网站完成改版

Published on Mar 15, 2021 UTC by Wang Ye <https://wangye.org>

时光荏苒，也不知从什么时候开始，很多感慨时间的文章都喜欢以此作为开头，依稀记得的是这个词汇应该是学生时代课文中学会的，原谅我词汇的匮乏，但是时间真的过的很快，转眼独立个人建站也有近16个年头了，这16年也见证了互联网时代的发展，从刚开始学生党疯狂的申请各种免费域名和主页空间，然后挂上几个静态页面而沾沾自喜，到正儿八经的跑银行汇款注册了人生第一个域名，认认真真的边看书边用ASP搭建了第一个真正个人网站，再后来转学PHP并使用成熟的WordPress建立了博客，至此我开始了内容创建为主的博客历史，截止WordPress博客停用大概也有三百多篇文章了，其中部分文章也得到了网友们中肯的评价。

常常怀念过去的时光，有人说当你开始怀念的时候说明你已经老了，确实人总是对于失去的东西念念不忘往往又不能珍惜拥有的，回顾过往，尤其是那段无忧无虑的学生时代，那种可以泡图书馆，混迹各大技术论坛、通宵写代码，约同学逛街的时代，也是那段时光是我技术突飞猛进的阶段，当工作以后疲于各种琐事时才突然发现最难获得的确是学生时代内心的那份宁静。网站很长时间不维护了，文章也没写几篇，新技术也学不进了，技术仿佛进入了瓶颈期，渐渐开始被互联网技术圈子所淘汰，就拿移动互联网来说，如果有朋友问我：你会不会做手机App啊？我会很尴尬的说不会。我知道朋友问的是iOS和Android两大阵营的App开发，虽然我以前做过手机程序，但那也是零几年多普达Windows Mobile系统上的，再后来的iOS和Android移动互联网浪潮我就没有跟上，很是遗憾。

如果有人问我这么多语言你最擅长那个？我可能会说是C语言，因为C语言较为底层且结构和语法简单，完全适合我这样的老古董，现在我也在重新学习C#，现代编程语言为了方面开发提高效率，增加了各种新特性和语法糖，现在的C#已经不同于我在.NET 2.0时代学习的C#，就好像ASP.NET 2.0时代网页开发沿袭了微软拖控件的思想且只能运行在Windows系统上，而现在脱胎于.NET Core技术的ASP.NET

5.0已经是标准MVC架构且跨多个平台的开源技术了。当然也正是因为.NET开源跨平台才让我重新关注C#语言发展，这个博客也是基于这项技术开发的，稍后再讲具体细节。回到C语言，之前陆陆续续用C语言开发过几款Win32桌面程序，其中也有大家关注的SrvanyUI，也就是对微软服务器工具srvany.exe的包装（Wrapper），很多人问我为什么不更新了呀，这个还真不得不抱歉，因为srvany.exe这个工具并不开源，我的软件也仅仅停留在外部调用包装阶段，所以源程序更新不太现实，如果后期有机会的话，我可能考虑根据原理重新写一个可维护的工具。

