实验报告

实验名称	实验四　Maven 的使用		
实验教室	丹青 922	实验日期	2021 年 6 月 6 日
学　　号	2019210173	姓　　名	刘思远
专业班级	计算机科学与技术 04 班		
指导教师	卢洋		

东北林业大学

信息与计算机科学技术实验中心

一、 实验目的

1. 掌握与课程相关的Maven工具的使用；

2. 掌握使用Maven制造Fat Jar包的方法。

二、 实验环境

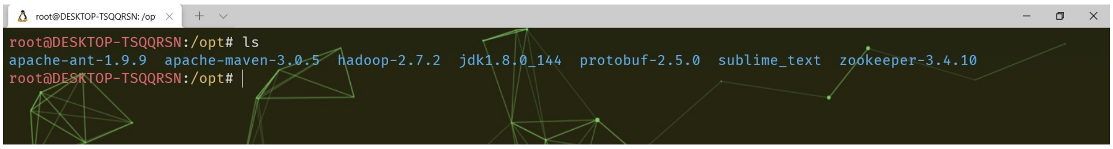
（1）计算机的硬件配置PC系列微机。

（2）计算机的软件配置VMware虚拟机软件及Ubuntu虚拟机。

三、 实验内容及结果

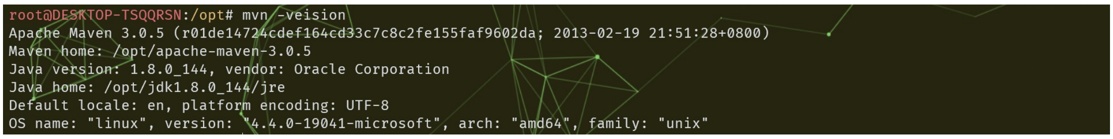
2.1 Maven的安装和配置

1. 下载安装超星在线课提供的 Maven ，版本号为 3.0.5 ；



```
root@DESKTOP-TSQQRSN:/opt# ls
apache-ant-1.9.9  apache-maven-3.0.5  hadoop-2.7.2  jdk1.8.0_144  protobuf-2.5.0  sublime_text  zookeeper-3.4.10
root@DESKTOP-TSQQRSN:/opt# 
```

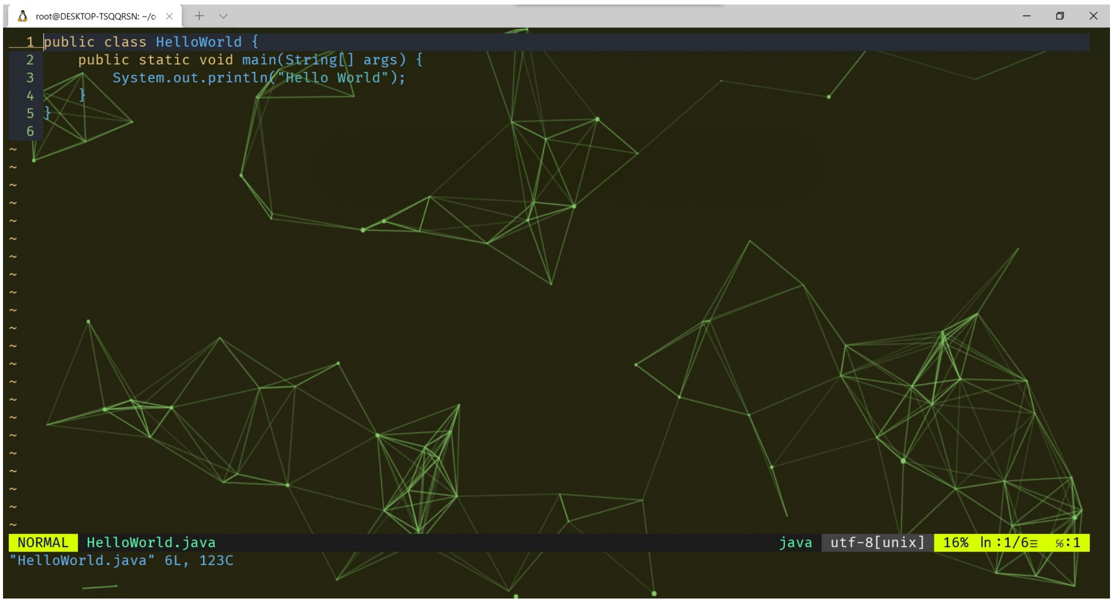
2. 验证所安装 Maven 的版本。



```
root@DESKTOP-TSQQRSN:/opt# mvn -veision
Apache Maven 3.0.5 (r01de14724cdef164cd33c7c8c2fe155faf9602da; 2013-02-19 21:51:28+0800)
Maven home: /opt/apache-maven-3.0.5
Java version: 1.8.0_144, vendor: Oracle Corporation
Java home: /opt/jdk1.8.0_144/jre
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "4.4.0-19041-microsoft", arch: "amd64", family: "unix"
```

