



2021年5月19日 21:48

## MIT Mini Cheetah Ubuntu 18.04 环境配置

自己第一次写的比较正式一点的文档吧，有什么笔误，或者技术方面的问题可以提醒我一下，我会注意的。

参考了MIT开源项目的一些网络资料。包括开发环境搭建的步骤，运行、遇到的问题以及解决方法：

[\(5条消息\) MIT Mini Cheetah Ubuntu 18.04 环境配置\\_sinat\\_23084397的博客-CSDN博客](#)

[\(5条消息\) MIT Cheetah编译环境搭建\\_onev2012的博客-CSDN博客](#)

[\(5条消息\) Mit Mini Cheetah开源代码开发环境搭建\\_Galaxy\\_Robot的博客-CSDN博客](#)

[华北舵狗王带你一起做四足机器人9 \(MIT Cheetah 开源 代码 仿真器编译\) - 知乎](#)

[\(zhihu.com\)](#)

首先提醒一下大家它适用于这样的配置：Ubuntu 18.04.4 Eigen 3.3.90 QT 5.14.0 lcm 1.3.1 java 1.8.0。所以首先检查一下自己系统版本和环境版本，我起初用的是Ubuntu20.04，但中间出现了无数个阴间bug，也遇到了不少坑，耗费了我三四天的时间，分享出来，希望对遇到类似问题的朋友有帮助。更多的内容后面持续分享出来。最后不得不放弃了，用18.04以后可以说是无障碍的完成了这个过程。（所以环境版本很重要！）

下面正式的开始。。。

1. 获取安装依赖包：

```
sudo apt install mesa-common-dev freeglut3-dev coinor-libipopt-dev libblas-dev liblapack-dev
gfortran liblapack-dev coinor-libipopt-dev cmake gcc build-essential libglib2.0-dev
```

2. 安装编译lcm1.3.1(it says Java 6, but you can use newer) (<https://lcm-proj.github.io/>)[命令安装](#)

[LCM库](#)

安装git: sudo apt-get install git

克隆lcm: git clone <https://github.com/lcm-proj/lcm.git>

进入lcm目录: cd lcm

在lcm文件夹下建build文件夹: mkdir build

进入build目录: cd build

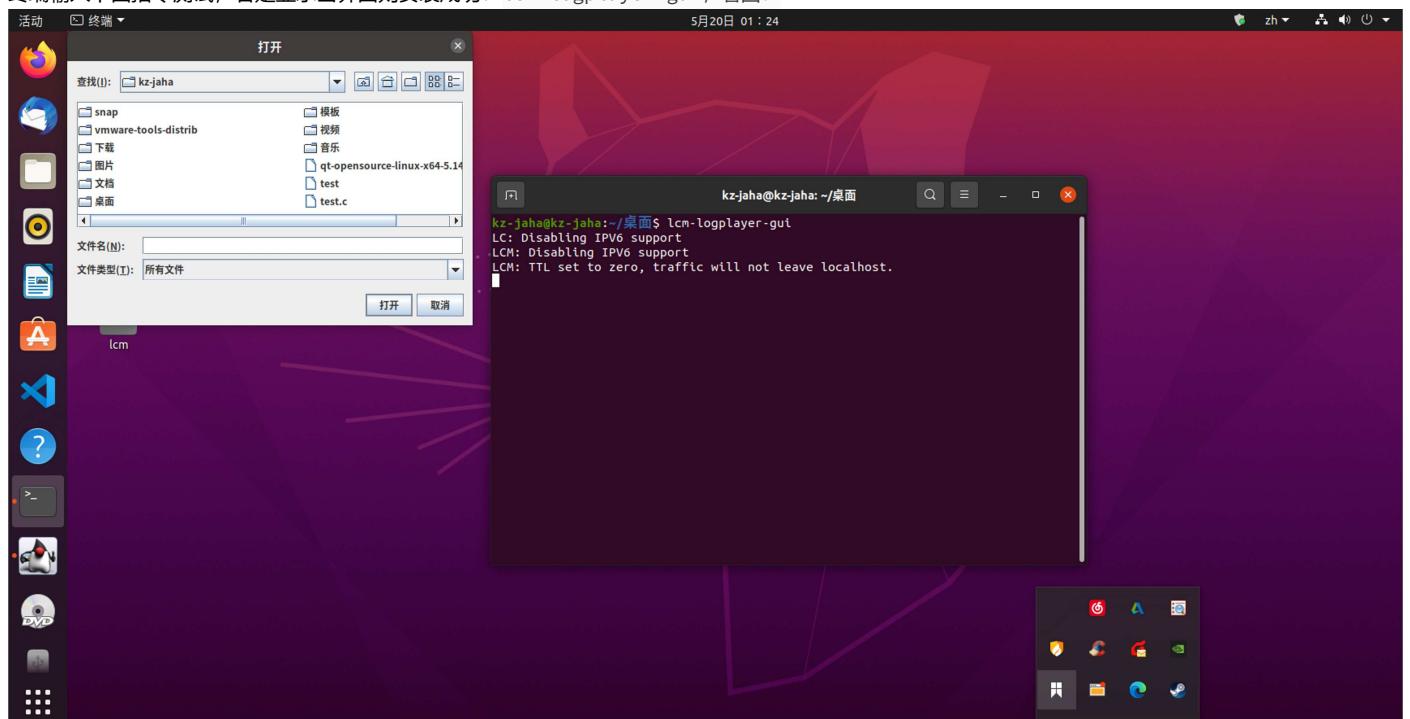
建立: cmake ..

