

# 망고100 보드로 놀아보자-6

U-boot build 환경 분석

# U-boot build 환경 분석

- Board/\$(Vendor)/\$(BOARD)/Config.mk 파일 분석  
(U-boot 가 실행하는 주소 정의)

```
ifndef TEXT_BASE
```

```
//TEXT_BASE = 0xc7e00000 :MMU 사용
```

```
TEXT_BASE = 0x2fe00000 # Physical Address
```

```
endif
```

# U-boot 1.3.4 Build 환경 분석

- #make mango100\_config 실행 시 분석

\$(Top dir)/Makefile 에 아래 코드 실행

```
mango100_config :          1          unconfig
    @$(MKCONFIG) $(@:_config=) 2 arm s5pc1xx mango100 samsung s5pc100 3 4 5 6 7
```

```
MKCONFIG := $(SRCTREE)/mkconfig
export MKCONFIG
```

- ① Configuration 할 보드 이름 정의
- ② mkconfig 로 \$1인자 의미
- ③ Architecture 인자 \$2인자
- ④ CPU :\$3인자
- ⑤ Board name :\$4
- ⑥ VENDOR :\$5
- ⑦ SOC :\$6

# U-boot build 분석

```
mango100_config :      unconfig
    @$(MKCONFIG) $(@:_config=) arm s5pc1xx mango100 samsung s5pc100
```

실행 순서는

① unconfig 실행

```
unconfig:
    arm -f $(obj)include/config.h $(obj)include/config.mk \
        $(obj)board/*/config.tmp $(obj)board/*/*/config.tmp \
        $(obj)include/autoconf.mk $(obj)include/autoconf.mk.dep
```

② @\$\$(MKCONFIG) \$(@:\_config=) arm s5pc1xx mango100 samsung s5pc100

•include/config.h : include/configs/mango100.h 를 include,mkconfig 에서 create (아래코드)

```
echo "/* Automatically generated - do not edit */" >>config.h
echo "#include <configs/$1.h>" >>config.h
```

•Include/config.mk:

```
ARCH = arm
CPU = s5pc1xx
BOARD = mango100
VENDOR = samsung
SOC = s5pc100
```

내용

•board/\*/config.tmp, board/\*/\*/config.tmp : 존재하면 삭제

•Include/autoconfig.mk :makefile 에서 생성

•Include/autoconf.mk.dep : makefile에서 생성

# U-boot build 환경 분석

1

```
mango100_config :      unconfig
    @$(MKCONFIG) $(@:_config=) arm s5pc1xx mango100 samsung s5pc100
```

2

3

4

5

6

7

- `$(@:_config=)` : @현재 target 값 , 즉 mango100\_config, \_config=는 null 변경하라는 의미
- 매크로 치환 (Macro substitution): <http://cafe.naver.com/embeddedcrazyboys/7000>
- <http://marvel.incheon.ac.kr/infomation/unix/makefile/GNU-Make-4.html>

```
[ "${BOARD_NAME}" || BOARD_NAME="$1"
[ $# -lt 4 ] && exit 1
[ $# -gt 6 ] && exit 1
echo "Configuring for ${BOARD_NAME} board..."
```

- “\$1”값은 mango100
- `${BOARD_NAME}` : mango100 값을 가진다.
- `$#` : 넘어오는 인자 수를 의미
- 출력은 “Configuring for mango100 board...”

# U-boot build 환경 분석

1

```
mango100_config :          unconfig
                        @$(MKCONFIG) $(@:_config=) arm s5pc1xx mango100 samsung s5pc100
```

2

3

4

5

6

7

```
if [ "$SRCTREE" != "$OBJTREE" ] ; then
    mkdir -p ${OBJTREE}/include
    mkdir -p ${OBJTREE}/include2
    cd ${OBJTREE}/include2
    rm -f asm
    ln -s ${SRCTREE}/include/asm-$2 asm
    LNPREFIX=" ../../include2/asm/"
    cd ../include
    rm -rf asm-$2
    rm -f asm
    mkdir asm-$2
    ln -s asm-$2 asm
else
    cd ../include
    rm -f asm
    ln -s asm-$2 asm
fi

rm -f asm-$2/arch

if [ -z "$6" -o "$6" = "NULL" ] ; then
    ln -s ${LNPREFIX}arch-$3 asm-$2/arch
else
    ln -s ${LNPREFIX}arch-$6 asm-$2/arch
fi
```

