Lab 3 – Boot Using NFS-mounted file system

Purpose

The purpose of this lab is to demonstrate how to boot the EVM when the file system resides on a different server that is mounted on the EVM, then develop a code on the Linux host and move it to the file system. The executable will be available to the ARM on the EVM. A debug session using gdb will be performed from the serial port terminal.

Task 1: Build a file system on a Linux host, use the NFS server

The NFS server is installed on the Ubuntu server in the directory /opt/filesys. Each student has a sub-directory where he or she builds the file server, and the Uboot is configured to reach this directory for each student.

1. Next the file system to be mounted should be built on the local Ubuntu machine.
    a. Create a directory where the file system resides; say /opt/filesys/studentN (where N is the student number. Note, this directory should be created already)
    b. Copy a tar version of the compressed file system tisdk-rootfs.tar.gz (part of the release in the images directory) into /opt/filesys/studentN
    c. Untar the file system -> "sudo tar zxf tisdk-rootfs.tar.gz "
    d. Delete the original compress file -> "sudo rm tisdk-rootfs.tar.gz "
    e. Add the file system directory to the exports list, open the file /etc/exports and add the following line to it. Note; just verify that this was already done.

    /opt/filesys *(rw,subtree_check,no_root_squash,no_all_squash,sync)

    The file /etc/exports looks like the following:

```
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subt
ree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
/opt/filesys *(rw,subtree_check,no_root_squash,no_all_squash,sync)
#
```

2. The instructor will start the NFS server -> "sudo /etc/init.d/nfs-kernel-server restart "

Task 2: Configure U-BOOT to mount the file server and boot

1. Power cycle the EVM, in the ARM tera-terminal stop the autoboot
2. Change the following environment variable
    a. Change the boot to be from the network -> "setenv boot net"
    b. Add the nfs server ip -> "setenv nfs_serverip xxx.xxx.xxx.xxx " where xxx.xxx.xxx.xxx is the IP address of the Ubuntu server on which the file system resides
    c. Define the file system root directory -> "setenv nfs_root /opt/filesys/studentN "
    d. Configure the arguments for the boot -> " setenv args_net 'setenv bootargs ${bootargs} rootfstype=nfs root=/dev/nfs rw nfsroot=${nfs_serverip}:${nfs_root},${nfs_options} ip=dhcp' "
    e. Save the new environment variables -> " saveenv"
3. Boot
    a. Note, if the DHCP does not supply an IP address to the EVM, the EVM will use its default IP address. This default IP address is define in the environment -> " printenv" as ipaddr. If this does not exist the user can configure it -> "setenv ipaddr yyy.yyy.yyy.yyy "
    b. In addition, if the DHCP does not provide IP address, follow the steps in the appendix for static IP address configuration.

Task 3: Build a new C program in the file system, and debug it

1. In a local Ubuntu terminal go to /opt/filesys/studentN and look at the file system
2. Follow the example simple code section of Lab 2, copy example1.c into one of the directories of the file system, for example into /opt/filesys/student/bin
3. Set the terminal in the bin directory -> "cd /opt/filesys/student/bin "
4. Compile and build the application similar to what you did in Lab2, but add the debug flag (-g) to the command – that is

~/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/bin/arm-linux-gnueabi-gcc **–g** -o example1 example1.c

5. Back to the tera-terminal, navigate to /bin -> "cd /bin "
    a. Make sure that example1.c and example1 are both in the bin directory -> "ls –ltr example1*
    b. Start a debug session -> " gdb example1 "
    c. Use the list command to see the source, use b to set a break point, use r to run to the break point
    d. Other simple gdb command s to step, n for next (step over), c to run to the next breakpoint, and finish to end
    e. There are many gdb quick guides on the Web. Here is a URL to one of them:

    http://condor.depaul.edu/glancast/373class/docs/gdb.html