

Harmony OS 网络编程 实验指南

netcat简介

netcat 是什么？

netcat是一个非常强大的网络实用工具，可以用它来调试TCP/UDP应用程序；

netcat 如何安装？

Linux上可以使用发行版的包管理器安装，例如Debian/Ubuntu上：

```
* `sudo apt-get install netcat`
```

Windows上，MobaXterm工具上也可以用 `apt-get install netcat` 安装；

netcat 如何使用？

开始之前，先简单介绍一下 netcat 的几种用法：

1. TCP服务端模式： `netcat -l 5678`，会启动一个TCP服务器，监听 `5678` 端口，你可以换成其他端口；
2. TCP客户端模式： `netcat localhost 5678`，`localhost` 是目标主机参数，可以换成其他你想要连接的主机（主机名、IP地址、域名都可以），`5678` 是端口；
 - 你如果在同一台机器的两个终端中分别执行上述两条命令，它们两者之间就会建立连接一条TCP连接，此时你在其中一个终端上输入字符，敲回车就会发送到另一个终端中；
3. UDP服务端模式： `netcat -u -l 6789`，没错，只需要加一个 `-u` 参数，就可以启动一个UDP服务端；
4. UDP客户端模式： `netcat -u localhost 6789`，
 - 类似的，在同一台机器的两个终端中分别执行上述两条命令，他们两者之间也可以收发消息，只不过是UDP报文；

LwIP简介

LwIP是什么？

A Lightweight TCP/IP stack
一个轻量级的TCP/IP协议栈

详细介绍请参考LwIP项目官网：<https://savannah.nongnu.org/projects/lwip/>

LwIP在openharmony上的应用情况

目前，openharmony源码树有两份LwIP：

- 1. third_party/lwip
 - 源码形式编译
 - 供liteos-a内核使用
 - 还有一部分代码在kernel/liteos_a中，一起编译
- 2. vendor/hisi/hi3861/hi3861/third_party/lwip_sack
 - hi3861-sdk的一部分
 - 静态库形式编译
 - 不可修改配置
 - 可以查看当前配置（vend

下载源码

建议将本教程的源码下载到鸿蒙源码树的顶层目录，即和 applications、 build 等目录平级的地方：

- 执行命令 `git clone https://gitee.com/hihopeorg/HarmonyOS-IoT-Application-Development.git`

文件说明

文件：

文件名	说明
BUILD.gn	鸿蒙构建脚本，支持Hi3861、Hi3518、Hi3516开发板
demo_entry_cmsis.c	鸿蒙liteos-m程序入口，支持Hi3861
demo_entry_posix.c	鸿蒙liteos-a和Unix系统程序入口，Hi3516、Hi3518、PC
Makefile	Unix系统构建脚本，支持Linux/Mac OS
net_common.h	系统网络接口头文件
net_demo.h	demo脚手架头文件
net_params.h	网络参数，包括WiFi热点信息，服务器IP、端口信息
tcp_client_test.c	TCP客户端
tcp_server_test.c	TCP服务端
udp_client_test.c	UDP客户端
udp_server_test.c	UDP服务端
wifi_connecter.c	鸿蒙WiFi STA模式API的封装实现文件，比鸿蒙原始接口更容易使用
wifi_connecter.h	鸿蒙WiFi STA模式API的封装头文件，比鸿蒙原始接口更容易使用

