

OpenCL环境配置参考

1. 不同芯片及操作系统的openc1支持情况

硬件平台	操作系统	openc1支持情况	其他说明
Intel CPU	Windows	支持	参考 https://www.intel.cn/content/www/cn/zh/developer/articles/technical/intel-cpu-runtime-for-openc1-applications-with-sycl-support.html 进行相关配置
Intel CPU	Linux/ WSL Linux	支持	参考配置示例
Intel GPU	Windows	支持	参考 https://www.intel.com/content/www/us/en/developer/articles/tool/openc1-drivers.html 进行相关配置
Intel GPU	Linux/ WSL Linux	支持	参考配置示例
Apple M1/M2 GPU	macOS	未测试；应该可以直接用	参考 https://support.apple.com/zh-cn/101525
NVIDIA GPU		支持	linux：配置方法可参考相关博客 window：配置方法可参考相关博客 (搜索关键词NVIDIA GPU + Linux/Windows + OpenCL)
AMD CPU		不推荐	AMD CPU弃用了OpenCL，但是可以使用pocl， (有时间的话，也可以尝试下intel cpu runtime是否对amd cpu有效)
AMD GPU		支持	linux：配置方法可参考相关博客 window：配置方法可参考相关博客

			(搜索关键词NVIDIA GPU + Linux/Windows + OpenCL)
Qualcomm Adreno GPU		支持	移动端GPU通常自带运行时，具体参考配置示例
ARM Mali GPU		支持	自带运行时，配置与Qualcomm Adreno GPU类似，只需要替换LibOpenCL.so即可
FPGA (Xilinx-amd、Altera-intel)		支持	intel、amd的fpga的开发套件包含opencl运行时，具体使用参考官方文档或博客

2. 完整配置流程示例

WSL2 (Win10/Win11) + Intel CPU / Intel GPU / Qualcomm Adreno GPU 的开发环境配置教程，本教程仅作为配置流程的参考；

注意事项：NVIDIA GPU + WSL2 对OpenCL的支持不完善，不建议在WSL2中使用OpenCL在英伟显卡上计算

2.1 安装WSL2

参考官方文档：<https://learn.microsoft.com/zh-cn/windows/wsl/install>

2.2 vscode配置

下载安装VScode，打开vscode - 搜索WSL，安装WSL插件，然后点击左下角的ssh -> Connect to WSL，就可以登陆这台 Linux 服务器了

2.3 WSL中git配置

2.3.1 配置ssh

```

1 1.生成密钥对
2 ssh-keygen -t ed25519 -C "2902139028@qq.com"
3
4 2.查看生成的密钥 复制
5 cat ~/.ssh/id_ed25519.pub
6
7 2.添加密钥到github的ssh密钥设置中
8 登录到 GitHub 账户，进入 Settings -> SSH and GPG keys -> New SSH key，将公钥粘贴到
   Key 字段中，并添加描述后点击 Add SSH key。

```

2.3.2 解决github无法访问问题

```
1 问题：
2 mahang@mahang:~$ ssh -vT git@github.com
3 OpenSSH_8.9p1 Ubuntu-3ubuntu0.10, OpenSSL 3.0.2 15 Mar 2022
4 debug1: Reading configuration data /etc/ssh/ssh_config
5 debug1: /etc/ssh/ssh_config line 19: include /etc/ssh/ssh_config.d/*.conf
   matched no files
6 debug1: /etc/ssh/ssh_config line 21: Applying options for *
7 debug1: Connecting to github.com [127.0.0.1] port 22.
8 debug1: connect to address 127.0.0.1 port 22: Connection refused
9 ssh: connect to host github.com port 22: Connection refused
10
11 解决：
12 hosts文件中添加域名解析
13 1. 查找hosts文件
14 位置：C:\Windows\System32\drivers\etc
15 2. 修改权限
16 右键文件属性添加写入权限
17 3. 添加GitHub ip地址和域名
18 140.82.112.3 github.com
19
20 地址查询：https://github.com/521xueweihan/GitHub520
21 4. 重试 ok
```

