[windows下编译安装nginx\_rtmp\_zqw\_4181的博客-CSDN博客](https://blog.csdn.net/zqw_4181/article/details/93196839?utm_medium=distribute.pc_relevant.none-task-blog-baidujs_title-1&spm=1001.2101.3001.4242)

**windows下搭建nginx-rtmp服务器**

**文章目录**

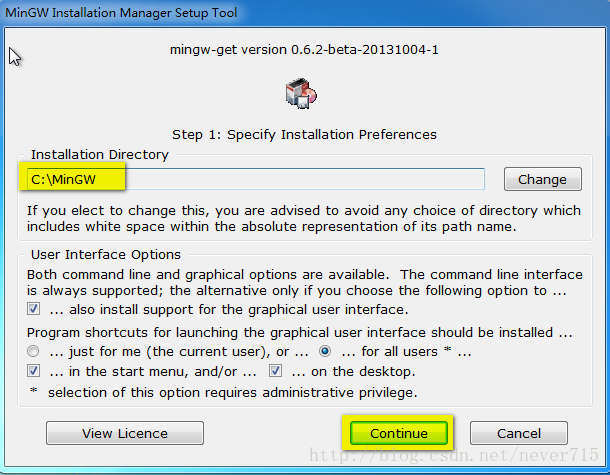
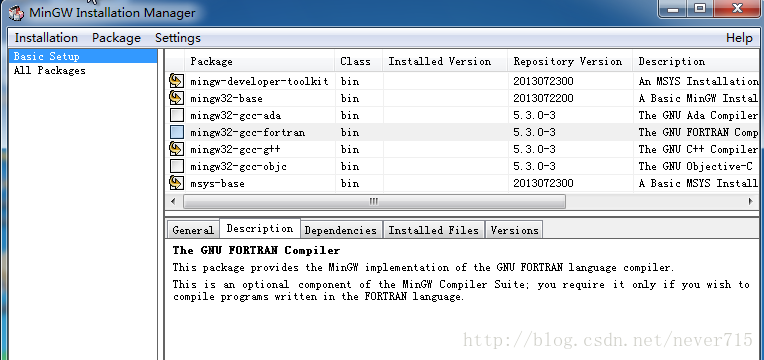
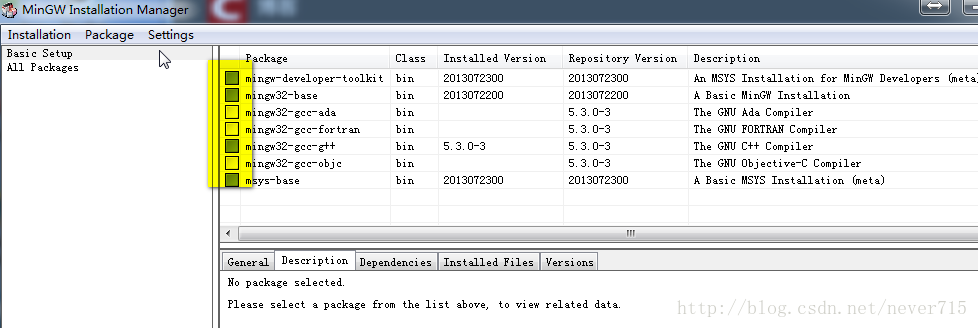
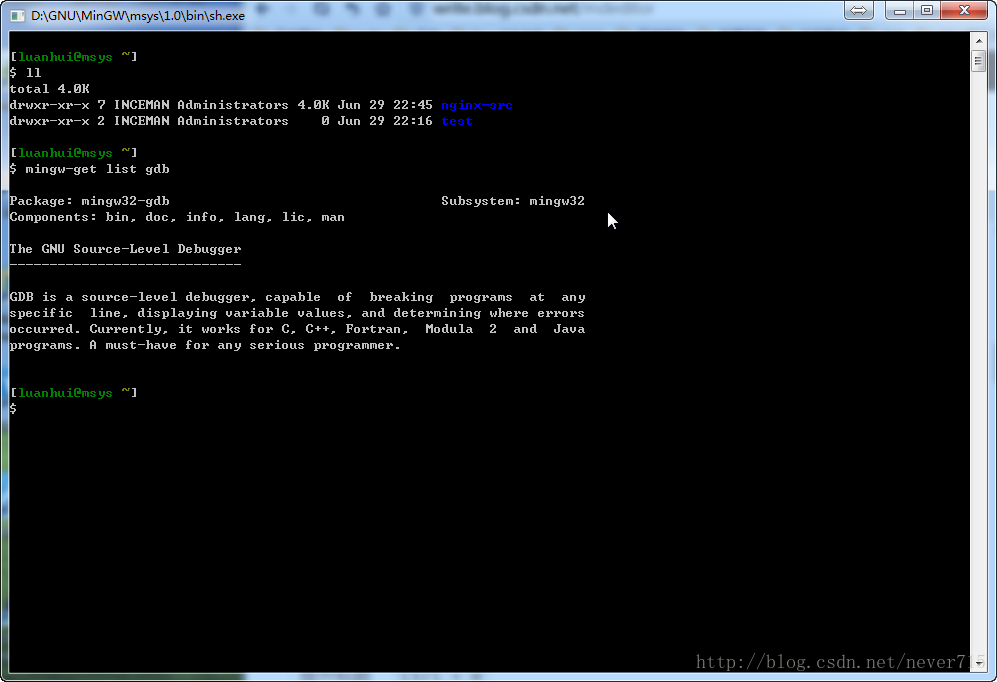
* [windows下搭建nginx-rtmp服务器](https://blog.csdn.net/zqw_4181/article/details/93196839?utm_medium=distribute.pc_relevant.none-task-blog-baidujs_title-1&spm=1001.2101.3001.4242#windowsnginxrtmp_0)
  + [编译nginx](https://blog.csdn.net/zqw_4181/article/details/93196839?utm_medium=distribute.pc_relevant.none-task-blog-baidujs_title-1&spm=1001.2101.3001.4242#nginx_73)
  + [nginx-rtmp环境搭建](https://blog.csdn.net/zqw_4181/article/details/93196839?utm_medium=distribute.pc_relevant.none-task-blog-baidujs_title-1&spm=1001.2101.3001.4242#nginxrtmp_114)
  + [效果测试](https://blog.csdn.net/zqw_4181/article/details/93196839?utm_medium=distribute.pc_relevant.none-task-blog-baidujs_title-1&spm=1001.2101.3001.4242#_123)
  + [最后说明](https://blog.csdn.net/zqw_4181/article/details/93196839?utm_medium=distribute.pc_relevant.none-task-blog-baidujs_title-1&spm=1001.2101.3001.4242#_204)

