GadgetPC
Single Board Computer

Debian Installation Guide

Document Revision: 1.03

Date: 20 October, 2009

BiPOM Electronics, Inc.

16301 Blue Ridge Road, Missouri City, Texas 77489
Telephone: 1-713-283-9970 Fax: 1-281-416-2806
E-mail: info@bipom.com
Web: www.bipom.com

1. Overview

Thank you for your purchase of the GadgetPC Single Board Computer.
GadgetPC is a powerful computer board that is capable of running high-level operating systems such as Linux.

This document is for advanced users who want to install Debian Linux system from scratch.

When GadgetPC is first powered, it goes through a boot sequence and executes various components in the following order:

ROM boot loader (built-in ROM)
AT91BootStrap (DataFlash)
U-boot (DataFlash)
Linux kernel (uImage file under USB FAT root)
Root FS (Debian on USB Flash drive)

ROM boot loader is built into the AT91SAM9260 microcontroller and cannot be changed. As soon as the board is powered the ROM boot loader starts. It downloads and runs an application (AT91BootStrap) from external storage media (DataFlash) into internal SRAM. AT91BootStrap has been developed by BiPOM Electronics specifically for GadgetPC.

AT91BootStrap is responsible for initializing hardware such as DataFlash, SDRAM, digital outputs, and USART0 serial port.

AT91BootStrap downloads to SDRAM and passes control to U-Boot which is a powerful boot loader that resides also in DataFlash. U-Boot performs many low-level tasks such as detecting USB hardware, reading Linux image from external USB flash drive, uncompressing Linux image to SDRAM, and passing control to Linux image in SDRAM.

U-Boot is open source system that may be upgraded from time to time by BiPOM Electronics for additional functionality. U-Boot can be upgraded by the user as described in this document.

Linux kernel and RootFS are the two main and largest components of the operating system. These are also the easiest to upgrade since they reside on an external USB flash drive. Linux and RootFS upgrades are provided by BiPOM Electronics. However, users may also build their own Linux kernel, copy to USB flash Drive and boot from their own Linux kernel.

When Linux kernel is started, it will mount USB flash drive as RootFS.
USB flash drive has to be DUAL partition.
The 1st partition has to be formatted to FAT32. U-boot can download Linux image only from FAT32.
The 2nd partition has to be formatted to EXT3. Linux will mount it as RootFS.

It is necessary to use a PC Linux machine to format USB Flash drive, copy files, etc.
Windows users can install FREE Vmware player and run the Linux virtual machine.

2. Debian RootFS.

2.1. Download and install FREE Vmware player from http://www.vmware.com/

2.2. Download Linux Ubuntu virtual machine (UbuntuARM_REV102_2HDD.zip) from BiPOM public FTP site at www.bipom.com.
FTP Username = bipomftp
FTP Password = guest123!

You can unzip to any suitable folder. For example, to C:\Virtual Machines\UbuntuARM_REV102

2.3. Download a Debian Lenny ARM image for QEMU from http://people.debian.org/~aurel32/qemu/armel/
http://people.debian.org/~aurel32/qemu/armel/debian_lenny_armel_small.qcow.gz
Uncompress the debian_lenny_armel_small.qcow.gz to get the debian_lenny_armel_small.qcow file.

