使用WSL2安装Ubuntu-18.04，环境配置，并复现项目Zyhlibrary/FAEP

---

1. 安装WSL与Ubuntu-20.04

可以参考官方文档：适用于 Linux 的 Windows 子系统文档 | Microsoft Learn

如果是第一次安装WSL，我们可以直接在在Powershell中执行命令来完成安装

系统中自带的Powershell是旧版，可以参考PowerShell 文档 - PowerShell | Microsoft Learn来安装新版Powershell~~（装不装都行，只是提一嘴）~~

注意，在安装Linux发行版前，最好关闭可能修改hosts文件的软件，比如Watt Toolkit的Hosts加速

因为这可能会让你的新系统的hosts文件变成一坨，如果这已经发生了，那么修改hosts文件就好

```
sudo nano /etc/hosts
```

1. 使用管理员身份运行Powershell
2. 执行命令

   ```
   # 安装WSL，但不安装Linux发行版
   wsl --install --no-distribution
   ```

3. 重启电脑，再次使用管理员身份运行Powershell
4. 执行命令

   ```
   # 安装Ubuntu-20.04
   wsl --install Ubuntu-20.04
   ```

5. 等待一段时间，安装完成后会自己启动WSL，需要新建用户名和密码

   注意：输入密码时为盲人键入，光标会保持在原处，正常键入密码即可

6. 关闭Power shell，可以直接搜索Ubuntu来打开其终端，使用Powershell可能会出现一些问题

   微软官方推荐使用Windows Terminal，Terminal的安装参见：Windows 终端概述 | Microsoft Learn

如果想要卸载Linux发行版，命令如下：

```
# 查看已安装的发行版信息
wsl --list --verbose
# 以Ubuntu-20.04为例
wsl --unregister Ubuntu-20.04
```

如果忘记了用户密码可以参照：技术|在 WSL 上忘记了 Linux 密码？下面是如何轻松重设的方法

解决：“wsl: 检测到 localhost 代理配置，但未镜像到 WSL。NAT 模式下的 WSL 不支持 localhost 代理”

如果想要安装其他Linux发行版系统，可以查看可下载的系统

```
# 查看可用发行版列表
wsl --list --online
```

2. 修改软件包源并安装ROS

1. 使用nano打开软件源文件，删除所有原有官方源

```
sudo nano /etc/apt/sources.list
```

2. 粘贴以下内容，注意，以下内容仅适用于Ubuntu-20.04

   如果需要其他版本系统的软件包镜像源，具体可查看ubuntu | 镜像站使用帮助 | 清华大学开源软件镜像站 | Tsinghua Open Source Mirror

```
# 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-backports main restricted universe multiverse

# 以下安全更新软件源包含了官方源与镜像站配置，如有需要可自行修改注释切换
deb http://security.ubuntu.com/ubuntu/ focal-security main restricted universe multiverse
# deb-src http://security.ubuntu.com/ubuntu/ focal-security main restricted universe multiverse

# 预发布软件源，不建议启用
# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-proposed main restricted universe multiverse
# # deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-proposed main restricted universe multiverse
```

   Ctrl+O并回车保存，Ctrl+X退出编辑器

3. 更新软件包目录及软件包

```
sudo apt update
sudo apt upgrade -y
```

4. 添加ROS安装源及配置密钥

```
sudo sh -c '. /etc/lsb-release && echo "deb http://mirrors.tuna.tsinghua.edu.cn/ros/ubuntu/ `lsb_release -cs` main" > /etc/apt/sources.list.d/ros-latest.list'
sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
sudo apt update
```

```
# 如果装不上密钥，可以尝试
wget http://packages.ros.org/ros.key -O - | sudo apt-key add -
```

ROS源的添加参考了：[ros | 镜像站使用帮助 | 清华大学开源软件镜像站 | Tsinghua Open Source Mirror]