• SRCTREE와 OBJTREE값은 동일 ,else문 수행

• \$2 값은 arm

Include/asm → Include/asm-arm

Symbol link

• [-z "\$6": 파일이 존재하지 않으면 참  
• -o : 논리적 OR 연산

Include/asm-arm/arch → Include/arch-s5pc1xxx

Symbol link

# U-boot build 환경 분석

1

```
mango100_config :          unconfig
    @$(MKCONFIG) $(@:_config=) arm s5pc1xx mango100 samsung s5pc100
```

2

3

4

5

6

7

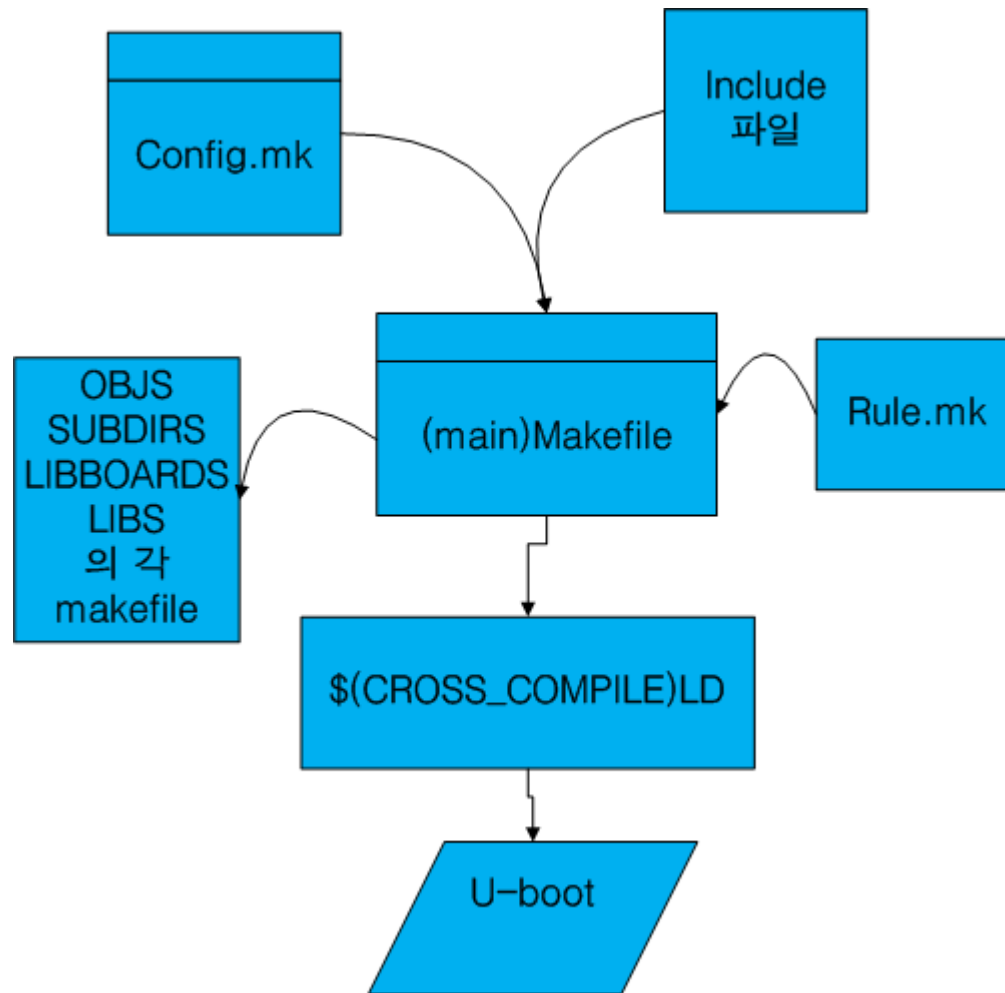
```
if [ "$APPEND" = "yes" ]      # Append to existing config file
then
    echo >> config.h
else
    > config.h                # Create new config file
fi
echo "/* Automatically generated - do not edit */" >>config.h
echo "#include <configs/$1.h>" >>config.h
exit 0
```