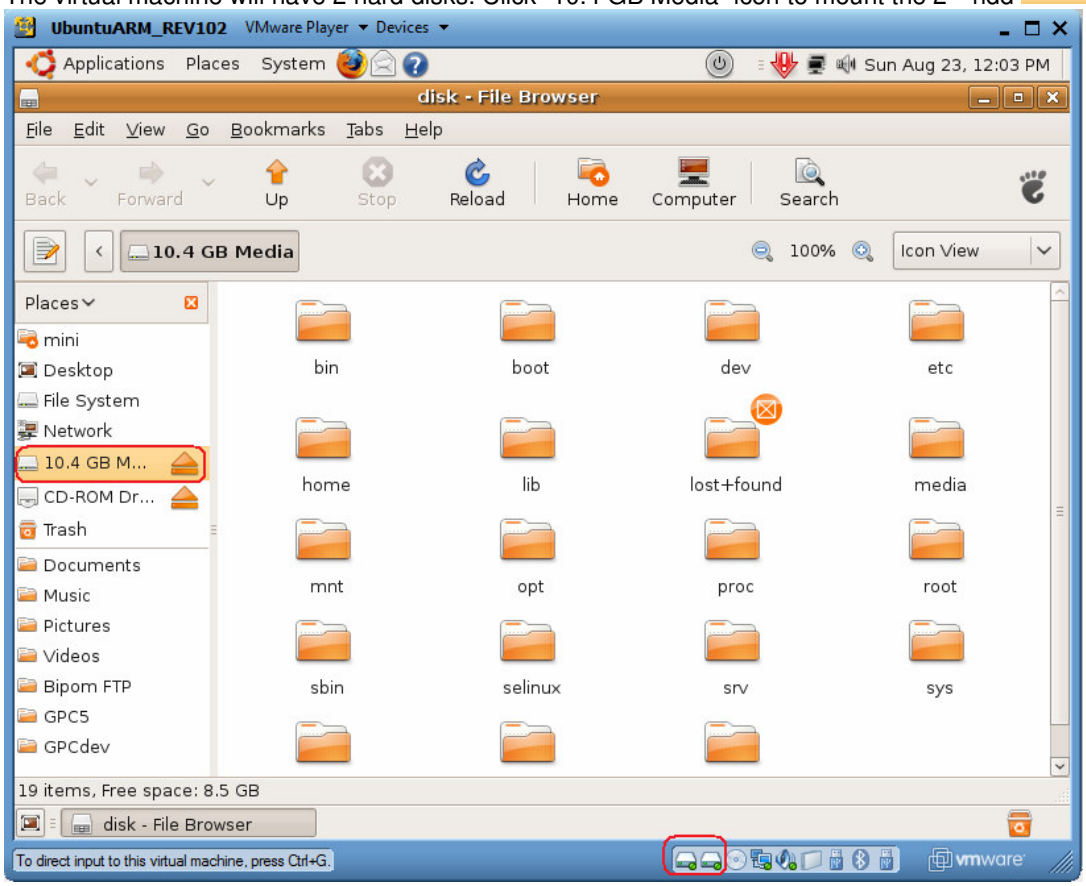
2.4. Download a small QEMU linux-test package from http://www.h7.dion.ne.jp/~qemu-win/
http://www.h6.dion.ne.jp/~kazuw/qemu-win/qemu-0.9.0-windows.zip
We need that package to obtain a special utility called qemu-img.exe
Using this utility we need to convert the debian_lenny_armel_small.qcow to Vmware disk image
Copy 2 files (qemu-img.exe, debian_lenny_armel_small.qcow) to C:\qemu
Run the following command using a simple BAT file

qemu-img.exe convert -f qcow debian_lenny_armel_small.qcow -O vmdk diskimage.vmdk

It will create diskimage.vmdk under C:\qemu folder.

