

# 数据挖掘课程实验

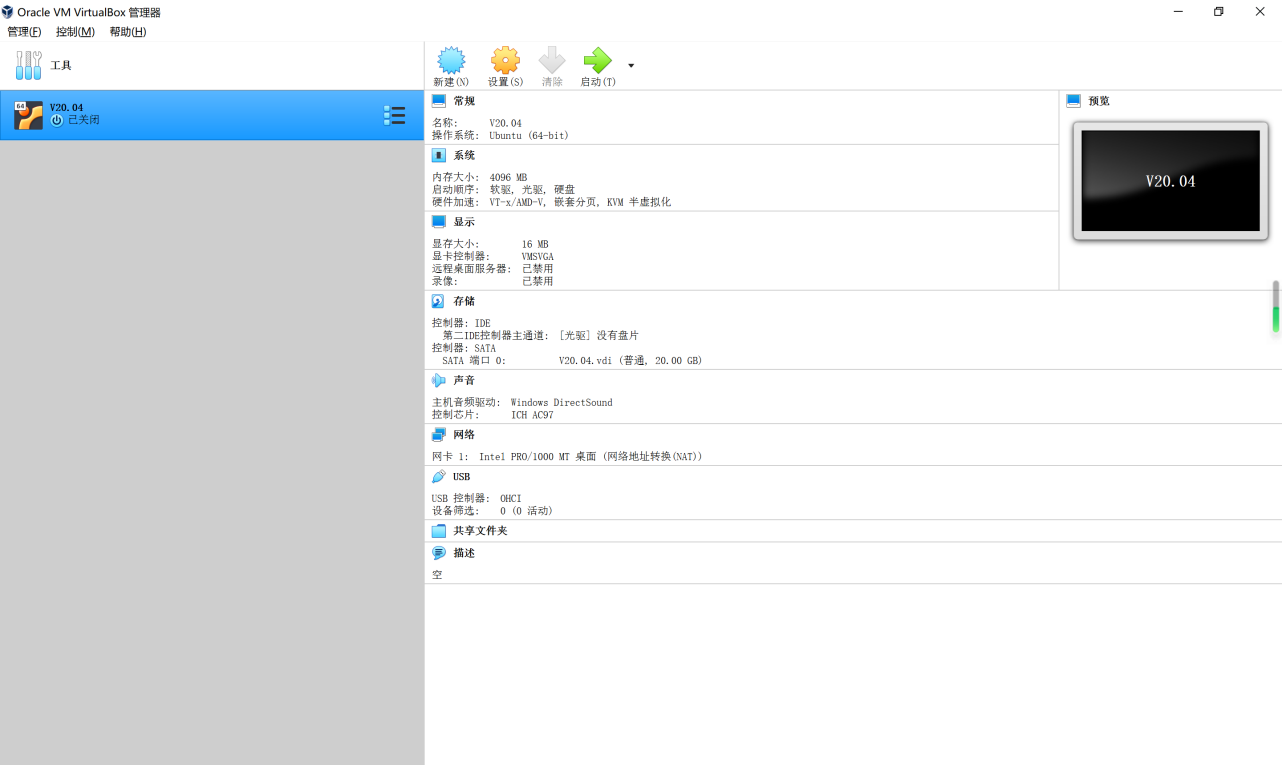
## 实验1 实验平台及环境安装

### 实验手册

计科210X 甘晴void 202108010XXX

### 1.安装虚拟机和Linux平台，熟悉Ubuntu环境。

（1）虚拟机使用Oracle VM VirtualBox。之前计算机系统和操作系统课程也使用的该平台。



、

（2）创建Linux操作系统64=位。使用xubuntu20.04版本。


## ← 新建虚拟电脑

## 虚拟电脑名称和系统类型

请选择新虚拟电脑的描述名称及要安装的操作系统类型。此名称将用于标识此虚拟电脑。

名称:

文件夹:

类型(T): Linux 

版本(V): Ubuntu (64-bit)

专家模式(E)

下一步(N)

取消

(3) 安装完系统之后立加装扩展功能。

## 2.在Linux平台上搭建Python平台，并安装Python环境工具anaconda。

Linux自带python平台，在终端输入

```
python3
```

查看本地python环境，得知是python3.8环境。

首先了解anaconda与miniconda的区别。

Anaconda是一个包含了conda、Python和超过150个科学包及其依赖项的科学Python发行版。它具有可视化图形用户界面（Anaconda Navigator）并且为了方便新手使用，预先包含了大量的库，如NumPy, Pandas, Scipy, Matplotlib等。

相较之下，Miniconda更加轻量级。它只包含了Python和Conda，但并没有预装其他的库。Miniconda用户需要手动安装他们需要的包，这使得Miniconda的环境更为简洁，可以根据实际需求来安装必要的包，避免不必要的存储占用。