Config.h 를 만든다

Config.h 파일 내용은 “include <configs/mango100.h>” 이다

```
[icanjji@localhost include]$ cat config.h
/* Automatically generated - do not edit */
#include <configs/mango100.h>
```

# U-boot Build 환경 구조도





# 리눅스 명령 find (TIP)

- #find . -name "\*.ch" -exec grep 문자열 {} \; -print
- 예) find . -name "\*.ch" -exec grep mango100 {} \; -print
- 스크립트 추가 방법

```
if [ "$1" != "" ] && [ "$2" != "" ]; then
    find . -name "$1" -exec grep "$2" {} \; -print
else
    echo "Usage: $0 *.* abcd"
    exit 1
fi
```

- /usr/bin 디렉토리 vi fin 추가
- #chmod 755 /usr/bin/fin

# U-boot build 실행 분석

#make 명령 실행

\$(TOPDIR)/config.mk  
\$(TOPDIR)/arm\_config.mk  
\$(TOPDIR)/rules.mk  
...

```
include $(obj)include/autoconf.mk.dep
```

```
all: $(ALL)
```

```
ALL += $(obj)u-boot.srec $(obj)u-boot.bin $(obj)System.map $(U_BOOT_NAND) $(U_BOOT_ONENAND) $(obj)u-boot.dis  
ifeq ($(ARCH),blackfin)  
ALL += $(obj)u-boot.ldr  
endif
```

```
$(obj)u-boot.srec: $(obj)u-boot  
$(OBJCOPY) ${OBJCFLAGS} -O srec $< $@
```

```
$(obj)u-boot: depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)  
UNDEF_SYM=`$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | \#  
sed -n -e 's/.*\#((${SYM_PREFIX})__u_boot_cmd_.*\#)/-u\#|/p'|sort|uniq`; \#  
cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) \#  
--start-group $(__LIBS) --end-group $(PLATFORM_LIBS) \#  
-Map u-boot.map -o u-boot
```

# U-boot build 실행 분석

```
sinclude $(obj)include/autoconf.mk.dep
```

```
$(obj)include/autoconf.mk.dep: $(obj)include/config.h include/common.h  
@$(XECHO) Generating $@ : \  
set -e ; \  
: Generate the dependancies ; \  
$(CC) -x c -DDO_DEPS_ONLY -M $(HOST_CFLAGS) $(CPPFLAGS) \  
-MQ $(obj)include/autoconf.mk include/common.h > $@
```

include/config.h == include/configs/mango100.h 내용 동일  
Include/common.h 두개의 파일의 내용을 가지고 온다

\$(CC): \$(CROSS\_COMPILE)gcc 값임  
-M 옵션: 의존성 있는 있는 파일을 구성해 준다.  
-MQ (target) : target 이름을 정의 해 준다.

## 두개의 예제를 결과 비교 실습

```
#> vi hello.c  
#include<stdio.h>  
int main(void) {  
    printf("Hello Mango");  
    return 0;  
}  
#> arm-linux-gcc -M hello.c
```

```
#> vi hello.c  
#include<stdio.h>  
int main(void) {  
    printf("Hello Mango");  
    return 0;  
}  
#> arm-linux-gcc -M -MQ foo.o hello.c
```

# U-boot build 실행 분석

```

1      2      3      4      5      6
$(obj)u-boot:      depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)
7      UNDEF_SYM=$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | \
      sed -n -e 's/.*\#($$SYM_PREFIX)__u_boot_cmd_.*\#)/-u\1/p'|sort|uniq;\#
      cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) \#
      --start-group $(__LIBS) --end-group $(PLATFORM_LIBS) \#
      -Map u-boot.map -o u-boot
  
```

```

depend dep:      $(VERSION_FILE)
      for dir in $(SUBDIRS) ; do $(MAKE) -C $$dir _depend ; done
  
