[14:48 0.23] kang39@WTH:d-oran $ pwd  
/home/kang39/work/ORAN/d-oran

* work/ORAN/에 d-oran 으로 git clone함.

[14:52 0.14] kang39@WTH:d-oran $ ifconfig

docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500

inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255

ether 02:42:85:35:e8:84 txqueuelen 0 (Ethernet)

RX packets 0 bytes 0 (0.0 B)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 0 bytes 0 (0.0 B)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s25: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.50.219 netmask 255.255.255.0 broadcast 192.168.50.255

inet6 fe80::7d08:6095:2982:f8e6 prefixlen 64 scopeid 0x20<link>

ether d8:9e:f3:21:40:ee txqueuelen 1000 (Ethernet)

RX packets 70680849 bytes 53789979395 (53.7 GB)

RX errors 0 dropped 3578516 overruns 0 frame 0

TX packets 6926636 bytes 3336436570 (3.3 GB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

device interrupt 20 memory 0xf7400000-f7420000

enp9s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.1.99 netmask 255.255.255.0 broadcast 192.168.1.255

inet6 fe80::da9e:f3ff:fe21:40f0 prefixlen 64 scopeid 0x20<link>

ether d8:9e:f3:21:40:f0 txqueuelen 1000 (Ethernet)

RX packets 14781000 bytes 8118302867 (8.1 GB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 4428377 bytes 4284606851 (4.2 GB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

device memory 0xf7300000-f737ffff

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536

inet 127.0.0.1 netmask 255.0.0.0

inet6 ::1 prefixlen 128 scopeid 0x10<host>

loop txqueuelen 1000 (Local Loopback)

RX packets 70738 bytes 20878282 (20.8 MB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 70738 bytes 20878282 (20.8 MB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

* 192.168.50.219 -> 개발서버 주소
* 192.168.1.99 -> 장비와 연결된 주소? 장비 주소는 1.99와 같이 1로 시작함. 장비에 build된 결과물 package를 장비에 넣을 때 장비에 접속해서 어떤 파일?의 주소를 99번으로 변경하고 적용하면 장비에 업로드되고 리셋 후 재시작함.

[14:52 0.13] kang39@WTH:d-oran $ \_sel

1. arm-none-linux-gnueabi ( 4.3.2 2008q3-72 ) - ( T2200 )

2. aarch64-xilinx-linux ( 8.2.0 - 2019.1 ) - ( DKK NOKIA LGE )

3. aarch64-xilinx-linux ( 11.2.0 - 2022.2 ) - ( UQ )

4. arm-none-linux-gnueabi ( 4.8.3 ) - ( JTower MU, HU )

5. aarch64-xilinx-linux ( 9.2.0 - 2020.2 ) - ( JTOWER OIU)

6. aarch64-xilinx-linux ( 12.2.0 - 2023.1 ) - ( UQ[DPD], N-DAS MU-RU, J-TOWER R1 )

7. aarch64-xilinx-linux ( 12.2.0 - 2023.2 ) - ( UPLUS\_5G\_MB\_VE )

8. aarch64-xilinx-linux ( 8.2.0 - 2019.1 ) - ( O-RAN(SSHv3) )

9. aarch64-xilinx-linux ( 8.2.0 - 2020.1 ) - ( O-RAN(MONOCLOCK))

10. arm-linux-gnueabi-gcc ( 7.5.0 ) - ( N-DAS HU )

11. aarch64-linux-gnu-gcc ( 8.2.0 - 2019.1 ) - ( DKK(old)A2K )

