Windows下编译Nginx及nginx-http-flv-module

# 前置说明

nginx-http-flv-module只是nginx的一个扩展模块，nginx带不带nginx-http-flv-module对编译步骤几乎没影响，但是要注意，要支持nginx-http-flv-module必须同时加入openssl。

# 参考链接

* [nginx/nginx](https://github.com/nginx/nginx) nginx源码
* [winshining/nginx-http-flv-module](https://github.com/winshining/nginx-http-flv-module)  nginx-http-flv-module源码
* [PCRE2Project/pcre2](https://github.com/PCRE2Project/pcre2) pcre2源码
* [zlib Home Site](http://zlib.net/) zlib源码
* [openssl/openssl](https://github.com/openssl/openssl) openssl源码
* [MSYS2](https://www.msys2.org/) 编译环境（只需安装，不需源码）
* [Strawberry Perl for Windows](https://strawberryperl.com/) Perl语言的编译器（只需安装，不需源码）
* [Download & Install Perl - ActiveState](https://www.activestate.com/products/perl/) 另一个Perl语言的编译器（只需安装，不需源码）
* [GnuWin](https://sourceforge.net/projects/gnuwin32/) 又称 sed（只需安装，不需源码）
* [Building nginx on the Win32 platform with Visual C](http://nginx.org/en/docs/howto_build_on_win32.html)官方说明如何在Windows下编译Nginx

# 源码准备

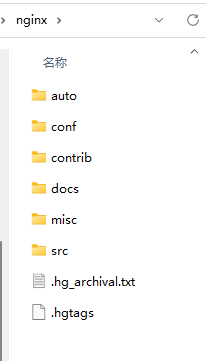
## 下载源码

需要用到的几个源码（都可在参考链接下载）：

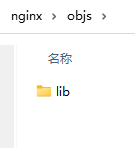
* nginx
* pcre2
* zlib
* openssl（非必须，如果你需要支持ssl或者nginx-http-flv-module就必须）
* nginx-http-flv-module（非必须）

## 放置源码

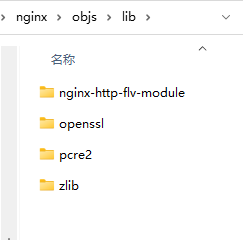
1. 创建一个文件夹，如nginx，里面存放nginx的源码，后续步骤都以nginx文件夹为例说明



1. nginx内创建一个文件夹objs，并再在objs内创建一个lib文件夹



1. 在lib文件夹内放置pcre2、zlib、openssl、nginx-http-flv-module的源码，按文件夹区分开。文件夹命名随意，后续步骤自己对应好哪个文件夹存了哪个库源码即可。



## 安装相关工具

### VS2019

包含单个组件：

* .NET Framework 4.6.1 目标包
* .NET Framework 4.6.1 SDK
* Windows 通用 C 运行时
* Windows 通用 CRT SDK
* MSVC v142 - VS 2019 C++ x64/x86 生成工具(v14.26)
* 对 v142 生成工具(14.21)的 C++/CLI 支持
* Clang compile for Windows
* Windows 10 SDK (10.0.16299.0)

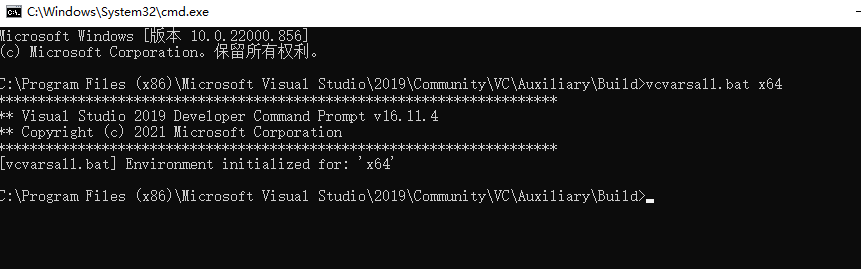
参考自[aushy/nginx-http-flv-win64: nginx-http-flv-module Windows builds. (github.com)](https://github.com/aushy/nginx-http-flv-win64)

