

Development Differences between Linux and Windows

OSQDU

Alex Zhang

Telephone 13905321042

CheungTiFan@GMail.com



Who am I

- 青岛大学化学化工与环境学院 07 化学 2 班
- OSQDU 核心成员



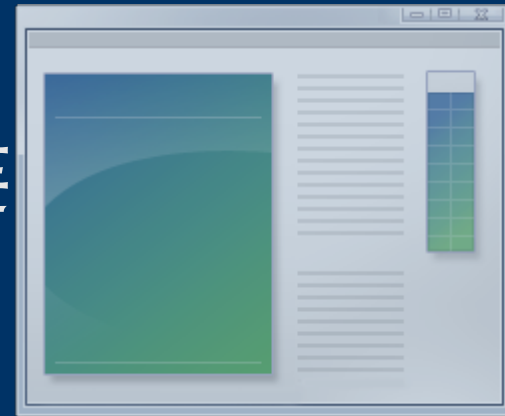
What I'm Talking About

- 简介 Windows 与 Linux 下的开发区别
- 为什么我们用开源平台而不是 MFC 等微软平台？



MFC

- Microsoft Fundamental Classes
- 微软推荐的程序开发平台
- 与 Win32 子系统密切相连



Why Open Source Platform

- 虽然代价巨大但是是大趋势
- 开放
- 容易移植
- 对开发者的吸引力
- 极高的开发效率



Comparing Gtk+ and MFC

- 两者都是基于面向对象设计的
- 两者都是基于消息驱动的
- 两者都不是线程安全的
- GTK+ 整合了一系列的基础函数库，功能强大，而 MFC 孤军作战，势单力薄。



Comparing Gtk+ and MFC

- GTK+ 是跨平台的，而 MFC 则不是。
- GTK+ 小巧，而 MFC 笨重。
- GTK+ 的使用简单，MFC 的使用繁琐。
- GTK+ 使用 signal 机制，解开消息源与消息目标之间耦合。而 MFC 使用消息，将消息源与消息目标硬编码在一起。



Comparing Gtk+ and MFC

- GTK+ 采用 layout 机制动态计算各子窗口的坐标位置，自适应屏幕大小的变化。而 MFC 要求子窗口的坐标位置硬编码，结果要适应不同分辨率的屏幕非常困难。
- GTK+ 采用容器机制来合理分离控件的职责，MFC 没有容器这个概念，很难实现递归组合。
- GTK+ 采用容器机制优先使用组合而不是继承，符合现代设计的原则。MFC 强制使用继承，使用麻烦而且耦合紧密。



Some Interesting Stuff

十六进制表示

0x00004E20

Windows 内存表示

20 4E 00 00

UNIX(R) 内存表示

00 00 4E 20

Windows 中存储方式和该整数的 16 进制表示是相反，是一种低位在前高位在后的存储顺序。Unix 下的存储顺序和正常的 16 进制表示的顺序相同，称为高位在前低位在后的顺序。



Issues

- 网络通信时
- 当 Windows 和 Unix 之间发生网络数据传输，传输一个整型数据（如一个数据包的长度）的时候，如果不经处理直接把内存中的数据传输过去，那么在对方看来完全是另一个数据，这样就会造成问题。如 Windows 下面发送过去一个 20000 (0x00004E20)，在 Unix 下面收到的数据就会被理解成 541982720 (0x204E0000)，这简直是天壤之别。



Issues

- 文件存储和读取时
- 跟网络传输类似，如果在 Windows 下面把某个整数写到了文件中，然后在 Unix 下面打开这个文件读取该数据，就会出现跟上面类似的问题。



Eggs

- 大头敲开还是小头敲开？
- “一场由鸡蛋引发的血案”



Disadvantages

- To play games...

```
% sudo -i
```

```
# cd ~/games/somegame/src
```

```
# ./configure
```

```
.....
```

```
# make
```

```
.....
```

```
# make install
```

```
.....
```

你还有兴趣么？



Programmers Need Entertainment

- Windows 是做什么的？
- 听歌、看碟、炒股票、打网络游戏、看电影、视频聊天
- 界面的美观
- 玩游戏不用一直在 console 下手忙脚乱的编译去
-
- 程序员也需要娱乐



Thanks

谢谢！

CheungTiFan@GMail.com

Distributed under Creative Commons

