

Weekly Meeting Memo

Date: Jun. 9, 2015

Participants: JI Xiang, WANG Shoubei, ZHOU Bofan

This memo was recorded by ZHOU Bofan.

Content:

1. Matlab 代码整合:

在对前一阶段已完成的 Matlab 代码各模块的整合过程中,发现这部分代码仍存在一些 问题:例如变量函数名的大小写,新增的代码对之前的模块造成了影响,调试阶段或无实际 功能的代码没有注释掉等等。这些问题产生的主要原因是代码调试的过程仍然存在问题,需 要利用回归测试和版本控制功能提高调试质量,进而保证代码正常工作。

2. dll 文件的生成:

为了更好地隐蔽代码内容,同时保证程序可以在不同的计算机和系统下使用,需要将已 经实现的类实现为 dll 文件。Qt 可以利用自带的 IDE 实现这一功能:

在生成类文件时,选择 Qt: C++ Library,即可生成类对应的 dll 文件,在调用类的程序 project(.pro)文件中,添加 LIBS += ..\..\dll 指令,同时将对应生成的 dll 文件放在主函数相同 的目录下,即可导入 dll 文件所包含类的功能。

在每人负责的各部分模块完成后,会将所有的类生成 dll 文件,通过 Qt 的 installer framework 工具得到系统的 release 版本。

3. project 文件的功能:

一个程序所调用到的所有 Qt 库文件都必须在 project 文件中通过+=指令进行导入,否则 这些模块无法被正常调用。在编写代码过程中,如果 QtIDE 没有在本应出现的正确指令后 生成备选,而是在指令下生成了红色波浪线,就可能是出现了之前的问题。

在 pro 文件中也可以通过 INCLUDEPATH 指令指定需要添加的 h 文件、cpp 文件或者 dll 文件所在文件夹,就可以不需要复制文件即可正常导入和使用类的功能。

4. 堆栈及其工作原理:

内存的存储空间分为 stack/栈和 heap/堆:栈中存储函数,堆中存储变量。地址空间的上 部被分配给堆,下部被分配给栈。如果两部分的存储空间产生重叠,就会导致后写入的内容 覆盖之前的内容,导致程序的 segment fault。栈的写入按照从下到上的顺序,取出或删除则 相反,按照从上到下的顺序。C 语言连续内存空间的分配是按照从小到大的顺序。

当函数或程序工作时,其原理也是按照栈的存取原理:最底层的被调用函数将其返回值 地址返回上一层函数,这层函数再将返回值地址向下返回,知道进行到栈的最下层,将其返 回值返回给系统接口(通常利用 main 函数)或 UI 界面。

Decisions:

本周任务为:

1. 继续各部分代码的编写和调试。
2. 尝试不借助 Qt Designer，直接使用指令实现 UI 功能。