编译: make

编译后的文件安装到系统: sudo make install

这是一个动态链接库管理命令，目的是为了让动态库为系统所共享: sudo ldconfig

终端输入下面指令测试，若是显示出界面则安装成功: lcm-logplayer-gui，看图:



然后测试一下python环境下：

Python3

```
>>>import lcm
```

```
>>>
```

3.Qt 5.10.0 or newer (requires the gamepad library) (<https://www.qt.io/download-qt-installer>) 可以去官网下载Qt5.10.0或更新版本的安装包，我在清华大学开源软件镜像站下载了qt5.14.0安装包进行安装。

比如清华镜像下载完之后进行安装：

```
sudo chmod -R 755 qt-opensource-linux-x64-5.14.0-rc_167.run  
.qt-opensource-linux-x64-5.14.0-rc_167.run
```

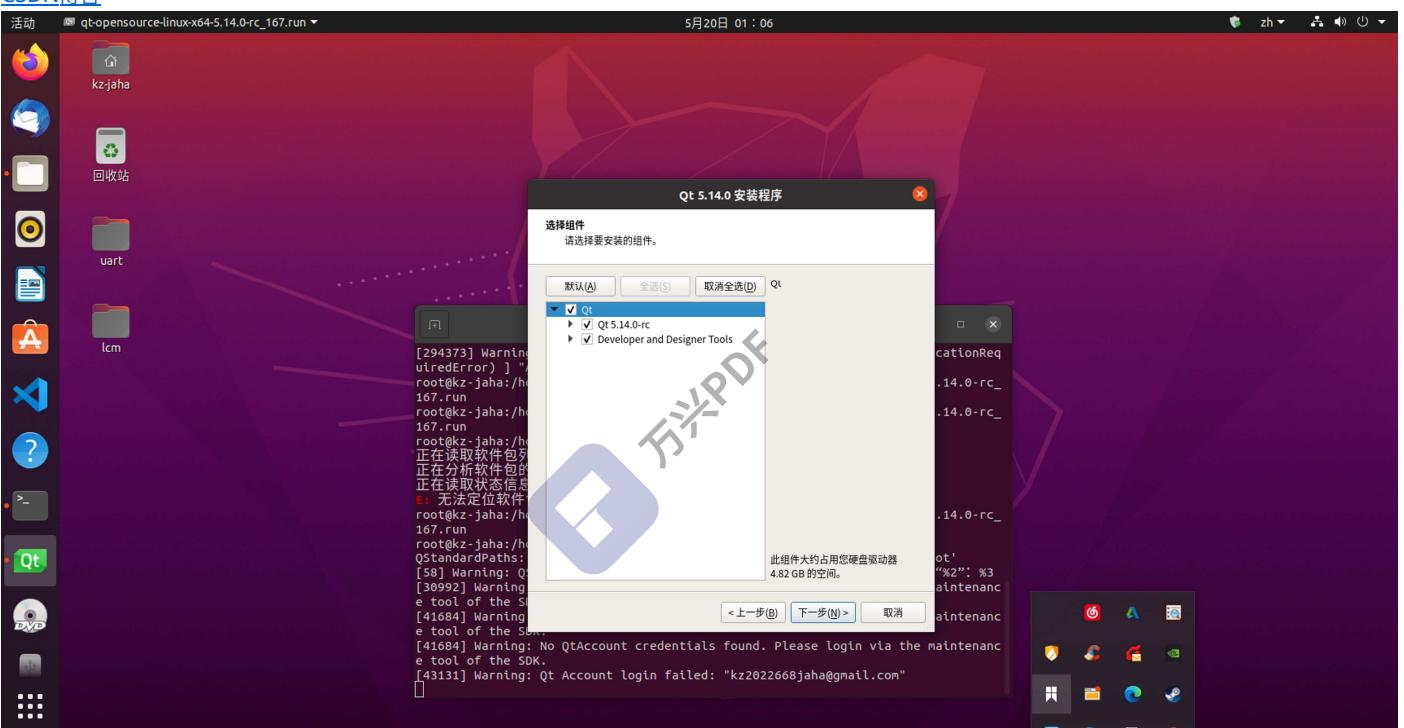
安装详细过程可以看：【Ubuntu使用清华源镜像安装QT-哔哩哔哩】<https://b23.tv/dcgLrM>，

但他讲的是5.15.2版本的，建议安装5.14.0。

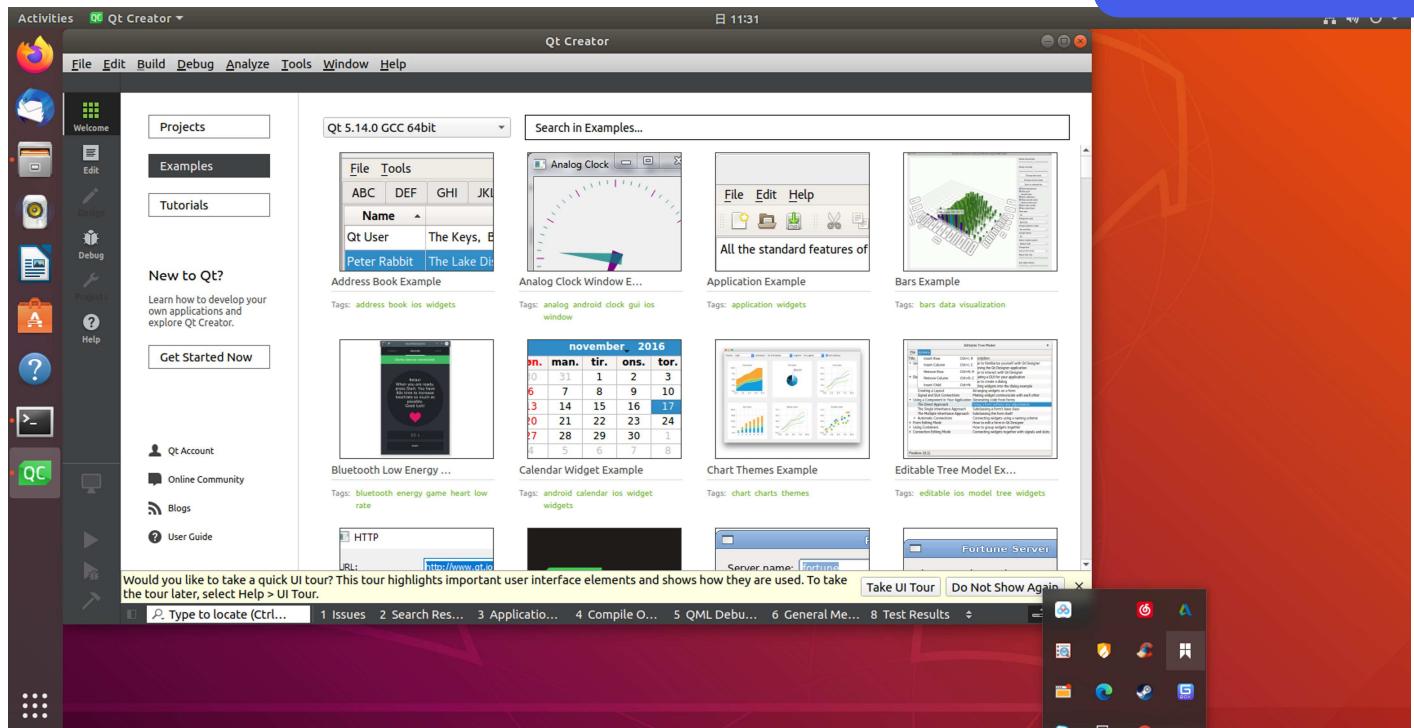
注意：它安装的默认路径是/opt/Qt5.14.0，我们要在/home/{username}/Qt 路径下安装