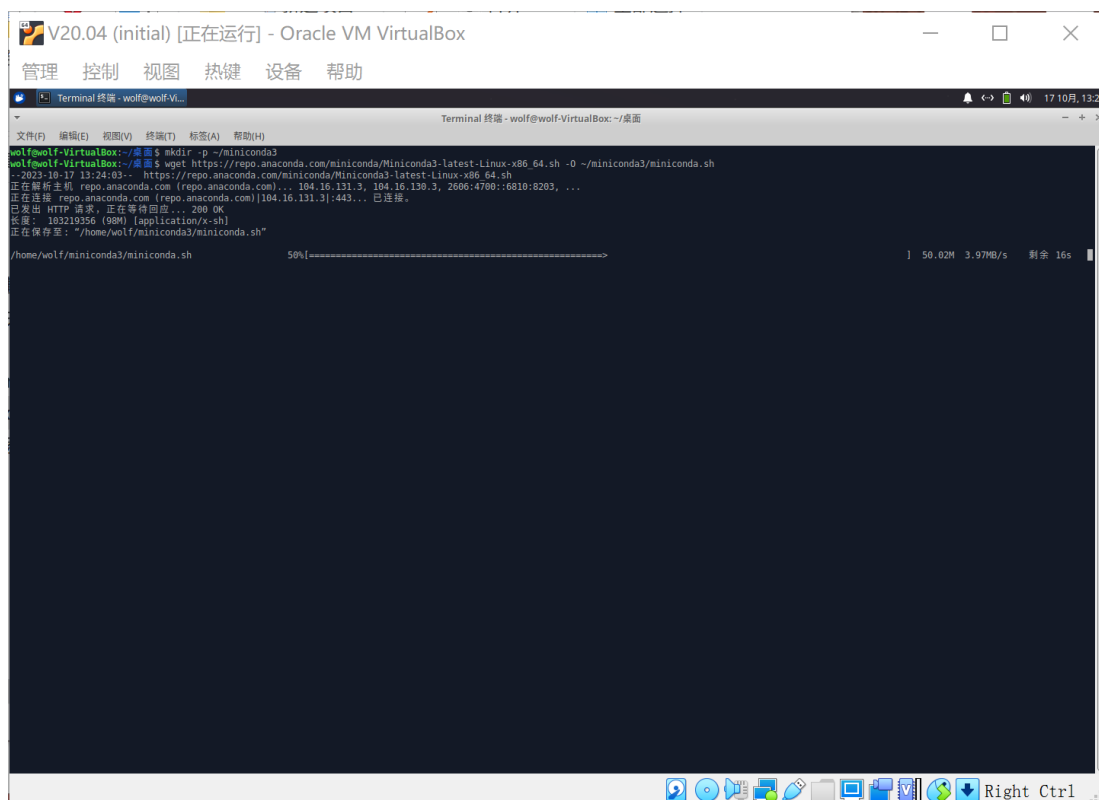
考虑到作为虚拟机的Linux系统实际上有的存储空间并不大，所以打算安装miniconda替代anaconda。

(1) 访问miniconda的官网<https://docs.conda.io/projects/miniconda/en/latest/>获取信息

(2) 在Linux下使用如下指令进行安装并初始化。

```
mkdir -p ~/miniconda3
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh -O ~/miniconda3/miniconda.sh
bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3
rm -rf ~/miniconda3/miniconda.sh
~/miniconda3/bin/conda init bash
~/miniconda3/bin/conda init zsh
```

步骤截图如下



### 3.掌握Anaconda下的Python环境安装，创建名称为emoji的python3.7环境。

安装了最新版本的miniconda之后，再次打开终端，会显示一个默认的(base)在前面，形如以下。

```
(base) wolf@wolf-virtualBox:~/桌面$
```

表示miniconda基本安装时 成功的，目前处于conda的环境下。

此时再次查看python3的版本，发现不知什么时候升级成3.11了。通过查阅资料发现，miniconda会自动为我们配置python环境，不需要手动再下载python版本。

