

Installing ANTLR4 in your Linux

Java

Make sure you have Java installed. Check it with

```
java -version
```

If you don't have it, install it normally with your distro package manager.

ANTLR4

Option 1: Using the package manager

Modern Linux distros already include a package for ANTLR, although it is probably too old.

```
sudo apt-get install antlr4
```

Option 2: Getting the jar file

Alternatively, you can go to <https://www.antlr.org> and follow the instructions there, which mainly consist of:

```
# Download antlr4 jar file to /usr/local/lib
cd /usr/local/lib
sudo wget http://www.antlr.org/download/antlr-4.13.2-complet
```

IMPORTANT: Do NOT use earlier versions than 4.13, the API is different and it will not work with our code.

You will also need to create a command file that will ease the call to Antlr4. Edit a file named `antlr4` and write into it:

```
#!/bin/bash
export CLASSPATH=".:usr/local/lib/antlr-4.13.2-complete.jar"
java -jar /usr/local/lib/antlr-4.13.2-complete.jar "$@"
```

save the file, and give it execution permission:

```
chmod +x antlr4
```

finally, move it to a folder in the default PATH

```
sudo mv antlr4 /usr/local/bin
```

C++ Runtime

Although Antlr4 is a Java-universe project, it can be instructed to generate C++ code. The resulting code uses some C++ classes included in a shared library (the *runtime*) that is not installed by default with Antlr4. To install it we can follow the steps below:

Option 1: Using the package manager

Some linux distributions have this library packaged. Installing the *dev* package should pull the binaries too. Make sure the version matches the 4.13 ANTLR version installed.

```
sudo apt-get install libantlr4-runtime-dev
```

Option 2: Compiling from source

If your distribution does not provide this package, you'll need to install it from source:

First, install required tools and dependencies:

```
sudo apt-get install cmake uuid-dev pkgconf
```

Then, download the source from

<https://www.antlr.org/download.html> and compile it.

```
# download source
mkdir antlr-src && cd antlr-src
wget http://www.antlr.org/download/antlr4-cpp-runtime-4.13.2
# unzip file
unzip antlr4-cpp-runtime-4.13.2-source.zip
# create build directory for cmake
mkdir build && cd build
# create makefiles
cmake .. -DANTLR_JAR_LOCATION=/usr/local/lib/antlr-4.13.2-co
# build and install
make
sudo make install
```

If you installed Antlr4 from a distribution package, your `ANTLR_JAR_LOCATION` will be something like `/usr/share/java/antlr4-4.13.2.jar`. Examine the contents of the package and set the right path in the `cmake` command.

Adapting Makefiles

Makefiles provided with the class exercises and lab project expect ANTLR4 runtime to be in `/assig/$USER/cl/runtime`

In order to run them in your computer, you need to edit the Makefile and replace line 80

```
ANTLR_ROOT := /assig/$(USER)/cl/runtime
```

with

- If you installed from distribution packages:

```
ANTLR_ROOT := /usr
```

- If you installed the jarfile directly, set it to the folder where you created the `antlr4` command file

```
ANTLR_ROOT := /usr/local
```

Once the Makefile is fixed, you should be able to compile the class examples:

```
cd intro/lab1/Calc
make antlr
make
./main calc1.txt
```

Setting up Library Paths

If you get errors when running your `./main` program, it may be that the loader is not finding the runtime.

If you installed the runtime in `/usr/local/lib` you may need to update loader paths:

```
sudo ldconfig
```

If you installed the runtime in some other folder, you need to set the right environment variables so that the loader finds it:

```
export LD_LIBRARY_PATH=/my/runtime/folder
```

You can add this line to your `.bashrc` to avoid having to do it for every session.

MacOS users: in MacOS the variable name is `DYLD_LIBRARY_PATH`

Using `grun` to visualize trees

You can use `grun` to visualize and debug the trees your grammar is creating.

Details on how to do so are given in the [Introduction to ANTLR4](#) page.

However, to run it in your computer you'll need to set up the paths for the java components of ANTLR4. Edit your `$HOME/.bashrc` file and add the commands:

```
export ANTLR4=/usr/share/java
if ( $?CLASSPATH ) then
    export CLASSPATH=.:$ANTLR4/antlr-4.13.2-complete.jar:$CL
else
    export CLASSPATH=.:$ANTLR4/antlr-4.13.2-complete.jar
fi
```

If you installed antlr4 by getting the jarfile instead of using the package manager, you may need to change the path `/usr/share/java` to another location (e.g. `/usr/local/share/java` or wherever you installed it)

Once `.bashrc` has been updated, open a new terminal to get the new configuration loaded.