# F. Y. M. Tech (ECE)

**Trimester: I Subject: Embedded System Design** (**ECE5015A)**)

# Name:Anupama Prasad Class: F.Y.M.Tech

**Roll No: 1032212131 Batch: 2021-2022**

# Assignment No: 06

**Name of the Assignment**: **Download pre-configured embedded Linux kernel image, file system, boot loader to target device- ARM9.**

|  |
| --- |
| **Marks** |
|  |
|  |

**Teacher’s Signature with date**

# Performed on: 27/10/2021

**Submitted on: 31/10/2021**

# Aim: Download pre-configured embedded Linux kernel image, file system, boot loader to target device- ARM9.

**Objective: Exercise on Porting of Linux on ARM9 board Procedure:**

**Installing the ARM9 tool chain**

This session explains how to install toolchain for ARM-9 target board

Note that separate version of Linux are used by the target and the host Linux PC.

The following Linux host operating system is supported for use with the ARM9 target board.

# o Open SUSE 11.0 or Open SUSE 11.1 To install the ARM9 toolchain, follow this step:-

1. Log in as root on your host Linux PC.
2. Copy the **build\_root.tar** from the “DVD” to the **/opt** folder.
3. Now type the following command to Un-tar (Un-zip) the build\_root.tar

[root@EDUTECH] tar vxf build\_root.tar.

1. This command will take approx 5 min for execution.
2. After executing this command you will see a folder “build\_root” in the /opt folder.
3. As the folder is created, it means your toolchain has successfully installed.

# Porting kernel on ARM9 target board

* There are many sites available for kernel source. However the most preferred site for the kernel source code is

# ftp://ftp.kernel.org/pub/linux/kernel/v2.6/

* You can find both current and historical versions of the Linux kernel, as well as numerous patches on the mentioned site.
* The architecture specific patches required for kernel porting can be derived from different manufacturers’ sites.
* Download kernel **"linux-2.6.20.4.tar.bz2"** or **"linux-2.6.20.4.tar.gz"** and copy it into your Linux PC's home directory using the following command.

# cp linux-2.6.20.4.tar.bz2 /home/edutech

**Note :** It is recommeded that you download the kernel from course download material to avoid confusion and work faster.

.............................

* Extract/untar the kernel source code using following command:

# HOME$ tar xjvf linux 2.6.20.4.tar.bz2 or HOME$ tar xzvf linux 2.6.20.4.tar.gz

**Note:**

1. It is recommeded that you should install the toolchain from the course material and copy that in the /opt folder of your Linux Host PC and ensure that your toolchain is working properly.
2. We will denote home directory by **"HOME$"** throughout this tutorial.

# Steps to apply the Patch to the raw kernel source:

Download the **linux-2.6.20.4.patch** file from the course download material provided to the

/opt directory where the kernel source code is extracted. This patch file contains all the required changes for configuring the kernel for our ARM9 target board. The details about the changes are going to discuss in the last session on the module. Now it is must to have the

* + patch file in the kernel source code's root directory. So copy **linux-2.6.20.4.patch** to the

**/home/edutech/linux-2.6.20.4/** directory as shown

# HOME$:cp linux-2.6.20.4.patch /home/edutech/linux-2.6.20.4/

* + Now go to **linux-2.6.20.4** directory by using following command

# HOME$:cd linux-2.6.20.4

* + Apply the patch by using following command

# HOME$/linux-2.6.20.4:patch -p 1 < linux-2.6.20.4.patch Follow the steps as shown to build the kernel.

* + Go to linux-2.6.20.4 using cd command in terminal window

# HOME$: cd linux-2.6.20.4

* + Now open Makefile by using following command

# vi Makefile

* + The following lines in the code are required to be modified. **ARCH ?= $(SUBARCH) CROSS\_COMPILE ?=**
  + Modify the above lines as shown below.

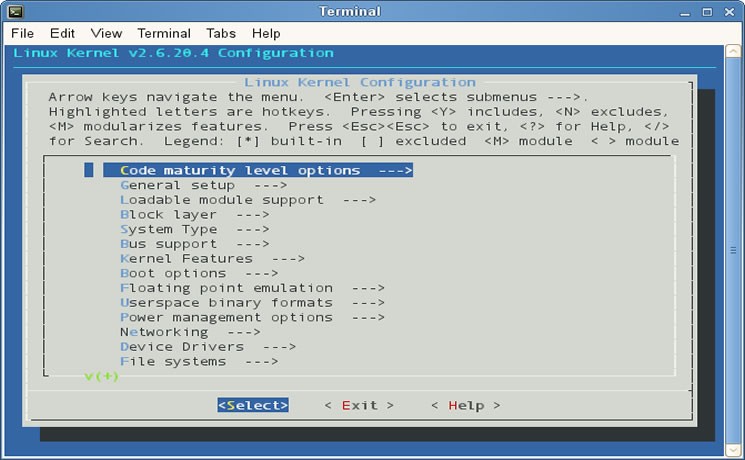
# ARCH ?= arm

**CROSS\_COMPILE ?= /opt/buildroot/build\_arm/staging\_dir/bin/arm-linux- Configuration**

Now to configure the kernel source code for the ARM9 target baord follow the steps as shown.