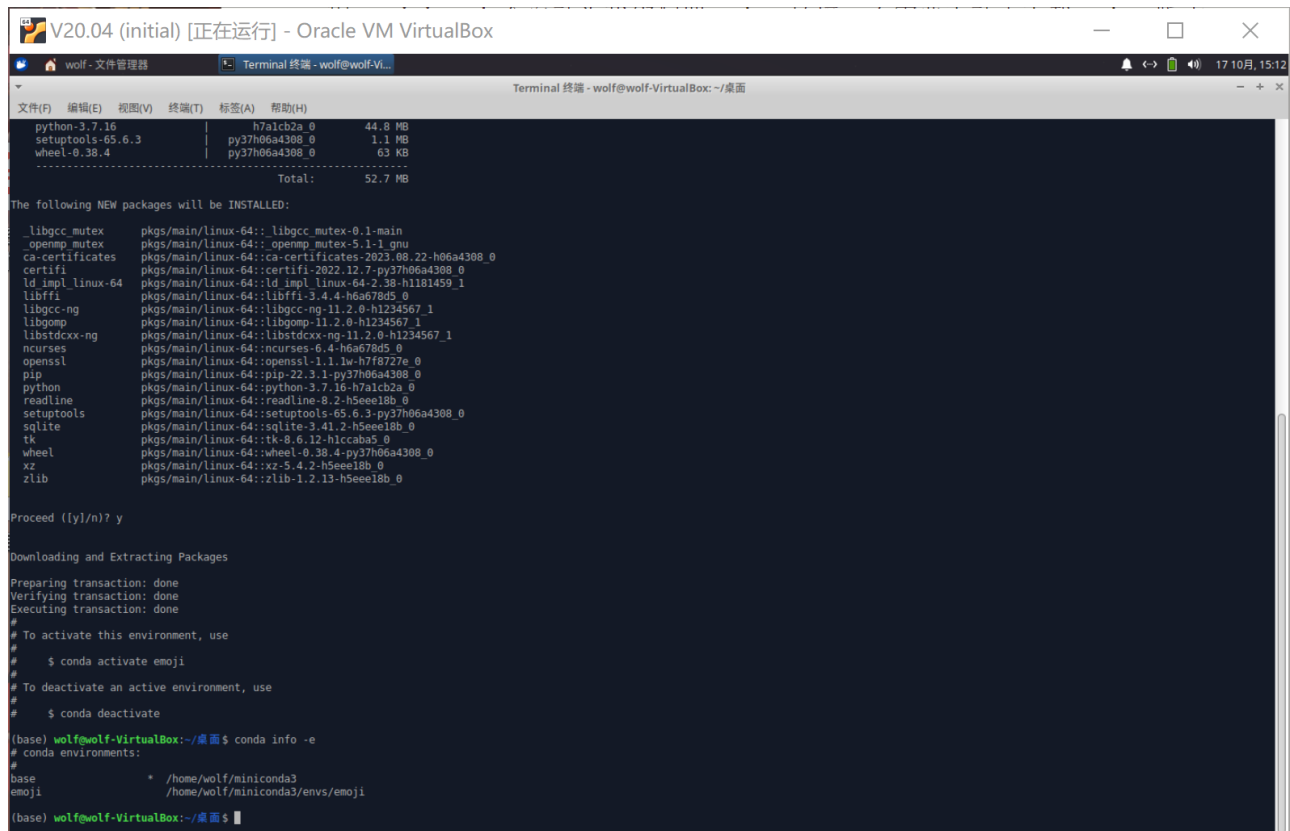
使用以下指令配置环境。

```
conda create -n emoji python=3.7
```

安装完成后使用如下指令查看

```
conda info -e
```

发现出现了原来的基础环境(base)和新建的环境(emoji)



```
V20.04 (initial) [正在运行] - Oracle VM VirtualBox
Terminal 终端 - wolf@wolf-VL...
Terminal 终端 - wolf@wolf-VirtualBox: ~/桌面
文件(F) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H)
python-3.7.16 | h7a1cb2a_0 | 44.8 MB
setuptools-65.6.3 | py37h06a4308_0 | 1.1 MB
wheel-0.38.4 | py37h06a4308_0 | 63 KB
-----
Total: 52.7 MB

The following NEW packages will be INSTALLED:
libgcc_mutex pkgs/main/linux-64::libgcc_mutex-0.1-main
openmp_mutex pkgs/main/linux-64::openmp_mutex-5.1.1-gnu
ca-certificates pkgs/main/linux-64::ca-certificates-2023.08.22-h06a4308_0
certifi pkgs/main/linux-64::certifi-2022.12.7-py37h06a4308_0
ld_impl_linux-64 pkgs/main/linux-64::ld_impl_linux-64-2.38-h1101459_1
libffi pkgs/main/linux-64::libffi-3.4.4-h6a678d5_0
libgcc-ng pkgs/main/linux-64::libgcc-ng-11.2.0-h1234567_1
libgomp pkgs/main/linux-64::libgomp-11.2.0-h1234567_1
libstdc++-ng pkgs/main/linux-64::libstdc++-ng-11.2.0-h1234567_1
ncurses pkgs/main/linux-64::ncurses-6.4-h6a078d5_0
openssl pkgs/main/linux-64::openssl-1.1.1w-h7f8727e_0
pip pkgs/main/linux-64::pip-22.3.1-py37h06a4308_0
python pkgs/main/linux-64::python-3.7.16-h7a1cb2a_0
readline pkgs/main/linux-64::readline-8.2-h5ee18b_0
setuptools pkgs/main/linux-64::setuptools-65.6.3-py37h06a4308_0
sqlite pkgs/main/linux-64::sqlite-3.41.2-h5ee18b_0
tk pkgs/main/linux-64::tk-8.6.12-h1ccab5_0
wheel pkgs/main/linux-64::wheel-0.38.4-py37h06a4308_0
xz pkgs/main/linux-64::xz-5.4.2-h5ee18b_0
zlib pkgs/main/linux-64::zlib-1.2.13-h5ee18b_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

# To activate this environment, use
#
#   $ conda activate emoji
#
# To deactivate an active environment, use
#
#   $ conda deactivate

(base) wolf@wolf-VirtualBox:~/桌面$ conda info -e
# conda environments:
#
base * /home/wolf/miniconda3
emoji /home/wolf/miniconda3/envs/emoji

(base) wolf@wolf-VirtualBox:~/桌面$
```

此时若使用

```
conda activate emoji //进入
conda deactivate //退出
conda config --set auto_activate_base true
conda config --set auto_activate_base false //取消自动进入
```