5. 安装ROS并配置环境变量

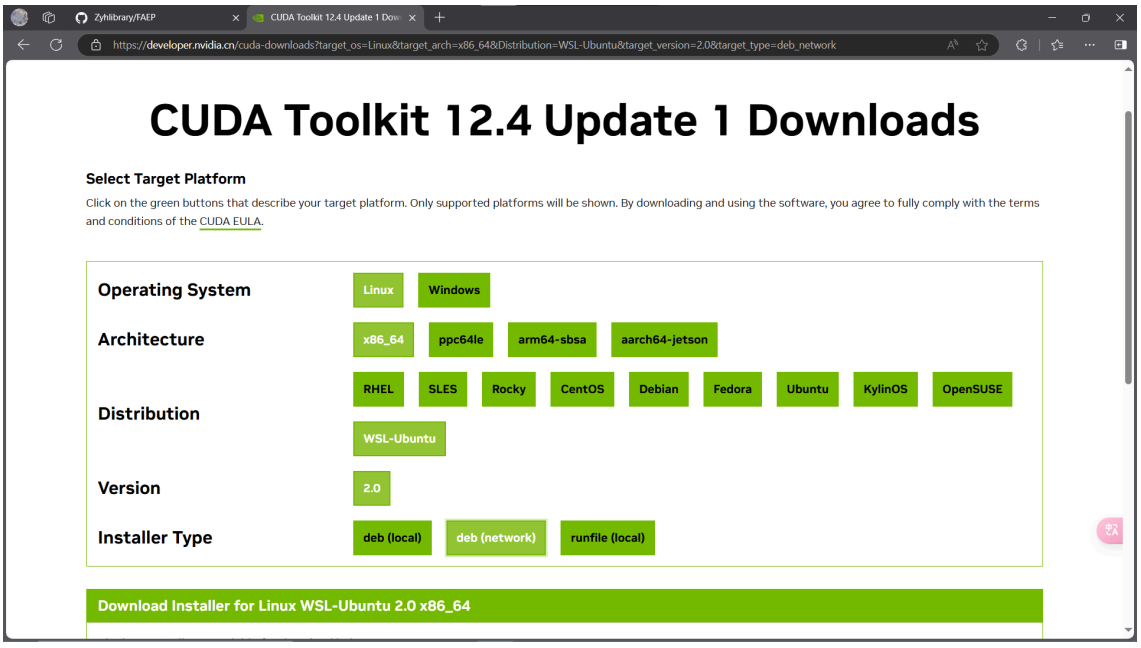
```
sudo apt install ros-noetic-desktop-full
echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

3. 安装CUDA Toolkit

1. 安装Toolkit

   CUDA Toolkit 12.4 Update 1 Downloads | NVIDIA 开发者

   做以下选择，可以得到安装命令，当然，可以选择其他cuda版本的Toolkit



   命令如下，一条一条执行即可：

```
wget https://developer.download.nvidia.com/compute/cuda/repos/wsl-ubuntu/x86_64/cuda-keyring_1.1-1_all.deb
sudo dpkg -i cuda-keyring_1.1-1_all.deb
sudo apt-get update
sudo apt-get -y install cuda-toolkit-12-4
```

2. 配置环境变量并启用

```
echo "export CUDA_HOME=/usr/local/cuda" >> ~/.bashrc
echo "export PATH=\$PATH:\$CUDA_HOME/bin" >> ~/.bashrc
echo "export LD_LIBRARY_PATH=/usr/local/cuda-12.4/lib64\${LD_LIBRARY_PATH:+:\${LD_LIBRARY_PATH}}" >> ~/.bashrc
source ~/.bashrc
```

3. 启用OpenGL硬件渲染

   更换为Ubuntu20.04的原因之一就在于此，Ubuntu18.04的Mesa版本为20.x，这不支持OpenGL硬件渲染，而使用llvm渲染会使得仿真过程很卡顿。

   当然，你可以选择继续使用Ubuntu18.04，并且如果想要更新Mesa，可以参照：如何在 Ubuntu 上安装 Mesa 驱动程序

```
sudo apt install mesa-utils
glxinfo | grep OpenGL
```

   从返回的信息我们可以看到OpenGL 渲染器使用了显卡

如果你有多张显卡或者被核显困扰，想要更改使用的显卡，可以参照：WSLg 中的 GPU 选择 · 微软/wslg Wiki

4. 复现FUEL

[HKUST-Aerial-Robotics/FUEL: An Efficient Framework for Fast UAV Exploration]

1. 安装依赖