这是编译好的下载地址，可自行下载 https://download.csdn.net/download/zqw\_4181/11049655

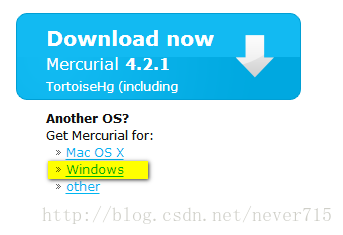
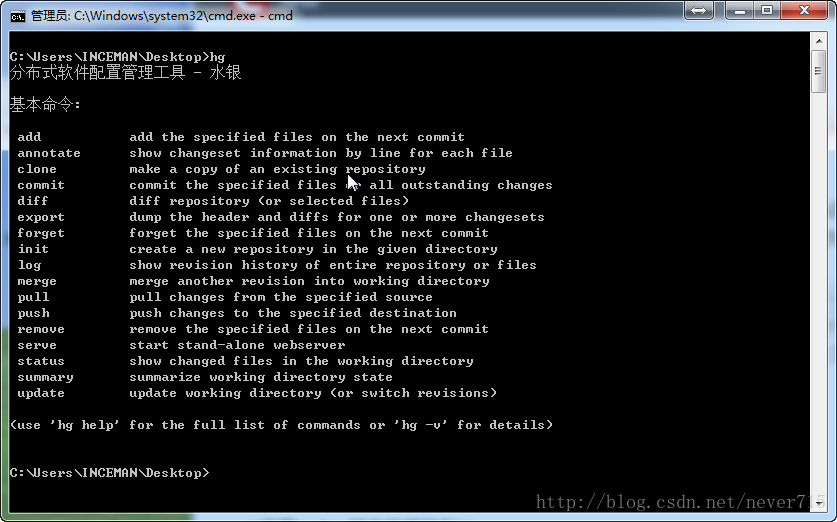
最近因为接触直播相关的技术工作，使用的也是基于高并发和高性能的nginx，首先需要做的当然是搭建一个nginx服务器。  
  
对于linux上nginx服务器的搭建此处就不再赘述，按照操作指引来的话一般都能顺理成章搭建起来。  
  
对于linux操作不熟练的手残党来说，当然是希望能在windows下也能用上这一套。刚开始在windows下搭建环境的时候遇到了很多问题，后面逐个解决花费不少功夫，怕以后忘了，就把我在windows下编译nginx和搭建RTMP直播服务器的过程记录下来，权当做个备忘。

