

# Filesystems

## Zilogic Systems

### 1. U-Boot Command Interface

- When U-boot boots up, before passing control to the kernel, displays

```
Hit any key to stop autoboot:  3
```

- Command interface can be accessed by pressing a key at this point.
- Several commands are available in the U-boot prompt. Type `help` to get the list of commands. The commands available can be configured during compile time.
- The command to boot the kernel is `boot`.

#### 1.1. Environment Variables

- U-boot has a set of variables that affects the behaviour of U-boot. These variables are stored in Flash memory and are called U-boot environment variables.
- To see the list of variables use the `printenv` command.

```
U-boot> printenv
bootdelay=3
baudrate=115200
hostname=sam9l9260
ethact=macb0
ethaddr=00:11:22:33:44:55
bootargs=console=ttyS0,115200 root=/dev/ram0
filesize=140790
fileaddr=21000000
netmask=255.255.0.0
ipaddr=172.16.1.10
serverip=172.16.0.2
bootcmd=bootm 0xD0042000
stdin=serial
stdout=serial
stderr=serial
```

- To change the value of a variable, the `setenv` command can be used.

```
U-boot> setenv variable value
```

- Values changed are stored in memory and are lost after, a reboot. To save the modified variables the `saveenv` command can be used.

```
U-boot> saveenv
Saving Environment to dataflash...
```

##### 1.1.1. Boot Related Variables

- |                        |   |
|------------------------|---|
| <code>bootdelay</code> | specifies how many seconds U-boot should wait for the user to press a key, before booting the kernel. |
| <code>bootargs</code>  | specifies the arguments to be passed to the kernel.   |

`bootcmd` specifies the U-boot command to be executed for booting the kernel. The value of the variable is the command to be executed when the user gives the `boot` command.

### 1.1.2. Network Related Variables

- U-boot is capable of transferring files through TFTP.
- TFTP requires a UDP/IP stack.
- The following variables have to be configured to be able to use TFTP.

`ethaddr` specifies the MAC address to be used for the first ethernet interface. This is only one time settable.

`ipaddr` specifies the IP address to be used.

`netmask` specifies the netmask, used to determine if the packet should be sent to the gateway.

`gatewayip` specifies the IP address of the system that acts as the gateway for the network.

`serverip` specifies the IP address of the TFTP server.

## 1.2. U-boot Commands

### 1.2.1. File Transfer Commands

- Files can be transferred from the TFTP server to the local memory using the `tftp` command.

```
U-boot> tftp 0x21000000 image
```

- Here `image` is the file to be transferred from the remote TFTP server.
- `0x21000000` is the address in which the downloaded image has to be stored. The address should correspond to a area in RAM.

### 1.2.2. Memory Related Commands

- To copy data from one memory area to another.

```
U-boot> cp.b 0x21000000 0xD0000000 0x1234
```

- The first argument is the source address.
- The second argument is the destination address.
- The third argument is the size of the data to be copied.
- Transferring data to dataflash.
- Dataflash is a serial flash and hence is not memory mapped.
- A logical address is assigned to each dataflash, and to read/write to the data flash, the corresponding logical address has to be read or written to.
- The Olimex board has one on-board 4MB dataflash. An external 8MB dataflash is connected to the dataflash socket.
- The 4MB dataflash is accessible from logical address 0xD0000000.
- The 8MB dataflash is accessible from logical address 0xC0000000.

### 1.2.3. Flashing Kernel and Filesystem

- The kernel and filesystem are located in the 4MB dataflash.
- The logical addresses are given below.

Kernel            D0042000

Filesystem       D0210000

- To transfer and flash the kernel

```
U-boot> tftp 0x21000000 uImage
macb0: link up, 100Mbps full-duplex (lpa: 0x45e1)
Using macb0 device
TFTP from server 172.16.0.2; our IP address is 172.16.1.10
Filename 'uImage'.
Load address: 0x21000000
Loading: #####
          #####
          #####
          #####
done
Bytes transferred = 1319724 (14232c hex)
U-Boot> printenv filesize
filesize=14232C
U-boot> cp.b 0x21000000 0xD0042000 $(filesize)
```

- The kernel should be present in the TFTP server's directory.

