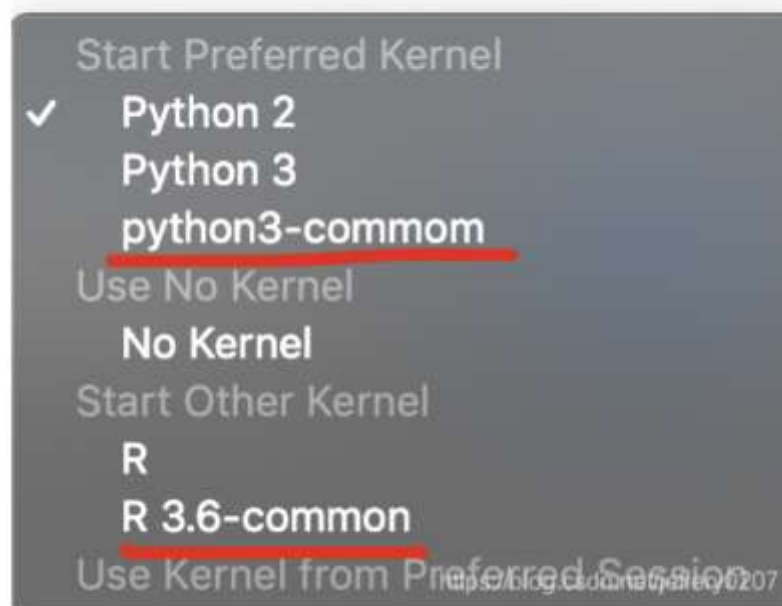
```
1 > install.packages("devtools")
2 > devtools::install_github('IRkernel/IRkernel')
3 > IRkernel::installspec(name = 'ir36-common', displayname = 'R 3.6-common')
4 [InstallKernelSpec] Installed kernelspec ir36-common in ~/.local/share/jupyter
```

Python

类似R语言，首先需要安装一个ipykernel，然后再安装一个kernel到
~/.local/share/jupyter/kernels 目录下，原理一模一样:

```
1 pip install ipykernel
2 python -m ipykernel install --user --name python3-commom
```

Select Kernel



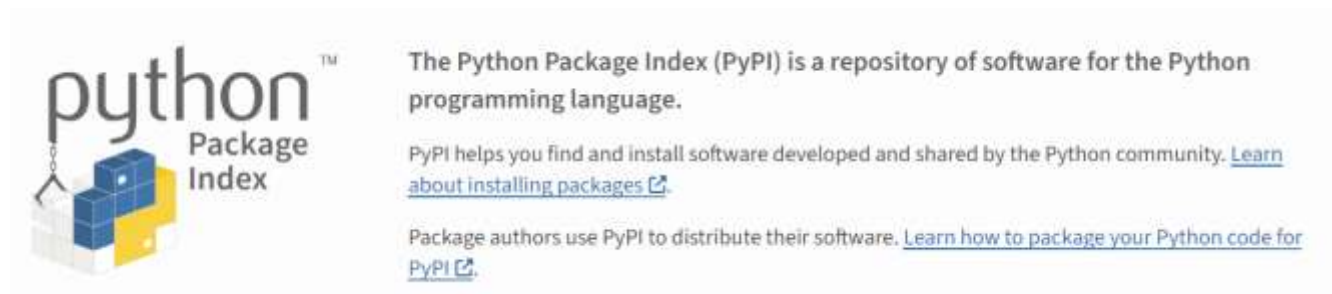
目录

❖ Linux 基本操作

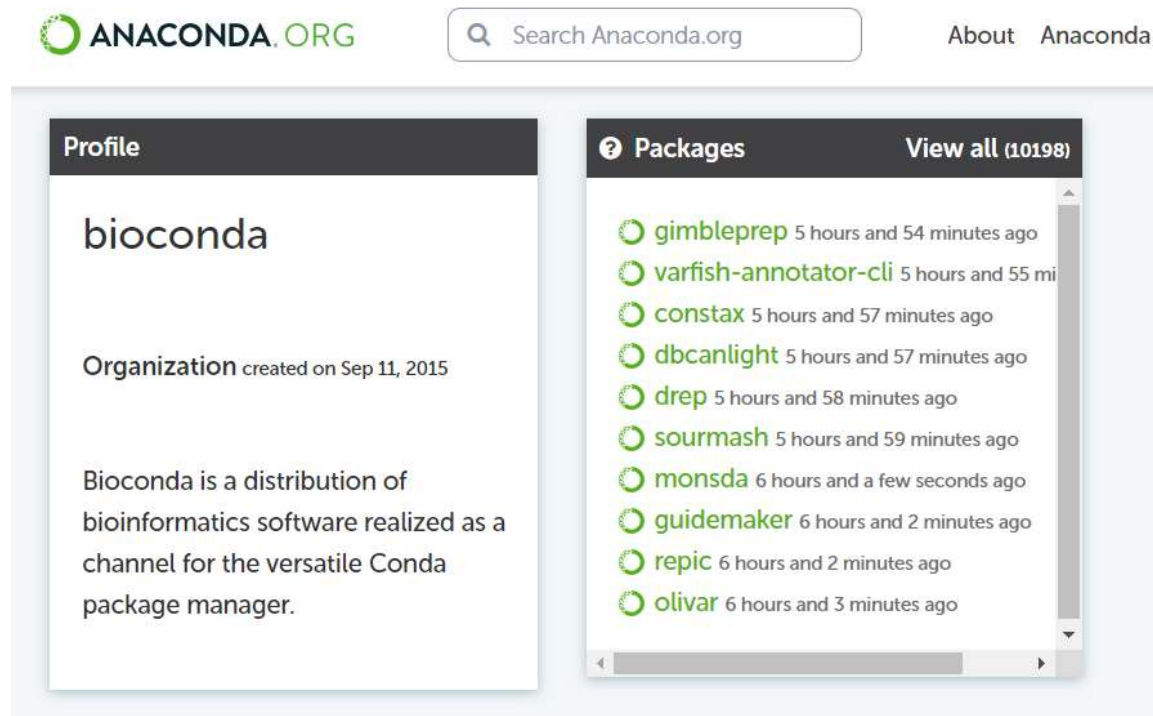
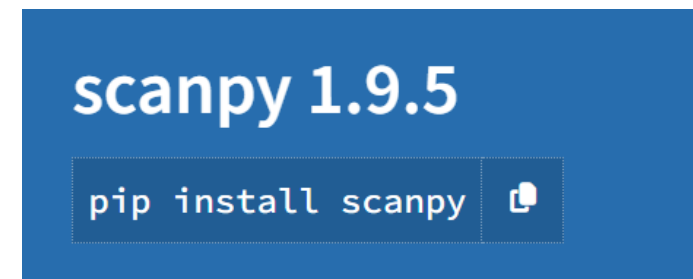
❖ Conda&Jupyter 安装(可视化分段编程)