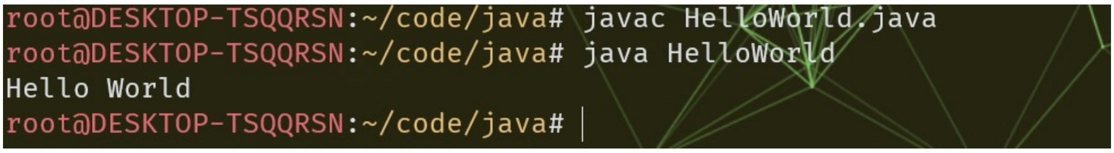
2.2 Maven的使用

1. 创建打印 Hello World 信息的 Java 程序，使用 javac 工具编译，并运行；



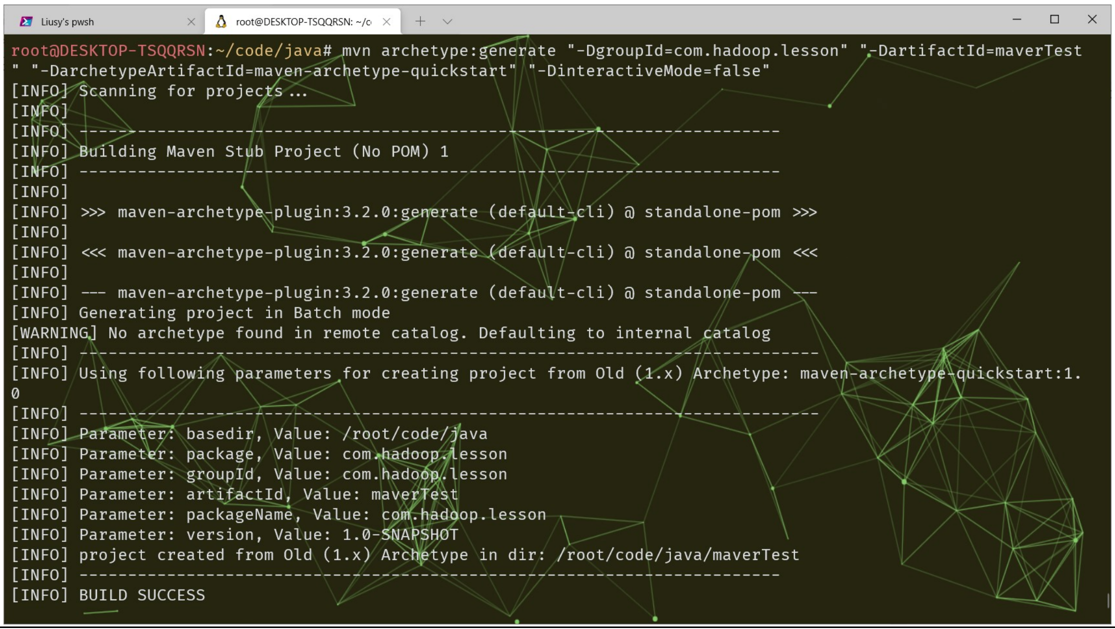
```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
 NORMAL  HelloWorld.java                                        java  utf-8[unix]  16%  ln :1/6≡  ℅:1
"HelloWorld.java" 6L, 123C
```