{username}是自己主机名字），并且别忘了勾选（看图）。如果安装错了，也没事，详细看这个文章([5条消息](#)) Mit Mini Cheetah开源代码开发环境搭建\_Galaxy\_Robot的博客

[CSDN博客](#)



安装陈工会打开一下这个页面：



#### 4. 安装Eigen (<http://eigen.tuxfamily.org/>)，采用指令行安装，eigen3 默认是安装

在/usr/include/下的，但是编译源码的时候是从/usr/local/include/路径下找的，所以要把

eigen3文件夹拷贝过去：

```
sudo apt-get install libeigen3-dev
```

安装eigen3的时候默认安装在/usr/include目录，而编译器编译时是按安装在

/usr/local/include目录进行编译的，然后用下面命令再将eigen3文件夹复制

到/usr/local/include目录下即可。不然的话编译时make会报错，看图。

```
sudo cp -r /usr/include/eigen3 /usr/local/include/eigen3
```

```
File Edit View Search Terminal Help
ndler.dir/ParamHandler.cpp.o
[ 12%] Linking CXX shared library libdynacore_param_handler.so
[ 12%] Built target dynacore_param_handler
[ 13%] Built target inih
[ 13%] Built target qdldlobj
[ 13%] Built target linsys_pardiso
[ 16%] Built target linsys_qdldl
[ 19%] Built target osqp
Scanning dependencies of target biomimetics
[ 20%] Building CXX object common/CMakeFiles/biomimetics.dir/src/Collision/CollisionBox.cpp.o
cc1plus: error: /usr/local/include/eigen3: No such file or directory [-Werror=missing-includes]
cc1plus: all warnings being treated as errors
common/CMakeFiles/biomimetics.dir/build.make:62: recipe for target 'common/CMakeFiles/biomimetics.dir/src/Collision/CollisionBox.cpp.o' failed
make[2]: *** [common/CMakeFiles/biomimetics.dir/src/Collision/CollisionBox.cpp.o] Error 1
CMakeFiles/Makefile2:1841: recipe for target 'common/CMakeFiles/biomimetics.dir/all' failed
make[1]: *** [common/CMakeFiles/biomimetics.dir/all] Error 2
Makefile:129: recipe for target 'all' failed
make: *** [all] Error 2
https://blog.csdn.net/Galaxy_Robot
```

#### 5. 下载，建立工程和编译minicheetah源码（在开始第一步已经获取了安装依赖包）

下载源码：git clone <https://github.com/mit-biomimetics/Cheetah-Software.git>

在用git下载源码时出现了这样错误，看图：

```
[sudo] kz-jaha@kz-jaha:~/桌面
kz-jaha@kz-jaha:~/桌面$ git clone https://github.com/mit-biomimetics/Cheetah-Software.git
正克隆到 'Cheetah-Software'...
fatal: 无法访问 'https://github.com/mit-biomimetics/Cheetah-Software.git/': Fail
ed to connect to github.com port 443: 拒绝连接
kz-jaha@kz-jaha:~/桌面$ nev|grep -i proxy

Command 'nev' not found, did you mean:

  command 'naev' from deb naev (0.7.0-3build1)
  command 'env' from deb coreutils (8.30-3ubuntu2)
  command 'rev' from deb util-linux (2.34-0.1ubuntu9.1)
  command 'net' from deb samba-common-bin (2:4.11.6+dfsg-0ubuntu1.8)
  command 'nes' from deb fceux (2.2.2+dfsg0-1build1)
  command 'nes' from deb mednafen (1.22.2+dfsg-1build1)
  command 'nes' from deb nestopia (1.50-1build1)
  command 'nex' from deb nvi (1.81.6-15build1)
  command 'new' from deb mmh (0.4-2)
  command 'new' from deb nmh (1.7.1-6)
  command 'ner' from deb frog (0.15-1build2)
  command 'xev' from deb x11-utils (7.7+5)
  command 'mev' from deb gpm (1.20.7-5)
  command 'ne' from deb ne (3.1.2-1)
```

原因不太清楚，论坛上大佬说网络被代理了，但问题还是用下面命令解决了，看图：

```
kz-jaha@kz-jaha:~/桌面
kz-jaha@kz-jaha:~/桌面$ env|grep -i proxy
kz-jaha@kz-jaha:~/桌面$ git clone https://github.com/mit-biomimetics/Cheetah-Software.git
正克隆到 'Cheetah-Software'...
remote: Enumerating objects: 1282, done.
remote: Total 1282 (delta 0), reused 0 (delta 0), pack-reused 1282
接收对象中: 100% (1282/1282), 4.50 MiB | 3.01 MiB/s, 完成。
处理 delta 中: 100% (215/215), 完成。
kz-jaha@kz-jaha:~/桌面$
```

接下来就是建立和编译：

cd Cheetah-Software

进入这个文件夹： cd scripts

你可能会看到一个错误，像"rm: cannot remove..."但这没问题： ./make\_types.sh

cd ..