❖ 简要环境安装

❖ 简要环境安装



<https://pypi.org/>



<https://anaconda.org/bioconda/>

⚡ Favorites	⚡ Downloads	⚡ Artifact (owner / artifact)
2	98132	bioconda / scanpy 1.7.2 Single-Cell Analysis in Python. Scales to >1M cells.
0	91991	conda-forge / scanpy 1.9.5 Single-Cell Analysis in Python. Scales to >1M cells.

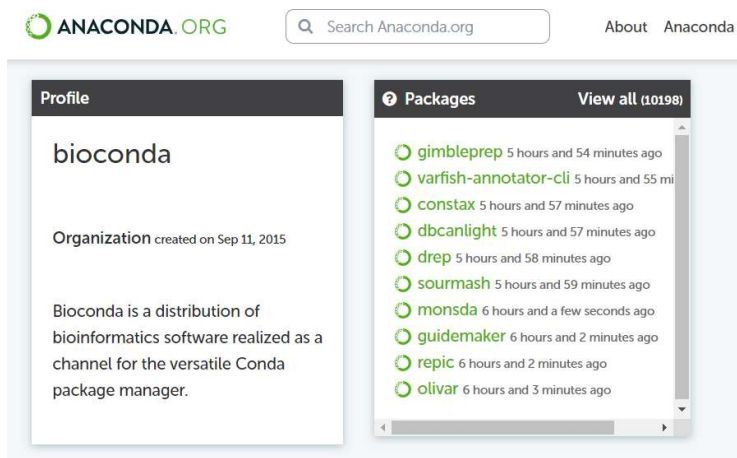
conda install ?

To install this package run one of the following:

```
conda install -c bioconda scanpy
```

```
conda install -c "bioconda/label/cf201901" scanpy
```

❖ 简要环境安装



<https://bioconductor.org/>

To install this package run one of the following:

```
conda install -c bioconda bioconductor-deseq2
```

```
conda install -c "bioconda/label/broken" bioconductor-deseq2
```

```
conda install -c "bioconda/label/cf201901" bioconductor-deseq2
```

```
conda install -c "bioconda/label/gcc7" bioconductor-deseq2
```

Installation

To install this package, start R (version "4.3") and enter:

```
if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install("DESeq2")
```

For older versions of R, please refer to the appropriate [Bioconductor release](#).

Documentation

To view documentation for the version of this package installed in your system, start R and enter:


```
browseVignettes("DESeq2")
```

[HTML](#) [R Script](#) Analyzing RNA-seq data with DESeq2

[PDF](#) Reference Manual


[Text](#) NEWS

❖ 简要环境安装

 **SPACEL** Public

main 7 branches 0 tags

Go to file Add file Code

 Bink98 fix simi_l	60cf6c1 on Jul 27	56 commits
SPACEL	fix simi_l	2 months ago
docs	fix simi_l	2 months ago
.gitignore	fix flat_model in Scube	2 months ago
.readthedocs.yml	add readthedocs	8 months ago
README.md	fix simi_l	2 months ago
environment.yml	remove rpy2 from pypi	2 months ago
readthedocs_environment.yml	remove rpy2 from pypi	2 months ago
requirements.txt	update requirements for windows	2 months ago
setup.py	update requirements for windows	2 months ago

Requirements

Note: The current version of SPACEL only supports Linux and MacOS, not Windows platform.

To install SPACEL, you need to install [PyTorch](#) with GPU support first. If you don't need GPU acceleration, you can just skip the installation for `cuda` and `cuda-toolkit`.

- Create conda environment for SPACEL:

```
conda env create -f environment.yml
```

or

```
conda create -n SPACEL -c conda-forge -c default cuda-toolkit=10.2 python=3.8 rpy2 m-base m-figdistplus
```

You must choose correct `PyTorch`, `cuda` and `cuda-toolkit` version dependent on your graphic driver version.

Note: If you want to run 3D expression GPR model in Scube, you need to install the [Open3D](#) python library first.

Installation

- Install SPACEL:

```
pip install SPACEL
```

- Test if `PyTorch` for GPU available:

```
python  
>>> import torch  
>>> torch.cuda.is_available()
```

If these command line have not return `True`, please check your gpu driver version and `cuda-toolkit` version. For more detail, look at [CUDA Toolkit Major Component Versions](#).

<https://github.com/QuKunLab/SPACEL>

❖ 简要环境安装