* 컴파일(build.sh)하기 전에 어느 SDK를 선택할지 컴파일 번호를 선택

텍스트, 폰트, 번호, 라인이(가) 표시된 사진

자동 생성된 설명

* gitlab에서 oran으로 검색. 담당해야 할 프로젝트가 oran으로 예상되므로.
* d-oran 프로젝트를 선택. SRU/RRU (NR방식 대응 3.4GHz대 저출력/고출력 무선 장치) - 본 프로젝트는 Docomo향 NR방식에 대응하는 3.4GHz대 무선 장치를 개발하는 것이며, SRU는 저출력, RRU는 고출력 무선 장치이다. Digital Hardware Block은 SRU와 RRU가 거의 유사하지만, 고출력 장치인 RRU에는 외부 안테나 장치와의 AISG(Antenna Interface Standards Group) interface가 추가로 존재한다.
* Source code는 gitlab server를 이용하여 이력 및 이슈가 관리되고 있으며, 여러 프로젝트들이 submodule로 관리된다. 따라서, main(or super) repository를 clone한 후에 추가적으로 submodule들도 가져와야 한다. 아래와 같이 clone한 후에 make update 명령을 사용하면 되며, 이후에는 항상 make update 명령만 사용하면 된다.

[15:04 0.09] kang39@WTH:ORAN $ git clone ssh://git@jupiter/sw/d-oran.git

'd-oran'에 복제합니다...

remote: Enumerating objects: 29, done.

remote: Counting objects: 100% (29/29), done.

remote: Compressing objects: 100% (27/27), done.

remote: Total 38960 (delta 9), reused 0 (delta 0), pack-reused 38931

오브젝트를 받는 중: 100% (38960/38960), 1.62 GiB | 27.78 MiB/s, 완료.

델타를 알아내는 중: 100% (27113/27113), 완료.

* 위와 같이 README 내용대로 해도 되고 아래와 같이 gitlab에서 clone 버튼을 눌러서 해도 된다.

[15:29 0.18] kang39@WTH:ORAN $ git clone git@121.0.140.43:sw/d-oran.git d-oran2

'd-oran2'에 복제합니다...

remote: Enumerating objects: 29, done.

remote: Counting objects: 100% (29/29), done.

remote: Compressing objects: 100% (27/27), done.

remote: Total 38960 (delta 9), reused 0 (delta 0), pack-reused 38931

오브젝트를 받는 중: 100% (38960/38960), 1.62 GiB | 27.42 MiB/s, 완료.

델타를 알아내는 중: 100% (27113/27113), 완료.

파일을 가져옵니다: 100% (826/826), 완료.

[15:32 0.20] kang39@WTH:d-oran2 $ pwd

/home/kang39/test/ORAN/d-oran2

[15:33 0.18] kang39@WTH:d-oran2 $ oran

[15:33 0.18] kang39@WTH:d-oran $ pwd

/home/kang39/work/ORAN/d-oran

* oran 은 작업 위치를 work/ORAN 디렉토리로 옮긴다.

[15:35 0.25] kang39@WTH:d-oran $ ll

합계 96

drwxrwxr-x 18 kang39 kang39 4096 9월 24 15:24 .

drwxrwxr-x 4 kang39 kang39 4096 9월 24 15:29 ..