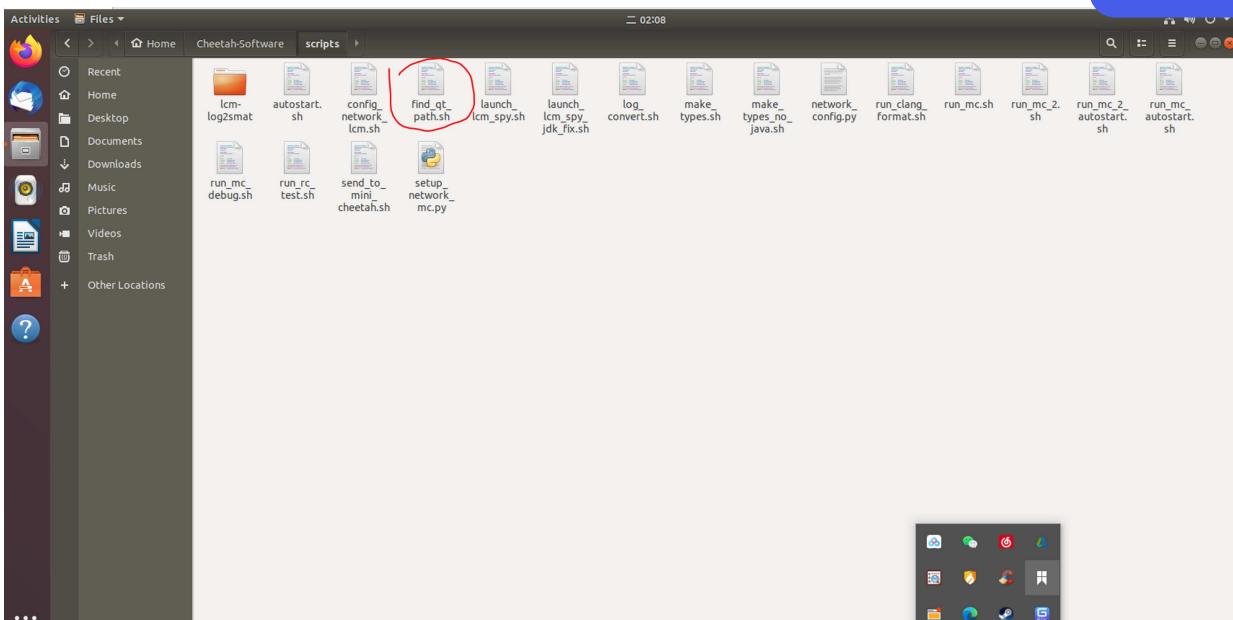
mkdir build

cd build

这里仍然有一些警告，可以按提示去消除警告，也可以不用： cmake ..

建立的时候最常见的错误应该是Qt报的错误，将/Cheetah-Software/scripts/find\_qt\_path.sh里面

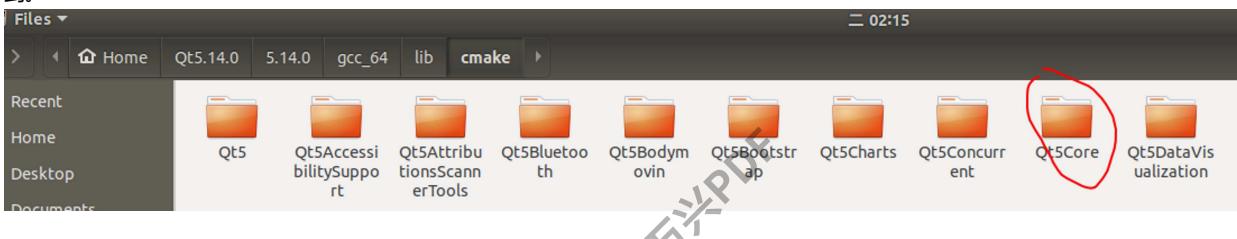
的相应路径和文件名改成你安装Qt的路径和文件名即可：



find.Qt\_path.sh里面只有三行代码,里面原来的内容是:

```
#!/bin/bash
QT_VER=$(ls ~/Qt/ | grep 5 -m1)
printf "${HOME}/Qt/${QT_VER}/gcc_64/"
```

下面第二个图是我填的安装Qt的位置,严格来说它应该需要这个文件,报错的时候可以看到。



```
#!/bin/bash
QT_VER=$(ls ~/Qt5.14.0 | grep 5 -m1)
printf "${HOME}/Qt5.14.0/${QT_VER}/gcc_64/"
```

然后就是建立成功:

```

book@100ask:~/nfs_rootfs/lib/modules$ cd 5.4.31-gf4b262578
book@100ask:~/nfs_rootfs/lib/modules/5.4.31-gf4b262578$ rm build source
book@100ask:~/nfs_rootfs/lib/modules/5.4.31-gf4b262578$ ls -la
drwxrwxr-x 3 book book 4096 May 19 07:55 .
drwxrwxr-x 3 book book 4096 May 19 07:49 ..
drwxrwxr-x 9 book book 4096 May 19 07:50 kernel
-rw-r--r-- 1 book book 159541 May 19 07:50 modules.alias
-rw-r--r-- 1 book book 171444 May 19 07:50 modules.alias.bin
-rw-rw-r-- 1 book book 19869 May 19 07:49 modules.builtin
-rw-r--r-- 1 book book 22616 May 19 07:50 modules.builtin.bin
-rw-rw-r-- 1 book book 105715 May 19 07:49 modules.builtin.modinfo
-rw-r--r-- 1 book book 47108 May 19 07:50 modules.dep
-rw-r--r-- 1 book book 70328 May 19 07:50 modules.dep.bin
-rw-r--r-- 1 book book 94 May 19 07:50 modules.devname
-rw-rw-r-- 1 book book 20715 May 19 07:49 modules.order
-rw-r--r-- 1 book book 85 May 19 07:50 modules.softdep
-rw-r--r-- 1 book book 71774 May 19 07:50 modules.symbols
-rw-r--r-- 1 book book 89075 May 19 07:50 modules.symbols.bin
book@100ask:~/nfs_rootfs/lib/modules/5.4.31-gf4b262578$ 