## 2. Mounting File-systems

- The mount command is used to attach another filesystem to a mount point.
- The general syntax of the mount command

```
mount -t fstype -o options device mount-point/
```

- `fstype` is the file-system type.
- `device` is the device containing the file-system
- `mount-point` is the directory from which the file-system should be available.
- Only root is allowed to mount filesystems without restrictions, using the mount command.

## 3. Ramdisk

### 3.1. Creating a Ramdisk

- Create the ramdisk using `genext2fs` utility. The utility is available from <http://genext2fs.sourceforge.net>. The advantage of the utility is that root privileges are not required for creating the ramdisk.

```
$ genext2fs -b 8192 -N 2048 -d /path/to/root/fs ramdisk.img
```

- |                 |  |
|-----------------|--|
| <code>-b</code> | specifies the size (in 1024 blocks) of the resulting ramdisk.          |
| <code>-N</code> | specifies the maximum no. of inodes that can be created.               |
| <code>-d</code> | specifies the location of the filesystem to be copied into the ramdisk |
| argument        | specifies the filename of the image to be created.                     |
- Compress it, using the `gzip` command.

```
$ gzip ramdisk.img
```

- Create an image suitable for U-boot, using the `mkimage` tool. The tool is available in U-boot source distribution.

```
$ mkimage -A arm -T ramdisk -n "ARM Ramdisk" -d ramdisk.img.gz uRamdisk
```

<code>-A</code>	specifies the architecture
<code>-T</code>	specifies the image type - <code>ramdisk</code> or <code>kernel</code>
<code>-n</code>	specifies a name for the image
<code>-d</code>	specifies the image source
argument	is the U-boot specific image to be created

### 3.2. Using Ramdisk

- The kernel should be passed the following boot arguments.

```
root=/dev/ram0
```

- Both the kernel and the ramdisk should be loaded to memory by the boot loader. The command to boot a system with a ramdisk.

```
bootm 0xD0042000 0xD0210000
```

- The first address is the kernel's address in memory.
- The second address is the ramdisk's address in memory.

## 4. Network File System

- Let's assume that the NFS server is configured correctly.
- The filesystem that is to be made available to board should be present in a directory and should be specified in the NFS server's configuration file.
- To boot the NFS filesystem, the ethernet interface in Linux should be properly configured — IP address, Netmask, etc.
- The kernel should also be provided with the details of the NFS server and directory.
- To configure the IP address the following `ip` argument should be passed to the kernel. The general format of the `ip` argument is

```
ip=<client-ip>:<nfs-serv-ip>:<gw-ip>:<netmask>:<hostname>:<device>:<autoconf>
```

- The `device` is the network interface name in Linux like `eth0`.
- The `autoconf` specifies if the network interface is to be statically configured or dynamically configured. The `autoconf` parameter can appear alone, as the value of `ip`. Meaningful values are
  - off or none      don't use autoconfiguration (do static IP assignment instead)
  - on or any        use any protocol available in the kernel (default)
  - dhcp            use DHCP
  - bootp          use BOOTP
  - rarp            use RARP
  - both            use both BOOTP and RARP but not DHCP
- An example of the `ip` argument with static configuration.

```
ip=172.16.1.10:172.16.0.2:172.16.0.6:255.255.0.0:atmel:eth0:off
```

- An example of the `ip` argument with DHCP configuration

```
ip=dhcp
```

- To specify that the root filesystem is NFS and the NFS parameters.

```
root=/dev/nfs nfsroot=172.16.0.2:/home/resources/tools/rootfs-1
```

- `nfsroot` specifies the server's IP and the directory in the server that contains the root filesystem.

## 5. Virtual Filesystems

- Mounting proc file-system

```
$ mount -t proc none /proc
```

- Mounting sysfs file-system

```
$ mount -t sysfs none /sys
```

- Mounting tmpfs file-system

```
$ mount -t tmpfs -o size=2m none /my/mnt/point
```

## 6. Further Reading

- U-boot commands <http://www.denx.de/wiki/view/DULG/UBootCommandLineInterface>
- Mounting the root filesystem via NFS - <https://www.kernel.org/doc/Documentation/filesystems/nfs/nfsroot.txt>
- For tmpfs filesystem related documentation <https://www.kernel.org/doc/Documentation/filesystems/tmpfs.txt>
- Getting genext2fs command: <http://genext2fs.sourceforge.net/>