Linux主机实验指南

编译测试程序

使用如下命令进行编译：

1. `make` 编译测试程序，该命令会生成4个可执行文件和几个.o文件
2. `make clean` 删除可执行程序 and .o文件

运行测试程序

Linux系统可以在同一台机器上，使用多个终端进行测试；

- TCP客户端测试：
 1. 在一个终端中使用netcat启动一个TCP服务端：`netcat -l 5678`；
 2. 在另一个终端中启动TCP客户端测试程序：`./tcp_client_test 5678 127.0.0.1`；
 3. 在netcat终端中应该会出现TCP客户端测试程序发来的：`Hello`，输入`world`并回车，`world`将会发送到测试程序所在终端；
- TCP服务端测试：
 1. 在一个终端中启动TCP服务端测试程序：`./tcp_server_test 5678`；
 2. 在另一个终端中使用netcat启动一个TCP客户端，并尝试连接测试程序：`netcat 127.0.0.1 5678`；
 3. 在netcat终端中输入`Hello`并回车，终端应该会再输出一行`Hello`，后面一行是TCP服务端测试程序发送回来的，同时终端上会有相关打印；
- UDP客户端测试：
 1. 在一个终端中使用netcat启动一个UDP服务端：`netcat -u -l 5678`；
 2. 在另一个终端中启动UDP客户端测试程序：`./udp_client_test 5678 127.0.0.1`；
 3. 在netcat终端中应该会出现UDP客户端测试程序发来的`Hello.`，输入`world`并回车，`world`将会发送到测试程序所在终端；
- UDP服务端测试：
 1. 在一个终端中启动UDP服务端测试程序：`./udp_server_test 5678`；
 2. 在另一个终端中使用netcat启动一个UDP客户端，并尝试连接测试程序：`netcat -u 127.0.0.1 5678`；
 3. 在netcat终端中输入`Hello.`并回车，终端应该会再输出一行`Hello.`，后面一行是UDP服务端测试程序是发送回来的，同时终端上会有相关打印；

Hi3861开发板实验指南

准备网络环境

在Hi3861开发板上运行上述四个测试程序之前，需要先准备网络环境：

1. 准备一个无线路由器，
2. 将Linux系统的PC连接到这个无线路由器，
 - 如果是虚拟机运行的Linux系统，需要通过网线连接路由器，并且将虚拟机网口设置为“桥接网卡”，确保**能够从路由器分到IP地址**（这一点非常重要，因为默认是NAT模式，只能通过虚拟机访问外部环境，外部环境无法通过网络访问虚拟机）；

修改网络参数

在Hi3861开发板上运行上述四个测试程序之前，需要根据你的无线路由、Linux系统IP修改 `net_params.h` 文件的相关代码：

- `PARAM_HOTSPOT_SSID` 修改为你的热点名称
- `PARAM_HOTSPOT_PSK` 修改为你的热点密码；
- `PARAM_SERVER_ADDR` 修改为你的Linux主机IP地址；

编译和烧录测试程序

在Hi3861开发板上运行上述四个测试程序，需要分别编译、烧录程序；

1. 需要修改 `BUILD.gn` 代码，取消你想要执行的测试程序的注释：
 - 编译TCP客户端测试程序，取消 `sources = ["tcp_client_test.c"]` 的注释，保留其余三行的 `sources` 注释；
 - 编译TCP服务端测试程序，取消 `sources = ["tcp_server_test.c"]` 的注释，保留其余三行的 `sources` 注释；
 - 编译UDP客户端测试程序，取消 `sources = ["udp_client_test.c"]` 的注释，保留其余三行的 `sources` 注释；
 - 编译UDP服务端测试程序，取消 `sources = ["udp_server_test.c"]` 的注释，保留其余三行的 `sources` 注释；
2. 确认你已经修改了hi3861开发板的编译配置文件 `build/lite/product/wifiot.json`：
 - `"//applications/sample/wifi-iot/app"` 替换为：`"//HarmonyOS-IoT-Application-Development:app"`
3. 使用 `python build.py wifiot` 进行编译；
4. 使用DevEco Device Tool或者HiBurn将二进制程序烧录到Hi3861开发板上；
5. 烧录成功后暂时不要复位程序；

运行测试程序

Hi3861开发板上运行测试程序的操作流程和Linux上大体相同，只是Linux终端执行测试程序变成了——开发板复位后自动运行；

- TCP客户端测试：
 1. 在Linux终端中使用netcat启动一个TCP服务端：`netcat -l 5678`；
 2. 连接开发板串口，复位开发板，板上程序启动后，首先会连接WiFi热点，然后会尝试连接到Linux上用netcat启动的TCP服务端；
 3. 在Linux终端中应该会出现开发板上TCP客户端通过发来的 `Hello`，输入 `world` 并回车，`world` 将会发送到开发板上，同时开发板的串口会有相关打印；
- TCP服务端测试：
 1. 重新修改 `BUILD.gn`，放开 `sources = ["tcp_server_test.c"]` 的注释，保留其他三行的注释，重新编译、烧录到开发板上；
 2. 连接开发板串口，复位开发板，板上程序启动后，会首先连接WiFi热点，然后会启动TCP服务端，并监听 `5678` 端口；
 - 这里需要从串口日志上查看开发板的IP地址，下一步操作需要用到

