Ubuntu (Linux) Neo4j-5.6 Disposition

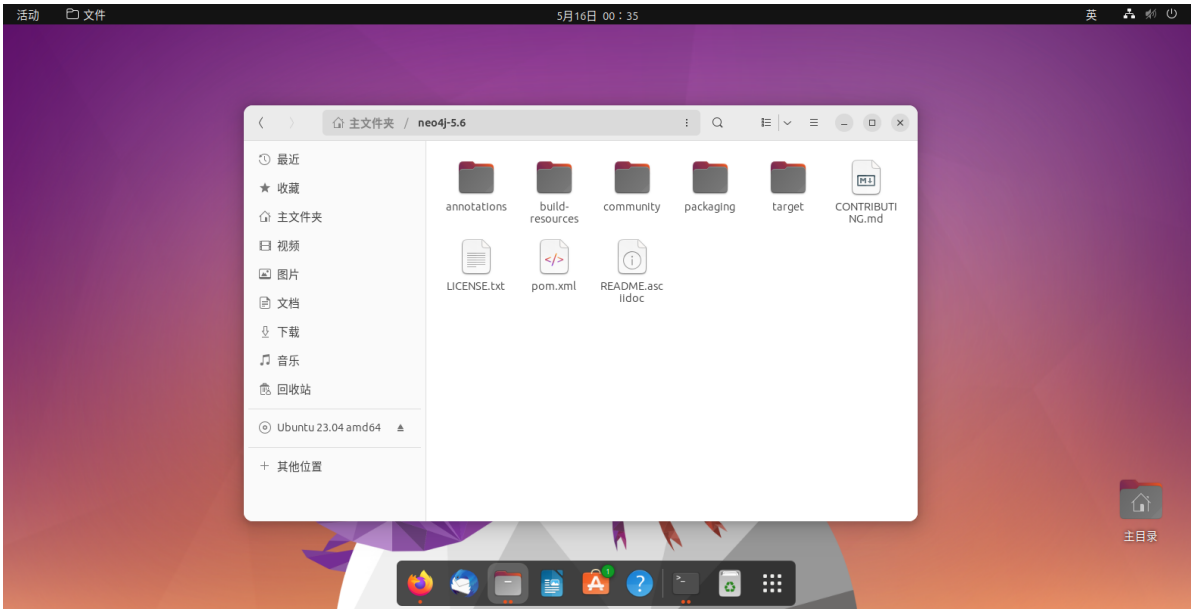
Hua Zhouqi, Department of Computer Science & Technology, Tongji University

System version	Ubuntu 23.04
Virtual machine version	VMware Workstation 17
Neo4j version	Neo4j-5.6

Reference: github https://github.com/neo4j/neo4j

Download source code

Download zip file from https://github.com/neo4j/neo4j and decompress into `\home\hzzzq\neo4j-5.6`



Download dependent files

Since the construction of neo4j relies on `Apache Maven` (project management and synthesis tool) , it's necessary to prepare the dependent environment before running the source code.

> `Maven`
>
> Based on the concept of the Project Object Model (**POM**) , Maven can manage project builds, reports, and files from a central slice. POM will be mentioned In subsequent problems.

> Learn more about Apache Maven: https://maven.apache.org/ or https://zhuanlan.zhihu.com/p/327818426

Open the terminal directly in Ubuntu (using `Ctrl+Alt+t`) and enter the following command:

```
sudo apt install maven openjdk-17-jdk
```

Neo4J is developed using the Java environment, so the version corresponding to the JDK is downloaded at the same time.

Build Neo4j project

According to the installation guide on GitHub, it's necessary to test the limit on the number of open files before building the project.

Choose build project after test or skip the test outright.