```
git clone -b v2.7.1 https://github.com/stevengj/nlopt.git
cd nlopt
mkdir build
cd build
cmake ..
make
sudo make install
sudo apt-get install libarmadillo-dev
```

2. 编译FUEL包

   回到用户目录下，新建工作空间

```
cd ~
mkdir -p fuel_ws/src
cd fuel_ws/src
```

   克隆并修改包

```
git clone https://github.com/HKUST-Aerial-Robotics/FUEL.git
```

   修改文件，在 `uav_simulator/local_sensing/CMakelist.txt` 中

   如果你的显卡是10系，其架构是Pascal，其算力值为sm_61

   如果你的显卡是20系，其架构是Turing，其算力值为sm_75

   如果你的显卡是30系，其架构是Ampere，其算力值为sm_86

   如果你的显卡是40系，其架构是Ada，其算力值为sm_89

   详见：[CUDA GPUs - Compute Capability | NVIDIA Developer]

```
  set(CUDA_NVCC_FLAGS
#       -gencode arch=compute_20,code=sm_20;
#       -gencode arch=compute_20,code=sm_21;
#       -gencode arch=compute_30,code=sm_30;
#       -gencode arch=compute_35,code=sm_35;
#       -gencode arch=compute_50,code=sm_50;
#       -gencode arch=compute_52,code=sm_52;
#       -gencode arch=compute_60,code=sm_60;
#       -gencode arch=compute_75,code=sm_75;
      -gencode arch=compute_86,code=sm_86;
#       -gencode arch=compute_89,code=sm_89;
  )
```

   回到工作空间下，编译包

   部分可能会出现Anaconda的环境变量导致编译失败的，可以删除Win中的Anaconda环境变量

```
cd ..
catkin_make
```

3. 运行仿真示例

   在工作空间下执行命令

```
source devel/setup.bash && roslaunch exploration_manager rviz.launch
```

   新终端，同样在工作空间下执行

```
source devel/setup.bash && roslaunch exploration_manager
exploration.launch
```

   使用 `2D Nav Goal` 触发仿真，在使用硬件加速后明显没有那么卡顿了

4. 选择不同的探索环境

   在`FUEL/fuel_planner/exploration_manager/launch/simulator.xml`中第22行

```
  <node pkg ="map_generator" name ="map_pub" type ="map_pub" output =
"screen" args="$(find map_generator)/resource/office.pcd"/>
```

   可用的环境在`FUEL/uav_simulator/map_generator/resource/`下

5. 复现FAEP

1. 安装依赖

```
sudo apt-get install libdw-dev
```

2. 创建工作空间

```
mkdir -p faep_ws/src
cd faep_ws/src
```

3. 克隆并修改包

```
git clone https://github.com/Zyhlibrary/FAEP.git
```

同样修改 `uav_simulator/local_sensing/CMakelist.txt` 文件

然后，步骤变多了，不知道为什么这个傻鸟作者好好的把FUEL中的配置给删了

使用FUEL对应位置的文件内容来替换 `"FAEP/faep_planner/bspline_opt/CMakeLists.txt"` 中的内容

在 `faep_ws/src/` 下，修改 `CMakelist.txt` 文件

这个是为了改起来方便，因为这个傻鸟把所有 `CMakelist.txt` 文件里的C++14都删了，懒得一个一个改

```
sudo nano CMakeLists.txt
```

在其中添加使用C++14

```
set(CMAKE_CXX_STANDARD 14)
```

4. 回到工作空间 `~/faep_ws` 下，编译包

部分可能会出现Anaconda的环境变量导致编译失败的，可以删除Win中的Anaconda环境变量

```
cd ..
catkin_make
```

5. 运行仿真示例

在工作空间下执行命令

```
source devel/setup.bash && roslaunch exploration_manager rviz.launch
```

新终端，同样在工作空间下执行

```
source devel/setup.bash && roslaunch exploration_manager
exploration.launch
```

使用 `2D Nav Goal` 触发仿真