```

```

$(VERSION_FILE):
      @( printf '#define U_BOOT_VERSION "U-Boot %s%s"\n' "$(U_BOOT_VERSION)" \
      '$(shell $(CONFIG_SHELL) $(TOPDIR)/tools/setlocalversion $(TOPDIR))' \
      ) > $$@.tmp
      @cmp -s $$@ $$@.tmp && rm -f $$@.tmp || mv -f $$@.tmp $$@
  
```

```

VERSION = 1
PATCHLEVEL = 3
SUBLEVEL = 4
EXTRAVERSION =
U_BOOT_VERSION = $(VERSION).$(PATCHLEVEL).$(SUBLEVEL)$(EXTRAVERSION)
VERSION_FILE = $(obj)include/version_autogenerated.h
  
```

cmp -s file1 file2 : 비교하여 같으면 0  
 && 연산자 : cmp 비교 결과 같으면 실행  
 || 연산자 : cmp 비교 결과 다르면 실행

# U-boot build 실행 분석

```

1      2      3      4      5      6
$(obj)u-boot:      depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)
7      UNDEF_SYM=$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | \#
      sed -n -e 's/.*\#((${SYM_PREFIX})__u_boot_cmd_.*\#)/-u\1/p'|sort|uniq`;\#
      cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) \#
      --start-group $(__LIBS) --end-group $(PLATFORM_LIBS) \#
      -Map u-boot.map -o u-boot
  
```

```

depend dep:      $(VERSION_FILE)
                for dir in $(SUBDIRS) ; do $(MAKE) -C $$dir _depend ; done
  
```

```

SUBDIRS = tools \
          examples \
          api_examples
  
```

\$\$dir :tools, examples api\_example 값을 가진다.

```

make -C tools _depend
make -C examples _depend
make -C api_example _depend
  
```

형식:make -C 실행디렉토리 타겟  
 즉, -C 옵션에 정의된 디렉토리로 가서 Makefile을 수행하며,  
 target은 \_depend 임

```

_depend:      $(obj).depend
$(obj).depend: $(src)Makefile $(TOPDIR)/config.mk $(SRCS)
                @rm -f $@
                @for f in $(SRCS); do \
                    g=`basename $$f | sed -e 's/\(.*\)\.o/'`; \
                    $(CC) -M $(HOST_CFLAGS) $(CPPFLAGS) -MQ $(obj)$$g $$f >> $@ ; \
                done
  
```

# U-boot build 실행 분석

```
1 2 3 4 5 6
$(obj)u-boot: depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)
7 UNDEF_SYM=$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | \
  sed -n -e 's/.*\#$(SYM_PREFIX)__u_boot_cmd_.*\#/-u\1/p'|sort|uniq;\#
  cd $(LNDIR) && $(LD) $(LD_FLAGS) $$UNDEF_SYM $(__OBJS) \#
  --start-group $(__LIBS) --end-group $(PLATFORM_LIBS) \#
  -Map u-boot.map -o u-boot
```

```
$(SUBDIRS): depend $(obj)include/autoconf.mk
$(MAKE) -C $@ all
```

Make -C \$SUBDIRS all 수행되면서 실제 빌드됨

```
$(obj)include/autoconf.mk: $(obj)include/config.h
$(X_ECHO) Generating $@ : \
set -e ; \
: Extract the config macros ; \
$(CPP) $(C_FLAGS) -DDO_DEPS_ONLY -dM include/common.h | \
sed -n -f tools/scripts/define2mk.sed > $@
```

결과

```
CONFIG_BOOTP_BOOTPATH=y
CONFIG_DISPLAY_CPUINFO=y
CONFIG_SETUP_MEMORY_TAGS=y
CONFIG_SERIAL2=y
CONFIG_CMD_LOADB=y
CONFIG_CMD_LOADS=y
CONFIG_CMD_IMI=y
CONFIG_ARM=y
CONFIG_CMD_BDI=y
CONFIG_SERVERIP="192.168.1.2"
CONFIG_BOOTP_SUBNETMASK=y
CONFIG_CMD_REGINFO=y
CONFIG_S3C_USBD=y
CONFIG_S5PC1XX=y
"include/autoconf.mk" 77L, 1781C
```