Test

The test only requires a single line of code in the terminal:

```
ulimit -n
```

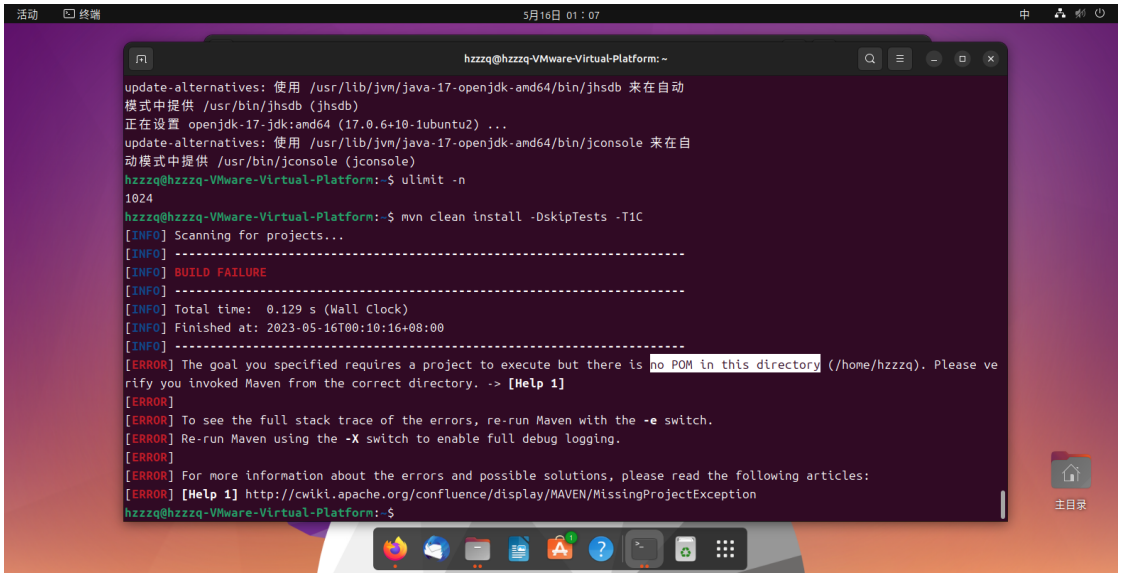
Skip Test

Run the following code to skip the test and directly get jars:

```
mvn clean install -DskipTests -T1C
```

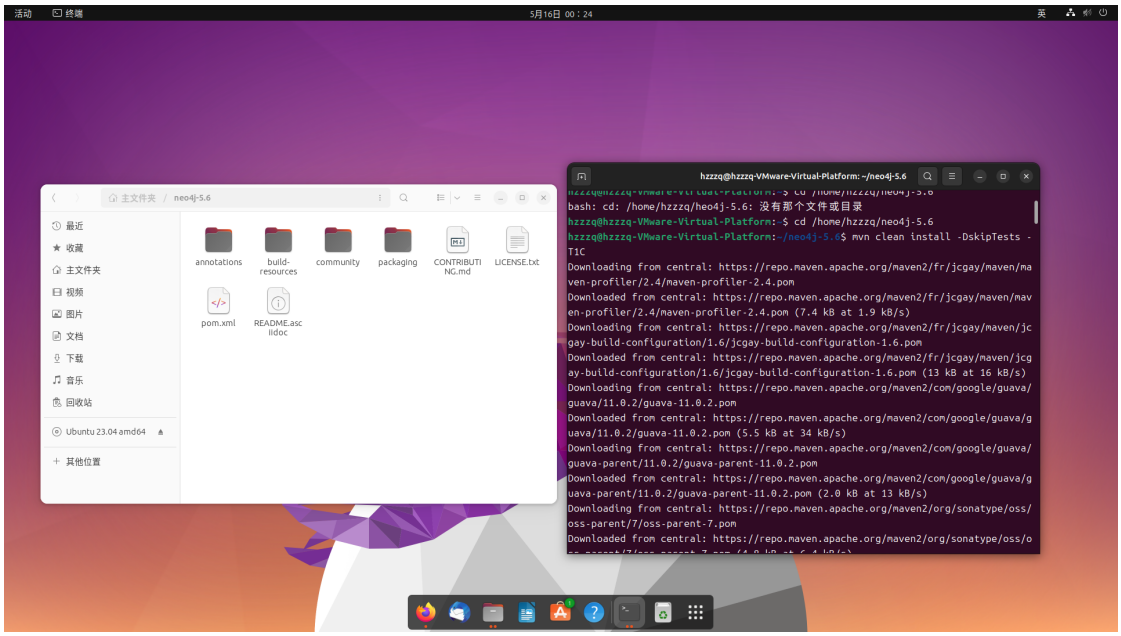
There a list of problem may occur: (or fortunately not happen)