drwxrwxr-x 8 kang39 kang39 4096 9월 24 15:24 .git

-rw-rw-r-- 1 kang39 kang39 213 9월 24 15:24 .gitattributes

-rw-rw-r-- 1 kang39 kang39 510 9월 24 15:24 .gitignore

-rw-rw-r-- 1 kang39 kang39 515 9월 24 15:24 .gitmodules

-rw-rw-r-- 1 kang39 kang39 6613 9월 24 15:24 Makefile

-rw-rw-r-- 1 kang39 kang39 1366 9월 24 15:24 README.md

drwxrwxr-x 6 kang39 kang39 4096 9월 24 15:24 app

drwxrwxr-x 2 kang39 kang39 4096 9월 24 15:24 configs

drwxrwxr-x 2 kang39 kang39 4096 9월 24 15:24 dhcpcd

drwxrwxr-x 5 kang39 kang39 4096 9월 24 15:24 hal

drwxrwxr-x 2 kang39 kang39 4096 9월 24 15:24 host

drwxrwxr-x 6 kang39 kang39 4096 9월 24 15:24 hw

drwxrwxr-x 2 kang39 kang39 4096 9월 24 15:24 images

drwxrwxr-x 4 kang39 kang39 4096 9월 24 15:24 libs

drwxrwxr-x 2 kang39 kang39 4096 9월 24 15:24 mk

drwxrwxr-x 2 kang39 kang39 4096 9월 24 15:24 nlib

drwxrwxr-x 2 kang39 kang39 4096 9월 24 15:24 pkg

drwxrwxr-x 2 kang39 kang39 4096 9월 24 15:24 plantuml

drwxrwxr-x 2 kang39 kang39 4096 9월 24 15:24 scripts

drwxrwxr-x 3 kang39 kang39 4096 9월 24 15:24 slab\_timing\_driver

drwxrwxr-x 5 kang39 kang39 4096 9월 24 15:24 tests

* 처음 clone했을 때 소스 구성

#

# GIT submodule settings

#

# ADRV9025 : ADI에서 받아서 setup

[submodule "hal/common/adrv9025"]

path = hal/common/adrv9025

url = ssh://git@jupiter/libs/adrv9025.git

branch = master

update = rebase

# internal developed N library

[submodule "nlib"]

path = nlib

url = ssh://git@jupiter/libs/nlib.git

branch = master

update = rebase

# https://github.com/NetworkConfiguration/dhcpcd.git

[submodule "dhcpcd"]