##准备工作

* ###**安装MinGW**###  
  在这些准备工作之前，你的电脑上至少安装有一款IDE，我的是VS2008。  
  首先安装MinGW，后面的nginx编译需要先在msys环境下进行。  
  安装方式可以有在线安装和离线安装两种方式，离线安装的话过程很繁琐，需要自己下载各种包解压、配置路径等等操作，所以此处采用无脑式的在线安装进行说明。  
  由于MinGW的很多资源包都是在sourceforge上下载的，国内虽然能访问，但是速度奇慢，在线安装过程中经常会出现下载组件失败，强烈建议翻墙安装。

1. 到[MinGW](http://www.mingw.org/)找到MinGW安装引导程序[mingw-get-setup.exe](https://sourceforge.net/projects/mingw/files/latest/download?source=files" \t "_blank)的链接地址并下载，下载的文件不到100KB。执行安装工具，开始选择安装路径时要注意，**不要将MinGW安装到带有空格或者中文路径的目录下！！**。  
   
2. 默认情况下，在下载完成后会在桌面生成一个MinGW Installer快捷图标，如果没有，也可以去MinGW安装目录中找到， $MinGW安装目录\libexec\mingw-get\guimain.exe。接下来运行安装之后的快捷图标，选择要在线安装的组件，此处只需选择Basic Setup下msys及gcc相关组件就行了。  
   然后，选择Installation菜单中Apply Changes进行在线下载和安装，整个过程可能会持续十几分钟。  
   当然后续有需要可以再利用这个安装界面进行组件的更改或者删除  
     
   如果安装过程中出现组件下载失败的情况，直接确定继续往下，安装程序会在所有的依赖包下载完成之后进行解压安装，执行完成后，在不关闭安装界面的前提下，回到上面的第二个步骤再来一遍，一直到没有安装错误。
3. 至此，MinGW的成功安装后，勾选的项目会变成浅绿色填充。  
     
   MinGW的安装目录下也会有msys的目录，进入该目录里面，会看到有个批处理文件 msys.bat，可以把它创建到桌面作为快捷方式，双击进去就进入msys环境了。使用上和linux终端类似，不过是个精简版的，命令没有linux下的全，可以使用mingw-get命令进行在线安装某些包，比如，要安装gdb，直接输入命令 mingw-get install gdb。  
     
   MinGW正常安装的话，一般会自动将路径加入到环境变量。如果在使用中出现问题，检查下是不是因为路径设置的原因，手动将 $MinGW安装目录\bin 和 $MinGW安装目录\msys\1.0\bin 加入到windows的PATH环境变量。

* ###**安装Mercurial**###  
  Mercurial是一个版本管理工具，后面我们需要用它下载windows下的nginx源码。

1. 到[Mercurial](https://www.mercurial-scm.org/)下载windows版并安装。  
   
2. 安装时最好将安装目录选择在没有空格和中文目录的路径下。安装完成后，在cmd中执行命令 hg，如果有以下显示则表示安装成功了。  
   

* ###**安装strawberryperl**###  
  Strawberry Perl是Windows环境的Perl环境，包含了运行和开发Perl应用程序所需的全部内容，它的目的是尽可能接近UNIX系统上的Perl环境，后面在编译nginx及其依赖库的时候需要用到。

1. 到[Strawberry Perl](http://strawberryperl.com/)下载并安装windows下的安装程序，选择安装路径时最好选择在没有空格和中文目录的路径下。

