

1. 开始, Lab目录的目录结构如下:

```
.
├─ Lab.iml
├─ Makefile
├─ Makefile.git
├─ src
├─ 201200000.pdf
├─ submit.sh
├─ tests
└─ test1.sysy
```

2. Lab目录下新建 pom.xml, 写入如下内容:

```
<?xml version="1.0" encoding="UTF-8"?>
<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/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

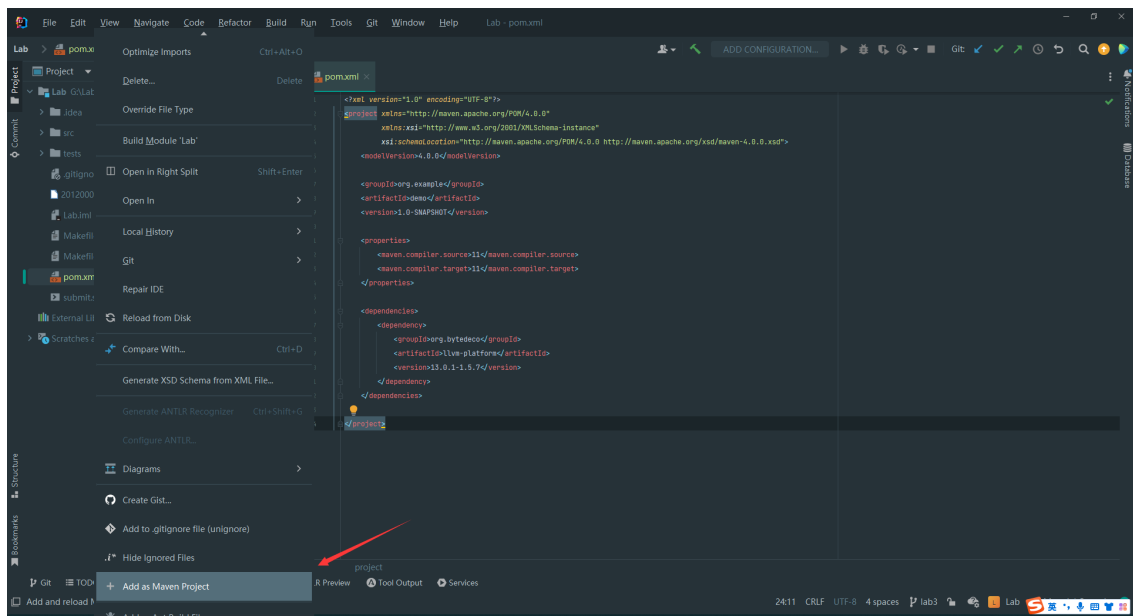
    <groupId>org.example</groupId>
    <artifactId>demo</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>11</maven.compiler.source>
        <maven.compiler.target>11</maven.compiler.target>
    </properties>

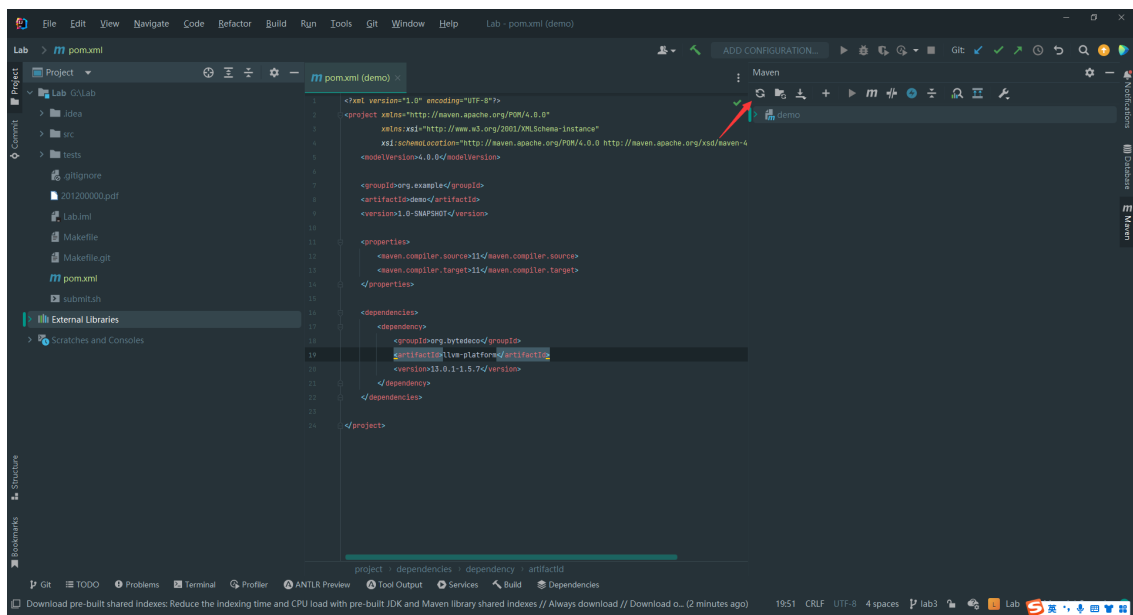
    <dependencies>
        <dependency>
            <groupId>org.bytedeco</groupId>
            <artifactId>llvm-platform</artifactId>
            <version>13.0.1-1.5.7</version>
        </dependency>
    </dependencies>

</project>
```