path = dhcpcd

url = ssh://git@jupiter/libs/dhcpcd.git

branch = v8.0/d-oran

update = rebase

* 처음 clone했을 때의 .gitmodules 내용

[15:38 0.08] kang39@WTH:d-oran $ gitb

\* master

remotes/origin/DKK/ASIA/m2.0-bw100m

remotes/origin/DKK/ASIA/master

remotes/origin/DKK/DOCOMO/700M\_2T2R

remotes/origin/DKK/DOCOMO/jig-capture

remotes/origin/DKK/DOCOMO/master

remotes/origin/DKK/DOCOMO/mplane\_2308

remotes/origin/DKK/DOCOMO/rpc\_timeout

remotes/origin/HEAD -> origin/master

remotes/origin/JT/DOCOMO/master

remotes/origin/LGE/2nd\_func

remotes/origin/LGE/CBRS

remotes/origin/LGE/master

remotes/origin/LGE/without\_synce\_2

remotes/origin/NOKIA/GLOBAL/P5G

remotes/origin/NOKIA/GLOBAL/adaptation-M2.0-M7.0

remotes/origin/NOKIA/GLOBAL/jesd204c\_fix

remotes/origin/NOKIA/GLOBAL/m2.0-bw100m

remotes/origin/NOKIA/GLOBAL/m2.0-bw80m-new-psu

remotes/origin/NOKIA/GLOBAL/m2.0-to-m7.0

remotes/origin/NOKIA/GLOBAL/master

remotes/origin/NOKIA/GLOBAL/mplane\_2303

remotes/origin/NOKIA/GLOBAL/mplane\_2305

remotes/origin/NOKIA/GLOBAL/mplane\_2308

remotes/origin/NOKIA/GLOBAL/mplane\_2401

remotes/origin/NOKIA/GLOBAL/mplane\_2403\_debugging

remotes/origin/NOKIA/GLOBAL/mplane\_2404

remotes/origin/NOKIA/GLOBAL/sfp\_10\_25\_dual\_mode

remotes/origin/NOKIA/SKT/m2.0-bw100m

remotes/origin/NOKIA/SKT/master

remotes/origin/NOKIA/UPLUS/m2.0-bw80m

remotes/origin/NOKIA/UPLUS/master

remotes/origin/master

* gitb 는 git branch -a 의 alias

[16:19 0.22] kang39@WTH:d-oran $ git branch -a

\* master

remotes/origin/DKK/ASIA/m2.0-bw100m

remotes/origin/DKK/ASIA/master

remotes/origin/DKK/DOCOMO/700M\_2T2R

remotes/origin/DKK/DOCOMO/jig-capture

remotes/origin/DKK/DOCOMO/master

remotes/origin/DKK/DOCOMO/mplane\_2308

remotes/origin/DKK/DOCOMO/rpc\_timeout

remotes/origin/HEAD -> origin/master

remotes/origin/JT/DOCOMO/master

remotes/origin/LGE/2nd\_func

remotes/origin/LGE/CBRS

remotes/origin/LGE/master

remotes/origin/LGE/without\_synce\_2

remotes/origin/NOKIA/GLOBAL/P5G

remotes/origin/NOKIA/GLOBAL/adaptation-M2.0-M7.0

remotes/origin/NOKIA/GLOBAL/jesd204c\_fix

remotes/origin/NOKIA/GLOBAL/m2.0-bw100m

remotes/origin/NOKIA/GLOBAL/m2.0-bw80m-new-psu

remotes/origin/NOKIA/GLOBAL/m2.0-to-m7.0

remotes/origin/NOKIA/GLOBAL/master

remotes/origin/NOKIA/GLOBAL/mplane\_2303

remotes/origin/NOKIA/GLOBAL/mplane\_2305

remotes/origin/NOKIA/GLOBAL/mplane\_2308

remotes/origin/NOKIA/GLOBAL/mplane\_2401

remotes/origin/NOKIA/GLOBAL/mplane\_2403\_debugging

remotes/origin/NOKIA/GLOBAL/mplane\_2404

remotes/origin/NOKIA/GLOBAL/sfp\_10\_25\_dual\_mode

remotes/origin/NOKIA/SKT/m2.0-bw100m

remotes/origin/NOKIA/SKT/master

remotes/origin/NOKIA/UPLUS/m2.0-bw80m

remotes/origin/NOKIA/UPLUS/master

remotes/origin/master

* d-oran 프로젝트에 존재하는 branch
* gitb alias를 사용해도 됨.
* Git clone한 후라 master 브랜치를 가리킴.

[16:20 0.07] kang39@WTH:d-oran $ git checkout DKK/DOCOMO/700M\_2T2R

'DKK/DOCOMO/700M\_2T2R' 브랜치가 리모트의 'DKK/DOCOMO/700M\_2T2R' 브랜치를 ('origin'에서) 따라가도록 설정되었습니다.

새로 만든 'DKK/DOCOMO/700M\_2T2R' 브랜치로 전환합니다

[16:23 0.03] kang39@WTH:d-oran $ git branch

\* DKK/DOCOMO/700M\_2T2R

master

* checkout

[16:28 0.06] kang39@WTH:d-oran $ make update

이미 업데이트 상태입니다.

'dhcpcd' 경로에 대해 'dhcpcd' (ssh://git@jupiter/libs/dhcpcd.git) 하위 모듈 등록

'hal/common/adrv9025' 경로에 대해 'hal/common/adrv9025' (ssh://git@jupiter/libs/adrv9025.git) 하위 모듈 등록

'm-plane\_libs/libnetconf2' 경로에 대해 'm-plane\_libs/libnetconf2' (git@121.0.140.43:libs/libnetconf2.git) 하위 모듈 등록

'm-plane\_libs/libpcre2' 경로에 대해 'm-plane\_libs/libpcre2' (git@121.0.140.43:libs/libpcre2.git) 하위 모듈 등록

'm-plane\_libs/libssh' 경로에 대해 'm-plane\_libs/libssh' (git@121.0.140.43:libs/libssh.git) 하위 모듈 등록

'm-plane\_libs/libyang' 경로에 대해 'm-plane\_libs/libyang' (git@121.0.140.43:libs/libyang.git) 하위 모듈 등록

'm-plane\_libs/netopeer2' 경로에 대해 'm-plane\_libs/netopeer2' (git@121.0.140.43:libs/netopeer2.git) 하위 모듈 등록

'm-plane\_libs/openssl' 경로에 대해 'm-plane\_libs/openssl' (git@121.0.140.43:libs/openssl.git) 하위 모듈 등록