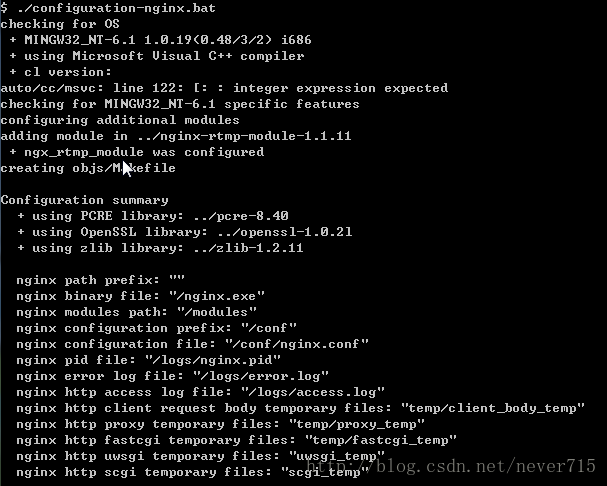
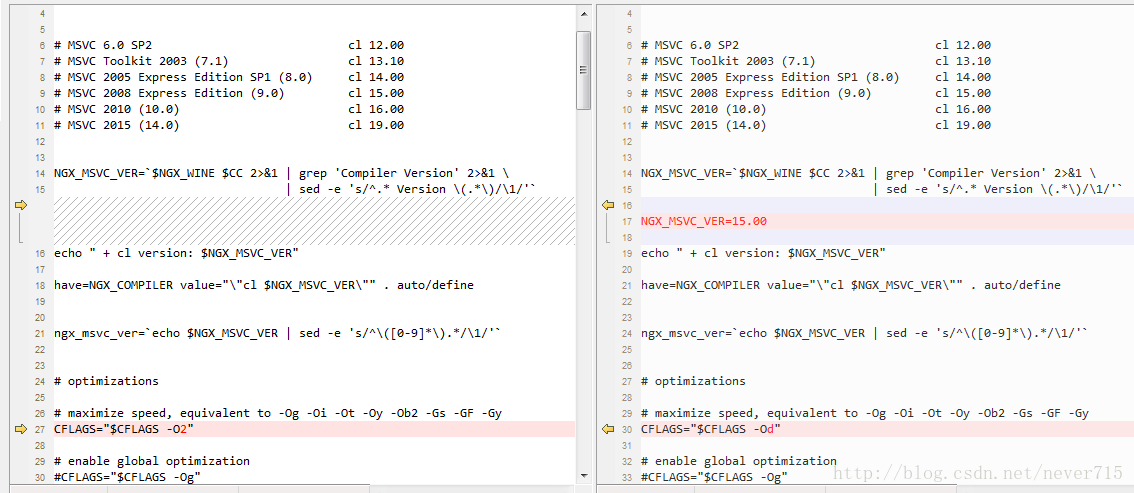
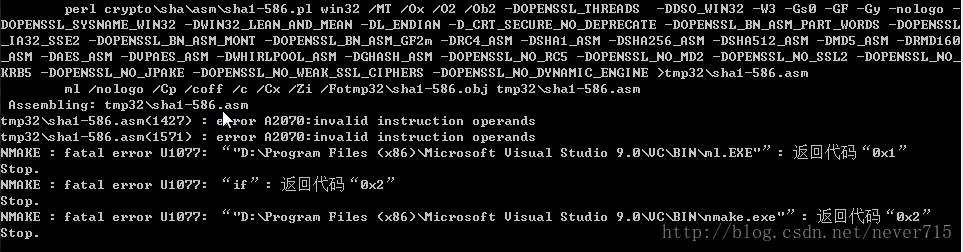
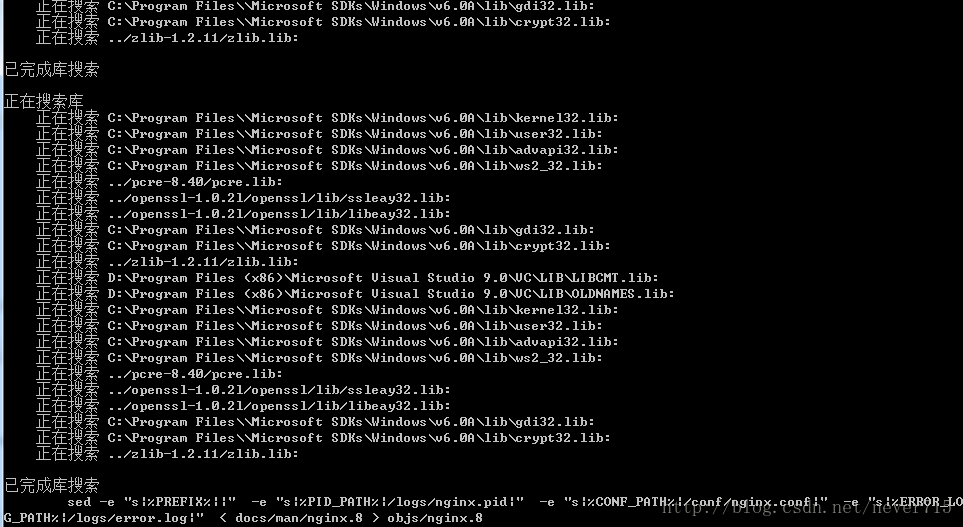
* ###**安装nasm**###  
  NASM全称The Netwide Assembler，是一款基于80x86和x86-64平台的汇编语言编译程序，其设计初衷是为了实现编译器程序跨平台和模块化的特性，至于为什么要安装这货，后面再说。

