build 构建脚本说明

Git clone http://xiaodong@Yilian-001/BNLM1969/build.git

目录结构如下:

├── build.sh

├── ndn\_workspace

│   └── build\_ndn.sh

└── workspace

├── build\_clone\_openwrt.sh

├── build\_noclone\_openwrt.sh

其中

build.sh为整体编译脚本，

build\_ndn.sh为ndn编译脚本，

build\_clone\_openwrt.sh为openwrt和bsdr&noded编译脚本,clone openwrt git库，将git库中all目录下的文件拷贝到openwrt的编译目录下。

build\_noclone\_openwrt.sh为openwrt和bsdr&noded编译脚本，不clone openwrt git库，只将 baar git 库中的代码拷贝到openwrt的编译目录下baar包的src目录：

package/system/baar/src。

在workspace目录下，需要手动构建openwrt编译环境，目录结构如下：

├── build

├── build\_clone\_openwrt.sh

├── build\_noclone\_openwrt.sh

├── src

└── zkopenwrt\_uclib //需要手动构建

参考：

http://172.16.23.248/BNLM1969/openwrt/blob/master/Openwrt 编译说明.docx

build.sh执行完毕后，会生成bcr.deb,bsdr.deb,openwrt映像文件。

build.sh有5个选项：all、no\_clone\_openwrt、no\_ndn、no\_ndn\_openwrt、deb

使用方法如下：

./build.sh all

所有项目都重新编译

./build.sh no\_clone\_openwrt

由于openwrt固定后，就没有必要每次都clone openwrt代码库，只需要将baar代码库中的内容拷贝到openwrt编译目录下的package/system/baar/src中。

./build.sh no\_ndn

ndn不重新编译

./build.sh no\_ndn\_openwrt

ndn不重新编译，同时不clone openwrt代码库，只需要将baar代码库中的内容拷贝到openwrt编译目录下的package/system/baar/src中。

./build.sh deb

不编译，只打包deb

编译完成后会在build.sh所在的目录下建立result目录，result目录中包含以日期构建的目录，此目录下包含以编译时间构建的目录，此目录下包含所有编译得到的文件。

result

└── 20170629

└── 20170629-16:18:50

├── bcr.deb

├── bndeb

├── bsdr.deb

└── openwrt-ramips-mt7621-zk-wac5080-squashfs-sysupgrade.bin

其中bndeb是构建beb包的原始文件

bcr.deb是 x86 bcr设备使用的deb包

bsdr.deb是 x86 bsdr设备使用的deb包

openwrt-ramips-mt7621-zk-wac5080-squashfs-sysupgrade.bin

是mtk7621baar使用的映像文件。

说明：

由于ndn\_workspace是基础编译环境，如果没有修改，编译一次后就不用再编译了，所以不需要再次编译。Openwrt也是如此。