

实验室指南

Author : Maplef

Email : maplefsnow@gmail.com

Date : 2024.03

目录

- 实验室指南
 - 目录
 - 前言
 - 建议提前掌握的知识/技能
 - 简洁且高效地使用现代操作系统 (Windows, Linux, ...)
 - Linux
 - Windows
 - 规范地管理项目和文件夹结构
 - 会使用终端 Terminal
 - 理解系统变量 Path
 - 了解并掌握面向对象程序设计思想
 - 了解代码编辑器 (Editor) 和集成开发环境 (IDE) 及掌握其使用方法
 - 简单了解程序编译过程和编译器
 - 使用 VPN 节点和代理软件访问国际网络
 - 获取代理软件
 - 获取 VPN 节点
 - Git
 - 安装
 - 配置
 - 学习
 - Microsoft Visio 流程图绘制
 - CMake
 - 实验室常用软件和开发套件
 - OpenCV - 机器视觉
 - 安装
 - 学习
 - Qt / LabView - GUI 操作界面设计
 - NI National Instruments DAQmx - NI 数据采集卡

- Advantech Common Motion API - 研华运动控制卡
 - 安装
 - 学习
- DaHeng Galaxy SDK - 大恒相机开发套件
- HIKVision SDK - 海康威视相机开发套件
- 开发环境配置
 - Visual Studio Code
 - Visual Studio
- 写在后面

前言

我时常怀疑，写下这些东西的意义，究竟会有多少。我时常怀疑，非计科科班出身而要掌握这些技能，是否会有失偏颇。

我曾目睹一些惨象，比如用记事本在没有后缀名的文档里敲代码，比如需要恢复之前删掉的某个模块而不断 ctrl-Z，比如混乱的项目库管理让整个项目几乎无法维护.....

许多看似常识一样的知识和技能，如果不去了解和学习或者甚至未曾听闻，便也会如“隔了一层可悲的厚障壁”，怎么也施展不开。除去研究方面的种种困难，我更不愿意看到的是由于低效和混乱的工作方式导致时间和精力被白白浪费。但这是一个水滴石穿的过程，只有具备一定的项目开发经验才会慢慢在实践中总结一套属于自己的 workflow，慌不择路往往是一个开发者的必经之路。

我常常回想：“若是当时写某某的时候如何如何，便好了”，“若是当时知道有某某，就好了”。诸如此类的想法变成了一点又一点微小的遗憾，一点又一点微小的遗憾聚沙成塔，最终变得无法撼动。我不想看到后来者继续积累这种微小的遗憾。

我并非计科科班出身，只依靠一厢情愿在这条路上摸爬滚打，并不见得有什么资格在这里指指点点，姑妄言之，姑妄听之。若是读后可以产生一些有益的思考或启发，便也不枉实验室经年来对我的种种栽培。

我在深夜里回顾过往，敲下这些文字。那些曾经做过的努力和探索，或心有不甘，或潦草收尾，都构筑成坚实而悲伤的基，不堪回首，回首不堪。而如今我即将离开这里，时间要把一切都雪藏。这些文字或许只是我自己的一厢情愿，连我自己都不知道它们究竟何去何从。但倘若尚有一个人读过，倘若只有一个人读过也好，也算作是我曾经存在于斯的今生今世的证明。

不要温和地走进，那个良夜。

献给勇敢的人

建议提前掌握的知识/技能

笔者整理了进入实验室以及参与各种竞赛以来，自身或其他同学因基本能力的欠缺而导致的低效工作情况，大致列举了在正式开始启动一个项目前需要掌握的基本技能，以供参考。

简洁且高效地使用现代操作系统 (Windows, Linux, ...)

