Lab – using android toolchain

1. Using gedit write a simple hello.c file

2. Update the path to include:
   ~/aosp/prebuilt/linux-x86/toolchain/arm-eabi-4.4.3/bin

3. Update and add agcc tool to the above directory

4. Compile using the command:
   agcc test.c -o app1

5. Push to the emulator and run
   adb push ./app1 [destination in target]

Lab – building android kernel

1. Change directory to ~/kernel/goldfish

2. Edit the kernel Makefile and add
   ARCH ?= arm
   CROSS_COMPILE ?= arm-eabi-

3. Load the default configuration for the board (emulator)
   make goldfish_armv7_defconfig

4. Configure the kernel – add/remove printk times (kernel hacking)
   make menuconfig

5. Build the kernel
   make

6. test the new kernel with the emulator
   cd [aosp]/
   ./out/host/linux-x86/bin/emulator -kernel ~/kernel/goldfish/arch/arm/boot/zImage &

7. Open adb shell and run dmesg

Lab – customizing the target – installing busybox

1. copy and extract busybox package
   tar -xvf ./busybox-1.20.2.tar.bz2

2. copy and install linux toolchain
   ./arm-2011.03-41-arm-none-linux-gnueabi.bin

3. configure busybox:
   1. make menuconfig
   2. under build options – set as static binary
   3. set the toolchain prefix to arm-none-linux-gnueabi-
   4. under general configuration set “do not use /usr”
   5. exit and save
   6. run make
   7. run make install
   8. copy bin folder to target /system/bin
   9. copy sbin folder to target /system/xbin
   10. overwrite ash with busybox (cp ./busybox ./ash)

4. build android and test the shell