# U-boot build 실행 분석

```
1      2      3      4      5      6
$(obj)u-boot:      depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)
7      UNDEF_SYM=$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | #
      sed -n -e 's/.*#$(SYM_PREFIX)__u_boot_cmd_.*#/-u#1/p'|sort|uniq`;#
      cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) #
      --start-group $(__LIBS) --end-group $(PLATFORM_LIBS) #
      -Map u-boot.map -o u-boot
```

```
$(OBJS):      depend $(obj)include/autoconf.mk
      $(MAKE) -C cpu/$(CPU) $(if $(REMOTE_BUILD),,$@,$(notdir $@))
```

CPU 값은 s5pc1xx

\$(if \$(REMOTE\_BUILD),,\$@,\$(notdir \$@)) 의미는

REMOTE\_BUILD 값이 있으면, OBJS 값을 그대로 대입,

없으면, 디렉토리를 제외한 파일이름만 추출하라는 의미

make -C cpu/s5pc1xx start.o 이 수행이 됨

# U-boot build 실행 분석(LIBBOARD)

- 1
- 2
- 3
- 4
- 5
- 6

```
$(obj)u-boot:          depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)
                        UNDEF_SYM=$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | #
                        sed -n -e 's/.*#$(SYM_PREFIX)__u_boot_cmd_.*#/-u#1/p'|sort|uniq`:#
7 cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) #
                        --start-group $(__LIBS) --end-group $(PLATFORM_LIBS) #
                        -Map u-boot.map -o u-boot
```

```
$(LIBBOARD):          depend $(LIBS) $(obj)include/autoconf.mk
                        $(MAKE) -C $(dir $(subst $(obj),,$@))
```

LIBBOARD = board/\$(BOARDDIR)/lib\$(BOARD).a  
LIBBOARD := \$(addprefix \$(obj),\$(LIBBOARD))

BOARDDIR = \$(VENDOR)/\$(BOARD)  
VENDOR=samsung , BOARD=mango100 값  
따라서 BOARDDIR은 samsung/mango100  
LIBBOARD는  
\$(obj)/board/samsung/mango100/libmango100.a



# U-boot build 실행 분석(LIBBOARD)

- 1
- 2
- 3
- 4
- 5
- 6

```
$(obj)u-boot: depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)
UNDEF_SYM=$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | \
sed -n -e 's/.*#$(SYM_PREFIX)__u_boot_cmd_.*#/-u#1/p'|sort|uniq`;\#
7 cd $(LNDIR) && $(LD) $(LD_FLAGS) $$UNDEF_SYM $(__OBJS) \#
--start-group $(__LIBS) --end-group $(PLATFORM_LIBS) \#
-Map u-boot.map -o u-boot
```

```
$(LIBBOARD): depend $(LIBS) $(obj)include/autoconf.mk
$(MAKE) -C $(dir $(subst $(obj),,$@))
```

```
depend dep: $(VERSION_FILE)
for dir in $(SUBDIRS) ; do $(MAKE) -C $$dir _depend ; done
```

```
$(obj)include/autoconf.mk: $(obj)include/config.h
$(XECHO) Generating $@ ; \
set -e ; \
: Extract the config macros ; \
$(CPP) $(CFLAGS) -DDO_DEPS_ONLY -dM include/common.h | \
sed -n -f tools/scripts/define2mk.sed > $@
```

```
$(LIBS): depend $(obj)include/autoconf.mk
$(MAKE) -C $(dir $(subst $(obj),,$@))
```

```
$(MAKE) -C $(dir $(subst $(obj),,$@))
```