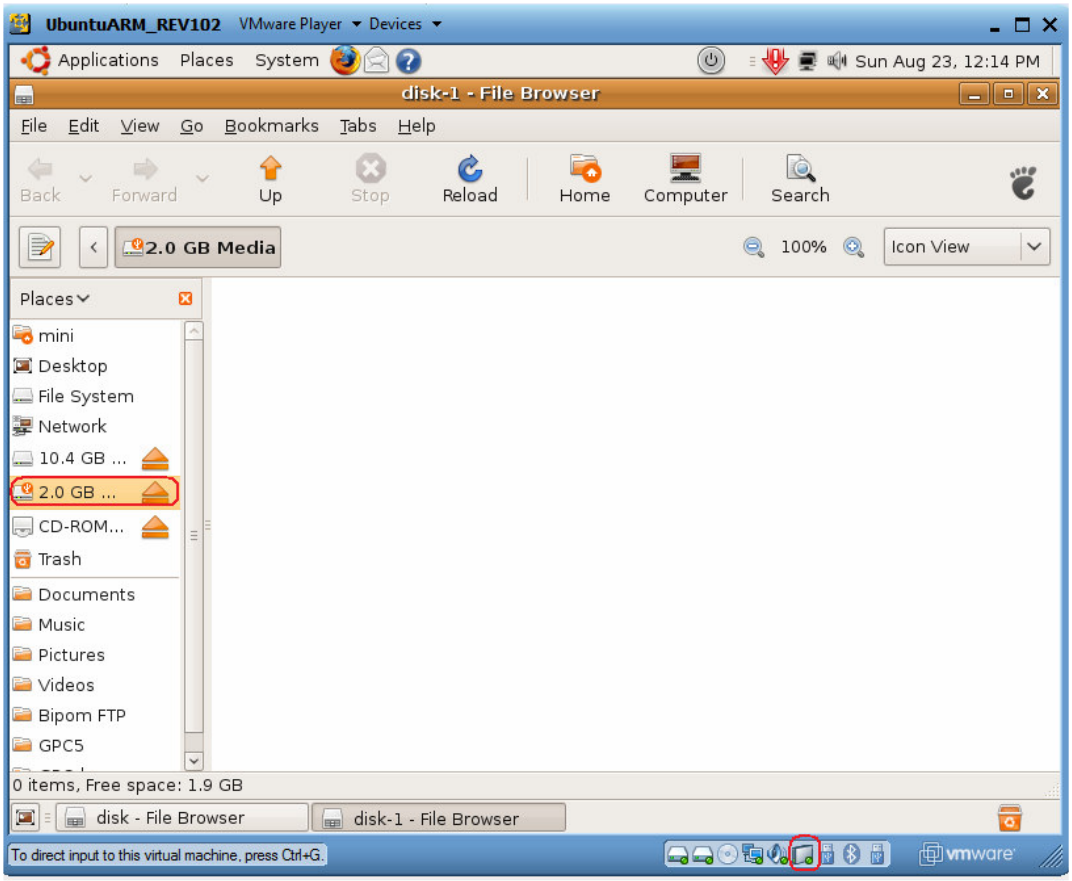
2.5. Run Vmware player and open UbuntuARM_REV102.vmx.
Username is **mini**
Password is **max**
Root password is **max**

The virtual machine will have 2 hard disks. Click “10.4 GB Media” icon to mount the 2nd hdd 