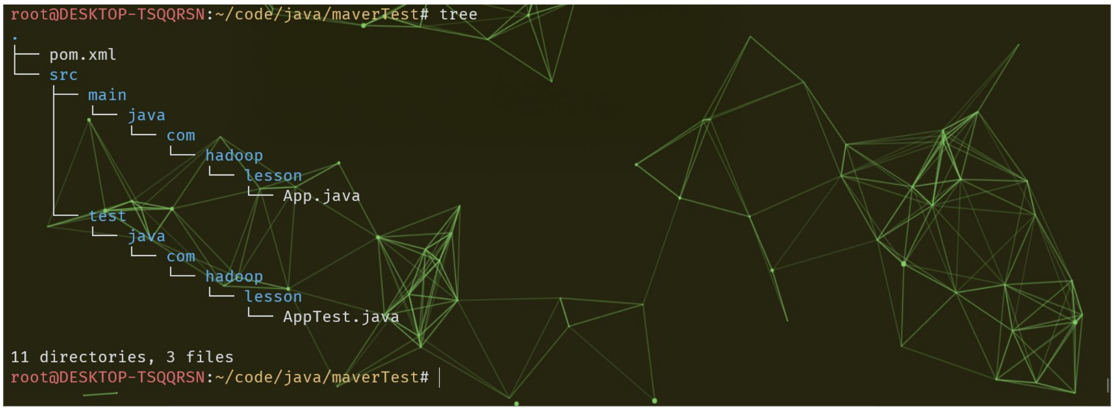


```
root@DESKTOP-TSQQRSN:~/code/java# javac HelloWorld.java
root@DESKTOP-TSQQRSN:~/code/java# java HelloWorld
Hello World
root@DESKTOP-TSQQRSN:~/code/java# 
```

2. 使用 mvn 命令创建一个项目，指定 groupId 的值为 com.hadoop.lesson ，并观察项目的结构；



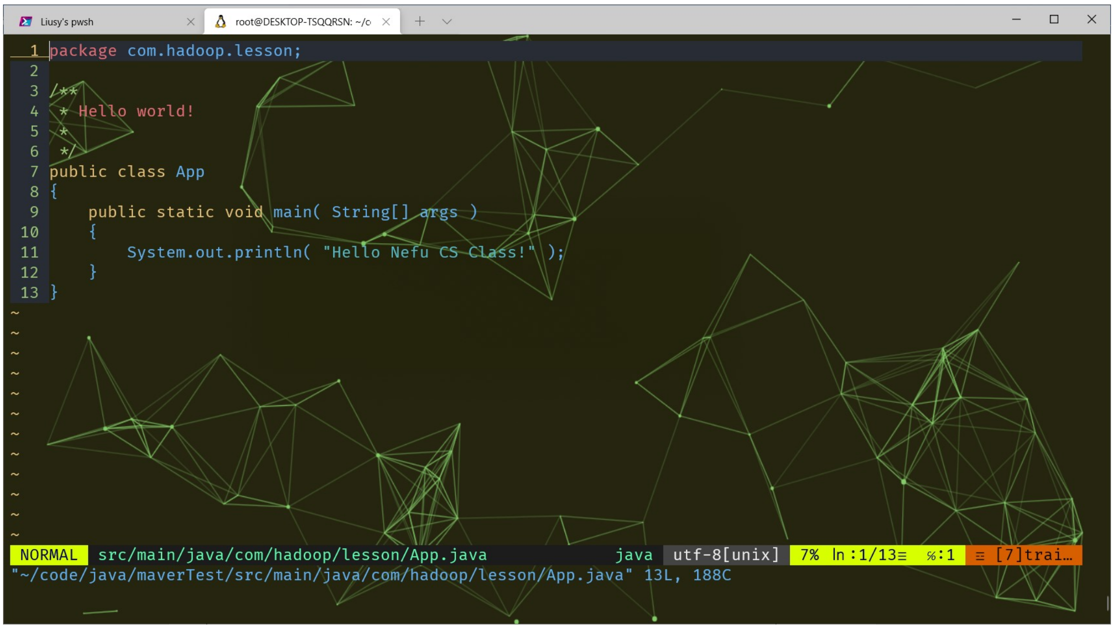
```
root@DESKTOP-TSQQRSN:~/code/java# mvn archetype:generate "-DgroupId=com.hadoop.lesson" "-DartifactId=maverTest
" "-DarchetypeArtifactId=maven-archetype-quickstart" "-DinteractiveMode=false"
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building Maven Stub Project (No POM) 1
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] >>> maven-archetype-plugin:3.2.0:generate (default-cli) @ standalone-pom >>>
[INFO]
[INFO] <<< maven-archetype-plugin:3.2.0:generate (default-cli) @ standalone-pom <<<
[INFO]
[INFO] --- maven-archetype-plugin:3.2.0:generate (default-cli) @ standalone-pom ---
[INFO] Generating project in Batch mode
[WARNING] No archetype found in remote catalog. Defaulting to internal catalog
[INFO] ----------------------------------------------------------------------------
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-quickstart:1.
0
[INFO] ----------------------------------------------------------------------------
[INFO] Parameter: basedir, Value: /root/code/java
[INFO] Parameter: package, Value: com.hadoop.lesson
[INFO] Parameter: groupId, Value: com.hadoop.lesson
[INFO] Parameter: artifactId, Value: maverTest
[INFO] Parameter: packageName, Value: com.hadoop.lesson
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: /root/code/java/maverTest
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
```



```
root@DESKTOP-TSQQRSN:~/code/java/maverTest# tree
.
├── pom.xml
└── src
    ├── main
    │   └── java
    │       └── com
    │           └── hadoop
    │               └── lesson
    │                   └── App.java
    └── test
        └── java
            └── com
                └── hadoop
                    └── lesson
                        └── AppTest.java

11 directories, 3 files
root@DESKTOP-TSQQRSN:~/code/java/maverTest# 
```

