服务器搭建ffmpeg软件步骤:

1. 首先进入cd/local/software下载ffmpeg tar包

命令: wget <http://ffmpeg.org/releases/ffmpeg-3.4.1.tar.gz>

解压tar包,tar -zxvf ffmpeg-3.4.1.tar.gz -C ../ 到/usr/local目录下

2.进入cd/local/software下载yasm tar包

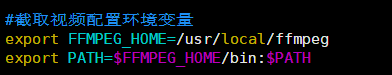
命令: wget <http://www.tortall.net/projects/yasm/releases/yasm-1.3.0.tar.gz>

解压tar包,tar -zxvf yasm-1.3.0.tar.gz -C ../ 到/usr/local目录下

3.配置环境变量

vim /etc/profile 在末尾加上

export FFMPEG\_HOME=/usr/local/ffmpeg   
export PATH=$FFMPEG\_HOME/bin:$PATH



1. 进入cd /usr/local/ffmpeg配置安装路径

./configure --enable-shared --prefix=/usr/local/ffmpeg

--enable-shared 参数据说是允许其编译产生动态库，在以后的编程中要用到这个几个动态库。  
如果出现异常，提示因为缺少yasm，需要添加参数，再执行以下命令

5.cd /usr/local/yasm-1.3.0/

命令: ./configure,然后make,在make install

6.在进入到/usr/local/ffmpeg执行命令:

./configure --enable-shared --disable-yasm --prefix=/usr/local/ffmpeg,安装完毕后进行make,在make install

1. 安装之后在/usr/local/ffmpeg会看到有三个目录  
   bin 执行文件目录  
   lib 静态，动态链接库目录  
   include 编程用到的头文件

8.执行./ffmpeg -version查看版本号,若出现error while loading shared libraries: libavdevice.so.52的错误  
修改/etc/ld.so.conf 在最后一行加上/usr/local/ffmpeg/lib

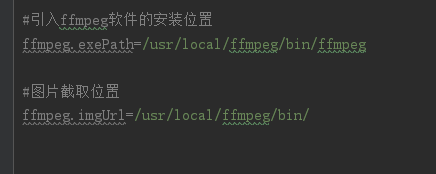
9.在执行ldconfig -v 让其配置生效

1. 最后修改 /usr/local/ffmpeg/lib目录下的文件权限为777

命令:chmod -R 777 /usr/local/ffmpeg/lib

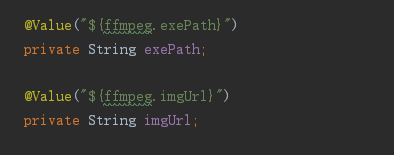
11.到这里我们就已经安装好了

12.环境配置

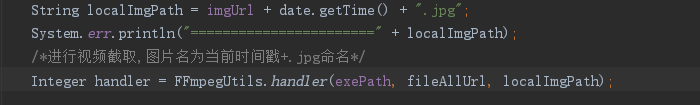


ffmpeg.exePath:是我们软件的安装路径

ffmpeg.imgUrl:是我们上传视频截取图片暂存的位置,当上传视频成功后,暂存图片会删除



为了避免图片名称重复,使用的当前时间命名



exePath：软件的安装路径

fileAllUrl:视频的url地址

LocalImgPath：图片的暂存路径（注：这里可以随便存放,记得删除就好！！）



图片上传成功后,会存储到fastdfs中,然后把暂存图片删除

1. 以下是上传视频的工具类

public static int handler(String ffmpegPath, String upFilePath, String mediaPicPath) {  
 List videoCapture = new ArrayList();  
 // 视频提取工具的位置  
 videoCapture.add(ffmpegPath);  
 // 添加参数＂-i＂，该参数指定要转换的文件  
 videoCapture.add("-i");  
 // 视频文件路径  
 videoCapture.add(upFilePath);  
 videoCapture.add("-y");  
 videoCapture.add("-f");  
 videoCapture.add("image2");  
 // 添加参数＂-ss＂，该参数指定截取的起始时间  
 videoCapture.add("-ss");  
 // 添加起始时间为第0.4秒  
 videoCapture.add("0.4");  
 // 添加参数＂-t＂，该参数指定持续时间  
 videoCapture.add("-t");  
 // 添加持续时间为1毫秒  
 videoCapture.add("0.001");  
 // 添加参数＂-s＂，该参数指定截取的图片大小  
 videoCapture.add("-s");  
 // 添加截取的图片大小为800\*600  
 videoCapture.add("800\*600");  
 // 添加截取的图片的保存路径  
 videoCapture.add(mediaPicPath);  
  
 boolean mark = true;  
 ProcessBuilder builder = new ProcessBuilder();  
 try {  
  
 builder.command(videoCapture);  
 builder.redirectErrorStream(true);  
 // 如果此属性为 true，则任何由通过此对象的 start() 方法启动的后续子进程生成的错误输出都将与标准输出合并，  
 // 因此两者均可使用 Process.getInputStream() 方法读取。这使得关联错误消息和相应的输出变得更容易  
 Process process = builder.start();  
 int i = process.waitFor();  
 return i;  
 } catch (Exception e) {  
 mark = false;  
 System.*out*.println(e);  
 e.printStackTrace();  
 }  
  
 return -1;  
}