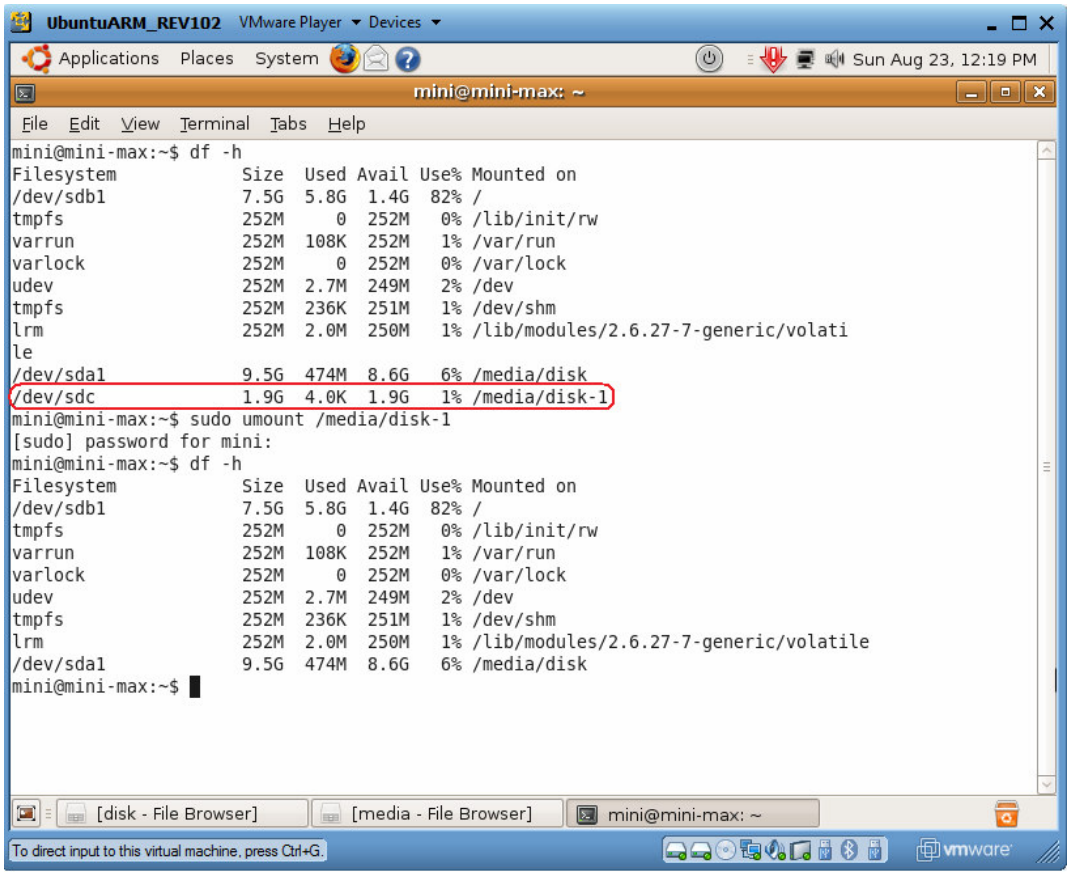


The 10.4.GB Media HDD with Debian RootFS will be mounted under /media/disk.

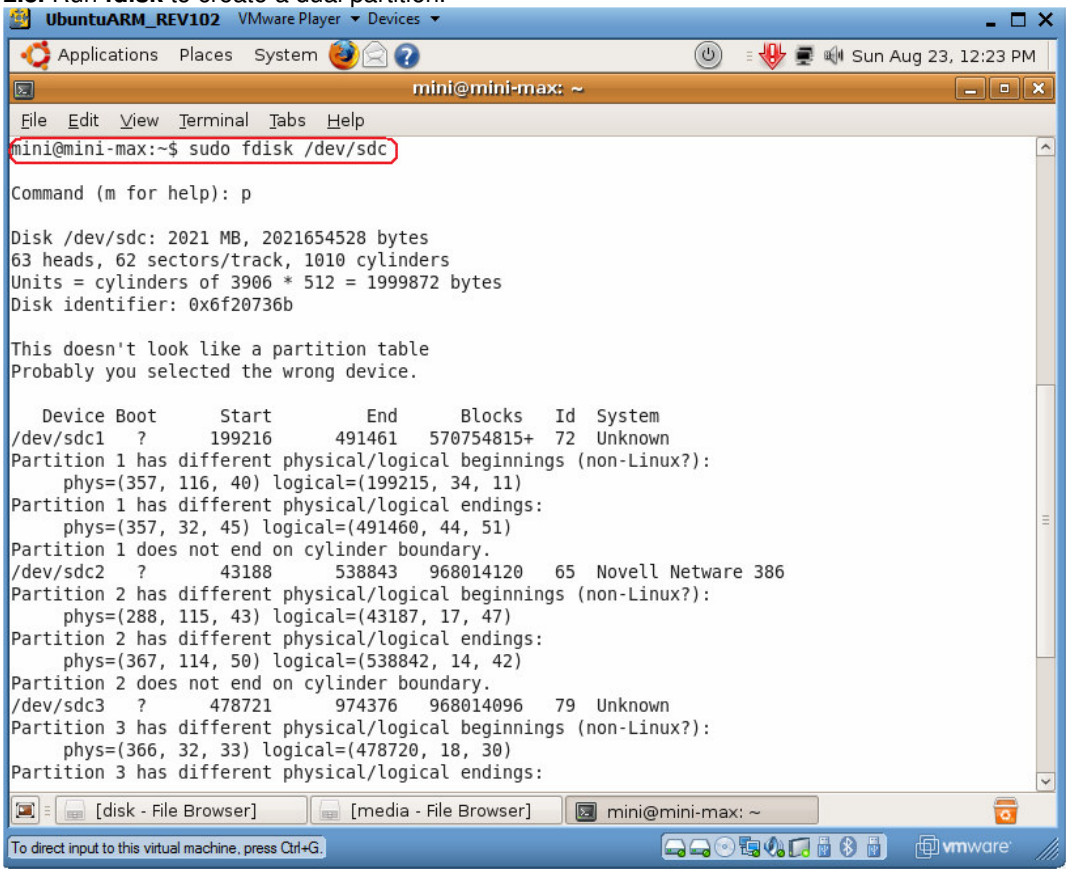
2.6. Install a USB Flash drive to a PC. Connect it to a virtual machine.