2.4 配置安卓开发环境

2.4.1 jdk配置

```
1 1. 下载jdk
2 官网下载，参考版本jdk-21.0.4
3 2. 压缩包解压
4 tar -xvf jdk-21.0.4-linux-x64.tar.gz
5 3. 配置环境变量
6 打开配置文件
7 vim ~/.bashrc
8 尾部添加：
9 export JAVA_HOME=/home/mahang/toolchains/jdk-21.0.4
10 export PATH=$JAVA_HOME/bin:$PATH
11 4. 重新加载 ~/.bashrc：
12 source ~/.bashrc
13 5. 测试配置是否成功：
14 java -version
```

2.4.2 sdk+ndk配置

```
1 1. 下载sdk
2 下载地址: https://developer.android.com/studio?hl=zh-cn
3 2. 解压文件到新建的sdk目录下
4 unzip commandlinetools-linux-11076708_latest.zip
5 3. 进入cmdline-tools/bin目录
6
7 4. 执行以下命令显示所有可下载工具
8 ./sdkmanager --sdk_root=/home/mahang/toolchains/ --list
9
10 5. 依次安装platforms, platform-tools, build-tools, emulator, ndk等:
11 ./sdkmanager --sdk_root=/home/mahang/toolchains/ "platforms;android-32"
12 ./sdkmanager --sdk_root=/home/mahang/toolchains/ "platform-tools"
13 ./sdkmanager --sdk_root=/home/mahang/toolchains/ "build-tools;35.0.0"
14 ./sdkmanager --sdk_root=/home/mahang/toolchains/ "emulator"
15 ./sdkmanager --sdk_root=/home/mahang/toolchains/ "ndk;27.0.12077973"
16
17 6. 添加环境变量
18 vim ~/.bashrc
19 export ANDROID_HOME=/home/mahang/toolchains/
20 export PATH=$PATH:$ANDROID_HOME/build-tools/35.0.0:$ANDROID_HOME/platform-
  tools:$ANDROID_HOME/ndk/27.0.12077973
21 export NDK_HOME="/home/mahang/toolchains/ndk/27.0.12077973"
22 5. 加载
23 source ~/.bashrc
24 6. 依次执行下列命令进行测试
25 adb
26 aapt
27 ndk-build
```

2.4.3 opengl配置

```
1 1. 拷贝OpenCL依赖库libOpenCL.so
2 因为google是不公开支持opengl的, 所以NDK中并不会会有libOpenCL.so, 这个库都是放在各个
  厂家的库中, 比较常见的位置是"system/vendor/lib/libOpenCL.so"
3
4 查看OpenCL在手机位置
5 因为厂商和安卓版本的差异存放位置可能有细微变化, 以下为一般情况下OpenCL所在位置:
6 低版本安卓设备
7 高通平台: /system/vendor/lib/libOpenCL.so
8 mali:/system/lib/egl/libGLES_mali.so
9 高版本安卓设备:
10 高通平台: /system/vendor/lib64/libOpenCL.so
11 mali:/system/lib64/egl/libGLES_mali.so
12
13 拷贝到指定(可任意指定)目录
```

```
14 adb pull /system/vendor/lib64/libOpenCL.so
15 本文存放的路径为：/home/mahang/toolchains/libOpenCL.so
16
17 2. 下载opencl-c-headers
18 在指定（可任意指定）目录下：
19 克隆https://github.com/KhronosGroup/OpenCL-Headers项目即可
20 本文存放的路径为：/home/mahang/toolchains/OpenCL-Headers/
21
22 3. 配置环境变量
23 export OPENCL_LIB="/home/mahang/toolchains/"
24 指向OpenCL依赖库libOpenCL.so位置
25 export OPENCL_HEADERS="/home/mahang/toolchains/OpenCL-Headers/"
26 指向OpenCL头文件位置
```