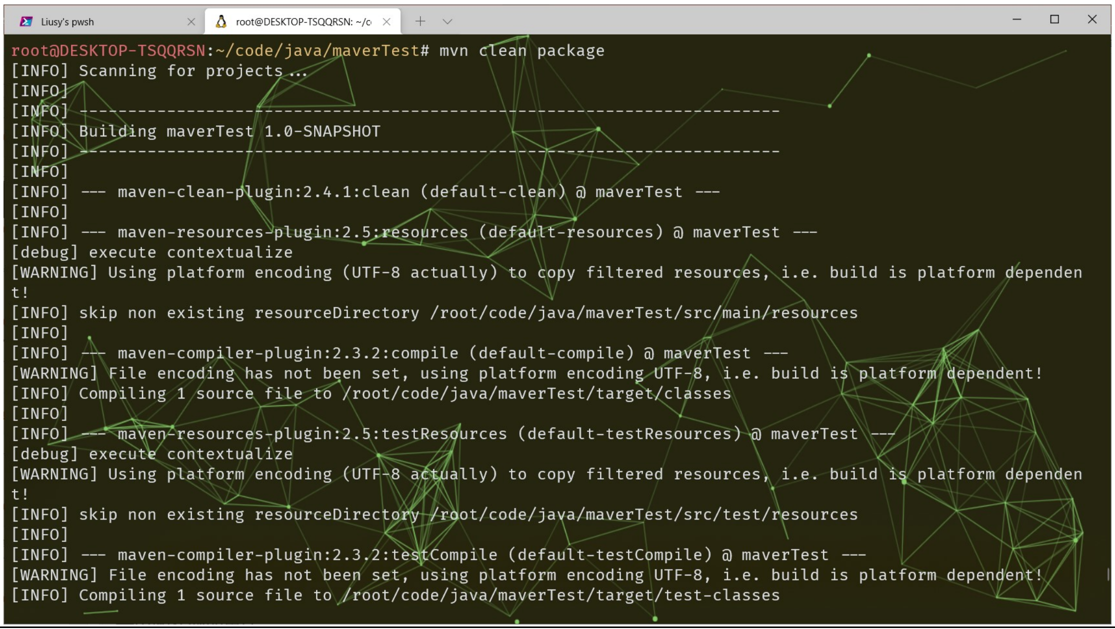
3. 在该项目下创建打印 Hello Nefu CS Class! 信息的程序；