1. `no POM in this directory`

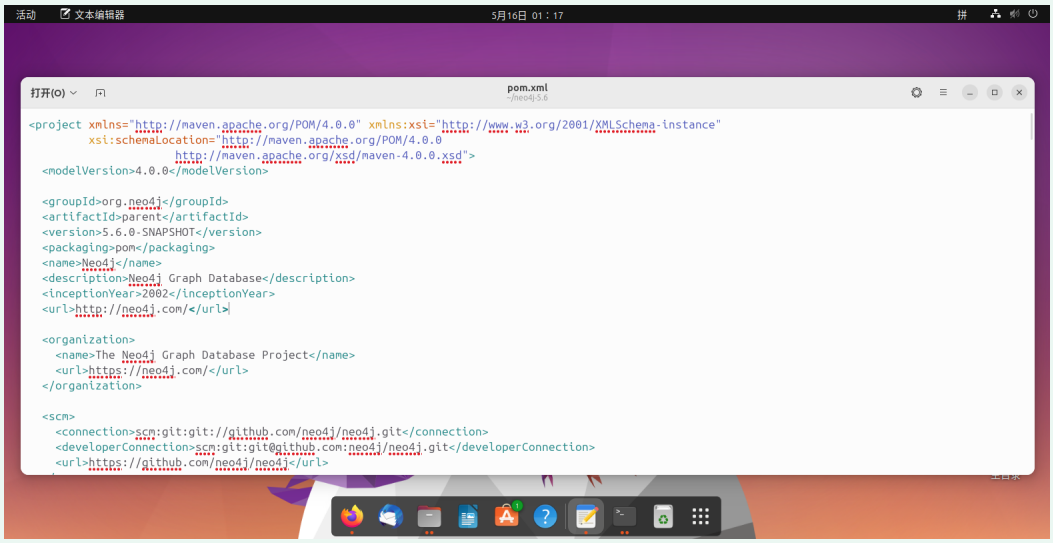


The reason for this error is that there is no pom.xml file in the current directory, and the Maven execution must pom.xml file, which means that it needs to be run in the pom.xml file directory of the project.

Since the full source code is downloaded, there is no problem with missing pom.xml files. So just locate the terminal operation location there using order `cd xxx`.



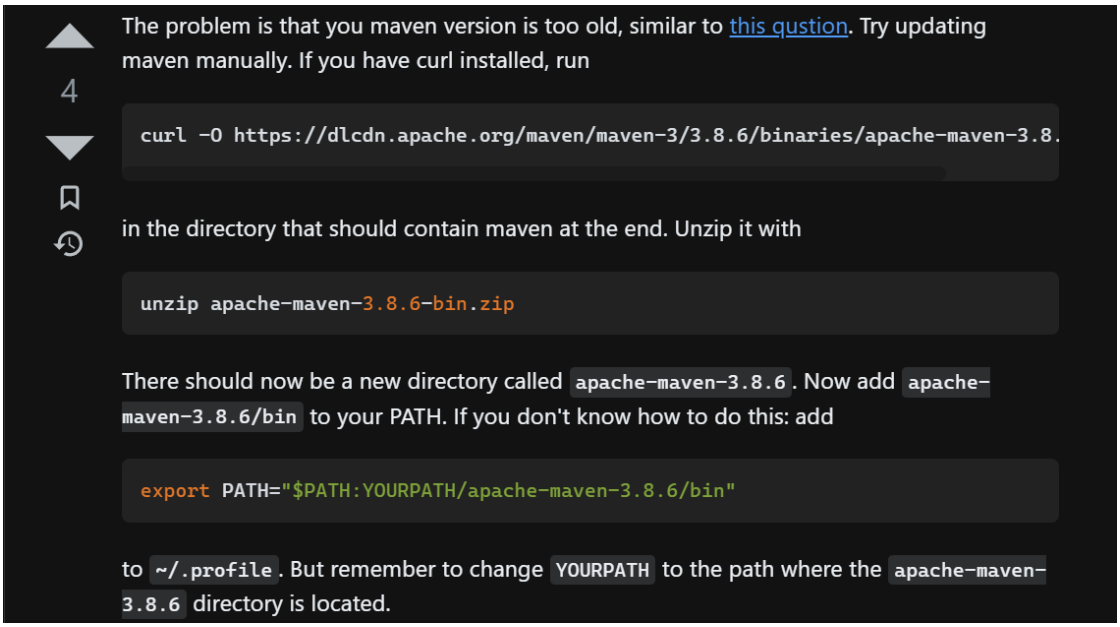
> `POM`
>
> POM is one of the core concepts in the Apache Maven project, which is an abbreviation for **Project Object Model**. POM is an XML file that describes the basic information of a Maven project, including project dependencies, plugins, version numbers, developer information, build configurations, etc.
>
> The role of POM includes:
>
> 1. Defines the **basic information and structure** of the project, including the project name, version number, author, license, and so on.
> 2. **Manage project dependencies**, define all dependent libraries and version numbers, and implement specific functions by configuring dependent libraries.
> 3. **Describes the build process** of the project, including how to compile, test, package, and so on.
> 4. Various **plugins are configured**, such as code inspection, unit testing, publishing, and so on.