2.4.4 测试安卓交叉编译

2.4.4.1 源码

测试源码文件地址：<https://github.com/ma-hang/OpenCL-Test>

克隆代码到本地。

2.4.4.2 主机端交叉编译

1. 赋予执行权限

```
chmod +x build_android.sh
```

2. 编译测试脚本

```
./build_android.sh
```

3. 传输到安卓设备上执行

```
1 执行命令：
2 adb push openc1_test /data/local/tmp/
3 adb shell chmod +x /data/local/tmp/openc1_test
4 adb shell /data/local/tmp/openc1_test
5 adb shell rm /data/local/tmp/openc1_test
```

2.5 配置intel GPU开发环境

2.5.1 openc1配置

```
1 sudo apt install opencl-headers
2 sudo apt install ocl-icd-libopencl1
3 sudo apt install ocl-icd-opencl-dev
4 sudo apt install clinfo
5 sudo apt install intel-opencl-icd #Install NEO OpenCL runtime for Intel GPU
```

2.5.2 下载测试源码

测试源码文件地址：<https://github.com/ma-hang/OpenCL-Test>

克隆代码到本地。

2.5.3 编译运行

1. 赋予执行权限

```
chmod +x build_gpu.sh
```

2. 编译测试脚本

因为不同设备的安装位置可能存在差异，编译前执行

```
dpkg -L ocl-icd-opencl-dev | grep libOpenCL.so
```

 确认脚本中OPENCL_LIB的路径正确；

执行：

```
./build_gpu.sh
```

3. 执行测试

```
./opencl_test
```

2.6 配置IntelCPU开发环境

2.6.1 opencl配置

1. 添加 Intel oneAPI 软件仓库
- 2 首先，需要添加 Intel 的 **oneAPI** 软件仓库来获得相关的软件包。以下是如何添加仓库并安装所需包的步骤：
- 3 `wget -q0 - https://apt.repos.intel.com/intel-gpg-keys/2023/intel-archive.asc | sudo tee /etc/apt/trusted.gpg.d/intel.asc`
- 4 `echo "deb https://apt.repos.intel.com/oneapi all main" | sudo tee /etc/apt/sources.list.d/intel-oneapi.list`
- 5 `sudo apt update`
- 6 2. 安装 Intel OpenCL 运行时
- 7 安装完成仓库添加之后，安装 `intel-oneapi-runtime-opencl` 包：
- 8 `sudo apt install intel-oneapi-runtime-opencl`
- 9 验证安装
- 10 安装完成后，验证 Intel OpenCL 是否成功安装。可以使用以下命令检查 OpenCL 平台：
- 11 `clinfo | grep "Platform Name"`

```
12
13 可能会遇到的问题：
14 导入 GPG 公钥
15 使用以下命令导入 Intel 的 GPG 公钥：
16 sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys BAC6F0C353D04109
17 这会从 Ubuntu 密钥服务器导入 Intel 公钥。
18 更新仓库
19 公钥导入后，执行以下命令更新包列表：
20 sudo apt update
21 安装 Intel OpenCL 运行时
22 更新仓库后，再次尝试安装 Intel OpenCL 运行时：
23 sudo apt install intel-oneapi-runtime-opencl
```

2.6.2 下载测试源码

测试源码文件地址：<https://github.com/ma-hang/OpenCL-Test>

克隆代码到本地。

2.6.3 编译运行

1. 赋予执行权限

```
chmod +x build_cpu.sh
```

2. 编译测试脚本

因为不同设备的安装位置可能存在差异，编译前执行

```
dpkg -L intel-oneapi-runtime-opencl | grep libOpenCL.so
```

 确认脚本中
OPENCL_LIB的路径正确；

执行：

```
./build_cpu.sh
```

3. 执行测试

```
./opencl_test
```