**1. 系统目录结构**

/ //系统根目录

~ //当前用户主目录

. //当前目录

.. //上一级目录

- //之前目录

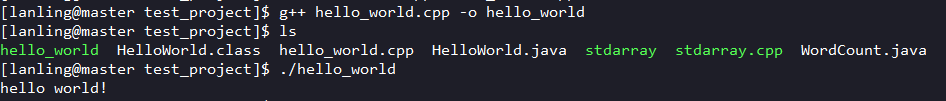
**2. 运行C++程序**

源文件hello\_world.cpp



# g++ -std=c++11 hello\_world.cpp –o hello\_world

# ./hello\_world



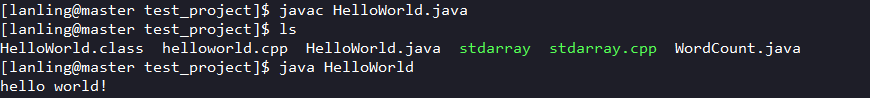
**3. 运行JAVA程序**

源文件HelloWorld.java



# javac HelloWorld.java

# java HelloWorld



**4. 运行Python程序**

Ubuntu安装python3

Ubuntu系统默认python版本为python2

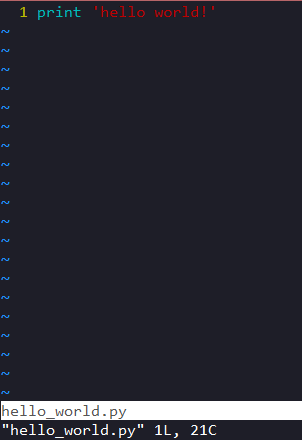
执行以下命令：

$ sudo apt-get install python3

$ sudo rm /usr/local/bin/python

$ sudo ln -s /usr/local/bin/python3.4 /usr/local/bin/python

源文件hello\_world.py



# python hello\_world.py



**5. 进程监管supervisor**

pip安装

$ pip install supervisor

$ cd /home/fanyuguang/Project/

$ mkdir supervisor-conf

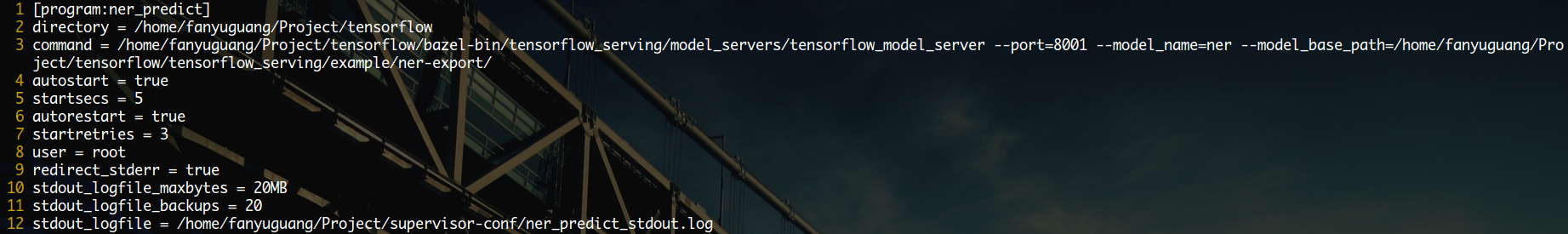
$ cd supervisor-conf

生成默认配置文件

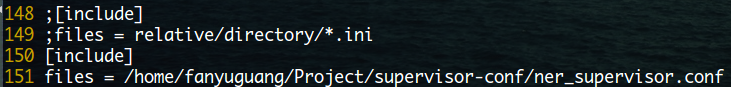
$ echo\_supervisord\_conf > supervisord.conf

创建需要监管的进程的配置文件，并添加该配置文件到默认配置文件中

$ vim ner\_supervisor.conf



$ vim supervisord.conf



启动监管程序，-c指向默认的配置文件

$ supervisord -c /home/fanyuguang/Project/supervisor-conf/supervisord.conf

查看进程状态

$ supervisorctl status

更新配置文件

$ supervisorctl reread

停止进程，其中ner\_predict为ner\_supervisor.conf中定义进程名

$ supervisorctl stop ner\_predict

启动进程

$ supervisorctl start ner\_predict

重启进程

$ supervisorctl restart ner\_predict

**6. 压缩及解压缩**

打包命令：将目录dir1和dir2打包成file.tar.gz

tar czvf file.tar.gz dir1 dir2

解包命令：将file.tar.gz解包至当前目录

tar xzvf file.tar.gz

**7. 编译安装gcc**

**7.1. 安装依赖**

编译安装gmp

$ wget wget ftp://ftp.gnu.org/gnu/gmp/gmp-4.3.2.tar.bz2

$ tar xvf gmp-4.3.2.tar.bz2

$ cd gmp-4.3.2

$ ./configure

$ make

$ make check

$ make install

编译安装mpfr

$ wget http://www.mpfr.org/mpfr-2.4.2/mpfr-2.4.2.tar.bz2

$ tar xvf mpfr-2.4.2.tar.bz2

$ cd mpfr-2.4.2

$ ./configure --with-gmp-include=/usr/local/include --with-gmp-lib=/usr/local/lib

$ make

$ make check

$ make install

编译安装mpc

$ wget http://www.multiprecision.org/mpc/download/mpc-0.8.1.tar.gz

$ tar xvf mpc-0.8.1.tar.gz

$ cd mpc-0.8.1

$ ./configure --with-mpfr-include=/usr/local/include --with-mpfr-lib=/usr/local/lib --with-gmp-include=/usr/local/include --with-gmp-lib=/usr/local/lib

$ make

$ make check

$ make install

**7.2. 编译安装gcc**

$ wget https://ftp.gnu.org/gnu/gcc/gcc-4.8.5/gcc-4.8.5.tar.gz

$ tar zxvf gcc-4.8.5.tar.gz

$ cd gcc-4.8.5

安装依赖，这一步往往失败，因此可按照7.1手动编译安装

$ ./contrib/download\_prerequisites

开始编译

$ mkdir gcc-build

$ cd gcc-build

$ ../configure --enable-checking=release --enable-languages=c,c++ --disable-multilib

$ make

$ sudo make install

更新libstdc++.so.6库

$ rm -rf /usr/lib64/libstdc++.so.6

$ ln -s /usr/local/lib64/libstdc++.so.6.0.19 /usr/lib64/libstdc++.so.6

更新动态库缓存

$ ldconfig

查看gcc 和动态库版本

$ gcc -v

$ g++ -version

$ strings /usr/lib64/libstdc++.so.6 | grep GLIBC

$ strings /lib64/libc.so.6 | grep GLIBC

**7.3. 编译安装gdb**

$ wget http://ftp.gnu.org/gnu/gdb/gdb-7.6.1.tar.gz

$ tar zxvf gdb-7.6.1.tar.gz

$ cd gdb-7.6.1

$ ./configure

$ make

$ sudo make install

$ sudo cp /home/work/fanyuguang/gdb-7.6.1/gdb/gdb /usr/bin/gdb

查看gdb版本

$ gdb -v