```

最激动人心的环节来了，编译：make -j4

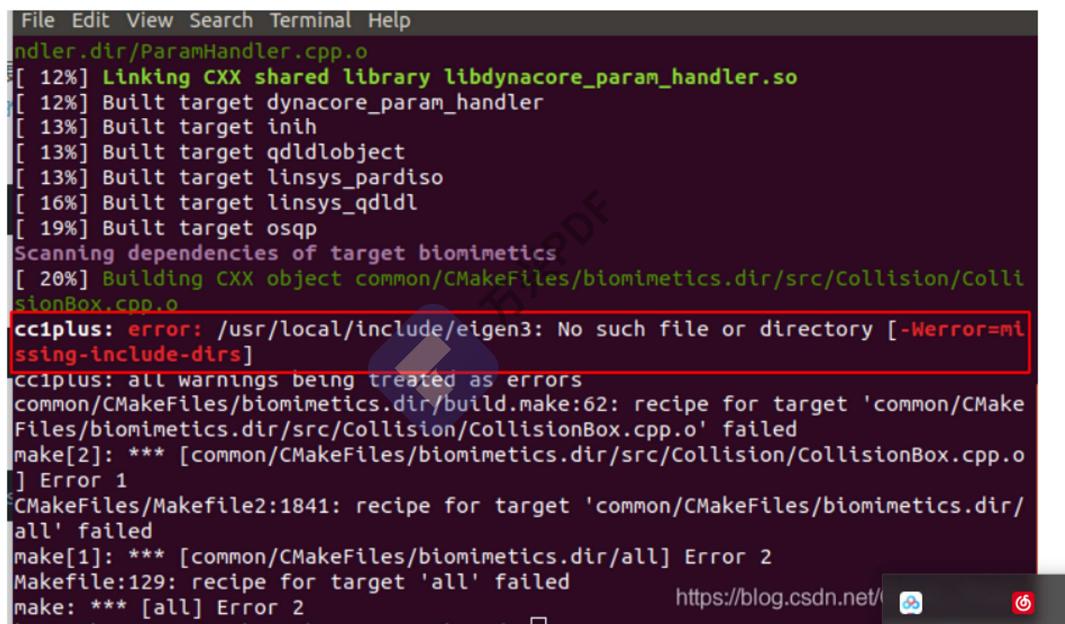
这时候如果你按上面步骤做的对的话，或者你电脑系统是Ubuntu18.04的话，恭喜你已经成功

80%了，基本上会出现下面两个问题：

## make错误及解决方法

执行 make -j4可能会出现如下错误，按照对应方法处理即可：

(1)cc1plus: error: /usr/local/include/eigen3: No such file or directory [-Werror=missing-include-dirs]



```

File Edit View Search Terminal Help
ndler.dir/ParamHandler.cpp.o
[ 12%] Linking CXX shared library libdynacore_param_handler.so
[ 12%] Built target dynacore_param_handler
[ 13%] Built target inih
[ 13%] Built target qdldlobject
[ 13%] Built target linsys_pardiso
[ 16%] Built target linsys_qdldl
[ 19%] Built target osqp
Scanning dependencies of target biomimetics
[ 20%] Building CXX object common/CMakeFiles/biomimetics.dir/src/Collision/CollisionBox.cpp.o
cc1plus: error: /usr/local/include/eigen3: No such file or directory [-Werror=missing-include-dirs]
cc1plus: all warnings being treated as errors
common/CMakeFiles/biomimetics.dir/build.make:62: recipe for target 'common/CMakeFiles/biomimetics.dir/src/Collision/CollisionBox.cpp.o' failed
make[2]: *** [common/CMakeFiles/biomimetics.dir/src/Collision/CollisionBox.cpp.o] Error 1
CMakeFiles/Makefile2:1841: recipe for target 'common/CMakeFiles/biomimetics.dir/all' failed
make[1]: *** [common/CMakeFiles/biomimetics.dir/all] Error 2
Makefile:129: recipe for target 'all' failed
make: *** [all] Error 2

```

这是由于安装eigen3的时候默认安装在/usr/include目录，而编译起器以为安装在 /usr/local/include目录，可以把eigen3文件夹直接复制到/usr/local/include目录下。

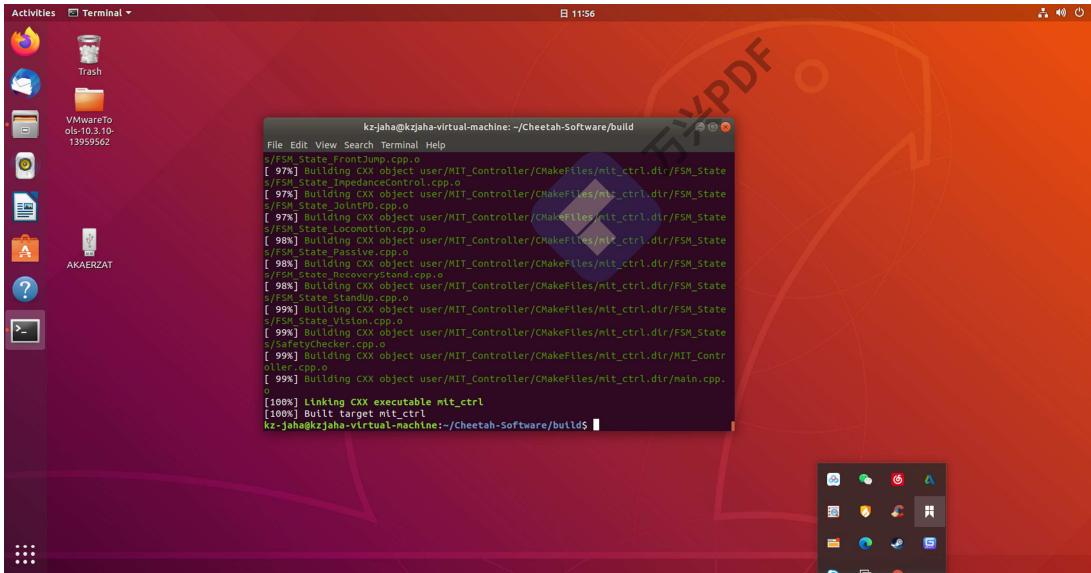
其实如果你按我的步骤来，不会出现上面这种错误因为这个我们已经在安装eigen3的时候做了，所以放心。