* + Type following command in console Terminal as shown

# make menuconfig



**Compilation**

The following steps are to be executed to compile the kernel source.

**Step1:** Compile the Kernel using make all command

# make all

**Note :** This command will take few minutes.

**Step 2 :** Login as super user by su command **HOME $ <kernel source directory>su password:**

**Step 3 :** Create the binary file using the command

# HOME $ <kernel source directory>/opt/buildroot/build\_arm/staging\_dir/bin/arm- linux-objcopy -O binary -R .note -R .comment vmlinux linux.bin

* + In the previous **objcopy** command, the **-O** option tells **objcopy** to generate a binary file,
  + the **-R** option removes the ELF sections named **.note** and **.comment** , And the **-S** option is the flag to strip debugging symbols.
  + Notice that **objcopy** takes the **vmlinux** ELF image as input and generates the target binary file called **Image/ linux.bin** . In summary, **Image/ linux.bin** is nothing more than

the kernel proper in binary form stripped of debug symbols and the **.note** and **.comment**

ELF sections.

**Step 4 :** Check whether **linux.bin** file is created or not using ls command.

**HOME $ <kernel source directory>ls Step 5 :** Compress linux.bin file by gzip command

**HOME $ <kernel source directory>gzip -9 linux.bin Step 6** : Go to u-boot/tools/directory using command window

# HOME$ <Kernel source directory> cd /opt/u-boot/tools

**Step7 :** Make bootable image with the help of u-boot command mkimage as shown

# HOME$/opt/u-boot/tools #./mkimage -A arm -O linux -T kernel -C gzip -a 0x20008000 - e 0x20008000 -n "Linux kernel Image " -d <path>/ Linux.bin.gz uImage

**Where:** A = Architecture

O = Operating System T = Type

C = compressed type. a = load address

e = entry point n = image name

d = image data file (PATH)

- Here add the path of your kernel source directory instead of the word <path>.

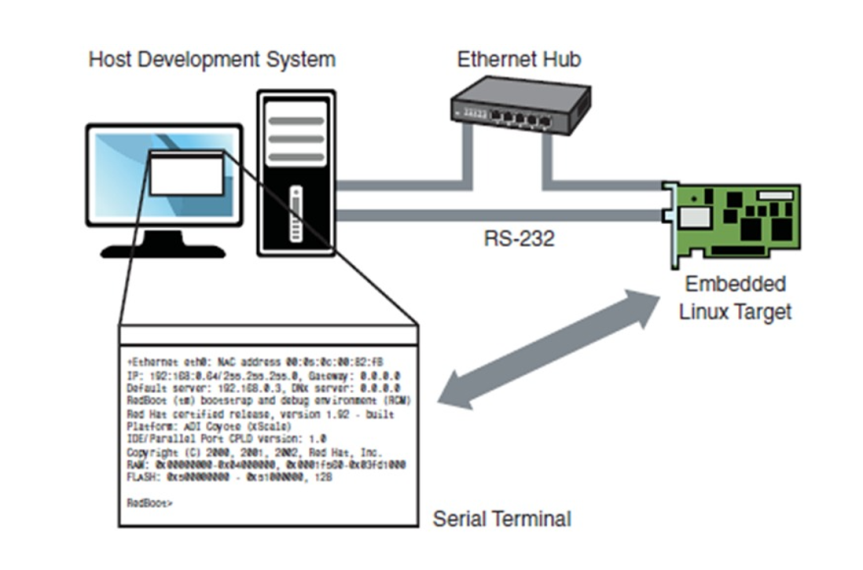
Mkimage utility is available in uboot source code, you can also seperatly download the this utility from **ftp://ftp.koansoftware.com/public/linux/mkimage/ and can build**

In that case we do not have to go in uboot directory.

**Step8:** The above command will create the uImage of kernel in "/opt/u-boot/tools" directory.

Check it using ls command.

Host /opt/u-boot/tools # **ls**

****

**Conclusion:**

Pre-configured embedded linux kernel image, file system and boot loader are downloaded to the target device ARM9. Status of Hardware parameters such as I/O devices, status of memory is observed on the terminal.