'm-plane\_libs/sysrepo' 경로에 대해 'm-plane\_libs/sysrepo' (git@121.0.140.43:libs/sysrepo.git) 하위 모듈 등록

'nlib' 경로에 대해 'nlib' (ssh://git@jupiter/libs/nlib.git) 하위 모듈 등록

'slab' 경로에 대해 'slab' (ssh://git@jupiter/libs/slab\_timing\_driver.git) 하위 모듈 등록

'sync' 경로에 대해 'sync' (git@121.0.140.43:libs/accutime.git) 하위 모듈 등록

'/home/kang39/test/ORAN/d-oran/dhcpcd'에 복제합니다...

'/home/kang39/test/ORAN/d-oran/hal/common/adrv9025'에 복제합니다...

'/home/kang39/test/ORAN/d-oran/m-plane\_libs/libnetconf2'에 복제합니다...

'/home/kang39/test/ORAN/d-oran/m-plane\_libs/libpcre2'에 복제합니다...

'/home/kang39/test/ORAN/d-oran/m-plane\_libs/libssh'에 복제합니다...

'/home/kang39/test/ORAN/d-oran/m-plane\_libs/libyang'에 복제합니다...

'/home/kang39/test/ORAN/d-oran/m-plane\_libs/netopeer2'에 복제합니다...

'/home/kang39/test/ORAN/d-oran/m-plane\_libs/openssl'에 복제합니다...

'/home/kang39/test/ORAN/d-oran/m-plane\_libs/sysrepo'에 복제합니다...

'/home/kang39/test/ORAN/d-oran/nlib'에 복제합니다...

'/home/kang39/test/ORAN/d-oran/slab'에 복제합니다...

'/home/kang39/test/ORAN/d-oran/sync'에 복제합니다...

하위 모듈 경로 'dhcpcd': '05fe2fedd8dd65c007cca073d98d7d70c2dc726d' 체크아웃

하위 모듈 경로 'hal/common/adrv9025': 'f2c692aa9160853f2499bfb962fa645aab9e83e1' 체크아웃

하위 모듈 경로 'm-plane\_libs/libnetconf2': 'aa3b6dec0938bbd849a172c33945479a2b167baa' 체크아웃

하위 모듈 경로 'm-plane\_libs/libpcre2': '73a5b03e7566ae0e5221a0970206ed5ac0f6309a' 체크아웃

하위 모듈 경로 'm-plane\_libs/libssh': 'e135765662da00f9964232c1de7f6ab27de29829' 체크아웃

하위 모듈 경로 'm-plane\_libs/libyang': 'b1280b93a331f14460d9ae0f3a7336eee2ac0ec4' 체크아웃

하위 모듈 경로 'm-plane\_libs/netopeer2': 'c970d9638ba651ca01bea6c8bd1323e020435a93' 체크아웃

하위 모듈 경로 'm-plane\_libs/openssl': 'b89a4f6f9bbeb24c610f8765948454cb1beec6fa' 체크아웃

하위 모듈 경로 'm-plane\_libs/sysrepo': 'd771cfe822ee4778d8b4ad05508c994628813834' 체크아웃

하위 모듈 경로 'nlib': 'a7aa0f86f4a2430380f2bdfa540055751f5dabd2' 체크아웃

하위 모듈 경로 'slab': 'af6b31b4625251bd7feb9f9b7fb027a76cc90f75' 체크아웃

하위 모듈 경로 'sync': '2a8091e5ab78bf070015fa2f56102de79b4861df' 체크아웃

* .gitmodules가 변경됨. 필요한 모듈을 연결시켜줌?
* m\_plane\_libs 나 slab, sync등의 소스가 보임.