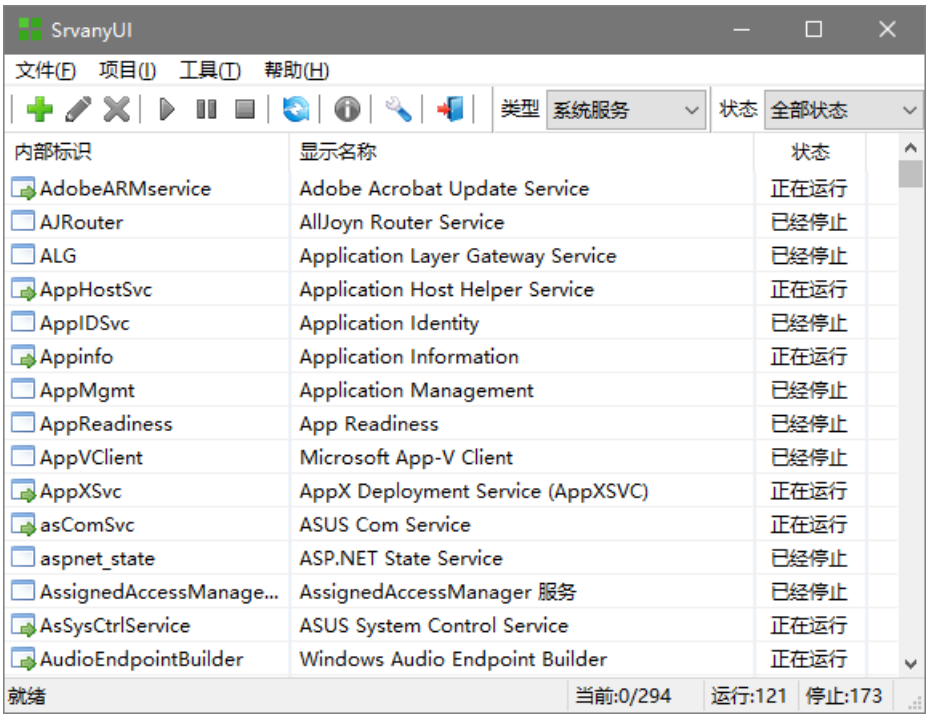

图1 SrvanyUI程序运行截图

C语言开发Windows窗体程序这个是我大学阶段就一直研究的课题，原本以为C语言只能开发黑乎乎的控制台程序，直到看到Notepad2代码后才发现原来还有开发可视界面这种操作，自此认真研究了Win32程序开发，C语言开发Win32程序最大问题就是例子很少，当时大部分是MFC的，MFC调用需要反推其使用的底层API，这点比较耗时耗精力，直到后来Windows 2000源代码泄露事件的发生，源代码中包含大量C/C++开发的实例，包含很多控件的使用、编程习惯等，有很大的参考价值，很长一段时间沉迷于C语言构建Win32程序，其中比较成功的有内部项目Simple Web

Server，这个项目基于Windows原生的HTTP.SYS驱动，通过HTTP API调用实现了高效的多线程网页服务器，期间参考了ASP技术接口，基本实现了Request、Response、Server、Session、Application对象，后期也会考虑整理开放出来供大家使用。