1. 到[NASM](http://www.nasm.us/)下载安装程序安装即可。
2. 安装完成后，将其安装路径加入到PATH环境变量。

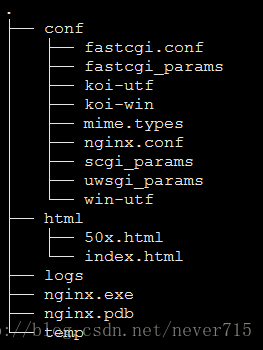
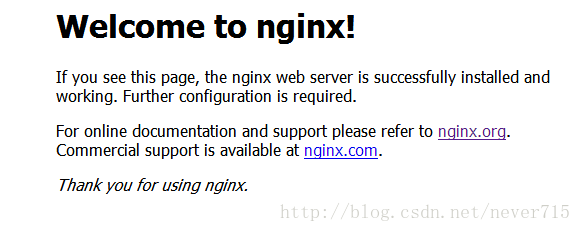
* ###**下载nginx相关的源码包**###  
  搭建nginx-rtmp服务器需要下载nginx源码和nginx-rtmp-module源码。以及nginx依赖的组件。 后面将要在msys下进行配置，不像linux下直接可以使用yum install xxx-devel的形式安装依赖的组件包头。所以需要将nginx相关的源码包悉数下载好，注意是源码包，因为nginx编译时需要引用这些依赖库的开发标头。
* [**nginx**](http://hg.nginx.org/nginx)  
  linux下的nginx源码包可以直接去[nginx网站](http://nginx.org/" \t "_blank)下载，但是上面的源码包只有linux的，源码的os文件夹下仅有unix平台的。而windows只有编译好的文件。这里就需要使用Mercurial下载带有win32平台的nginx源码包。  
  在cmd中执行命令 hg clone http://hg.nginx.org/nginx nginx-win32-src 下载最新的nginx代码，耐心等待源码down到本地，也可以下载指定的版本，具体参考 hg help 。
* [**nginx-rtmp-module**](https://github.com/arut/nginx-rtmp-module)  
  作为nginx外接的扩展模块，将用作后面搭建RTMP直播环境的一部分  
  使用Git或者SVN等工具或者直接去GitHub上下载nginx-rtmp-module
  + [**pcre**](https://ftp.pcre.org/pub/pcre/)
  + [**zlib**](http://www.zlib.net/)
  + [**openssl**](http://distfiles.macports.org/openssl/)

以下是我使用的几个源码包的版本：  
[nginx-release-1.31.1 tag](http://hg.nginx.org/nginx/rev/b4cb75066262)  
[nginx-rtmp-module-1.1.11.tar.gz](https://codeload.github.com/arut/nginx-rtmp-module/tar.gz/v1.1.11)  
[pcre-8.40.tar.gz](https://ftp.pcre.org/pub/pcre/pcre-8.40.tar.gz)  
[zlib-1.2.11.tar.gz](https://sourceforge.net/projects/libpng/files/zlib/1.2.11/zlib-1.2.11.tar.gz/download?use_mirror=nchc&download=)  
[openssl-1.0.2l.tar.gz](http://distfiles.macports.org/openssl/openssl-1.0.2l.tar.gz)

**编译nginx**

1. 将上述下载好的五个源码包拷贝到msys目录下 $MinGW安装目录\msys\1.0\home\$USERNAME\。
2. 打开msys批处理文件，分别解压之前下载的源码包。
3. 进入nginx源码目录。新建一个configuration-nginx.bat文件，里面内容如下，后面的将几个依赖包加入的配置代码根据实际情况填写不同路径。  
     
   auto/configure --with-cc=cl --builddir=objs --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  
   –with-cc-opt=-DFD\_SETSIZE=1024  
   –with-select\_module  
   –with-debug  
   –with-http\_ssl\_module  
   –with-pcre=…/pcre-8.40 --with-zlib=…/zlib-1.2.11 --with-openssl=…/openssl-1.0.2l  
   –add-module=…/nginx-rtmp-module-1.1.11
4. 此时执行该批处理文件，稍稍等待几秒钟，会出现nginx配置完成的信息。  
     
   这里会发现有一个异样的提示，auto/cc/msvc: line 117: [: : integer expression expected ，只要修改文件 $nginx源码目录\auto\cc\msvc 即可，在 echo " + cl version: $NGX\_MSVC\_VER" 的前面加入一行 NGX\_MSVC\_VER=15.00 ，当然不修改也不会影响后续的编译。  
   如果需要指定后续编译nginx时禁用优化，则将其中的 CFLAGS="$CFLAGS -O2"修改为 CFLAGS="$CFLAGS -Od" 。  
     
   到这里为止，msys的使命就完成了。
5. 打开Visual Studio 2008 命令提示，在命令提示符中进入到nginx源码目录，执行命令 nmake /f objs\MakeFile ，会发现编译报错。  
     
