



链滴

关于 rtty 在 Ubuntu 和嵌入式板卡上的客户端搭建以及在阿里云上服务端的搭建

作者: [yuren0705](#)

原文链接: <https://ld246.com/article/1563626442507>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>第一次发布博客本人萌新,不足之处望谅解,有意见尽管回复,本人——收下。

不久前在 GitHub 上看到一份文件 https://github.com/zhaojh329/rtty 是一个反向代理 WebTTY。它由客户端和[服务端]组成。服务端采用 Go 语言实现, 以及使用了 vue + iview。你可以基于你设置的设备 ID (不设置则为设备的 MAC 地址) 通过 Web 浏览器访问你的任意一台终端。本人分别在自己的虚拟机(ubuntu) 以及在嵌入式板卡 (imx6ul) 上建了客户端, 以及在自己的阿里云上搭建了服务端, 测试通过;

#####服务端搭建#####

https://github.com/zhaojh329/rttys

本人使用的已经编译好的文件

1.在阿里云克隆到本地 wget https://github.com/zhaojh329/rttys/files/3129765/rttys-linux-amd64.tar.gz

2.压缩

3.进入目录后发现四个文件 (nuhup.out 文件是本人后台运行服务端时产生的文件)

在.conf 中可以查看你的一些用户信息, 后面登陆需要

4.启动

基于 Ubuntu 的客户端开启 ssl: ./rttys

登陆 https://www.yuren.fun (这里为你自己服务器的 ip,本人用自己购买并且解析的阿里云域名)</p>
<p>基于嵌入式的客户端未开启 ssl: ./rttys -ssl-cert string

登陆 http://www.yuren.fun

登陆界面如下 用户名密码在.conf 文件中

#####客户端搭建#####

https://github.com/zhaojh329/rtty

一、ubuntu 客户端搭建

安装编译工具

sudo apt install gcc cmake git

编译和安装依赖软件包

sudo apt install libjson-c-dev

git clone https://git.openwrt.org/project/libubox.git

cd libubox && cmake -DBUILD_LUA=OFF . && sudo make install

git clone https://github.com/zhaojh329/libuwsc.git

cd libuwsc && cmake -DUWSC_SSL_SUPPORT=OFF . && sudo make install

编译安装

git clone https://github.com/zhaojh329/rtty.git

cd rtty && cmake . && sudo make install

运行 (嵌入式运行过程一致注意 ssl 的使用)

sudo rtty -l 'My-device-ID' -h 'your-server' -p 5912 -a -v -s -d 'My Device Description' (使用 ssl)

sudo rtty -l 'My-device-ID' -h 'your-server' -p 5912 -a -v -d 'My Device Description' (未使用 ssl)

My-device-ID: 设置设备的 ID (随便填--123,qqq 都可以; web 端会显示连接的设备 ID)

'your-server':你的服务端的 IP 要和服务端的 IP 一致

-p 5912:端口号 (注意在阿里云控制台的安全组中打开这个端口号)

My Device Description:向你的设备添加描述

具体的指令如下

>
服务端和客户端都打开以后可以查看是否连接成功

这里代表已经连接, 我们在进入 web 界面查看是否可以使用

到这里 ubuntu 已经搭建完了

二、嵌入式客户端搭建

https://github.com/zhaojh329/rtty/blob/master/CROSS_COMPILE.md

(其实发帖的最重要原因是网上基本没有关于嵌入式客户端的编译过程, 最初在虚拟机上交叉编译过程中出现了很多问题, 之后在 GitHub 上发现了发布此项目人留了邮箱, 打扰很久后在他重新更新提交一份新的交叉编译过程后才编译通过接下来说明一下搭建过程, 本人萌新摸索很久如果不足之处希望家之处(#^.^#))

1.Build libev

git clone https://github.com/enki/libev.git

cd libev

./configure --host=arm-linux-gnueabi

(此处的 arm-linux-gnueabi 交叉编译环境为自己嵌入式板卡支持的编译版本, 比如我的编译环境为 rm-linux-gnueabihf 则使用的指令为./configure --host=arm-linux-gnueabihf)

DESTDIR=/tmp/rtty_install make install

2.Build libuwsc

git clone --recursive https://github.com/zhaojh329/libuwsc.git

(如果在此过程中遇到编译错误请克隆 git clone --recursive https://github.com/zhaojh329/libuwsc)

cd libuwsc

cmake . -DCMAKE_C_COMPILER=arm-linux-gnueabi-gcc -DCMAKE_FIND_ROOT_PATH=/tmp/rtty_install -DUWSC_SSL_SUPPORT=OFF (同理这边的 arm-linux-gnueabi-gcc 换成你自己的交叉编译环境)

DESTDIR=/tmp/rtty_install make install

 <https://b3logfi e.com/file/2019/07/10-c854175f.png?imageView2/2/interlace/1/format/jpg>

3.Build rtty

git clone https://github.com/zhaojh329/rtty.git

cd rtty

cmake . -DCMAKE_C_COMPILER=arm-linux-gnueabi-gcc -DCMAKE_FIND_ROOT_PATH=/tmp/ tty_install

(同理这边的 arm-linux-gnueabi-gcc 换成你自己的交叉编译环境)

 <https://b3logfi e.com/file/2019/07/13-9b983f2b.png?imageView2/2/interlace/1/format/jpg>

DESTDIR=/tmp/rtty_install make install

 <https://b3logfi e.com/file/2019/07/14-6a6752a5.png?imageView2/2/interlace/1/format/jpg>

搭建完成后会在/tmp/目录下看到编译好的 rtty 客户端

 <https://b3logfi e.com/file/2019/07/11-cf152f32.png?imageView2/2/interlace/1/format/jpg>

接下来你/tmp/rtty_install 这些文件复制到设备的相应目录中

 <https://b3logfi e.com/file/2019/07/12-0ffdf94e.png?imageView2/2/interlace/1/format/jpg>

唔....编译好的 rtty 文件放入你嵌入式设备的/usr/bin/目录下

将 lib 中的文件放入你嵌入式设备的/lib 目录下, 记得创建软连接呀

(如果自己板卡上面缺少运行的共享库参考这篇文章 https://blog.csdn.net/rauaning/article/details/42463593)

将文件放好以后可以看下文件的版本信息查看是否可以执行

基本上显示版本都表示可以运行如果还有问题请查看自己的交叉环境, 是否缺少 rtty 运行库, ssl....
 <https://b3logfi e.com/file/2019/07/16-10bc5ccc.png?imageView2/2/interlace/1/format/jpg>

上边提到了在交叉编译过程中本人并没有加入 sll, 所以客户端, 服务端启动时都不要加入 ssl 启动

如果有人知道怎么在交叉编译过程中加入 sll 功能希望你可以留言给我大家一起学习学习 ((#^.^#))

呼.....到这里基本结束了, 如果有问题也可以私信, 留言, 或者邮箱联系 707201132@qq.com (#^.^#</p></p>