[16:39 0.09] kang39@WTH:d-oran $ cd m-plane\_libs/

* open stack이 있는 곳

[16:40 0.07] kang39@WTH:m-plane\_libs $ make clean

Clean: libyang

Clean: sysrepo

Clean: libssh

Clean: libnetconf2

Clean: lib\_pubkey

make[1]: 디렉터리 '/home/kang39/test/ORAN/d-oran/m-plane\_libs/lib\_pubkey' 들어감

rm -rf lib\_pubkey.o libpubkey.so

make[1]: 디렉터리 '/home/kang39/test/ORAN/d-oran/m-plane\_libs/lib\_pubkey' 나감

Clean: netopeer2

Clean: libpcre2

* make clean

[16:43 0.14] kang39@WTH:m-plane\_libs $ \_sel

1. arm-none-linux-gnueabi ( 4.3.2 2008q3-72 ) - ( T2200 )

2. aarch64-xilinx-linux ( 8.2.0 - 2019.1 ) - ( DKK NOKIA LGE )

3. aarch64-xilinx-linux ( 11.2.0 - 2022.2 ) - ( UQ )

4. arm-none-linux-gnueabi ( 4.8.3 ) - ( JTower MU, HU )

5. aarch64-xilinx-linux ( 9.2.0 - 2020.2 ) - ( JTOWER OIU)

6. aarch64-xilinx-linux ( 12.2.0 - 2023.1 ) - ( UQ[DPD], N-DAS MU-RU, J-TOWER R1 )

7. aarch64-xilinx-linux ( 12.2.0 - 2023.2 ) - ( UPLUS\_5G\_MB\_VE )

8. aarch64-xilinx-linux ( 8.2.0 - 2019.1 ) - ( O-RAN(SSHv3) )

9. aarch64-xilinx-linux ( 8.2.0 - 2020.1 ) - ( O-RAN(MONOCLOCK))

10. arm-linux-gnueabi-gcc ( 7.5.0 ) - ( N-DAS HU )

11. aarch64-linux-gnu-gcc ( 8.2.0 - 2019.1 ) - ( DKK(old)A2K )

6

다음 명령어를 실행하세요

. /opt/petalinux/2023.1/environment-setup-cortexa72-cortexa53-xilinx-linux

. /opt/petalinux/2023.1/settings.sh

-v

PATH => /opt/petalinux/2023.1/sysroots/x86\_64-petalinux-linux/usr/bin/aarch64-xilinx-linux/

PetaLinux environment set to '/opt/petalinux/2023.1'

INFO: Checking free disk space

INFO: Checking installed tools

INFO: Checking installed development libraries

INFO: Checking network and other services

* \_sel 로 6번 선택?

[16:47 0.14] kang39@WTH:m-plane\_libs $ ./build.sh

아래 옵션 중 한 가지를 입력하여 진행합니다.

(옵션이 없이 Enter 시 install 및 SSLv1.x로 진행)

Option:

- clean: clean libs.

- ssl3: OpenSSL v3.x 적용.

Option을 입력 하세요:

./make\_toolchain.sh

TOOLCHAIN :/home/kang39/test/ORAN/d-oran/m-plane\_libs/out/ToolChain.cmake

DESTDIR :/home/kang39/test/ORAN/d-oran/m-plane\_libs/out

…

…

…

[ 95%] Building C object cli/CMakeFiles/netopeer2-cli.dir/\_\_/compat/compat.c.o

[100%] Linking C executable ../netopeer2-cli

[100%] Built target netopeer2-cli

-rw-rw-r-- 1 kang39 kang39 7767390 9월 24 16:50 out/mplane-lib.tar.gz

out/mplane-lib.tar.gz was generated

* mplane-lib.tar.gz 파일이 만들어짐.

[16:50 1.00] kang39@WTH:m-plane\_libs $ cm

/home/kang39/test/ORAN/d-oran/m-plane\_libs

* cm은 .bashrc에 있는 alias이다. 'rm -f ../libs/m-plane/mplane-lib.tar.gz;cp out/mplane-lib.tar.gz ../libs/m-plane; cd ../libs/m-plane;./update.sh ; cd -'
* update.sh의 위치는 /home/kang39/work/ORAN/d-oran2/libs/m-plane/update.sh

[18:31 0.01] kang39@WTH:m-plane\_libs $ cd ..