由于我电脑上已经有装了各种组件的vs，所以无法验证具体需要哪些。

安装完vs后，命令行执行路径

C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\VC\Auxiliary\Build

下的vcvarsall.bat，并传入参数x64，具体操作可参考如下

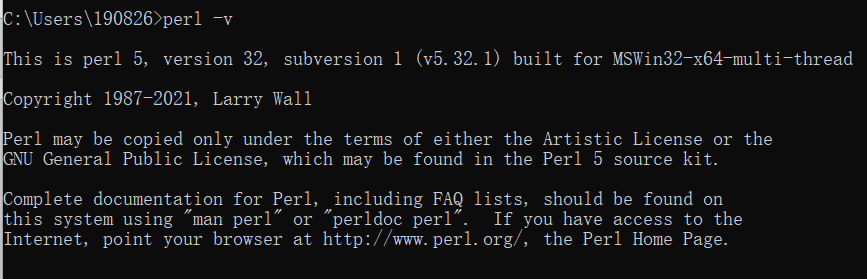


### MSYS2

无脑下一步即可，装完无需其他操作

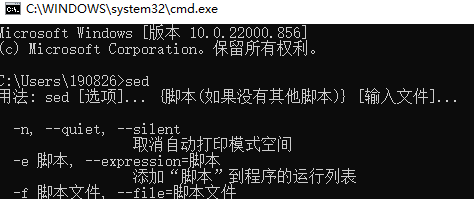
### Perl

Perl特指perl语言的编译器，装ActivePerl 或者 Strawberry Perl其中一个即可。建议Strawberry Perl，我装ActivePerl老是提示环境异常导致失败，Strawberry Perl一次过。装完无需其他操作，但需要检查一下环境变量，如果没有自动配置好，需要自己手动配置。