2. `java.lang.IllegalStateException`

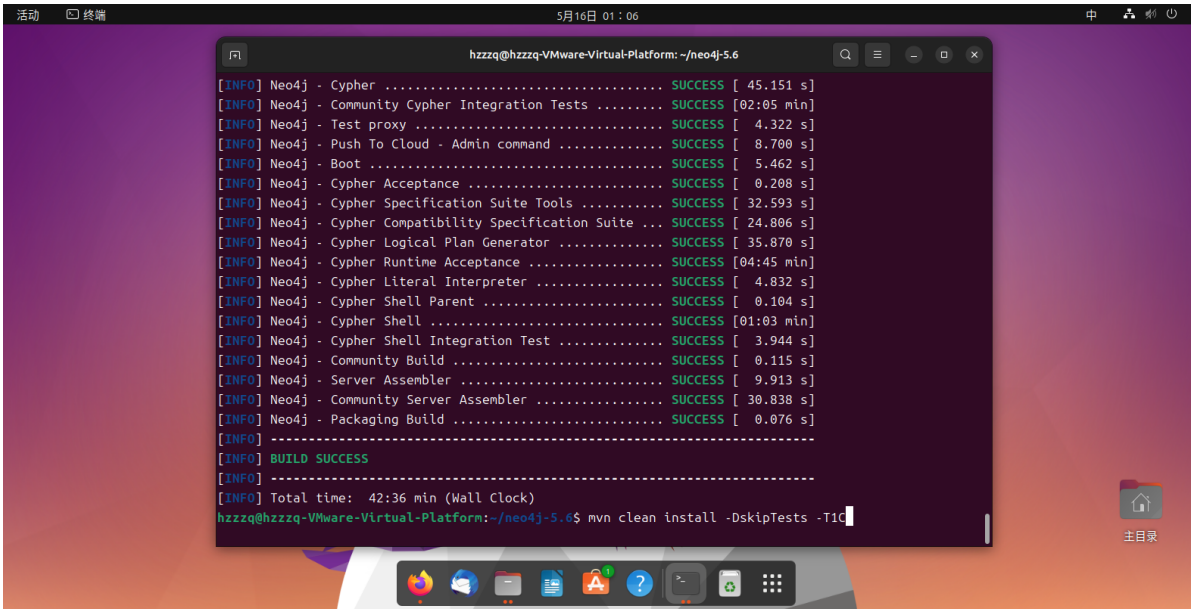
```
hzzzq@hzzzq-virtual-machine:~$ mvn clean install -DskipTests -T1C
[ERROR] Error executing Maven.
[ERROR] java.lang.IllegalStateException: Unable to load cache item
[ERROR] Caused by: Unable to load cache item
[ERROR] Caused by: Could not initialize class com.google.inject.internal.cglib.core.$MethodW
rapper
```

The cause of the problem is that the Java version and the Maven version do not match. Updating the Java version solves the problem.



(Thanks to senior *Zhu Tongtong* helps find the solution)

It takes a long time to run correctly :



BUILD SUCCESS Takes me **42min 32s** (no matter, I write this document during this period 😜) .