c++: internal compiler error: Killed (program cc1plus)

```
brian@brian-pc: ~/Cheetah-Software/build
File Edit View Search Terminal Help
See <file:///usr/share/doc/gcc-7/README.Bugs> for instructions.
third-party/JCQP/CMakeFiles/JCQP.dir/build.make:86: recipe for target 'third-par
ty/JCQP/CMakeFiles/JCQP.dir/ProblemGenerator.cpp.o' failed
make[2]: *** [third-party/JCQP/CMakeFiles/JCQP.dir/ProblemGenerator.cpp.o] Error
4
make[2]: *** Waiting for unfinished jobs....
[ 35%] Building CXX object user/MIT_Controller/Controllers/WBC/WBIC/CMakeFiles/W
BIC.dir/WBIC.cpp.o
c++: internal compiler error: Killed (program cc1plus)
Please submit a full bug report,
with preprocessed source if appropriate.
See <file:///usr/share/doc/gcc-7/README.Bugs> for instructions.
third-party/JCQP/CMakeFiles/JCQP.dir/build.make:110: recipe for target 'third-pa
rty/JCQP/CMakeFiles/JCQP.dir/eigenvalues.cpp.o' failed
make[2]: *** [third-party/JCQP/CMakeFiles/JCQP.dir/eigenvalues.cpp.o] Error 4
CMakeFiles/Makefile2:953: recipe for target 'third-party/JCQP/CMakeFiles/JCQP.di
r/all' failed
make[1]: *** [third-party/JCQP/CMakeFiles/JCQP.dir/all] Error 2
make[1]: *** Waiting for unfinished jobs....
c++: internal compiler error: Killed (program cc1plus)
Please submit a full bug report,
with preprocessed source if appropriate.
See <file:///usr/share/doc/gcc-7/README.Bugs> for instructions.
https://blog.csdn.net/Galaxy\_Robot
user/MIT_Controller/Controllers/WBC/WBIC/CMakeFiles/WBIC.dir/build.make:62: reci
```

我在第一步强调过了把虚拟机内存给大一点。因为这是由于内存不足引起的。第一个解决办法是把虚拟机内存设置更大(把虚拟机内存设置为8GB就很轻松编译完成了)。第二个办法是直接用make指令编译，去掉-j4选项，j4是指定4个线程进行编译，这样只是编译时间会很久。

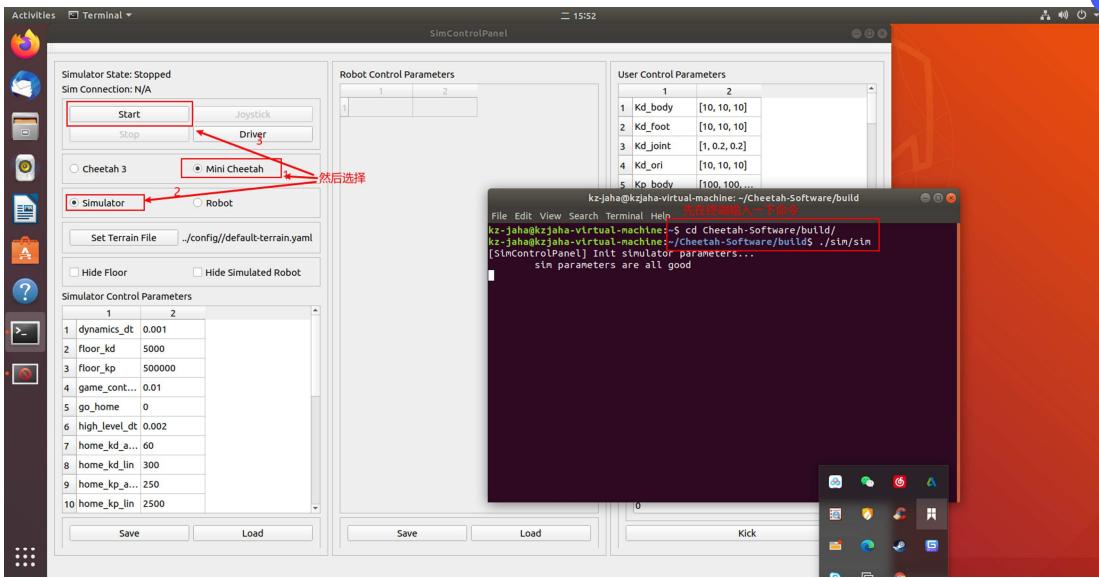
然后就是编译成功 (是不是很开心！) :



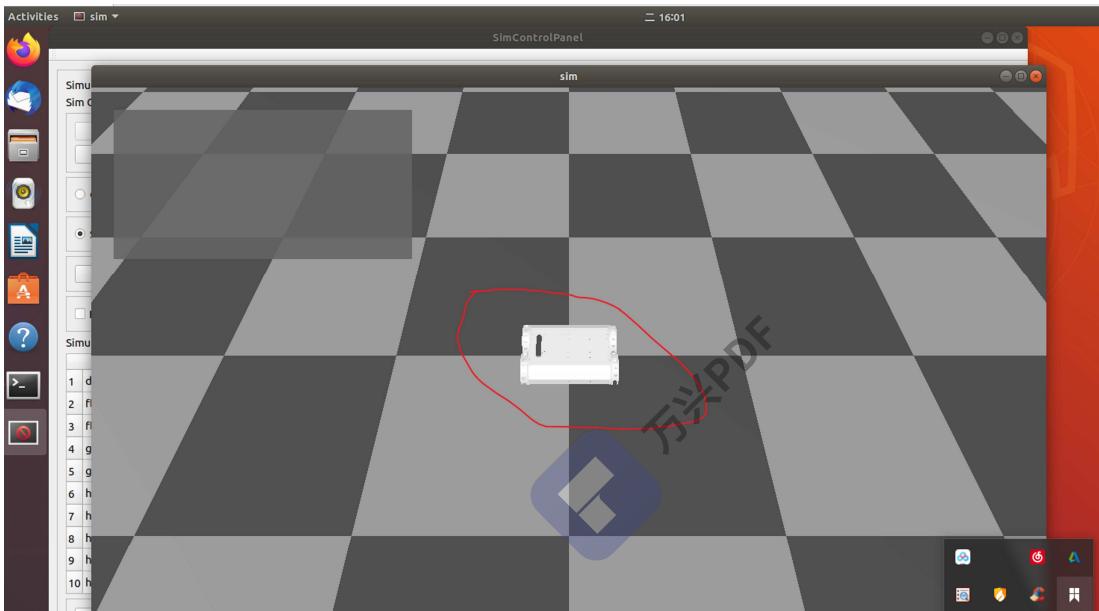
## 6.下面是运行仿真环境：

```
cd Cheetah-Software/build/
./sim/sim
```

下面弹出一个Sim 控制面板，选自一下，最后点击start。



点击start以后，出现下面这个页面，这时的狗就是还没起来，像小盒子一样在哪儿躺着。



在mini cheetah控制台,要把cheater\_mode 和 control\_mode 设为1,user\_rc设为0,不然会失败.

