93. Application guide

First build를 하게되면 ${HOME} 에 riscv-toolchain 이 생성되도록 되어있다.

따라서 application 개발시 이 toolchain을 사용하면된다.

이후 application또는 user library를 embedding 하고자 한다면 yocto 에 recipe를 만들어주면 된다.

1. riscv-toolchain 사용

command	description	
$ cd myapp $ export RISCV=$HOME/riscv-toolchain $ export PATH=$PATH:$RISCV/bin	환경 변수 설정	
$ vim helloworld.c	코드 작성	물론 makefile을 만들어 사용해도 된다.
$ riscv64-unknown-elf-gcc helloworld.c -o helloworld	compile	
$ cp helloworld ~/tmp/mnt	sdcard에 copy DroneSoC board에서 실행해본다.	

2. yocto recipe 작성 방법

이미 yocto/riscv-poky/meta-nexell 에 example 이 구현되어있다. (yocto/riscv-poky/meta-nexell/recipes-application 참고)

따라서 makefile 이 존재하는 경우에 대해서 example을 추가하는 방법을 설명한다.

순서	description	
1	$ mkdir -p yocto/riscv-poky/meta-nexell/recipes-application/myapp $ cd yocto/riscv-poky/meta-nexell/recipes-application/myapp	다른 recipes dir에 넣어도 상관없다.

2	$ vi myapp.bb
	
	1 #
	2 # This file was derived from the 'Hello World!' example recipe in the
	3 # Yocto Project Development Manual.
	4 #
	5
	6 SUMMARY = "Simple myapp application"
	7 SECTION = "examples"
	8 LICENSE = "MIT"
	9 LIC_FILES_CHKSUM = "file://${COMMON_LICENSE_DIR}/MIT;md5=0835ade698e0bcf8506ecda2f7b4f302"
	10
	11 SRC_URI = "file://myapp.c ₩
	12 file://Makefile ₩
	13 "
	14
	15 S = "${WORKDIR}"
	16
	17 do_compile() {
	18 oe_runmake CC="$CC" clean
	19 oe_runmake CC="$CC"
	20 }
	21
	22 do_install() {
	23 install -d ${D}${bindir}
	24 install -m 0755 myapp ${D}${bindir}
	25 }
	26
	27 #NEXELL appended code
	28 INSANE_SKIP_${PN} = "ldflags"
	29 PACKAGE_ARCH = "${MACHINE_ARCH}"
	30 FILES_${PN} = "${bindir}"
		recipe 작성

3	$ mkdir -p yocto/riscv-poky/meta-nexell/recipes-application/myapp/files
	
	$ cd yocto/riscv-poky/meta-nexell/recipes-application/myapp
	
	$ vim Makefile
	
	1 TOOLCHAIN ?= riscv64-unknown-elf-
	2 CC ?= $(TOOLCHAIN)gcc
	3 OBJS = myapp.o
	4 TARGET = myapp
	5
	6 .SUFFIXES : .c .o
	7
	8 all : $(TARGET)
	9
	10 $(TARGET): $(OBJS)
	11 $(CC) -o $@ $(OBJS)
	12
	13 clean :
	14 rm -f $(OBJS) $(TARGET)
		Makefile 작성

4	$ vim yocto/riscv-poky/meta-nexell/recipes-core/packagegroups/packagegroup-nexell.bb 9 RDEPENDS_${PN} = " ₩ 10 helloworld-nexell ₩ 11 android-tools-nexell ₩ 12 bash-completion ₩ 13 myapp ₩ 14 "	myapp 추가

5	$ cd repoRISCV $ ./tools/build.sh -b drone -t yocto $ ./tools/build.sh -b drone	yocto build를 실행하면 ramdisk가 생성된다. ramdisk는 vmlinux 생성시 import시켜주어야 하기때문에 yocto build후 full build를 한번 더 실행해 준다.
6	(on DroneSoC board 에서) riscv64 login: root (automatic login) root@riscv64:~# cd /usr/bin/ root@riscv64:/usr/bin# ./m mapscrn md5sum mesg mesg.sysvinit microcom mkfifo myapp root@riscv64:/usr/bin# ./myapp Hello World with RISCV! yocto by NEXELL root@riscv64:/usr/bin#	board에서 실행 확인