   此时，之前安装的NASM就该登场了。  
   打开文件 $nginx源码目录\auto\lib\openssl\makefile.msvc，将其中的  
   ms\do\_ms  
   修改为  
   ms\do\_nasm  
   然后，删除openssl源码目录中的out32和tmp32两个临时目录，重新启动Visual Studio 2008 命令提示，再次执行前面的第五步就可以了。
6. 成功编译之后，最终会有以下打印信息，并且在nginx源码目录的objs目录中可以看到最终生成的nginx.exe文件。  
   

**nginx-rtmp环境搭建**

1. 可以在msys的根目录 / 下新建一个nginx目录，然后将生成的nginx.exe，nginx源码目录下的conf目录以及docs目录下的html目录都拷贝到当前目录，并手动新建logs和temp两个目录，形成最终的目录结构。  
   
2. 随后修改conf/nginx.conf，关于如何配置和修改不在本文范畴内，so，略过~。  
   在当前nginx目录中，启动windows命令行，运行 start /b nginx.exe -c conf\nginx.conf 这样nginx服务就在后台启动了。在浏览器中输入地址进入到nginx服务的主页，如果可以看到如下内容，说明整个环境已经OK，接下来可以使用推流和拉流看看实际效果了。最后使用命令 nginx.exe -s stop 优雅地终止nginx服务。  
   

**效果测试**

接下来的效果测试，对环境稍微做了一点点改动，使用nginx-rtmp-module中提供的JWPlayer的Play页面进行拉流或者Record页面进行推流。

1. 首先是在nginx/html/目录下新建一个文件夹 nginx-rtmp-module，将之前解压的nginx-rtmp-module-1.1.11目录下的stat.xsl文件以及test文件夹拷贝到新建的目录nginx-rtmp-module下。
2. 修改nginx/html/nginx-rtmp-module/test/nginx.conf文件，关闭nginx的守护和master-worker模式，由于本机有SVN占据了80端口，所以修改http的服务端口为8080，同时将nginx服务的访问目录修改为刚才nginx-rtmp-module中指向的index文件。再加入RTMP模块的配置，最终的配置文件内容如下:
3. worker\_processes 1;
4. daemon off;
5. master\_process off;
   * 1
   * 2
   * 3

error\_log logs/error.log info;

events {  
worker\_connections 1024;  
}

rtmp {  
server {  
listen 1935;

application live {

live on;

}

}

* 1
* 2
* 3
* 4

}

http {  
server {  
listen 8080;

location /stat {

rtmp\_stat all;

rtmp\_stat\_stylesheet stat.xsl;

}

location /stat.xsl {

root html/nginx-rtmp-module/;

}

location /control {

rtmp\_control all;

}

location /rtmp-publisher {

root html/nginx-rtmp-module/test;