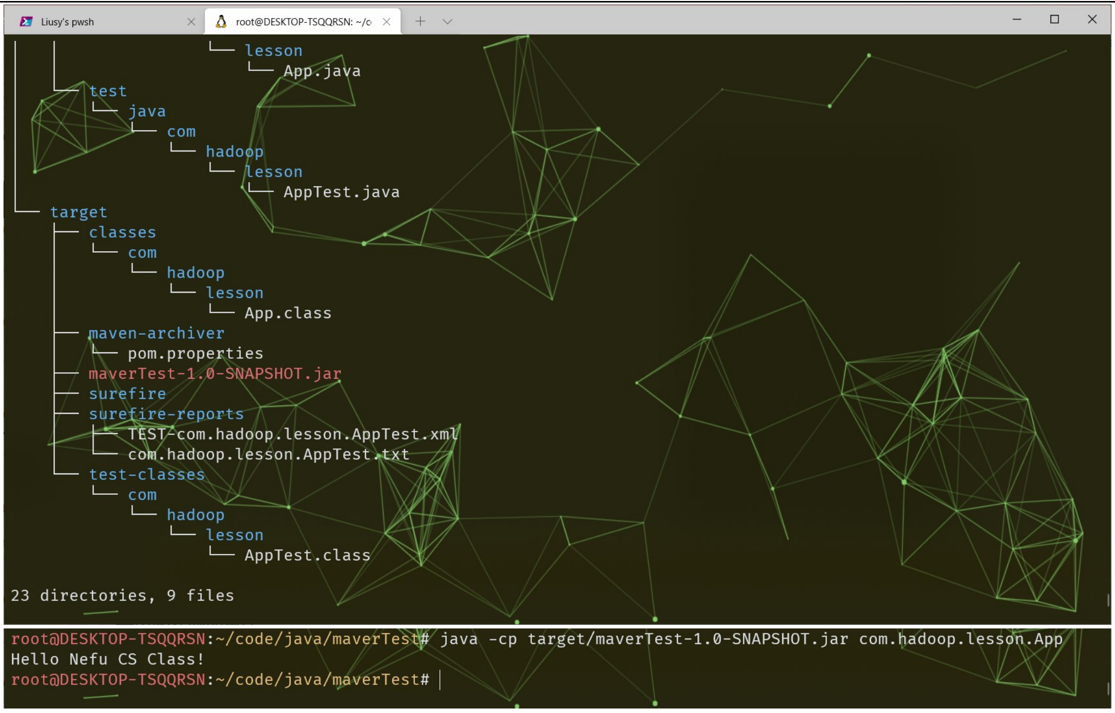
```
package com.hadoop.lesson;

/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args )
    {
        System.out.println( "Hello Nefu CS Class!" );
    }
}
~
~
~
~
~
~
~
~
~
~
~
~
 NORMAL  src/main/java/com/hadoop/lesson/App.java          java  utf-8[unix]  7%  ln :1/13≡  ℅:1  ≡ [7]trai…
"~/code/java/maverTest/src/main/java/com/hadoop/lesson/App.java" 13L, 188C
```

4. 创建项目并执行。



```
root@DESKTOP-TSQQRSN:~/code/java/maverTest# mvn clean package
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building maverTest 1.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.4.1:clean (default-clean) @ maverTest ---
[INFO]
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @ maverTest ---
[debug] execute contextualize
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependen
t!
[INFO] skip non existing resourceDirectory /root/code/java/maverTest/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @ maverTest ---
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 1 source file to /root/code/java/maverTest/target/classes
[INFO]
[INFO] --- maven-resources-plugin:2.5:testResources (default-testResources) @ maverTest ---
[debug] execute contextualize
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependen
t!
[INFO] skip non existing resourceDirectory /root/code/java/maverTest/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @ maverTest ---
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 1 source file to /root/code/java/maverTest/target/test-classes
```