★这里还应该加一步换源（换用清华源）

```
pip install pip -U  
pip config set global.index-url  
https://pypi.tuna.tsinghua.edu.cn/simple
```

#### 4.熟练安装pycharm和jupyter notebook。

使用Linux访问pycharm官方网址

```
https://www.jetbrains.com/pycharm/download/?section=linux
```

下载Linux下的pycharm，注意不要下载成Professional版本，要下载community版本的。

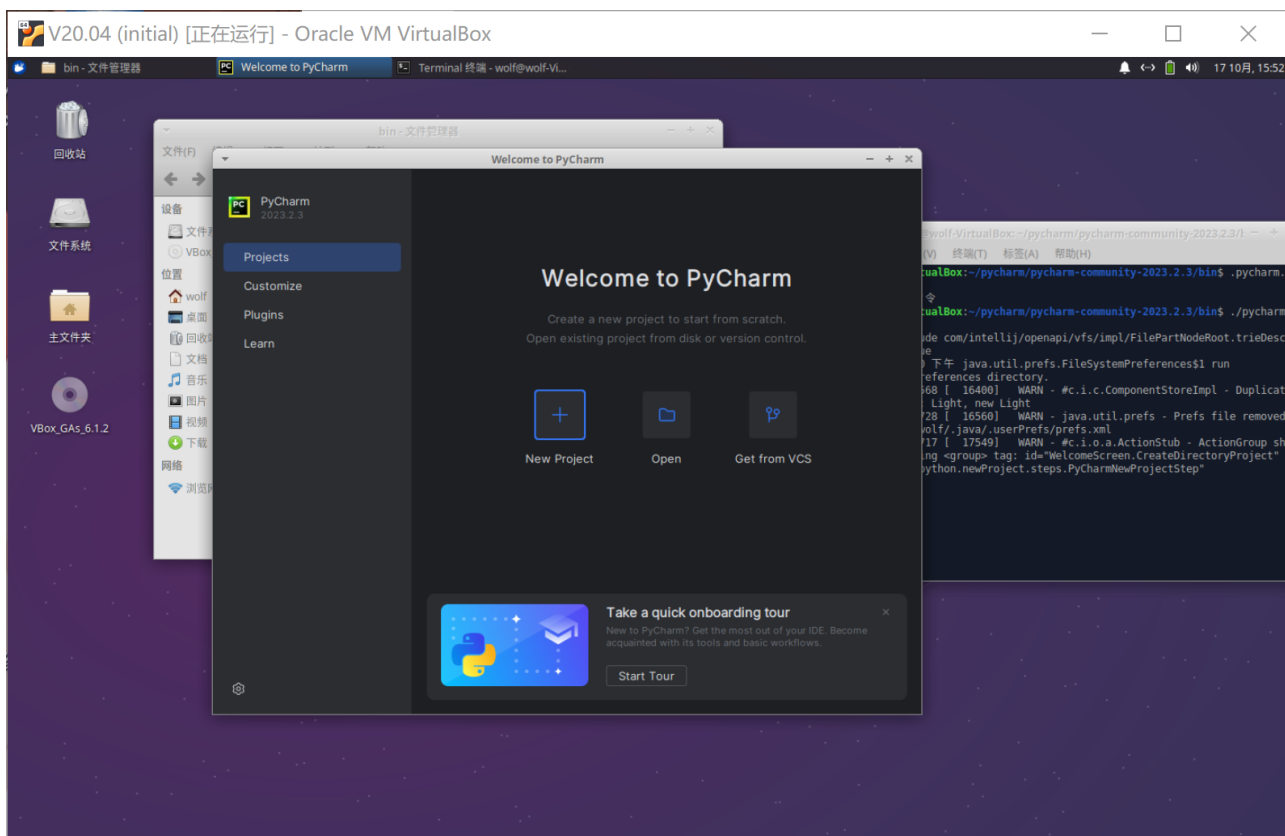
安装完毕后找到位置，解压该压缩包。

```
tar -zxvf pycharm-community-2023.2.3.tar.gz
```

