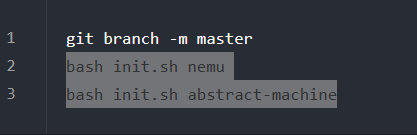
今天看了ysyx的PA0的最后一小节，因为在完成这一节之前，去看了git的一些讲解文章，所以在完成这一节的时候更明白了一些，克隆了一生一芯的仓库之后，进行用户的global配置，之后进入仓库进行一些子项目的初始化，似乎这两个命令后：



就能够从github中提取一些子项目。

之后source刷新，之后运行一些简单的git命令。

随后解决make menuconfig的问题，但是其实我并没有真正找到这个问题的答案，但是我当时觉得自己确实找了很久这个问题的答案，还是没有找到，感觉很难受。当询问一起学习的一个同学的时候他知道一个博主的解决方案，我问他为什么他能找到我找了这么久都没有找到解决方法，他说你怎么知道我就没有经历过呢，最终也是一个一个试最后成功的呢。

当然还是感觉蛮丢人的，更加坚定了我好好学习坚定自己独立解决问题的决心！

随后进行编译，之后似乎源代码的变化将被git追踪。

最后似乎也只有用git commit --allow-empty才能手动提交成功，因为每次的更改其实都已经通过跟踪开发系统提交，所以commit并不会有修改，此时我们加上这个参数这才会在 git log中显示出来，其实他似乎也只相当于一个空修改的标记，其他的都会之间经过开发跟踪记录。

（在某些开发追踪系统中，可能会自动记录代码更改，包括无更改的更改。这是因为开发追踪系统可能在每次编译成功后都会记录代码状态，而无论是否实际更改了代码。在这种情况下，如果开发者手动提交代码，Git 可能会拒绝空的提交（无更改的提交），因为这被认为是没有意义的提交。

为了允许这样的提交，开发者使用 git commit --allow-empty 命令，以便 Git 接受空的提交。这样，开发者就可以将开发追踪系统记录的代码状态（包括无更改的状态）提交到版本控制系统中，并在 git log 中看到相应的提交信息（标记有学生ID和姓名）。）

所以：，git commit --allow-empty 允许提交空的更改，这在某些开发追踪系统下可能是为了兼容记录代码状态的需求。这样的提交可以在 git log 中显示出来，并带有学生ID和姓名标记。而且似乎git log并不会显示那些跟踪开发系统的记录，只会展示git commit --allow-empty之后的空修改的标签。

我想之后我就应该继续mit那套课程的学习了