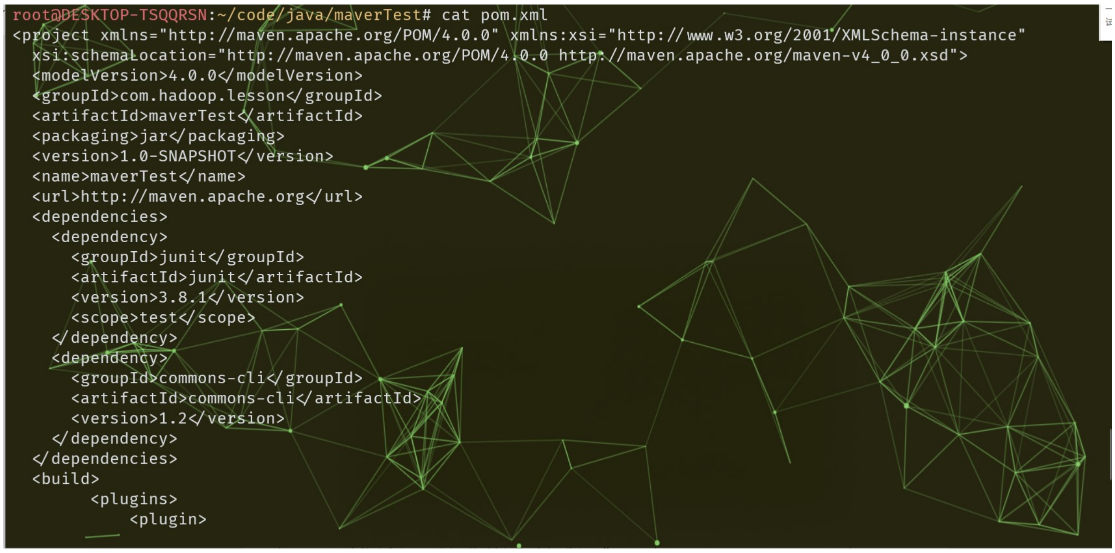
```
│   │               └── lesson
│   │                   └── App.java
│   └── test
│       └── java
│           └── com
│               └── hadoop
│                   └── lesson
│                       └── AppTest.java
└── target
    ├── classes
    │   └── com
    │       └── hadoop
    │           └── lesson
    │               └── App.class
    ├── maven-archiver
    │   └── pom.properties
    ├── maverTest-1.0-SNAPSHOT.jar
    ├── surefire
    ├── surefire-reports
    │   ├── TEST-com.hadoop.lesson.AppTest.xml
    │   └── com.hadoop.lesson.AppTest.txt
    └── test-classes
        └── com
            └── hadoop
                └── lesson
                    └── AppTest.class

23 directories, 9 files
root@DESKTOP-TSQQRSN:~/code/java/maverTest# java -cp target/maverTest-1.0-SNAPSHOT.jar com.hadoop.lesson.App
Hello Nefu CS Class!
root@DESKTOP-TSQQRSN:~/code/java/maverTest# 
```

2.3 引入外部依赖

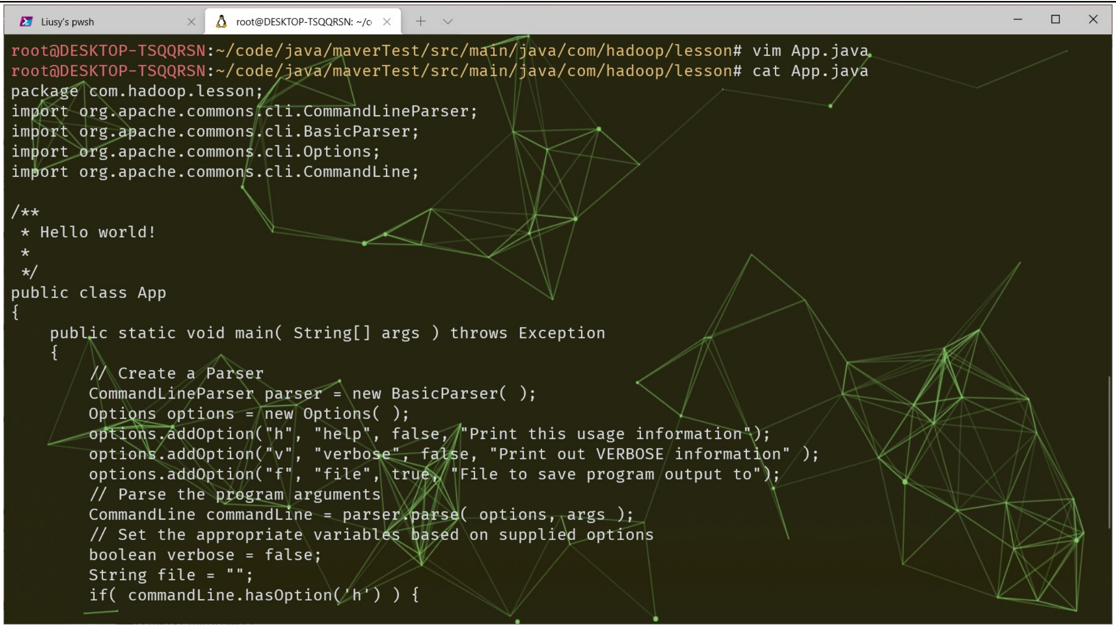
可以使用课程中讲解的命令行参数解析，也可以使用其它包。

1. 更新 pom.xml 文件；



```
root@DESKTOP-TSQQRSN:~/code/java/maverTest# cat pom.xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.hadoop.lesson</groupId>
  <artifactId>maverTest</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>maverTest</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>commons-cli</groupId>
      <artifactId>commons-cli</artifactId>
      <version>1.2</version>
    </dependency>
  </dependencies>
  <build>
        <plugins>
            <plugin>
```