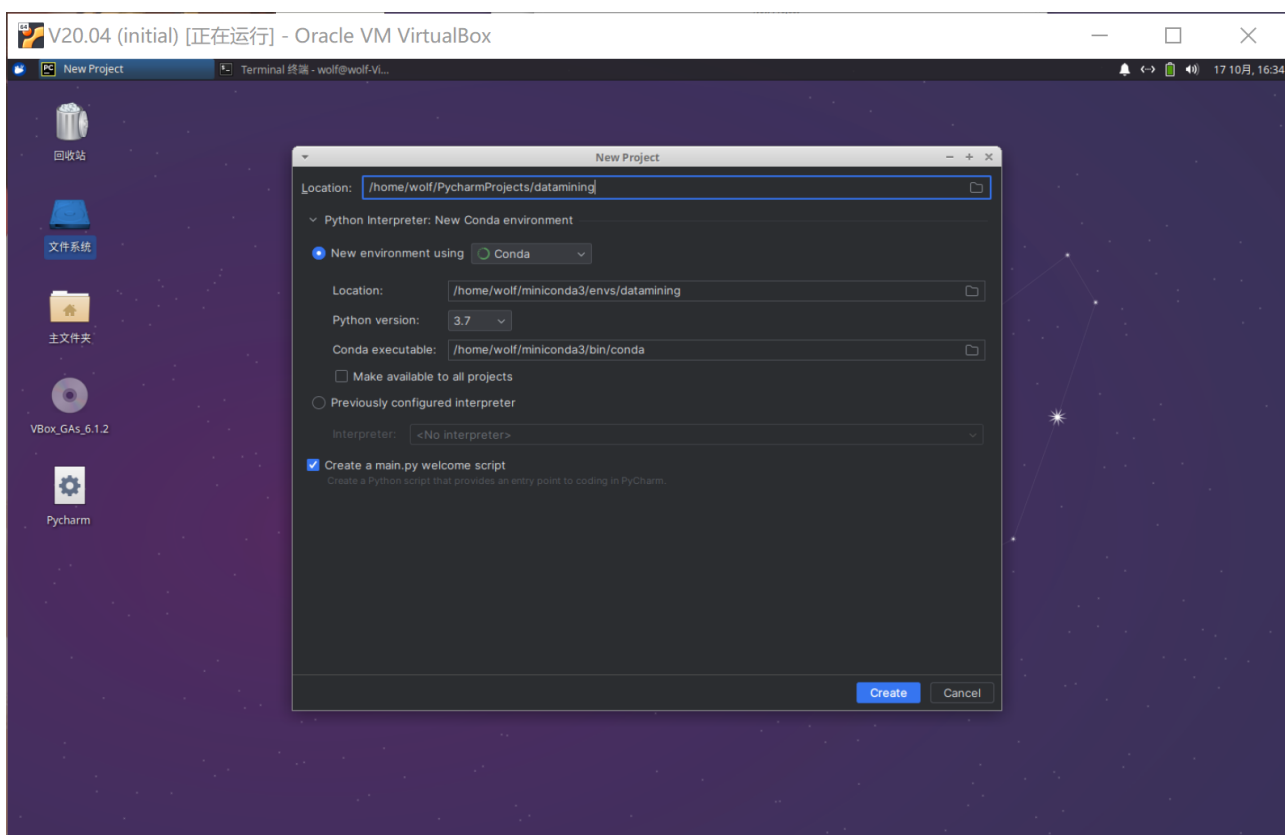
进入bin文件夹

```
./pycharm.sh
```

即可进行安装，安装后就可以打开pycharm，可以看见与windows下是一致的。

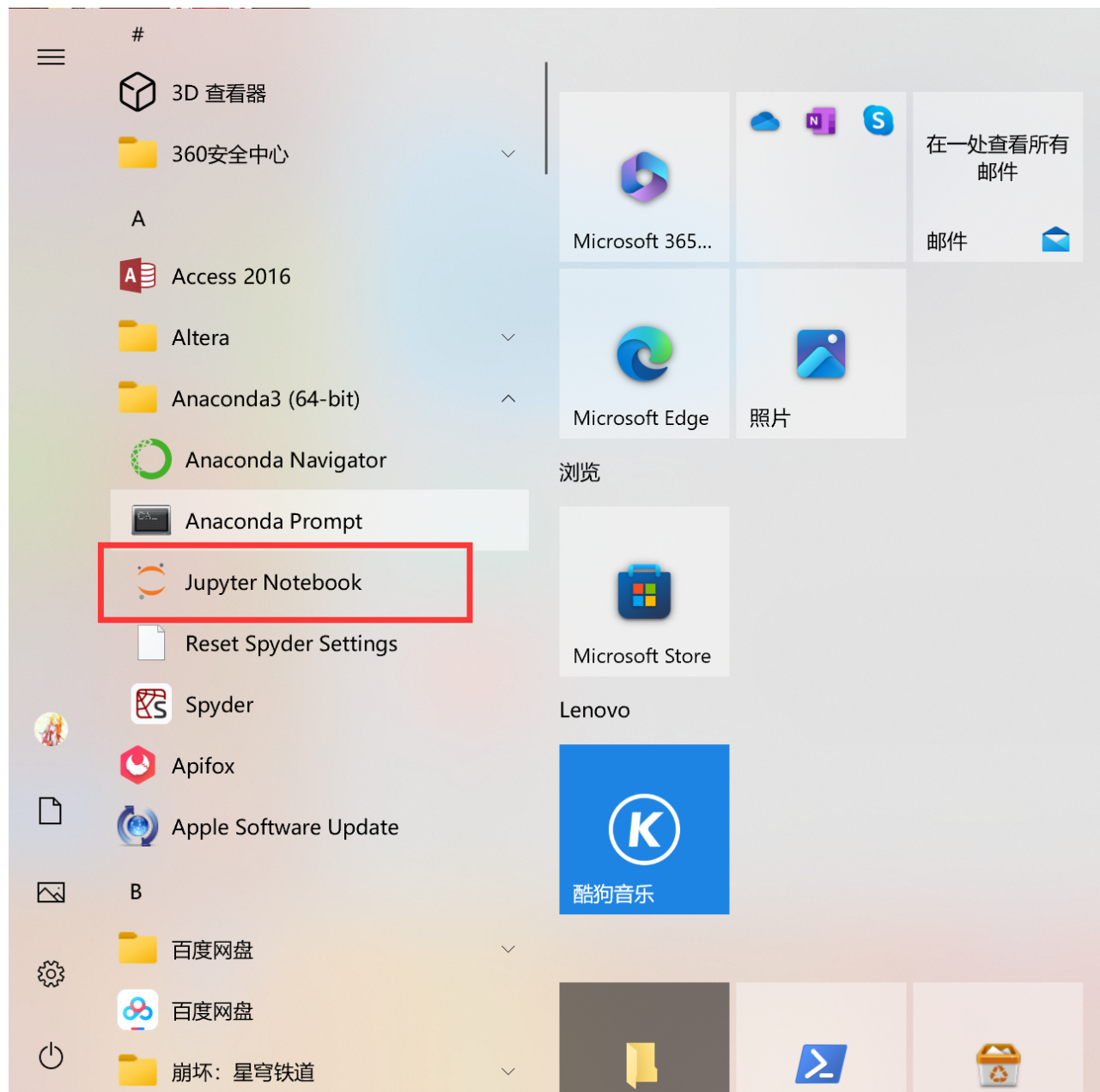


接下来为pycharm配置conda的环境。即pycharm作为编辑器，打开conda环境下的python工程。选择conda环境和对应版本即可。



这里我们发现很不方便，每次打开pycharm都需要到里面去打开，故可以创建桌面的快捷方式。

