1 版本号

2 MAKEFLAGS变量

make支持递归调用,可以用以下指令执行子目录下地Makefile

$(MAKE) –C subdir

$(MAKE)为make指令，-C指定子目录

有时候需要向子make传递变量，这个时候使用”export”来到处要传给子make的变量

export VARIABLE…

不想向子make传递变量时

unexport VARIABLE

SHELL和MAKEFLAGS为特殊，除非对其使用unexport，否则，他们的值会自动传递给子make

**3 获取主机架构和系统**



uname –m获取系统架构，输出x86\_64

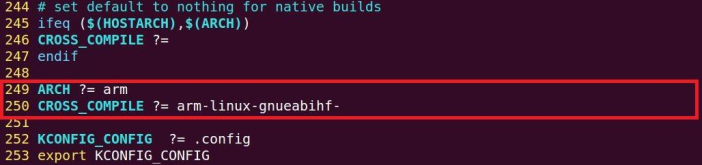
227行有一个符号”|”，shell中的”|”表示管道，意思是将左边的输出作为右边的输入，

sed –e s/i.86/x86 就是将i.86替换成x86

237行uname –s输出主机OS，为Linux，“tr '[:upper:]' '[:lower:]'”表示将所有的大写字母替换为小写字母，因此得到“linux”。最后同样使用管道，将“linux”作为“sed -e 's/\(cygwin\).\*/cygwin/'”的输入，用于将cygwin.\*替换为 cygwin。因此， HOSTOS=linux。

**4 设置目标架构、交叉编译器和配置文件**

为了方便，直接在顶层Makefile指明架构和交叉编译工具链



252行的.config是执行命令”make xxx\_defconfig”过程中自动生成的文件。defconfig是一些初始配置，.config里的才是实时有效的配置。

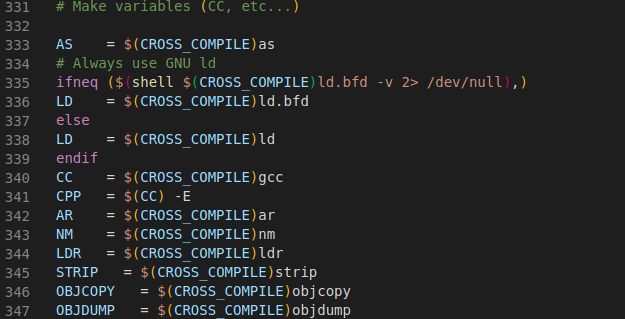
5 调用scripts/Kbuild.include

顶层Makefile会调用文件scripts/Kbuild.include.

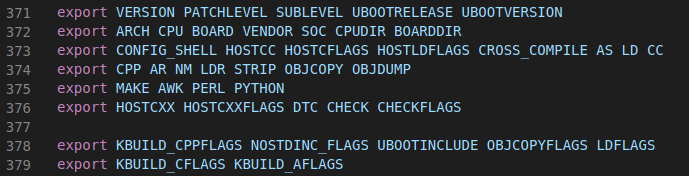


uboot编译会用到文件中定义的变量

6 交叉编译工具变量设置

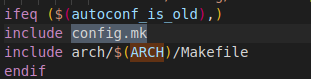


7 导出其他变量

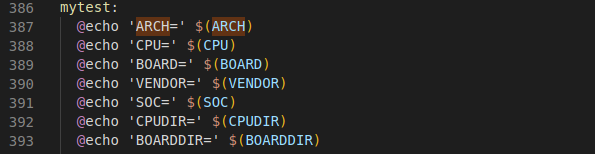


大部分变量在Makefile中已经定义，我么重点关注ARCH CPU BOARD VENDOR SOC CPUDIR BOARDDIR着几个变量

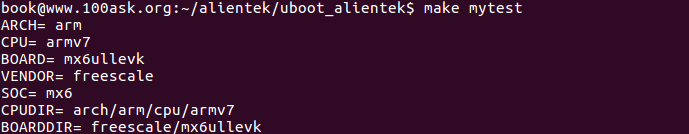
在顶层Makefile中找不到这几个变量，说明这几个变量由外部文件定义，它们定义在config.mk中，在顶层Makefile中，config.mk被include



我们可以用以下代码查看上述ARCH CPU..这几个变量的值

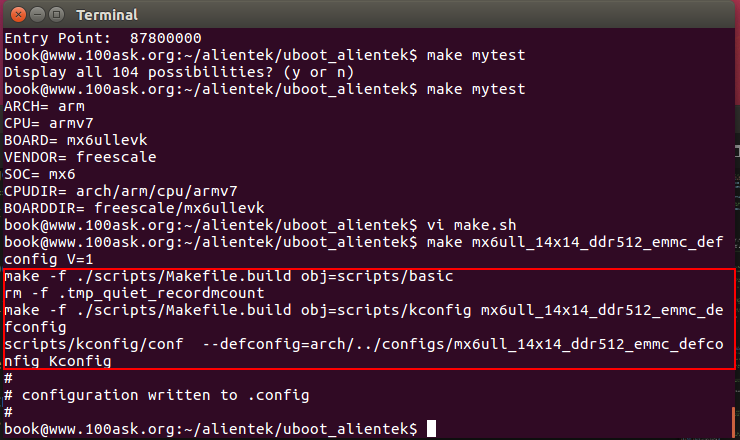


命令行中执行make mytset输出以下内容：



8 make xxx\_defconfig过程

韦东山有一种方法：想要知道make 执行的过程，就先执行make。我们先执行make mx6ull\_14x14\_ddr512\_emmc\_defconfig V=1，命令台输出以下内容：



**9 Makefile.build分析**

执行make xxx\_defconfig会执行以下两行命令：

@make -f ./scripts/Makefile.build obj=scripts/basic  
@make -f ./scripts/Makefile.build obj=scripts/kconfig xxx\_defconfig

我们来逐一分析上述两条命令

**9.1 cripts/basic目标对应的命令**