# U-boot build 실행 분석

- 1
- 2
- 3
- 4
- 5
- 6

```
$(obj)u-boot:          depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)
                        UNDEF_SYM=$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | #
                        sed -n -e 's/.*#($$SYM_PREFIX) __u_boot_cmd_.*#/-u#1/p'|sort|uniq`;#
7 cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $$(_OBJS) #
                        --start-group $$(_LIBS) --end-group $(PLATFORM_LIBS) #
                        -Map u-boot.map -o u-boot
```

```
$(LDSCRIPT):          depend $(obj)include/autoconf.mk
                        $(MAKE) -C $(dir $@) $(notdir $@)
```

```
ifndef LDSCRIPT
#LDSCRIPT := $(TOPDIR)/board/$(BOARD)/u-boot.lds.debug
ifeq ($(CONFIG_NAND_U_BOOT),y)
LDSCRIPT := $(TOPDIR)/board/$(BOARD)/u-boot-nand.lds
else
LDSCRIPT := $(TOPDIR)/board/$(BOARD)/u-boot.lds
```

LDSCRIPT값은  
/board/samsung/mango100/u-boot.lds  
따라서,  
\$(MAKE) -C /board/samsung/mango100 u-boot.lds  
실행이 됨

# U-boot.Ids 분석

```
OUTPUT_FORMAT("elf32-littlearm", "elf32-littlearm", "elf32-littlearm")
/*OUTPUT_FORMAT("elf32-arm", "elf32-arm", "elf32-arm")*/
OUTPUT_ARCH(arm)
ENTRY(_start)
SECTIONS
{
    . = 0x00000000;

    . = ALIGN(4);
    .text :
    {
        cpu/s5pc1xx/start.o (.text)
        cpu/s5pc1xx/s5pc100/cpu_init.o (.text)
        board/samsung/mango100/lowlevel_init.o (.text)
        cpu/s5pc1xx/nand_cp.o (.text)
        cpu/s5pc1xx/movi.o (.text)
        *(.text)
    }

    . = ALIGN(4);
    .rodata : { *(.rodata) }

    . = ALIGN(4);
    .data : { *(.data) }

    . = ALIGN(4);
    .got : { *(.got) }

    __u_boot_cmd_start = .;
    .u_boot_cmd : { *(.u_boot_cmd) }
    __u_boot_cmd_end = .;

    . = ALIGN(4);
    .mmudata : { *(.mmudata) }

    . = ALIGN(4);
    __bss_start = .;
    .bss : { *(.bss) }
    _end = .;
}
```

: 섹션의 시작 번지를 0x00000000로 설정. 실제로 컴파일된 실행코드인 text가 로딩되고 배치될 번지는 config.mk에 지정된 TEXT\_BASE + 0x00000000이 된다.

4byte align

Text section의 맨앞에 cpu/s5pcu/start.o를 배열 (Start-up code)

\_\_u\_boot\_cmd\_start와 \_\_u\_boot\_cmd\_end사이의 .u\_boot\_cmd는 u\_boot에서 사용되는 user interface command structure를 이 사이에 배열

\_\_u\_boot\_cmd\_start, \_\_u\_boot\_cmd\_end, \_\_bss\_start, \_\_end는 나중에 C code, start-up code에서 주소계산을 위해 사용함.

# U-boot build 실행 분석

```
1 2 3 4 5 6
$(obj)u-boot: depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)
7 UNDEF_SYM=$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | \
  sed -n -e 's/.*\($(SYM_PREFIX)__u_boot_cmd_.*\)/-u\1/p'|sort|uniq` ; \
  cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) \
  --start-group $(__LIBS) --end-group $(PLATFORM_LIBS) \
  -Map u-boot.map -o u-boot
```

```
UNDEF_SYM=`$(OBJDUMP) -x $(LIBBOARD) $(LIBS) \
sed -n -e 's/.*\($(SYM_PREFIX)__u_boot_cmd_.*\)/-u\1/p'|sort|uniq`
```

```
UNDEF_SYM=`/usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-objdump -x
board/samsung/mango100/libmango100.a lib_generic/libgeneric.a cpu/s5pc1xx/libs5pc1xx.a
cpu/s5pc1xx/s5pc100/libs5pc100.a lib_arm/libarm.a fs/cramfs/libcramfs.a fs/fat/libfat.a fs/fdos/libfdos.a fs/jffs2/libjffs2.a
fs/reiserfs/libreiserfs.a fs/ext2/libext2fs.a net/libnet.a disk/libdisk.a drivers/bios_emulator/libatibiosemu.a
drivers/block/libblock.a drivers/dma/libdma.a drivers/hwmon/libhwmon.a drivers/i2c/libi2c.a drivers/input/libinput.a
drivers/misc/libmisc.a drivers/mmc/libmmc.a drivers/mtd/libmtd.a drivers/mtd/nand/libnand.a
drivers/mtd/nand_legacy/libnand_legacy.a drivers/mtd/onenand/libonenand.a drivers/mtd/spi/libspi_flash.a
drivers/net/libnet.a drivers/net/sk98lin/libsk98lin.a drivers/pci/libpci.a drivers/pcmcia/libpcmcia.a drivers/spi/libspi.a
drivers rtc/librtc.a drivers/serial/libserial.a drivers/usb/libusb.a drivers/video/libvideo.a common/libcommon.a
libfdt/libfdt.a api/libapi.a post/libpost.a | \
  sed -n -e 's/.*\(__u_boot_cmd_.*\)/-u\1/p'|sort|uniq`
```