[18:31 0.01] kang39@WTH:d-oran $ git status

현재 브랜치 DKK/DOCOMO/700M\_2T2R

브랜치가 'origin/DKK/DOCOMO/700M\_2T2R'에 맞게 업데이트된 상태입니다.

커밋하도록 정하지 않은 변경 사항:

(무엇을 커밋할지 바꾸려면 "git add <파일>..."을 사용하십시오)

(작업 폴더의 변경 사항을 버리려면 "git checkout -- <파일>..."을 사용하십시오)

수정함: libs/m-plane/aarch64/rootfs/usr/local/bin/netopeer2-cli

수정함: libs/m-plane/aarch64/rootfs/usr/local/bin/netopeer2-server

수정함: libs/m-plane/aarch64/rootfs/usr/local/bin/pcre2grep

수정함: libs/m-plane/aarch64/rootfs/usr/local/bin/pcre2test

수정함: libs/m-plane/aarch64/rootfs/usr/local/bin/sysrepo-plugind

수정함: libs/m-plane/aarch64/rootfs/usr/local/bin/sysrepocfg

수정함: libs/m-plane/aarch64/rootfs/usr/local/bin/sysrepoctl

수정함: libs/m-plane/aarch64/rootfs/usr/local/bin/yanglint

수정함: libs/m-plane/aarch64/rootfs/usr/local/bin/yangre

수정함: libs/m-plane/aarch64/rootfs/usr/local/lib/libnetconf2.so.3.6.1

수정함: libs/m-plane/aarch64/rootfs/usr/local/lib/libpcre2-8.a

수정함: libs/m-plane/aarch64/rootfs/usr/local/lib/libpcre2-posix.a

수정함: libs/m-plane/aarch64/rootfs/usr/local/lib/libssh.so.4.9.0

수정함: libs/m-plane/aarch64/rootfs/usr/local/lib/libsysrepo.so.7.19.3

수정함: libs/m-plane/aarch64/rootfs/usr/local/lib/libyang.so.2.38.14

커밋할 변경 사항을 추가하지 않았습니다 ("git add" 및/또는 "git commit -a"를

사용하십시오)

* build 결과로 나온 파일의 압축을 풀면서 변경이 생긴 것 같음.

[17:05 0.05] kang39@WTH:out $ pwd

/home/kang39/work/ORAN/d-oran/out

[17:05 0.05] kang39@WTH:out $ ll

합계 19056

drwxrwxr-x 3 kang39 kang39 4096 9월 24 11:48 .

drwxrwxr-x 24 kang39 kang39 4096 9월 24 15:38 ..

drwxrwxr-x 9 kang39 kang39 4096 9월 24 11:29 app-rru

-rw-rw-r-- 1 kang39 kang39 9006 9월 24 11:40 autoconf-rru.h

-rw-rw-r-- 1 kang39 kang39 19487306 9월 24 11:48 rru\_ap.pkg

* out 디렉토리에 package 파일이 들어 있음?

[m-plane\_libs 디렉토리]

* M-plane 지원을 위한 라이브러리들
* Build: Petalinux 2019.1 설치 및 환경 설정이 되어 있어야 한다.
* M-plane 라이브러리와 프로그램 빌드: build.sh 명령이 성공적으로 진행되면 out/m-plane.tar.gz파일이 생성된다.
* 생성된 tarball을 application source tree에 설치하여 사용하면 되며, d-oran application project의 경우 libs/m-plane 폴더에 생성된 tarball을 복사한 후에 같은 폴더에 있는 update.sh 스크립트를 실행하면 된다.

[hw 디렉토리]

* 이 폴더에서는 각 보드별로 달라져야하는 파일들을 관리한다.
* 각 보드별로 장비에 다르게 설치되어야 하는 파일들은 보드 이름 아래에 rootfs/ 폴더에 장비와 동일한 folder tree를 생성한 후에 넣어두면 AP package에 그대로 포함되어 장비에 설치된다.

[configs/our\_defconfig]

* Machine info 등을 입력?