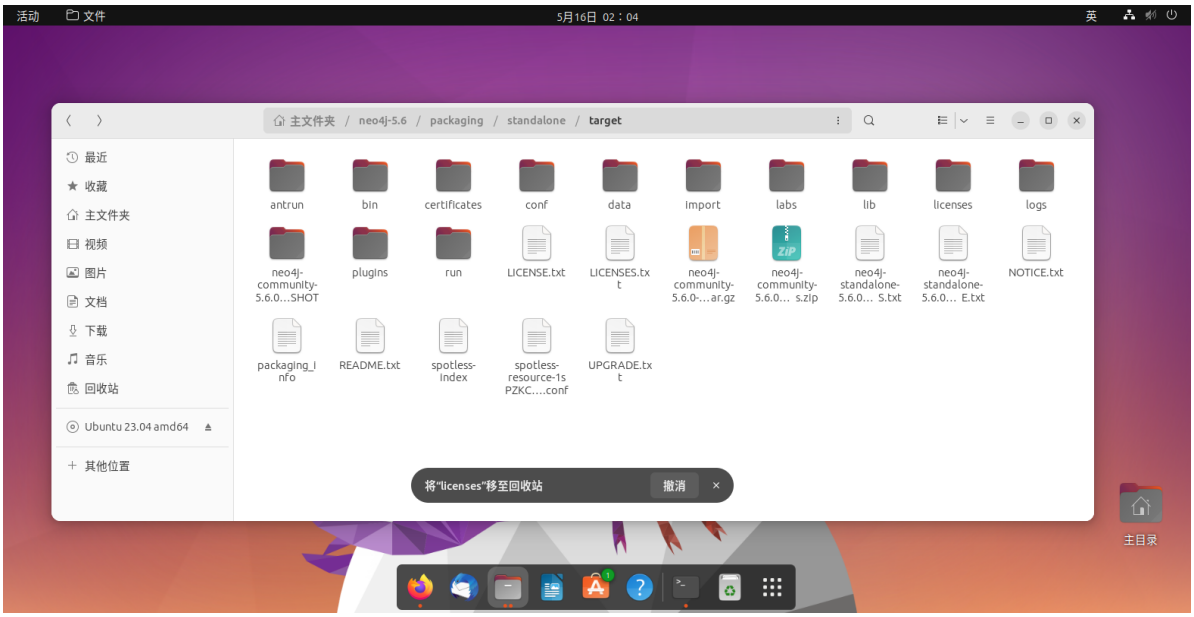
Then use the following command to **increase the amount of memory** available to Maven:

```
1   export MAVEN_OPTS="-Xmx2048m"
```

Run the project

All the preparation is complete! And now it's time to run the project code.

First, enter the following directories and extract the corresponding version of the package (unzip and put everything in a directory): `cd packaing/standalone/target`



Start the program with the following command from this directory terminal：

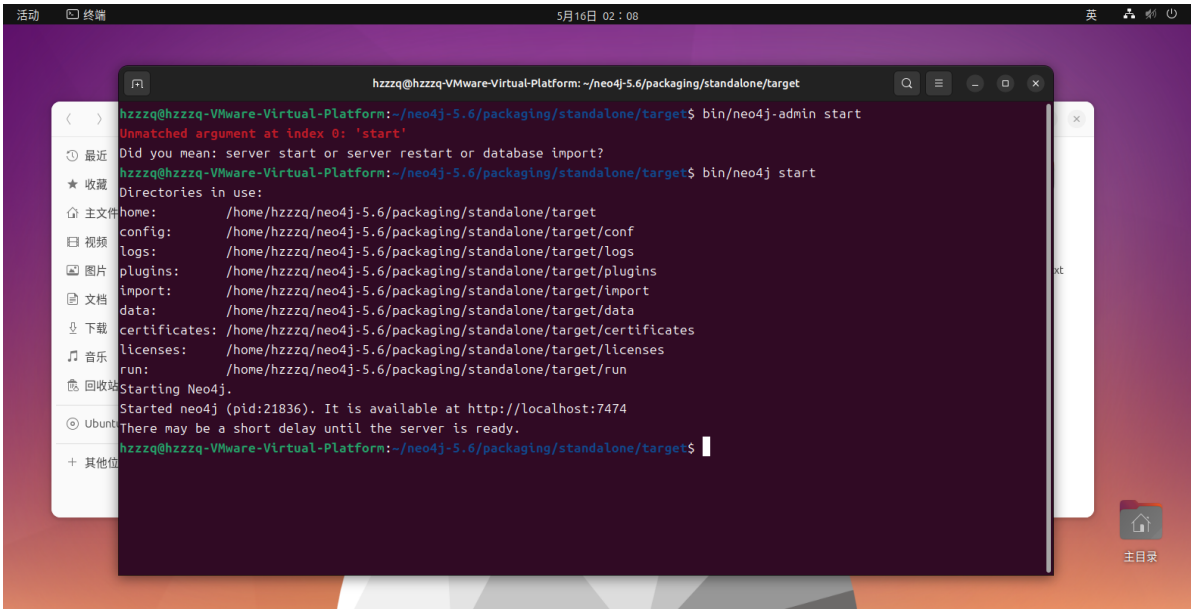
```
1   bin/neo4j start
```

or

```
1   bin/neo4j-admin start
```

Note that the Ubutunu system can only extract files in `.tar` archive, otherwise the command cannot be found.

The result of a successful run in the terminal:



-----END-----