<http://www.mangoboard.com>

실습

```
#echo abcd123 | sed 's/\([a-z]*\).*/\1/'
```

# U-boot build 실행 분석

- 1
- 2
- 3
- 4
- 5
- 6

```
$(obj)u-boot: depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)
UNDEF_SYM= $(OBJDUMP) -x $(LIBBOARD) $(LIBS) | \
sed -n -e 's/.*\#$(SYM_PREFIX)__u_boot_cmd_.*\#/-u\#1/p'|sort|uniq`;\#
cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) \#
--start-group $(__LIBS) --end-group $(PLATFORM_LIBS) \#
-Map u-boot.map -o u-boot
```

7

cd \$(LNDIR) && \$(LD) \$(LDFLAGS) \$\$UNDEF\_SYM \$(\_\_OBJS) \  
--start-group \$(\_\_LIBS) --end-group \$(PLATFORM\_LIBS) \  
-Map u-boot.map -o u-boot

cd /home/icanjji/work/u-boot-work/mango100-uboot-2010-06-29 && /usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-ld  
-Bstatic -T /home/icanjji/work/u-boot-work/mango100-uboot-2010-06-29/board/samsung/mango100/u-boot.lds -Ttext 0x2fe00000  
\$UNDEF\_SYM cpu/s5pc1xx/start.o \  
←

```
--start-group lib_generic/libgeneric.a cpu/s5pc1xx/libs5pc1xx.a  
cpu/s5pc1xx/s5pc100/libs5pc100.a lib_arm/libarm.a fs/cramfs/libcramfs.a  
fs/fat/libfat.a fs/fdos/libfdos.a fs/jffs2/libjffs2.a fs/reiserfs/libreiserfs.a  
fs/ext2/libext2fs.a net/libnet.a disk/libdisk.a drivers/bios_emulator/libatibiosemu.a  
drivers/block/libblock.a drivers/dma/libdma.a drivers/hwmon/libhwmon.a drivers/i2c/libi2c.a  
drivers/input/libinput.a drivers/misc/libmisc.a drivers/mmc/libmmc.a drivers/mtd/libmtd.a  
drivers/mtd/nand/libnand.a drivers/mtd/nand_legacy/libnand_legacy.a drivers/mtd/onenand/libonenand.a  
drivers/mtd/spi/libspi_flash.a drivers/net/libnet.a drivers/net/sk98lin/libsk98lin.a  
drivers/pci/libpci.a drivers/pcmcia/libpcmcia.a drivers/spi/libspi.a drivers rtc/librtc.a  
drivers/serial/libserial.a drivers/usb/libusb.a drivers/video/libvideo.a common/libcommon.a  
libfdt/libfdt.a api/libapi.a post/libpost.a board/samsung/mango100/libmango100.a --end-group  
-L /usr/local/arm/4.2.2-eabi/usr/bin-ccache/./lib/gcc/arm-unknown-linux-gnueabi/4.2.2 -lgcc \  
-Map u-boot.map -o u-boot
```