关于jupyter-notebook，这个在我的windows系统下的anaconda环境中是已经存在的，我认为再安装jupyter的意义不是很大，故没有在这里安装。需要用到的时候我会去再进行安装的。



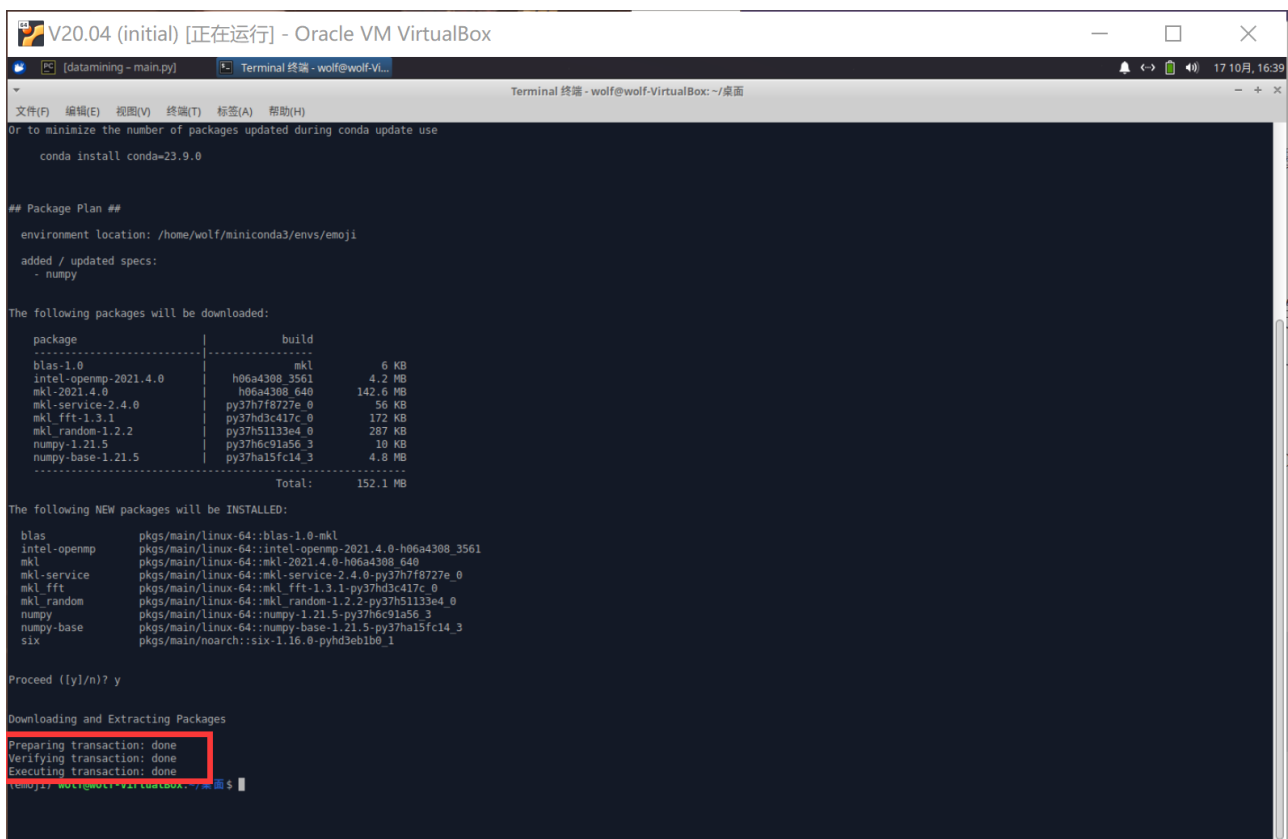
## 5.掌握pip和conda命令安装常用软件包。比如numpy、pandas、tensorflow、h5py、mygene matplotlib、seaborn、umap-learn等。