}

location / {

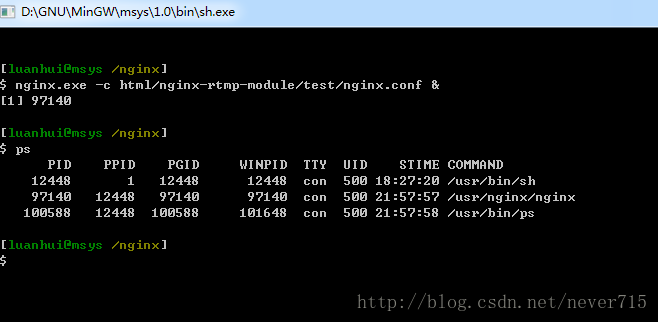
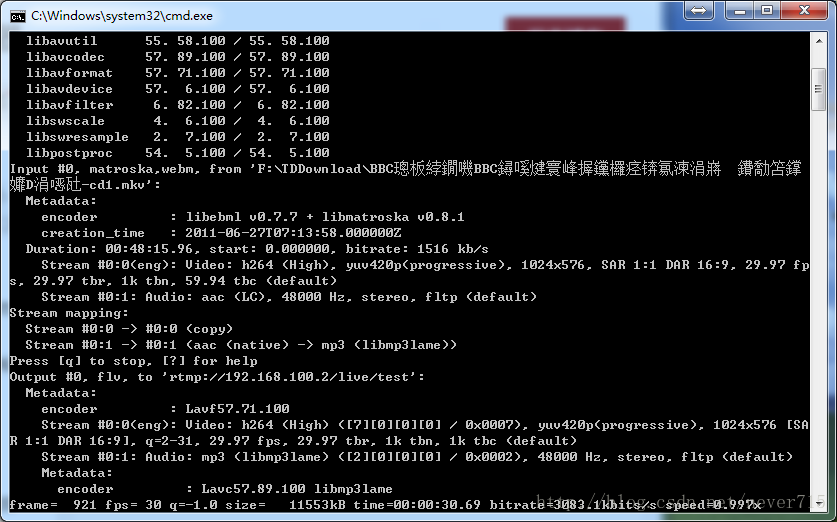
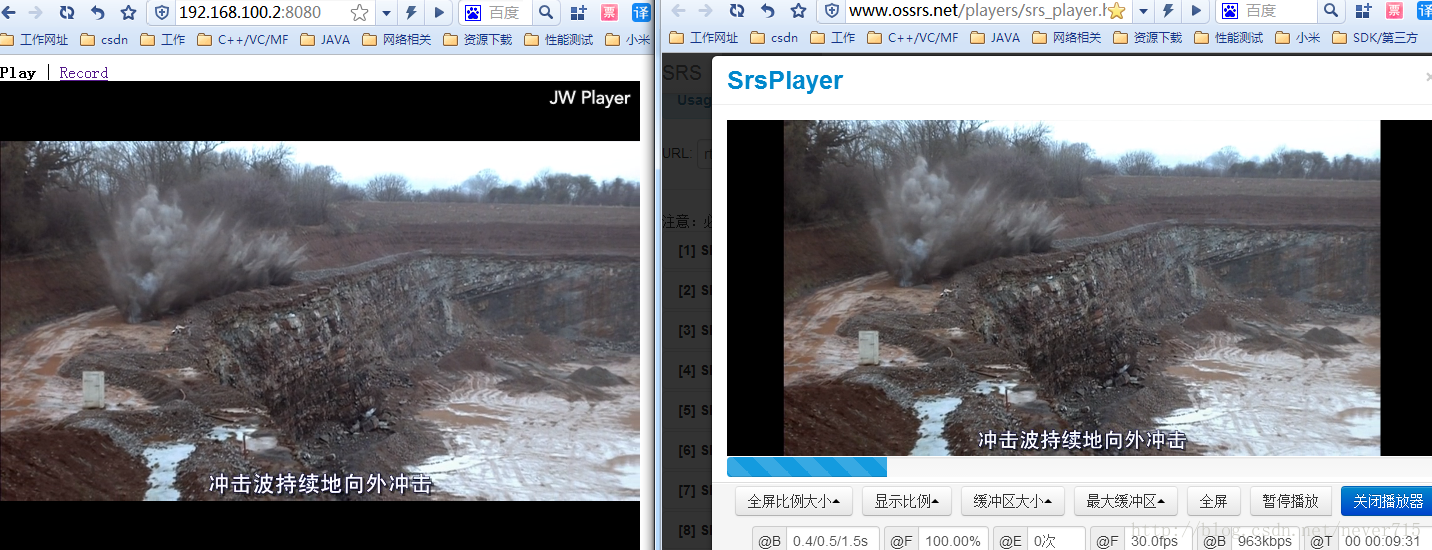
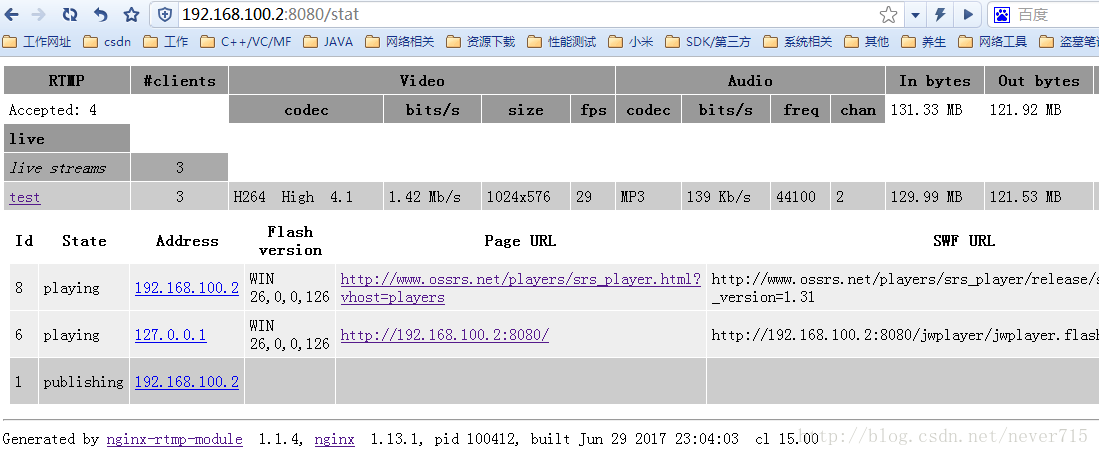
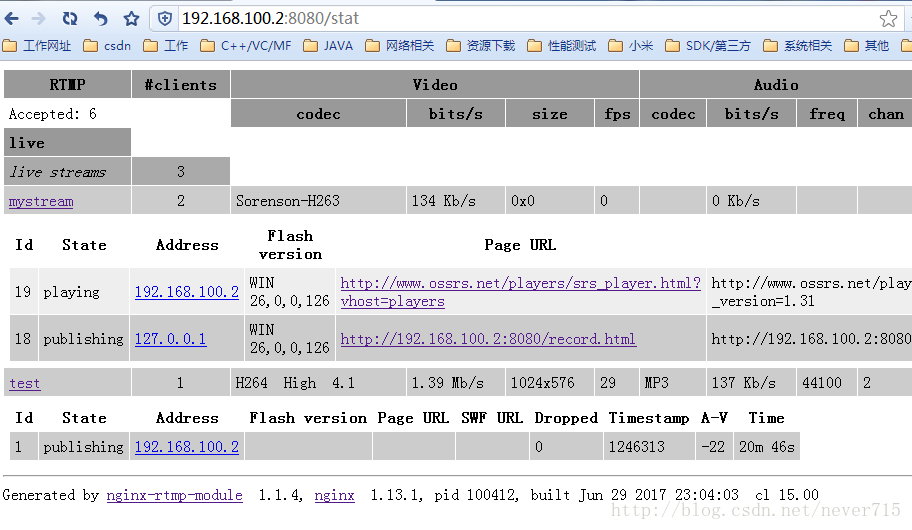
root html/nginx-rtmp-module/test/www;