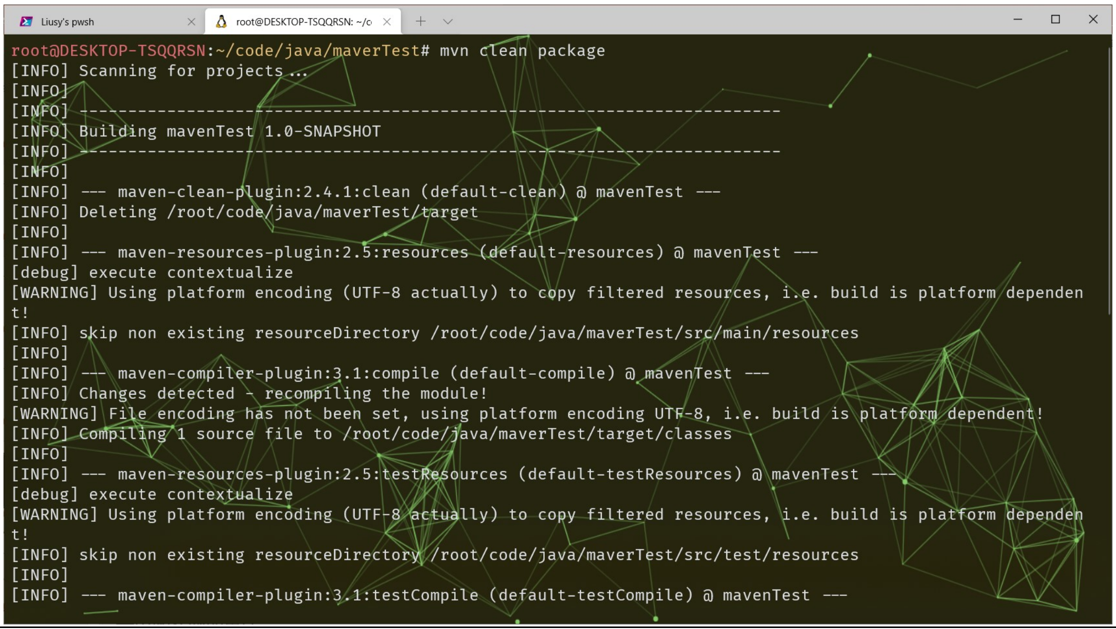
2. 修改源程序；



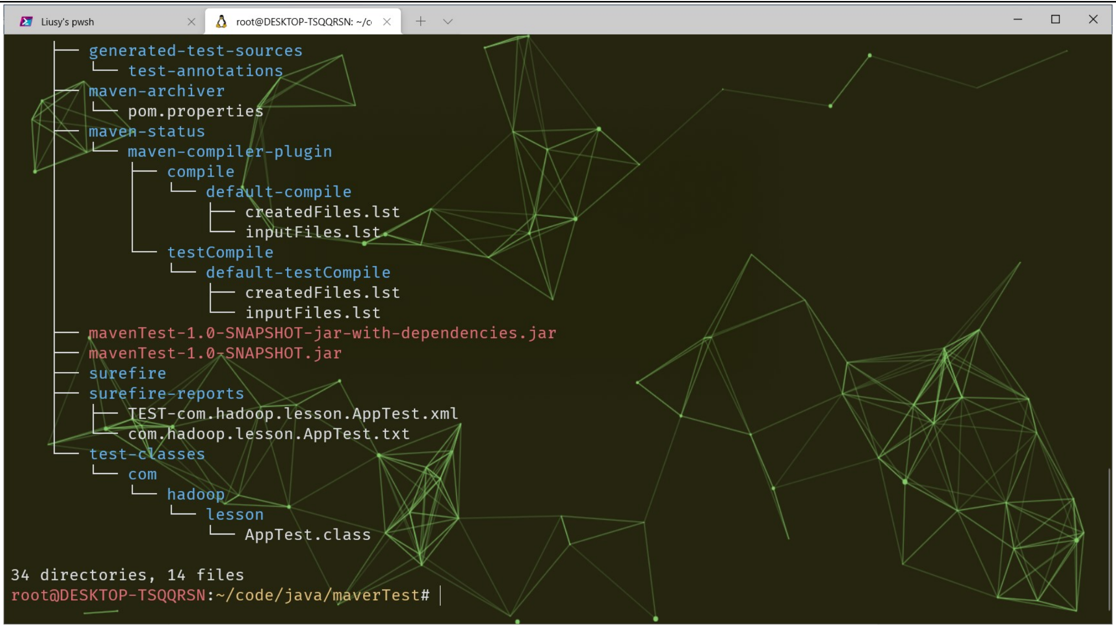
```
root@DESKTOP-TSQQRSN:~/code/java/maverTest/src/main/java/com/hadoop/lesson# vim App.java
root@DESKTOP-TSQQRSN:~/code/java/maverTest/src/main/java/com/hadoop/lesson# cat App.java
package com.hadoop.lesson;
import org.apache.commons.cli.CommandLineParser;
import org.apache.commons.cli.BasicParser;
import org.apache.commons.cli.Options;
import org.apache.commons.cli.CommandLine;

/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args ) throws Exception
    {
        // Create a Parser
        CommandLineParser parser = new BasicParser( );
        Options options = new Options( );
        options.addOption("h", "help", false, "Print this usage information");
        options.addOption("v", "verbose", false, "Print out VERBOSE information" );
        options.addOption("f", "file", true, "File to save program output to");
        // Parse the program arguments
        CommandLine commandLine = parser.parse( options, args );
        // Set the appropriate variables based on supplied options
        boolean verbose = false;
        String file = "";
        if( commandLine.hasOption('h') ) {
```