这一步就比较基础了，在之前windows下的anaconda环境中，我们也做过类似的事情。接下来逐个安装即可。

进入emoji环境。

```
conda activate emoji
conda install numpy
conda install pandas
pip install tensorflow #使用conda安装失败
conda install h5py
conda install matplotlib
conda install seaborn
pip install umap-learn #使用conda安装失败
conda list
```

出现以下三个done这样就表示这个包安装成功了。



```
V20.04 (initial) [正在运行] - Oracle VM VirtualBox
[datamining - main.py] Terminal 终端 - wolf@wolf-Vi...
Terminal 终端 - wolf@wolf-VirtualBox: ~/桌面

Or to minimize the number of packages updated during conda update use
conda install conda=23.9.0

## Package Plan ##
environment location: /home/wolf/miniconda3/envs/emoji
added / updated specs:
- numpy

The following packages will be downloaded:
-----
package | build | size
-----
blas-1.0 | mkl | 6 KB
intel-openmp-2021.4.0 | h06a4308_3561 | 4.2 MB
mkl-2021.4.0 | h06a4308_640 | 142.6 MB
mkl-service-2.4.0 | py37h7f8727e_0 | 56 KB
mkl_fft-1.3.1 | py37hd3c417c_0 | 172 KB
mkl_random-1.2.2 | py37h51133e4_0 | 267 KB
numpy-1.21.5 | py37h6c91a56_3 | 10 KB
numpy-base-1.21.5 | py37ha15fc14_3 | 4.8 MB
-----
Total: 152.1 MB

The following NEW packages will be INSTALLED:
blas pkgs/main/linux-64::blas-1.0-mkl
intel-openmp pkgs/main/linux-64::intel-openmp-2021.4.0-h06a4308_3561
mkl pkgs/main/linux-64::mkl-2021.4.0-h06a4308_640
mkl-service pkgs/main/linux-64::mkl-service-2.4.0-py37h7f8727e_0
mkl_fft pkgs/main/linux-64::mkl_fft-1.3.1-py37hd3c417c_0
mkl_random pkgs/main/linux-64::mkl_random-1.2.2-py37h51133e4_0
numpy pkgs/main/linux-64::numpy-1.21.5-py37h6c91a56_3
numpy-base pkgs/main/linux-64::numpy-base-1.21.5-py37ha15fc14_3
six pkgs/main/noarch::six-1.16.0-pyhd3eb1b0_1

Proceed ((y)/n)? y

Downloading and Extracting Packages
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
emoji:~$
```

其中tensorflow没有成功安装，故使用pip进行安装。



```
V20.04 (initial) [正在运行] - Oracle VM VirtualBox
[datamining - main.py] Terminal 终端 - wolf@wolf-Vi...
Terminal 终端 - wolf@wolf-VirtualBox: ~/桌面

The following NEW packages will be INSTALLED:

bottleneck      pkgs/main/linux-64::bottleneck-1.3.5-py37h7deecbd_0
numexpr         pkgs/main/linux-64::numexpr-2.8.4-py37he184ba9_0
packaging       pkgs/main/linux-64::packaging-22.0-py37h06a4308_0
pandas          pkgs/main/linux-64::pandas-1.3.5-py37h8c16a72_0
python-dateutil pkgs/main/noarch::python-dateutil-2.8.2-pyhd3eb1b0_0
pytz            pkgs/main/linux-64::pytz-2022.7-py37h06a4308_0

Proceed ([y]/n)? y

Downloading and Extracting Packages

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
(emoji) wolf@wolf-VirtualBox:~/桌面$ conda install tensorflow
Collecting package metadata (current repodata.json): done
Solving environment: unsuccessful initial attempt using frozen solve. Retrying with flexible solve.
Solving environment: unsuccessful attempt using repodata from current_repodata.json, retrying with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: \ \ / | - \ \ / - | - | / \ unsuccessful initial attempt using frozen solve. Retrying with flexible solve.

CondaError: KeyboardInterrupt

(emoji) wolf@wolf-VirtualBox:~/桌面$ pip install tensorflow
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Collecting tensorflow
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/42/24/830571895f0927fe205a23309b136520c7914921420bde1e81aff1da47bb1/tensorflow-2.11.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (588.3 MB)
Requirement already satisfied: numpy>=1.20 in /home/wolf/miniconda3/envs/emoji/lib/python3.7/site-packages (from tensorflow) (1.21.5)
Collecting tensorflow-io-gcs-filesystem<=0.23.1
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/47/45/f8aeca557bbd5fb505363520fec96cdec7246772sec4bc12fa24372b011a/tensorflow_io_gcs_filesystem-0.34.0-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (2.4 MB)
Collecting google-pasta<=0.1.1
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/a3/de/c648ef6835192e6e2cc03f40b19eeda4382c49b5bafb43d88b931c4c74ac/google_pasta-0.2.0-py3-none-any.whl (57 kB)
Collecting grpcio<2.0.0,>=1.24.3
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/7a/2c/be8c3cdc25d9946b67688f0712d3b4550d472c1267313ebd8c96f9c2122e/grpcio-1.59.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (5.3 MB)
Collecting tensorflow-estimator<2.12.0,>=2.11.0
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/bb/e2/8bf618c7c30a525054230ee6d40b036d3e5abc2c4ff67c7c7420a519204/tensorflow_estimator-2.11.0-py2.py3-none-any.whl (439 kB)
Collecting typing-extensions<=3.6.6
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/ec/6b/63cc3df74987c36fe26157ee12e09e8f9db4de771e0f3404263117e75b95/typing_extensions-4.7.1-py3-none-any.whl (33 kB)
Collecting keras<2.12.0,>=2.11.0
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/de/44/bf1b0eef5b13e6201aef076ff34b91bc4baace8591cd273c1c2a94a9cc00/keras-2.11.0-py2.py3-none-any.whl (1.7 MB)
Collecting gast<=0.4.0,>=0.2.1
```