### GnuWin(Sed)

安装完后需要自己添加环境变量



# 生成Makefile

1. 运行MSYS2 MINGW64，并进入nginx文件夹，如nginx文件夹放在d盘，cd命令如下

cd /d/nginx

1. 在MINGW64窗口执行脚本

auto/configure \

--with-cc=cl \

--with-debug \

--prefix= \

--conf-path=conf/nginx.conf \

--pid-path=logs/nginx.pid \

--http-log-path=logs/access.log \

--error-log-path=logs/error.log \

--sbin-path=nginx.exe \

--http-client-body-temp-path=temp/client\_body\_temp \

--http-proxy-temp-path=temp/proxy\_temp \

--http-fastcgi-temp-path=temp/fastcgi\_temp \

--http-scgi-temp-path=temp/scgi\_temp \

--http-uwsgi-temp-path=temp/uwsgi\_temp \

--with-cc-opt=-DFD\_SETSIZE=1024 \

--with-pcre=objs/lib/pcre2 \

--with-zlib=objs/lib/zlib \

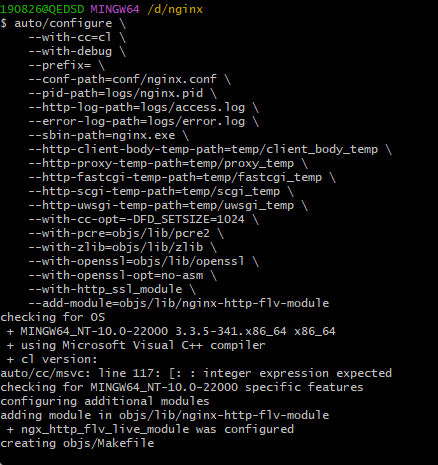
--with-openssl=objs/lib/openssl \

--with-openssl-opt=no-asm \

--with-http\_ssl\_module \

--add-module=objs/lib/nginx-http-flv-module

回车执行脚本并等待创建

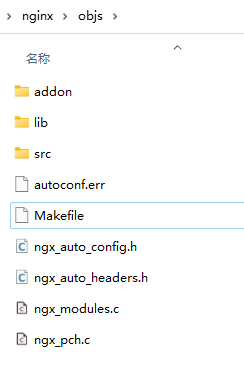


* 注意每个库的源码文件夹要对应上**放置源码**步骤的文件夹
* 如果不需要nginx-http-flv-module模块，把最后一行

--add-module=objs/lib/nginx-http-flv-module

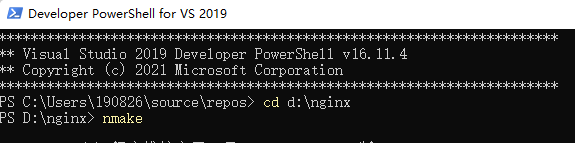
去掉即可，其他pcre\zlib\openssl同理

1. 执行完后会在objs文件夹下出现Makefile及一些其他文件即为完成



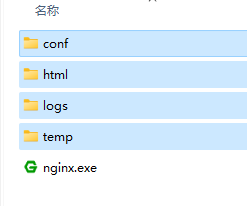
# 编译

打开Developer PowerShell for VS 2019并cd到nginx文件夹，然后执行nmake等待编译



编译耗时取决于电脑和你添加的模块，几分钟到半小时都可能。编译完成后会在objs文件夹下生成nginx.exe，至此即完成了整个nginx的编译。

至于nginx其他文件夹及配置文件



编译并不会一并创建，可以使用已有的，整个nginx核心在于nginx.exe，其他都通用。

# 可能碰到的问题

## Nmake过程出现将警告视为错误

打开objs的Makefile文件，修改CFLAGS行，在WX后加多符号-

原本：

CFLAGS = -O2 -W4 -WX -nologo -MT -Zi -Fdobjs/nginx.pdb -DFD\_SETSIZE=1024 -DNO\_SYS\_TYPES\_H

修改后：

CFLAGS = -O2 -W4 -WX- -nologo -MT -Zi -Fdobjs/nginx.pdb -DFD\_SETSIZE=1024 -DNO\_SYS\_TYPES\_H

参考自

[Windows下nginx-http-flv-module编译\_KayChanGeek的博客-CSDN博客\_nginx-http-flv-module 编译](https://blog.csdn.net/kaychangeek/article/details/105095844)

## Nmake过程提示未知 “perl”、”sed”

环境变量问题

# 其他参考

* [aushy/nginx-http-flv-win64: nginx-http-flv-module Windows builds. (github.com)](https://github.com/aushy/nginx-http-flv-win64)
* [nmake fatal error u1077:path/c1.exe 返回代码0x2解决思路\_m4sean的博客-CSDN博客\_u1077](https://blog.csdn.net/sean4m/article/details/60143222)
* [Win10编译Nginx-1.19.6详细配置并推流 - 简书 (jianshu.com)](https://www.jianshu.com/p/2a7cfcab5f43)

# 扩展：ffmpeg推流http-flv

1. ffmpeg推流命令参考（以rtmp协议推流）

ffmpeg -re -i 1.mp4 -vcodec h264 -f flv rtmp://127.0.0.1/live/uav

其中live表示nginx的app名称，uav表示streamname，不可省略

1. nginx.conf参考(同时开启rtmp和http-flv)

worker\_processes  1;

error\_log  logs/error.log info;

events {

    worker\_connections  1024;

}

rtmp {

    server {

        listen 1935;

        application live {

            live on;

        }

    }

}

http {

    server {

        listen      8088;

        location /live { # 拉流时的 uri ，可以自行修改

            flv\_live on; # 打开 http-flv 服务

            chunked\_transfer\_encoding on;

            add\_header 'Access-Control-Allow-Origin' '\*'; # 允许跨域

            add\_header 'Access-Control-Allow-Credentials' 'true';

        }

    }

}

1. 观看视频地址

* rtmp://127.0.0.1/live/uav
* http://127.0.0.1:8088/live?app=live&stream=uav