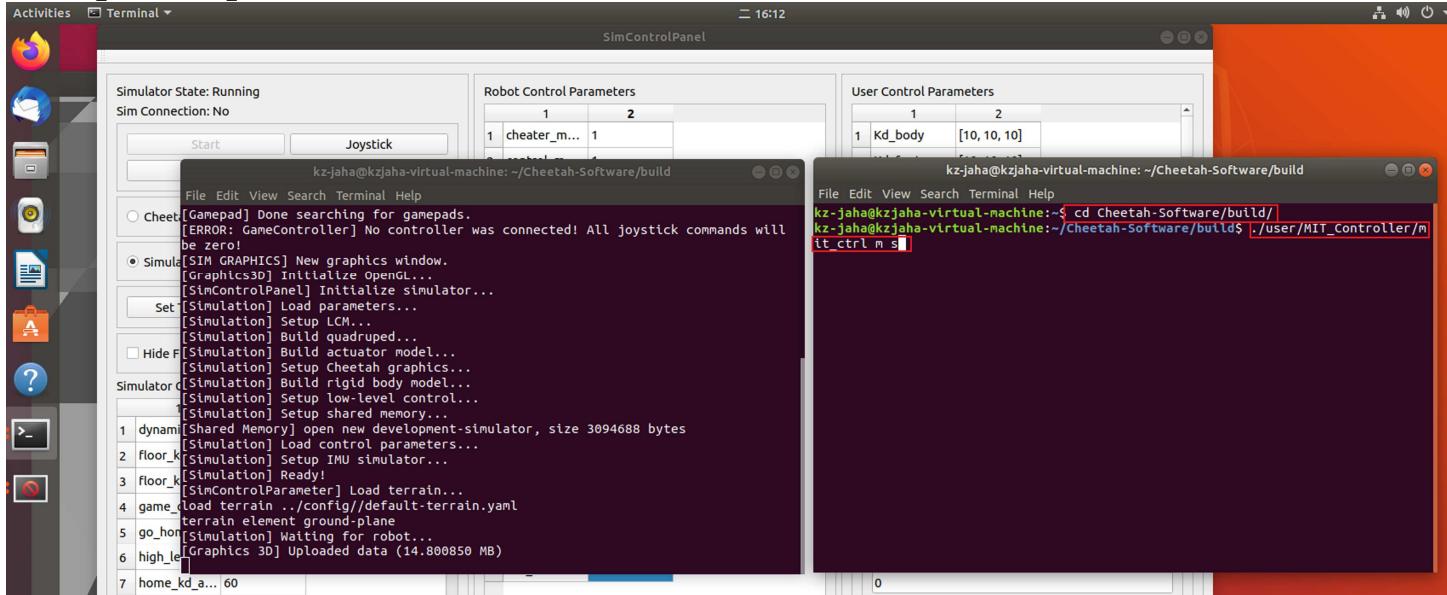
改的以后的参数		默认状态下的参数	
Robot Control Parameters		Robot Control Parameters	
1	cheater_m...	1	0
2	control_m...	1	0
3	controller_dt	0.002	0.002
4	foot_height...	0.001	0.001
5	foot_proce...	0.002	0.002
6	foot_senso...	0.001	0.001
7	foot_senso...	0.1	0.1
8	imu_proce...	0.02	0.02
9	imu_proce...	0.02	0.02
10	kdBase	[20, 10, 10]	[20, 10, 10]
11	kdCOM	[10, 10, 10]	[10, 10, 10]
12	kpBase	[300, 200, ...]	[300, 200, ...]
13	kpCOM	[50, 50, 50]	[50, 50, 50]
14	myValue	2222	2222
15	stand_kd_c...	[2.5, 2.5, 2.5]	[2.5, 2.5, 2.5]
16	stand_kp_c...	[50, 50, 50]	[50, 50, 50]
17	testValue	23	23
18	use_rc	0	1

重新打开一个终端，启动一个控制台

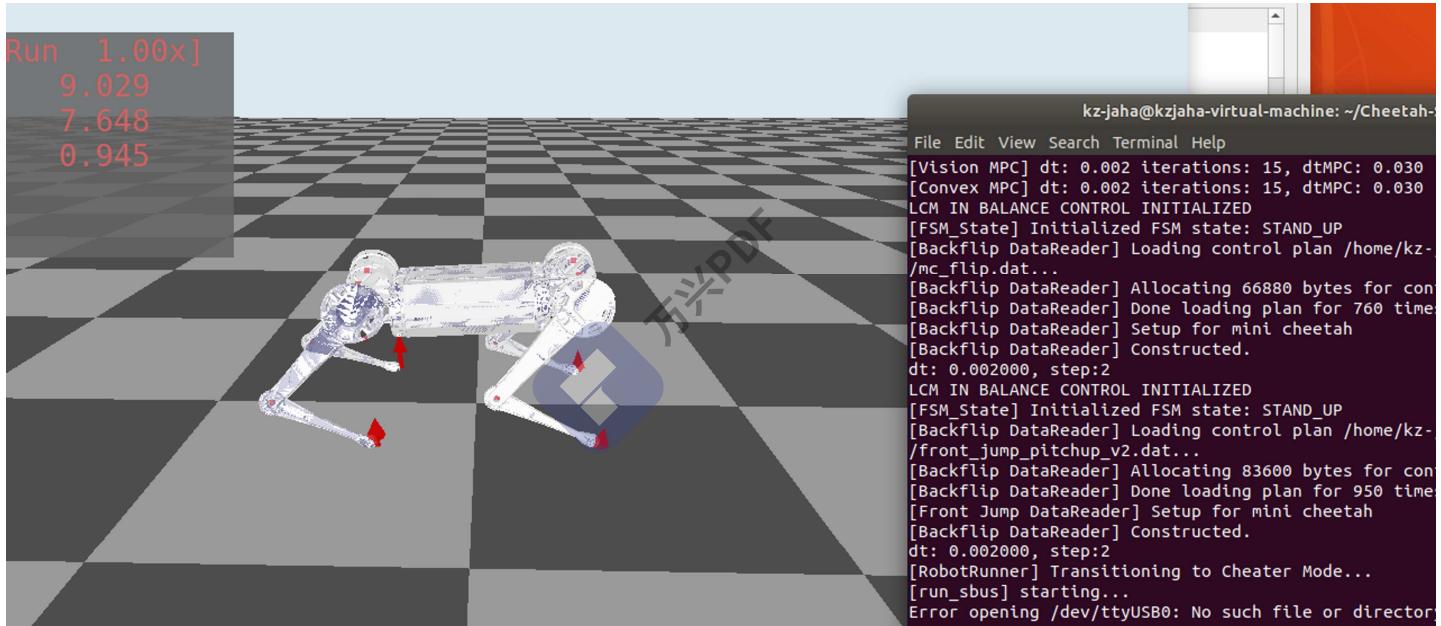
./user/\${controller\_folder}/\${controller\_name} \${robot\_name} \${target\_system}