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 配置拉流名字，修改nginx/html/nginx-rtmp-module/test/www/index.html文件，将  
  file: "rtmp://localhost/myapp?carg=1/mystream?sarg=2" 修改为  
  file: "rtmp://localhost/live?carg=1/test?sarg=2" 以保持和上面的配置文件一致。
* 配置推流名称，修改nginx/html/nginx-rtmp-module/test/www/record.html文件，将开始的  
  'streamer': 'rtmp://localhost/myapp', 修改为  
  'streamer': 'rtmp://localhost/live' 以保持和上面的配置文件一致。
* 在msys下进入nginx目录，执行 nginx.exe -c html/nginx-rtmp-module/test/nginx.conf & nginx服务会在后台运行。  
  
* 推流，推流的手段有很多，此处使用ffmpeg将本地的文件流化并推到nginx服务。  
  ffmpeg -re -i "F:\TDDownload\BBC记录片\BBC理查德哈蒙德：看不见的世界BD中字-cd1.mkv" -vcodec copy -f flv rtmp://192.168.100.2/live/test  
  
* 拉流，拉流方式也很多，浏览器中打开nginx服务的主页面192.168.100.2:8080，可以看到出现的视频播放窗口，点击Play即可拉流，此时的流地址就是上面配置好的rtmp://192.168.100.2/live/test ，也可在[SRS播放器页面](http://www.ossrs.net/players/srs_player.html?vhost=players" \t "_blank)输入流地址rtmp://192.168.100.2/live/test 拉流，还可利用 ffplay rtmp://192.168.100.2/live/test 拉流。  
  
* 打开nginx服务的统计页面192.168.100.2:8080/stat也可以看到当前的服务运行状态。  
  
* 使用JWPlayer的Record页面采集摄像头和麦的数据推流到nginx服务，在[SRS播放器页面](http://www.ossrs.net/players/srs_player.html?vhost=players" \t "_blank)拉流，由于前面配置的Record页面推流的名字是mystream，所以此处输入 rtmp://192.168.100.2/live/mystream。  
    
  
* **最后说明**

本来还试着在移动端试试播放效果，不过JWPlayer貌似是不支持直接在html页面进行rtmp直播，不过可以下载一个app(易推流)进行rtmp的直播测试。

有关nginx-rtmp服务器的搭建过程到此就算全部完成了，在此过程中当然也不全是靠我自己的摸索，也参考了前辈或同行的经验，在此感谢那些曾经的贡献者，参考资料如下：

<http://blog.csdn.net/felixking/article/details/51981794>  
<http://nginx.org/en/docs/howto_build_on_win32.html>

</div>

<link href="https://csdnimg.cn/release/phoenix/mdeditor/markdown\_views-e44c3c0e64.css" rel="stylesheet">

</div>