Linux

如果读者是 Linux 发行版 (如 Arch Linux、Ubuntu 等) 的惯用者, 笔者认为读者已经不必继续阅读本章节的剩余部分, 可直接阅读 [实验室常用软件和开发套件](#) 一章。

Windows

有许多个人电脑常用的系统管理方法可能并不适用于工控机/开发环境, 由于 Windows 系统十分宏大和繁杂, 每个人都有不同的理解, 笔者在此给出一些建议, 仅供参考。

- 开发环境应竭力遵循系统的整洁和最小化原则, 不要安装不必要的大型应用软件, **特别是 360、电脑管家等所谓的安全软件**
- 确保开发套件、SDK、驱动、编译器等重要系统组件从**官方网站**下载, 不要使用第三方下载站或分发页。若官网无法访问, 参考 [使用 VPN 节点和代理软件访问国际网络](#) 一章
- 开发环境可能被多人使用, 不要留下自己的个人信息 (如浏览器记住密码等功能), 信息安全很重要
- 尽量不要使用中文路径名, 一律使用英文命名文件夹和源文件。路径名中严禁出现半角单引号(')、半角双引号(")或其他可能引起歧义的特殊字符。路径名中尽量不要带有空格, 如确有需要可使用半角减号(-)或下划线(_)代替
- 合理管理硬盘空间, 可对系统硬盘进行必要的分区加以区分。如可参考分为系统(system)、环境(env)、软件(software)、项目(projects)、文档(doc)五区
- 尽管系统回收站是最后的后悔药, 删除每一个文件之前也都要三思

规范地管理项目和文件夹结构

会使用终端 (Terminal)

理解系统变量 (Path)

了解并掌握面向对象程序设计思想

笔者认为, 对于开发者而言, 学习并具有面向对象的程序设计思维非常重要。**面向对象是一种思维, 并不局限于某种特定的语言**。学习面向对象的思想可以让代码编写能力更上一个台阶, 并且对整个代码世界和现实世界都有更进一步的认识。

学习面向对象程序设计, 学习封装、继承、多态等特性, 可以提高代码的复用性, 让整个工程井井有条, 省去重复开发相似功能的时间。

推荐阅读菜鸟教程中关于 C++ 面向对象部分的介绍: <https://www.runoob.com/cplusplus/cpp-classes-objects.html>。其他文档或视频教程均可, 可以自行搜索。

了解代码编辑器 (Editor) 和集成开发环境 (IDE) 及掌握其使用方法

简单了解程序编译过程和编译器

使用 VPN 节点和代理软件访问国际网络

使用 VPN 节点, 俗称“翻墙”, 可以自由地访问国际网络和服务, 如 [Google](#)、[Arxiv](#) 等。

获取代理软件

笔者常用的代理软件为 Clash，但 Clash 官方内核仓库因不可抗力被迫删除，目前可在此镜像站下载 Clash For Windows 版本：<https://www.clash.la/releases/>。

其他代理软件，如 ShadowRocket、v2ray 等亦可，可根据喜好自行探索。

获取 VPN 节点

有许多个人或国际组织对中国大陆用户提供 VPN 节点服务，笔者常用的代理服务由 XSUS 提供，可在此进行购买：<https://xsus.wiki/>。其他运营商亦可，自行探索。

简而言之，购买节点后会获得一串订阅链接，将此订阅链接粘贴至代理软件相应的配置项中即可拉取节点信息，选择节点并开启系统代理后即可自由访问国际网络。相关教程很多，可以自行搜索。

Git

Git 官网：<https://git-scm.com/>

学会使用 Git 管理自己的项目，是笔者认为每个编写代码的人应该具备的基本素质。Git 提供了完善的版本管理机制，允许开发者以“提交(commit)”为单位对项目代码进行回溯，更有全球性的代码仓库 GitHub 可作为远程(remote)服务器使用，大大降低了因本地硬件故障造成的项目丢失可能性。

安装

在 Windows 下，官网下载 Git 的安装程序并按照提示安装即可。Git 的安装程序可配置项较多，全部使用默认设置一路 next 即可。

将 Git 的安装目录加入系统变量(PATH)中。

配置

打开终端，对 Git 进行基本的配置。执行以下命令：

```
# 设置 git 全局代理，代理端口根据代理软件给出的端口设置，一般为 7890
git config --global https.proxy https://127.0.0.1:7890
git config --global http.proxy http://127.0.0.1:7890

# 设置 git 提交时使用的用户名和邮箱，否则无法进行提交
# 如果开发环境是公共的且所有人共用一个账户，不要填写自己的个人信息，填一个公用的即可（乱填也行）
git config user.name <your_name>
git config user.email <your_email>
```

