WEEK1 NOTE

你应该已经在 Week0 中完成了关于 Linux 环境的安装，那么现在来尝试下在 Linux下进行开发吧🤓。

你将会学到：

工具的选择

这里使用 vscode 作为编辑器，gcc 作为编译器。

> ⚠ **Warning**
>
> 你可能会想，为什么不使用“现代化”的IDE，而是要麻烦的自己操作。作为学习 c 的程序员，搞明白 c 语言底层逻辑、与系统的交互等等是很重要的，作为一个系统级语言，光跑起来是不够的，要明白是如何跑的。你以后可能在写代码的时候发生许多“玄学问题”，如果没有底层能力将会十分煎熬。
>
> 下面这段出自 learn c the hard way
>
> IDE，或者“集成开发工具”，会使你变笨。如果你想要成为一个好的程序员，它会是最糟糕的工具，因为它隐藏了背后的细节，你的工作是弄清楚背后发生了什么。如果你试着完成一些事情，并且所在平台根据特定的IDE而设计，它们非常有用，但是对于学习C编程（以及许多其它语言），它们没有意义。
>
> > 注
> >
> > 如果你玩过吉他，你应该知道TAB是什么。但是对于其它人，让我对其做个解释。在音乐中有一种乐谱叫做“五线谱”。它是通用、非常古老的乐谱，以一种通用的方法来记下其它人应该在乐器上弹奏的音符。如果你弹过钢琴，这种乐谱非常易于使用，因为它几乎就是为钢琴和交响乐发明的。
> >
> > 然而吉他是一种奇怪的乐器，它并不能很好地适用这种乐谱。所以吉他手通常使用一种叫做TAB（tablature）的乐谱。它所做的不是告诉你该弹奏哪个音符，而是在当时应该拨哪根弦。你完全可以在不知道所弹奏的单个音符的情况下学习整首乐曲，许多人也都是这么做的，但是如果你想知道你弹的是什么，TAB是毫无意义的。
> >
> > 传统的乐谱可能比TAB更难一些，但是会告诉你如何演奏音乐，而不是如果玩吉他。通过传统的乐谱我可以在钢琴上，或者在贝斯上弹奏相同的曲子。我也可以将它放到电脑中，为它设计全部的曲谱。但是通过TAB我只能在吉他上弹奏。
> >
> > IDE就像是TAB，你可以用它非常快速地编程，但是你只能够用一种语言在一个平台上编程。这就是公司喜欢将它卖给你的原因。它们知道你比较懒，并且由于它只适用于它们自己的平台，他们就将你锁定在了那个平台上。
> >
> > 打破这一循环的办法就是不用IDE学习编程。一个普通的文本编辑器，或者一个程序员使用的文本编辑器，例如Vim或者Emacs，能让你更熟悉代码。这有一点点困难，但是终结果是你将会熟悉任何代码，在任何计算机上，以任何语言，并且懂得背后的原理。

vscode 的配置

我们为你导出了一个通用的基础配置，当然，我们同时也欢迎你自己去配置你喜欢的、你需求的东西。vscode 作为一个拥有相当丰富的插件库的应用，大部分的配置将会很愉悦（如果你想使用某插件有疑问， RTFM）。但记住，vscode 本质上还是一个文本编辑器，请不要试图当作 IDE 使用（如配置大量的“自动编译、执行”插件等等），你要做的是改善自己的打字体验。至于编译，交给 gcc 吧。

Git

你是否有这种情况：写代码的时候发现自己很大一段都写错了，想回到曾经的版本发现单纯的撤回完全做不到。或者团队开发项目的时候不知道怎么同步每个人的代码。那么这个时候，你就会想到 Git 的好处了。

什么是 Git: 建议 STFW

我该如何使用 Git 这里是最基础的 Git 使用教程，也就是俗称的三板斧。

```
# 在你的终端中
git clone url # clone 远程仓库
git add .# 这个命令用于添加所有更改
git commit -m "你的提交文本" # 这个命令用于提交
git log # 这个命令用于查看你的提交记录
git push # 这个命令用于把本地仓库提交给远程仓库
git pull # 这个命令用于获取远程仓库的提交
```

显然，上面列举的只是最基本最基本的内容，对于真正想使用 Git 的人，其实最推荐的是先学会再使用。但显然我们并没有那么多耐心。所以，当你感觉缺某部分的时候，STFW吧。

> **Note**
>
> 推荐学习:

> **Important**
>
> 小作业，学习如何回退到自己的某一个提交，看到这里就开一个仓库练手吧

> **Caution**
>
> 为什么这里没有如何创建一个仓库的内容？因为 Github 和 Git 创建的仓库远程默认名称不同，第一次用你可能会晕（当然，最好的方法当然是尝试下），所以这次课程先会建议你先在 Github 上面创建仓库，然后 clone 下来（但我们仍推荐做全部的尝试，因为提前的尝试可以减少未来遇见的恐慌）。

> **Warning**
>
> 记住，在项目中每做一部分就要提交，不然你一定会后悔的（等项目炸了就老实了）