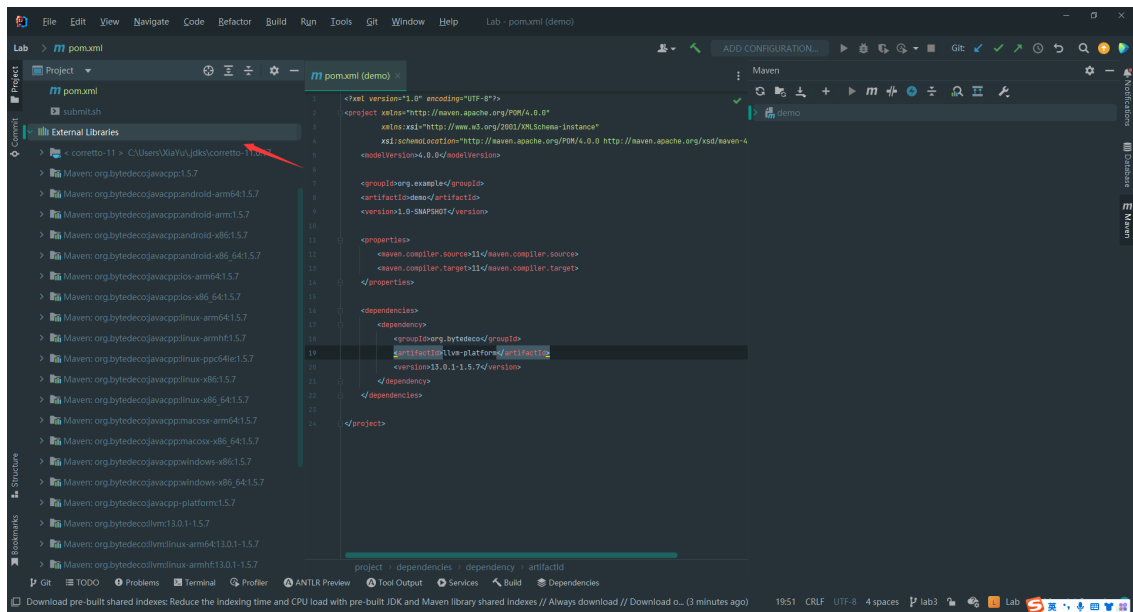
3. 右键 pom.xml, 选择 Add as Maven Project



点击同步，下载/加载依赖的 Jar 包

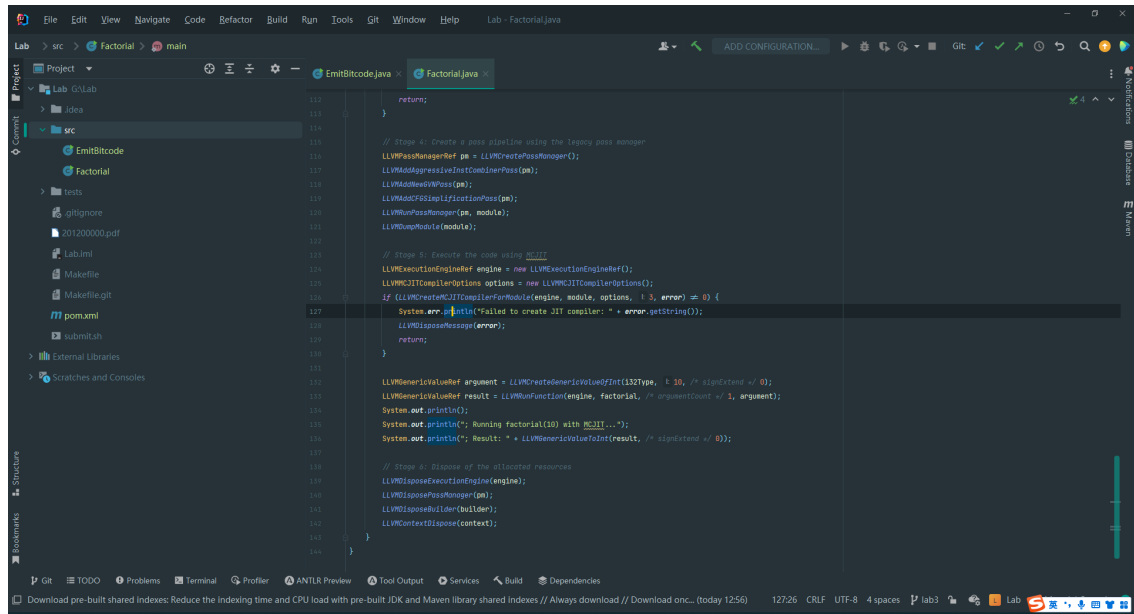


可以发现 Jar 包已导入。

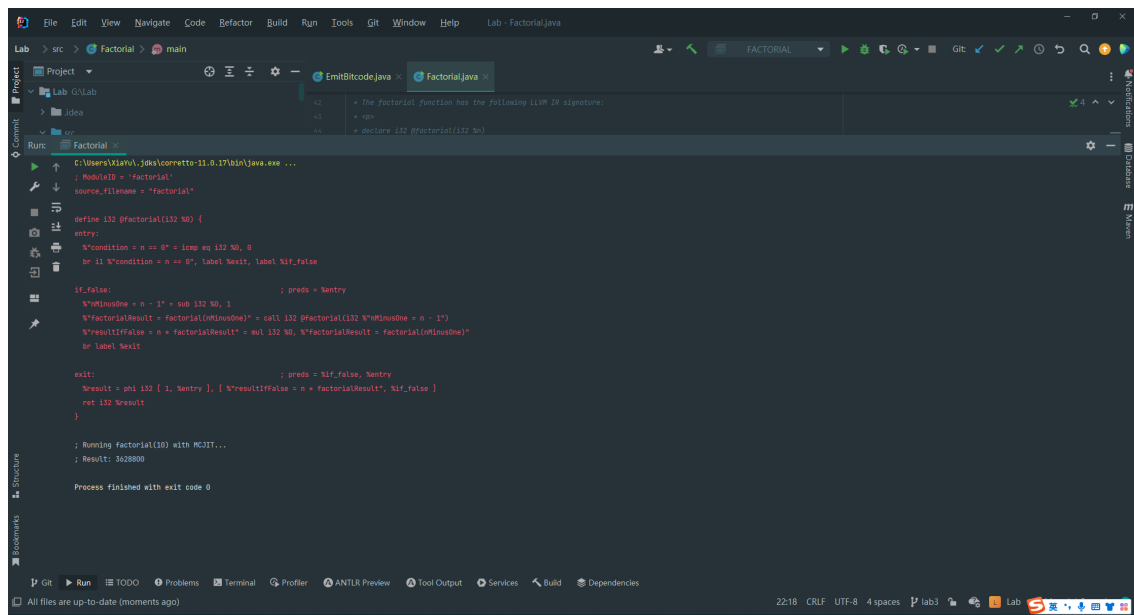


提醒：记得按之前教的方式手动导入 `antlr.jar` 包，不建议使用 `maven` 方式。因为 `Lab` 现在变成了一个 `maven` 项目，所以你需要向 `.gitignore` 文件中添加包括但不限于 `pom.xml`, `/target` 等内容，具体添加什么取决于你本地运行程序时多产生了什么文件/文件夹。

4. 从 [这里](#) 下载 `LLVM JAR` 包的使用样例 `EmitBitcode` 和 `Factorial`，拷贝到你本地的 `src` 目录下。



5. `Factorial` 可以直接执行它的 `main` 方法，观察到有如下输出。



这就成功生成了 `LLVM IR`

6. 对于 `EmitBitcode`，它接受一个为 `-emit` 或 `-evaluate`。

`-emit`：生成 `sum.bc`（`LLVM IR` 的二进制表示形式文件）和 `sum.o`（`linux` 可执行文件）文件。

`-evaluate`：执行 `-emit` 生成的 `sum.bc` 文件并输出结果

7. 实验环境搭建完成。希望你对照这两份源代码好好学习一下 `Jar` 包的简单使用，了解一下 `API`，对照实验文档以及助教提供的手册完成实验。（请记得提交代码前删除你下载的两个示例文件）。