3. 在Linux终端中使用netcat启动一个TCP客户端，并尝试连接到开发板：`netcat board_ip 5678`，其中 `board_ip` 是开发板的IP地址；
 4. 在Linux终端中输入 `Hello` 并回车，终端应该会再输出一行 `Hello`，后面一行是开发板上TCP服务端序发送回来的，同时开发板的串口会有相关打印；
- UDP客户端测试：
 1. 重新修改BUILD.gn，放开 `sources = ["udp_client_test.c"]` 的注释，保留其他三行的注释，重新编译、烧录到开发板上；
 2. 在Linux终端中使用netcat启动一个UDP服务端：`netcat -u -l 5678`；
 3. 连接开发板串口，复位开发板，板上程序启动后，首先会连接WiFi热点，然后会尝试连接到Linux上用netcat启动的UDP服务端；
 4. 在Linux终端中应该会出现UDP客户端测试程序发来的 `Hello.`，输入 `world` 并回车，`world` 将会发送到开发板上，同时开发板的串口会有相关打印；
 - UDP服务端测试：
 1. 重新修改BUILD.gn，放开 `sources = ["udp_server_test.c"]` 的注释，保留其他三行的注释，重新编译、烧录到开发板上；
 2. 连接开发板串口，复位开发板，板上程序启动后，会首先连接WiFi热点，然后会启动UDP服务端，并监听 5678 端口；
 3. 在Linux终端中使用netcat启动一个UDP客户端，并尝试连接到开发板：`netcat -u 127.0.0.1 5678`；
 4. 在Linux终端中输入 `Hello.` 并回车，终端应该会再输出一行 `Hello.`，后面一行是UDP服务端测试程序是发送回来的，同时开发板的串口会有相关打印；

Hi3516开发板实验指南

准备网络环境

和在Hi3861开发板上运行测试程序类似，在Hi3516开发板上运行四个测试程序之前，需要先准备网络环境：

1. 准备一个无线路由器，
2. 将Linux系统的PC连接到这个无线路由器，
 - 如果是虚拟机运行的Linux系统，需要通过网线连接路由器，并且将虚拟机网口设置为“桥接网卡”，确保**能够从路由器分到IP地址**（这一点非常重要，因为默认是NAT模式，只能通过虚拟机访问外部环境，外部环境无法通过网络访问虚拟机）；

编译测试程序

1. 确认你已经修改了Hi3516开发板的编译配置文件 `build/lite/product/ipcamera_hi3516dv300.json`：
 - 向 `"//applications/sample/camera/app:camera_app"` 所在行之前添加一行配置：

```
{ "name": "net_test", "dir": "//HarmonyOS-IoT-Application-Development:app", "features": [] },
```
2. 使用 `python build.py ipcamera_hi3516dv300` 进行编译；
3. 编译成功后在 `out/ipcamera_hi3516dv300/bin` 目录下会生成四个测试程序：

```
$ find out/ipcamera_hi3516dv300/ -name *_test
out/ipcamera_hi3516dv300/bin/udp_server_test
out/ipcamera_hi3516dv300/bin/udp_client_test
out/ipcamera_hi3516dv300/bin/tcp_client_test
out/ipcamera_hi3516dv300/bin/tcp_server_test
```

将测试程序转移到开发板上

有多种方式可以将测试程序从你的Linux主机转移到Hi3516开发板上：

- 重新烧写rootfs.img映像文件到开发板；
 - 这种方式会刷写整个rootfs分区；
- 通过T-Flash拷贝；
 - 先将上述四个测试程序从Linux主机拷贝到T-Flash卡上，
 - 再将T-Flash卡从Linux主机上拔下来，插到Hi3516开发板上；
 - 然后将测试程序从T-Flash卡中拷贝到开发板文件系统上；
- 通过网络和TFTP传输；
 - 先确认Linux主机和开发板直接的IP地址在同一个网段；
 - 可以用ping命令互相测试；
 - 将Linux主机上的几个测试程序拷贝到tftp目录下；
 - Linux系统上安装和配置tftp服务，可参考：<https://community.arm.com/developer/tools-software/oss-platforms/w/docs/495/tftp-remote-network-kernel-using-u-boot>
 - 在开发上，通过tftp命令从Linux主机下载测试程序；
 - 先cd到一个有写权限的位置，例如 /usrdata/
 - 执行tftp下载命令：`tftp -g -l tcp_client_test -r tcp_client_test $LINUX_IP`，其中 `$LINUX_IP` 是你的Linux主机的IP地址