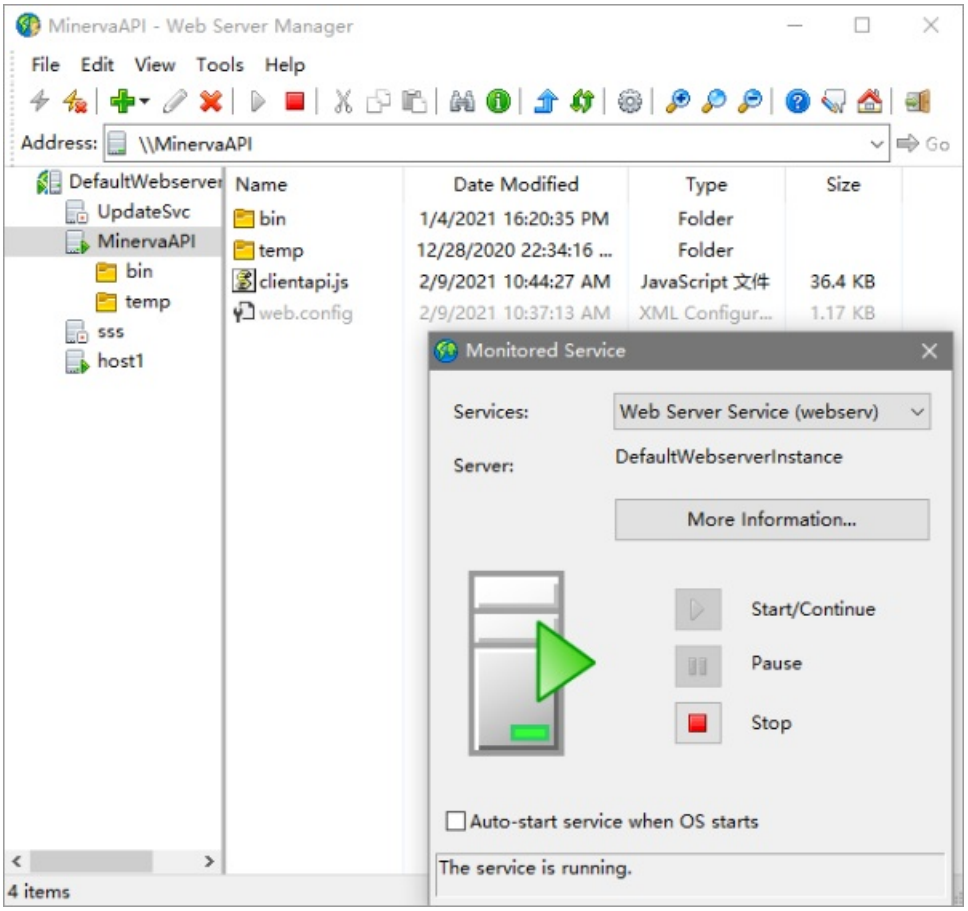

图2 Simple Web Server 程序运行截图

关于这次博客的改版，长期关注我博客的朋友们可能知道，之前的博客架构在WordPress框架下，最早运行在128MB VPS主机，稳定运行了很长时间，后来随着WordPress和PHP升级，对硬件的要求也越来越高，尤其是从32位Debian 5升级到64位后续版本，PHP-CGI升级到PHP-FPM，128MB内存的VPS显得捉襟见肘，后来博客整体迁移到512MB内存的主机空间上，原来VPS很长一段时间承担了Primary DNS的作用。再后来PHP陆续发布了版本7以及后续的版本8，由于很多原先开发的应用基于5.6，所以一直没有升级，直到后来最新版本WordPress也提示PHP版本过

旧存在风险，PHP的升级也提上日程，不过在调研过程中我发现升级后WordPress也变得臃肿起来，我开始考虑是不是应该换个平台了。要不要坚持MySQL+PHP的架构基础？是时候尝试一些新玩意儿了，正值微软破天荒的发布了跨平台的.NET Core技术，什么？.NET可以跨平台运行在Linux下了？荒废多年的.NET重新被我拾了起来，Kindle上下单了C# 7.0相关基础书籍，工作之余认真看了一遍，然后翻了翻MSDN文档，参考了几个开源的.NET博客平台，于是着手开发新的博客系统，是的，继十几年前用ASP自己写博客系统现在又开始了这条老路，只是这个路走的并不顺利，由于是边学边做，遇到很多问题，幸好最后都得以解决，但最大的难题是工作的繁忙导致这个项目无限期托更了下去，从.NET Core 2.0一直到.NET 5.0，甚至期间还抽精力用.NET Core为单位开发了两个内部使用的平台，当然这两个平台也权当练手了，终于最后我看着半成品的网站，想在这么下去.NET 6.0都要出了，网站还是不能上线，愧对网友支持，于是就匆匆上线，就有了这版网站，当然我从这里开始决定还是称呼“网站”而不是“博客”，毕竟网站包含的范围更广一些，除了“博客”这个基本功能外，后期我还将陆续上线自己开发的软件展示下载页面，甚至包含讨论板块，也请大家期待！

这次网站改版应用的部分新特性主要有：

所有基础设置架构全部部署在Docker上，这也是首次尝试这么部署，虽然中间掉过坑，但是Docker技术的应用确实给运维带来极大方便，至少VPS之间迁移更便捷了；
删除了TLS 1.2以下版本的SSL支持，并且全站强制http跳转https（以前这个是由浏览器决定），这也就意味着老版本的浏览器将无法访问本站，实验性开启HTTP3 QUIC支持；
重新设计的视觉效果，自适应的前端设计，本来想应用Bootstrap的，后来想这个框架太重，于是就剪裁了部分实现，并参考了部分主流网站，另外之前也考虑过完全剥离jQuery，毕竟现在主流浏览器支持的特性已经足够（参考），但后来由于一些库调用，这个没有能够实现；
基于.NET 5.0技术开发，后台数据库基于PostgreSQL，其实我感觉PostgreSQL比MySQL更省内存，受限于VPS配置，我对内存使用比较敏感。.NET技术构建网站的

优势就是安全和高效，毕竟可执行档全部是编译的DLL，不存在上传个php木马就能攻陷服务器的情况，尤其是分离上传存储功能后，整个网站应用文件夹全是只读的，杜绝上传漏洞攻击。当然不方便的地方就是每次更新需要重新编译，并停止整个网站，不能像PHP这种脚本语言可以热更新；
部分前端资源的优化，比如Brotli压缩算法（由nginx扩展提供），WebP压缩图片格式，关于这个WebP压缩格式，本站实验性采用了自适应算法，比如虽然请求的图片格式是png，但如果程序后端发现浏览器支持WebP（通过Accept消息头），则发送WebP版本的图片。

当然还有一些新的特性，具体还是实验性的，新网站的问题还有很多，比如contact，以前的contact是用PHP开发的，现在是用.NET改写的，问题就是那个GnuPG加密无法正常使用，虽然可以加密，但是我却无法解密，所以这个功能我暂时去除了，后期我将继续研究，另外contact验证码使用的是Google reCAPTCHA v3，目前感觉误判较高，后期我将继续测试，如果问题依旧我将考虑使用v2版本或者其他替代方案。

好了，如果大家有其他好的意见或者建议也欢迎提出。

External References (Links)

参考
http://youmightnotneedjquery.com/