学习

推荐一个可视化学习 Git 的网站：https://learngitbranching.js.org/?locale=zh_CN，这个网站基本囊括了 Git 使用的所有入门/初级知识，值得一学。

Microsoft Visio 流程图绘制

和微软 Office 家族御三家的操作逻辑相同，可以自行学习。

CMake

实验室常用软件和开发套件

笔者整理了进入实验室以来所接触到的各种硬件软件的开发套件，并给予简要介绍和说明。

有关如何使用 VS(Visual Studio) 或 VS Code(Visual Studio Code) 进行相关软件环境和项目的配置，参见后文。

OpenCV - 机器视觉

安装

OpenCV 官方下载页: <https://opencv.org/releases/>

运行下载好的 exe 程序，选择解压路径即可完成安装。

有关如何将 OpenCV 集成至自己的项目中，参见 [开发环境配置](#) 一章。

学习

关于 OpenCV 的学习，市面上有很多教程，可以自行搜索。

一些有用的资料：

- OpenCV 官方文档: <https://docs.opencv.org/4.9.0/index.html>
- OpenCV Tutorial C++: <https://www.opencv-srf.com/p/introduction.html>
- Learn OpenCV C++ and Python Examples:
<https://github.com/spmallick/learnopencv>

Qt / LabView - GUI 操作界面设计

由于笔者仅进行过 Qt 的开发，故本节内容只介绍 Qt。

Qt Online Installer 下载: <https://www.qt.io/download-qt-installer-oss>

由于在线安装默认连接的官方下载源速度过于缓慢，需要手动指定国内中科大镜像源，所以不要通过双击此 exe 的方式进行下载。在下载目录打开终端，键入以下命令启动下载器：

```
# 由于版本可能不同，下载器文件名按照实际名字更改
.\qt-unified-windows-x64-4.7.0-online.exe --mirror
https://mirrors.ustc.edu.cn/qtproject
```

按照下载器给出的提示进行安装，在指定安装路径界面选择 **Qt x.x for desktop development** 安装方式，之后全部以默认项安装即可。

Qt 自带了 mingw1120_64 编译器和 cmake，（也许）针对 Qt 进行了特别的优化，推荐在配置 Qt 环境时直接指定 Qt 自带的编译器和 cmake，无需额外安装。

- Qt 自带的 mingw 编译器位置: `../Qt/Tools/mingw1120_64/`
- Qt 自带的 cmake 位置: `../Qt/Tools/CMake_64/bin/`

NI(National Instruments) DAQmx - NI 数据采集卡

Advantech Common Motion API - 研华运动控制卡

安装

Advantech Common Motion Driver 下载: <https://www.advantech.com.cn/zh-cn/support/details/driver?id=1-RNJ0CF>

下载后安装软件，软件安装完成后会自动安装驱动，保持默认设置即可。

研华官方提供的实例软件 Common Motion Utility 的路径为 `../Advantech/Common Motion/Utility/Common Motion Utility.exe`，可在此进行基本的板卡功能测试。

学习

Common motion API manual: <https://www.advantech.com/en-us/support/details/software-api-manual?id=1-11T0QZZ>

自行阅读并学习该开发手册。

DaHeng Galaxy SDK - 大恒相机开发套件

HIKVision SDK - 海康威视相机开发套件

开发环境配置

配置开发环境向来是让人头痛的事情，笔者深有体会。不同的系统版本、软件版本、集成开发环境版本下都会产生难以预期的奇怪报错，且通常由于问题的特殊性难以找到相应的教程，笔者对此心有戚戚。笔者在此撰写的教程总有一天会过时或失效，笔者希望读者在配置环境的过程中逐渐掌握一般方法和常见问题的解决方式，学会看报错信息和 log，在试错中不断学习和进步，以期能够自由地组装趁手的开发环境。

由于笔者对 Visual Studio 的硕大体量和 msvc 编译器望而却步，且 VS 中出现的问题往往更加古怪且难以定位，故本章笔者着重介绍使用 VS Code 进行环境配置的方法。

Visual Studio Code

Visual Studio

从略。

写在后面

这里是写在后面。