运行测试程序

Hi3861开发板上运行测试程序的操作流程和Linux上几乎完全相同，只是Linux终端执行测试程序变成了——在开发板上的串口终端中执行；

- TCP客户端测试：
 1. 在Linux终端中使用netcat启动一个TCP服务端：`netcat -l 5678`；
 2. 在开发板串口终端中启动TCP客户端测试程序：`./tcp_client_test 5678 $LINUX_IP`，其中 `$LINUX_IP` 是你的Linux主机的IP地址；
 3. 在netcat终端中应该会出现TCP客户端测试程序发来的：`Hello`，输入 `world` 并回车，`world` 将会发送到测试程序所在终端；
- TCP服务端测试：
 1. 在开发板串口终端中启动TCP服务端测试程序：`./tcp_server_test 5678`；
 2. 在Linux终端中使用netcat启动一个TCP客户端，并尝试连接测试程序：`netcat $BOARD_IP 5678`，其中 `$BOARD_IP` 是开发板的IP地址；
 3. 在Linux终端中输入 `Hello` 并回车，终端应该会再输出一行 `Hello`，后面一行是开发板上TCP服务端测试程序发送回来的，开发板串口终端上会有相关打印；
- UDP客户端测试：
 1. 在Linux终端中使用netcat启动一个UDP服务端：`netcat -u -l 5678`；

2. 在开发板串口终端中启动UDP客户端测试程序: `./udp_client_test 5678 $LINUX_IP`, 其中 `$LINUX_IP` 是你的Linux主机的IP地址;
 3. 在Linux终端中应该会出现UDP客户端测试程序发来的 `Hello.`, 输入 `world` 并回车, `world` 将会发送到开发板上, 开发板串口终端上会有相关打印;
- UDP服务端测试:
 1. 在开发板串口终端中启动UDP服务端测试程序: `./udp_server_test 5678`;
 2. 在Linux终端中使用netcat启动一个UDP客户端, 并尝试连接测试程序: `netcat -u $BOARD_IP 5678`, 其中 `$BOARD_IP` 是开发板的IP地址;
 3. 在Linux终端中输入 `Hello.` 并回车, 终端应该会再输出一行 `Hello.`, 后面一行是UDP服务端测试程序是发送回来的, 开发板串口终端上会有相关打印;

Hi3518开发板实验指南

准备网络环境

和在Hi3861开发板上运行测试程序类似, 在Hi3518开发板上运行四个测试程序之前, 需要先准备网络环境:

1. 准备一个无线路由器,
2. 将Linux系统的PC连接到这个无线路由器,
 - 如果是虚拟机运行的Linux系统, 需要通过网线连接路由器, 并且将虚拟机网口设置为“桥接网卡”, 确保**能够从路由器分到IP地址** (这一点非常重要, 因为默认是NAT模式, 只能通过虚拟机访问外部环境, 外部环境无法通过网络访问虚拟机);

Hi3518开发板连接WiFi热点

Hi3518连接WiFi热点需要使用wpa_supplicant和相关配置文件;

- 编译wpa_supplicant程序:
 1. 打开编译wpa_supplicant编译配置:
 - 修改配置文件 `applications\sample\camera\communication\BUILD.gn` 中的 `features`, 向其中添加:
`"wpa_supplicant:wpa_sample"`
 2. 执行编译命令: `python build.py ipcamera_hi3518ev300`
- 准备wpa_supplicant配置文件;

```
country=GB
ctrl_interface=udp
network={
    ssid="example" # 热点名称
    # key_mgmt=NONE # for OPEN Access Point
    psk="12345678" # 热点密码
}
```

- 将编译生成的wpa_supplicant程序和准备好的wpa_supplicant.conf拷贝到T-Flash卡, 然后将T-Flash卡插到开发板上;
- 运行wpa_supplicant程序, 开发板启动后, 在串口终端中:

- 执行命令：`cd sdcard`，跳转到T-Flash卡所在目录，
- 执行命令：`wpa_supplicant -i wlan0 -c wpa_supplicant.conf`，如果配置文件中的热点名称和密码没错的话，将会连接上WiFi热点；
- 执行命令：`ifconfig`，查看网络配置信息，连接WiFi热点成功之后，可以查看到IP地址；

编译测试程序

和Hi3516开发板的操作流程一致，参考前面的描述；

将测试程序转移到开发板上

和Hi3516开发板的操作流程一致，参考前面的描述；

运行测试程序

和Hi3516开发板的操作流程一致，参考前面的描述；