其他都成功安装。

安装完毕之后使用

```
df -TH
conda list
```

分别查看Linux文件系统剩余空间和conda列表

```
V20.04 (initial) [正在运行] - Oracle VM VirtualBox
[datamining - main.py] Terminal 终端 - wolf@wolf-Vi...
Terminal 终端 - wolf@wolf-VirtualBox: ~/桌面
文件(F) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H)

Created wheel for pynndescent: filename=pynndescent-0.5.10-py3-none-any.whl size=55623 sha256=28d8a025a9f1af91f8342328610570fb0feb5c9dd57b75c150cd8d52669977a
Stored in directory: /home/wolf/.cache/pip/wheels/a6/a9/60/5cd68551c81ea035e291ffff3f29f4ab83d08ad06e83ecba93
Successfully built umap-learn pynndescent
Installing collected packages: tbb, tqdm, threadpoolctl, scipy, llvmlite, joblib, scikit-learn, numba, pynndescent, umap-learn
Successfully installed joblib-1.3.2 llvmlite-0.39.1 numba-0.56.4 pynndescent-0.5.10 scikit-learn-1.0.2 scipy-1.7.3 tbb-2021.10.0 threadpoolctl-3.1.0 tqdm-4.66.1 umap-learn-0.5.4
(wolf@wolf-VirtualBox:~/桌面) $ df -TH
文件系统 类型 容量 已用 可用 已用% 挂载点
udev devtmpfs 2.1G 0 2.1G 0% /dev
tmpfs tmpfs 411M 1.2M 410M 1% /run
/dev/sda5 ext4 21G 18G 1.8G 92% /
tmpfs tmpfs 2.1G 0 2.1G 0% /dev/shm
tmpfs tmpfs 5.3M 4.1k 5.3M 1% /run/lock
tmpfs tmpfs 2.1G 0 2.1G 0% /sys/fs/cgroup
/dev/sda1 vfat 536M 4.1k 536M 1% /boot/efi
tmpfs tmpfs 411M 13k 411M 1% /run/user/1000
(wolf@wolf-VirtualBox:~/桌面) $ conda list
# packages in environment at /home/wolf/miniconda3/envs/emoji:
#
# Name Version Build Channel
libgcc_mutex 0.1 main
openmp_mutex 5.1 1 gnu
absl-py 2.0.0 pypi_0 pypi
astunparse 1.6.3 pypi_0 pypi
blas 1.0 mkl
bottleneck 1.3.5 py37h7deecbd_0
brotli 1.0.9 h5eee18b_7
brotli-bin 1.0.9 h5eee18b_7
ca-certificates 2023.08.22 h06a4308_0
cachetools 5.3.1 pypi_0 pypi
certifi 2022.12.7 py37h06a4308_0
charset-normalizer 3.3.0 pypi_0 pypi
cycler 0.11.0 pyhd3eb1b0_0
cyrus-sasl 2.1.28 h9c9eb46_1
dbus 1.13.18 hb2f28db_0
expat 2.5.0 h6a678d5_0
flatbuffers 23.5.26 pypi_0 pypi
fontconfig 2.14.1 h4c34cd2_2
fonttools 4.25.0 pyhd3eb1b0_0
freetype 2.12.1 h4a9f257_0
gast 0.4.0 pypi_0 pypi
girlib 5.2.1 h5eee18b_3
glib 2.69.1 he621ea3_2
google-auth 2.23.3 pypi_0 pypi
google-auth-oauthlib 0.4.6 pypi_0 pypi
google-pasta 0.2.0 pypi_0 pypi
grpcio 1.59.0 pypi_0 pypi
gst-plugins-base 1.14.1 h6a678d5_1
gstreamer 1.14.1 h5eee18b_1
h5py 3.8.0 pypi_0 pypi
icu 58.2 he6710b0_3
idna 3.4 pypi_0 pypi
importlib-metadata 6.7.0 pypi_0 pypi
intel-openmp 2021.4.0 h06a4308_3561
joblib 1.3.2 pypi_0 pypi
jpeg 9e h5eee18b_1
```

可以看到我20G的空间啊!!! 都被装满了。