例如，启动mini cheetah，可以输入

./user/MIT\_Controller/mit\_ctrl m s



回车发现他站了起来。



1. 将 control\_mode 改为10 (JOINT\_PD), 机器人电机会上电切处于准备站立的姿态
2. 将 control\_mode 改为1 (STAND\_UP), Mini Cheetah会站立，而Cheetah 3 有Bug不会动
3. 将 controlmode 改为4 (LOCOMOTION), 上述的MPC控制器和阻抗控制器会开启，默认进行以Trot 步态运行，此时可以用手柄操作
4. 可以通过修改 cmcp\_gait 来改变步态
5. 可以通过改变cheater\_mode(0/1)来开关状态估计的结果的显示
6. 机械狗还有个后空翻模式,修改control\_mode为6,机械狗会停下,再修改control值为9,机械狗会趴下,然后来个后空翻,帅爆了.

手柄链接：遥控器后部在X档，同时前面MODE灯为灭，连接成功之后重新做上面第六运行仿真环境步骤就可以了。

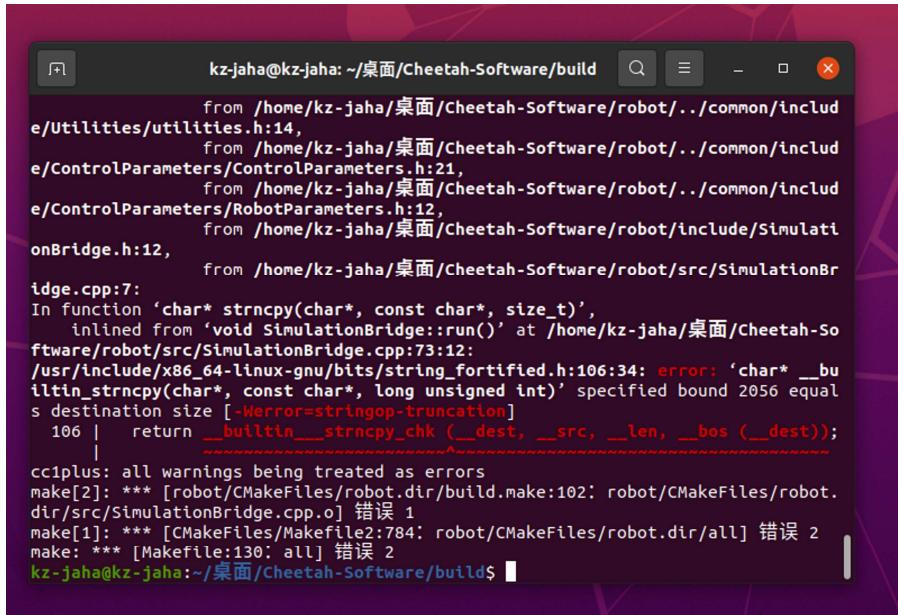
6.如果你系统不是Ubuntu18.04，像我这样用20.04的话，那只能看你的运气了，我一开始用20.04，但弄了四天还是出现各种阴间bug，试了无数种办法但最终还是放弃了（可能时因为我是刚接触Linux的菜鸟）。用18.04按照步骤来可以说是轻松完成（愉快！），但我还是想把这次碰运气时遇到的问题放在这里，如果有大佬用20.04成功了，请别忘了写一篇文章（恳求！）

主要是下面这个错误：

**stropts.h : No such file or directoryH: 没有这样的文件或目录**

我狂了各大论坛，找了各种方法来试了试运气但问题没有解决，有人说这不是error，是warning，make把警告看成错误，他给我推荐的解决方法是在对应的源文件的makefile里面找"-Werror"这个语句（标志），然后把它删掉或注释掉就可以了，但我再对应的makefile文件里没找到这种标志，而且还试了其他的一些方法，具体我也没记下来，可以Google一下，发现里面很多人遇到了这个问题，可以去试一下。

第二个问题是下面这个错误，学长说这可能是gcc的版本而导致的问题，我的是gcc9，后来我用gcc7编译的时候这错误消失了，但出现了其他一些错误，反正各种阴间bug。



```
kz-jaha@kz-jaha: ~/桌面/Cheetah-Software/build
从 /home/kz-jaha/桌面/Cheetah-Software/robot/../common/include/Utilities/utilities.h:14,
      from /home/kz-jaha/桌面/Cheetah-Software/robot/../common/include/ControlParameters/ControlParameters.h:21,
      from /home/kz-jaha/桌面/Cheetah-Software/robot/../common/include/ControlParameters/RobotParameters.h:12,
      from /home/kz-jaha/桌面/Cheetah-Software/robot/include/SimulationBridge.h:12,
      from /home/kz-jaha/桌面/Cheetah-Software/robot/src/SimulationBridge.cpp:7:
In function 'char* strncpy(char*, const char*, size_t)',
  inlined from 'void SimulationBridge::run()' at /home/kz-jaha/桌面/Cheetah-Software/robot/src/SimulationBridge.cpp:73:12:
/usr/include/x86_64-linux-gnu/bits/string_fortified.h:106:34: error: 'char* __builtin_strncpy(char*, const char*, long unsigned int)' specified bound 2056 equals destination size [-Winvalid-stringop-truncation]
  106 |   return __builtin_strncpy_chk (__dest, __src, __len, __bos (__dest));
          |                                         ^
cc1plus: all warnings being treated as errors
make[2]: *** [robot/CMakeFiles/robot.dir/build.make:102: robot/CMakeFiles/robot.dir/src/SimulationBridge.cpp.o] 错误 1
make[1]: *** [CMakeFiles/Makefile2:784: robot/CMakeFiles/robot.dir/all] 错误 2
make: *** [Makefile:130: all] 错误 2
kz-jaha@kz-jaha:~/桌面/Cheetah-Software/build$
```

还有一些问题我没记下来，后续找到了截图和笔记还会补上的。最后Good luck!