PROBLEM IN FOLLOWING C PROGRAM

```
#include <stdio.h>
int main(int argc, char **argv)
{
printf(“Hello World\n”);
return 0;
}
```

- hello.c:5: warning: implicit declaration of function ‘printf’
- This introduces some interesting questions:

  Where is the file stdio.h located, and how is it found?

  Where is the printf() function object code stored on your system, and how is this reference resolved in the binary executable?

- HOST SYSTEM REQUIREMENT

- Cross toolchain and libraries
- Target system packages, including programs, utilities, and libraries
- Host tools such as editors, debuggers, and utilities
- Servers for hosting your target board, as covered in the next section

- sudo apt-get install tftpd-hpa

TFTP Configuration

- #Defaults for tftpd-hpa
- RUN_DAEMON=”yes”
- OPTIONS=”-l -c -s /tftpboot”

HOSTING TARGET BOARDS

- sudo /etc/init.d/tftpd-hpa restart
- /etc/init.d/dhcpd start
- /etc/init.d/nfs start

A Universal Bootloader: Das U-Boot

- The official name for this bootloader is Das U-Boot.
- It is maintained by Wolfgang Denk and hosted on SourceForge at http://u-boot.sourceforge.net/.
- U-Boot has support for multiple architectures and has a large following of embedded developers and hardware manufacturers who have adopted it for use in their projects and have contributed to its development.

System Configuration: U-Boot

- When configuring U-Boot for one of its supported platforms, issue this command:

  $ make <platform>_config

- These platform-configuration targets are listed in the top level U-Boot makefile.
- For example, to configure for the Spectrum Digital OSK, which contains a TI OMAP 5912 processor, issue this command:

  $ make omap5912osk_config

U-Boot Command Sets

- **Command Set Commands**
- CFG_CMD_FLASH : Flash memory commands
- CFG_CMD_MEMORY : Memory dump, fill, copy, compare, and
- so on
- CFG_CMD_DHCP : DHCP Support
- CFG_CMD_PING : Ping support
- CFG_CMD_EXT2 : EXT2 File system support

Network Operations

- BOOTP (Bootstrap Protocol) and DHCP(Dynamic Host Control Protocol) are protocols that enable a target device with an Ethernet port to obtain an IP address and other network-related configuration information from a central server.
- TFTP(Trivial File Transfer Protocol) allows the target device to download files (such as a Linux kernel image) from a TFTP server.

DHCP Target Specification on DHCP server

- host coyote {
- hardware ethernet 00:0e:0c:00:82:f8;
- netmask 255.255.255.0;
- fixed-address 192.168.1.21;
- server-name 192.168.1.9;
- filename "coyote-zImage";
- option root-path "/home/chris/sandbox/coyote-target";
- }
- ...
- //This example is from Fedora DHCP implementation

DHCP Target Parameters

DHCP Target Parameter	Purpose	Comments
host	Hostname	Symbolic label from DHCP configuration file
hardware ethernet	Ethernet hardware address	Low-level Ethernet hardware address of the target's Ethernet interface
fixed-address	Target IP address	The IP address that the target will assume
netmask	Target netmask	The IP netmask that the target will assume
server-name	server-name	The IP address to which the target will direct requests for file transfers, root file system, and so on
filename	TFTP filename	The filename that the bootloader can use to boot a secondary image (usually a Linux kernel)

Storage Subsystems

- Many boot loaders support the capability of booting images from a variety of nonvolatile storage devices in addition to the usual Flash memory.
- U-Boot can load an image from a specified raw partition or from a partition with a file system structure.
- The board must have a supported hardware device (an IDE subsystem) and Uboot must be so configured.
- Adding CFG_CMD_IDE to the board-specific configuration file enables support for an IDE interface.
- Adding CFG_CMD_BOOTD enables support for booting from a raw partition.

Booting from Disk: U-Boot

- U-Boot supports several methods for booting a kernel image from a disk subsyste

  => **diskboot 0x400000 0:0**
- The 0:0 in this example specifies the device and partition.
- In this simple example, U-Boot performs a raw binary load of the image found on the first IDE device (IDE device 0) from the first partition found on this device.
- The image is loaded into system memory at physical address 0x400000.

- After the kernel image has been loaded into memory, the U-Boot bootm command (boot from memory) is used to boot the kernel:

- => **bootm 0x400000**