3. 构建项目，得到具备外部依赖的 Jar 包；



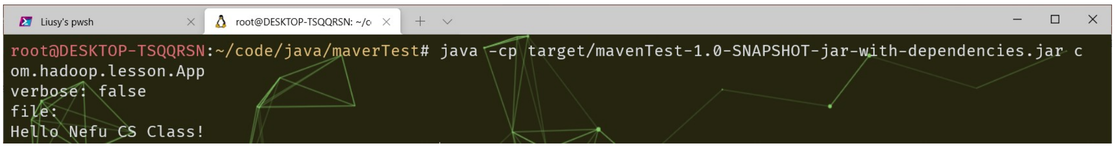
```
root@DESKTOP-TSQQRSN:~/code/java/maverTest# mvn clean package
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building mavenTest 1.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.4.1:clean (default-clean) @ mavenTest ---
[INFO] Deleting /root/code/java/maverTest/target
[INFO]
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @ mavenTest ---
[debug] execute contextualize
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependen
t!
[INFO] skip non existing resourceDirectory /root/code/java/maverTest/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ mavenTest ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 1 source file to /root/code/java/maverTest/target/classes
[INFO]
[INFO] --- maven-resources-plugin:2.5:testResources (default-testResources) @ mavenTest ---
[debug] execute contextualize
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependen
t!
[INFO] skip non existing resourceDirectory /root/code/java/maverTest/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ mavenTest ---
```



```
    ├── generated-test-sources
    │   └── test-annotations
    ├── maven-archiver
    │   └── pom.properties
    ├── maven-status
    │   └── maven-compiler-plugin
    │       ├── compile
    │       │   └── default-compile
    │       │       ├── createdFiles.lst
    │       │       └── inputFiles.lst
    │       └── testCompile
    │           └── default-testCompile
    │               ├── createdFiles.lst
    │               └── inputFiles.lst
    ├── mavenTest-1.0-SNAPSHOT-jar-with-dependencies.jar
    ├── mavenTest-1.0-SNAPSHOT.jar
    ├── surefire
    ├── surefire-reports
    │   ├── TEST-com.hadoop.lesson.AppTest.xml
    │   └── com.hadoop.lesson.AppTest.txt
    └── test-classes
        └── com
            └── hadoop
                └── lesson
                    └── AppTest.class

34 directories, 14 files
root@DESKTOP-TSQQRSN:~/code/java/maverTest#
```

4. 运行项目并观察结果。



```
root@DESKTOP-TSQQRSN:~/code/java/maverTest# java -cp target/mavenTest-1.0-SNAPSHOT-jar-with-dependencies.jar c
om.hadoop.lesson.App
verbose: false
file:
Hello Nefu CS Class!
```

四、 实验过程分析与讨论

学会 maven 项目的使用，对 Java 项目的管理和构建非常方便。在动手实验中，安装了 maven，并配置了 pom.xml。自己创建了 maven 仓库，并运行了 Java 程序，之后引入依赖，减少了项目的内存大小，并成功运行。

五、指导教师意见

指导教师签字：卢洋