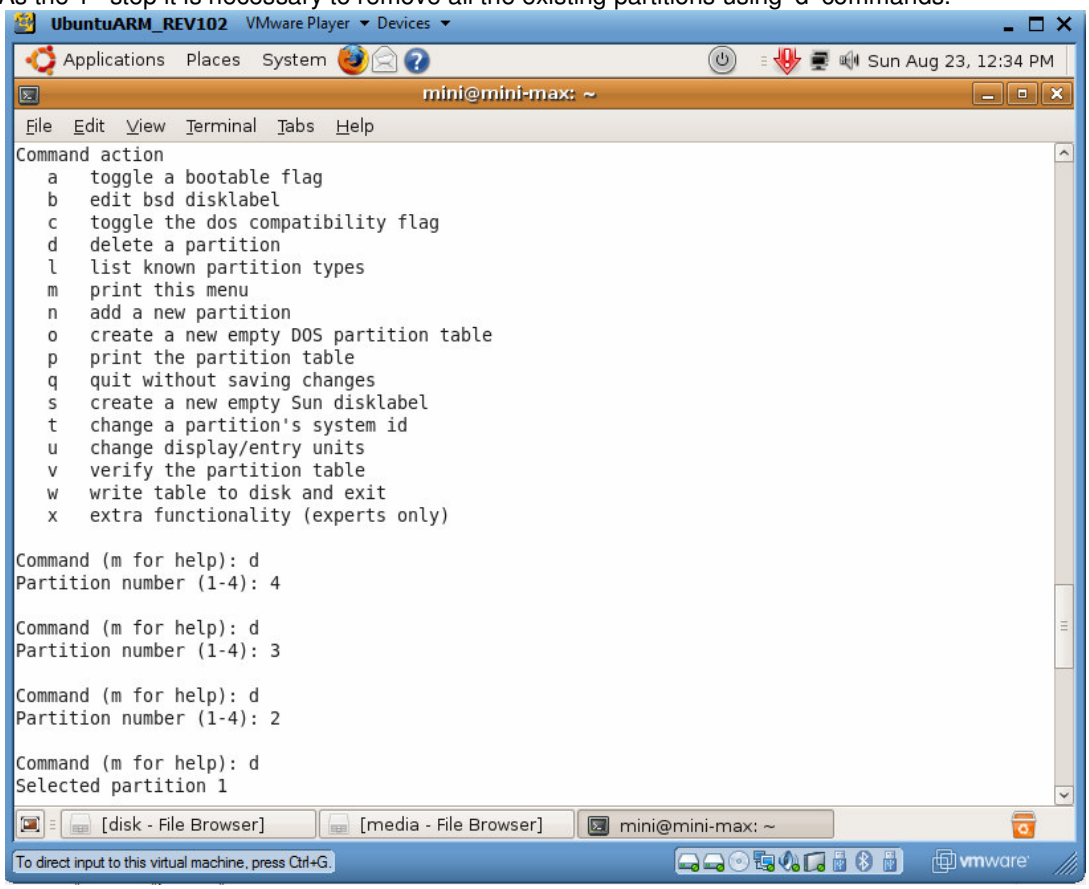
2.7. Unmount /media/disk-1.



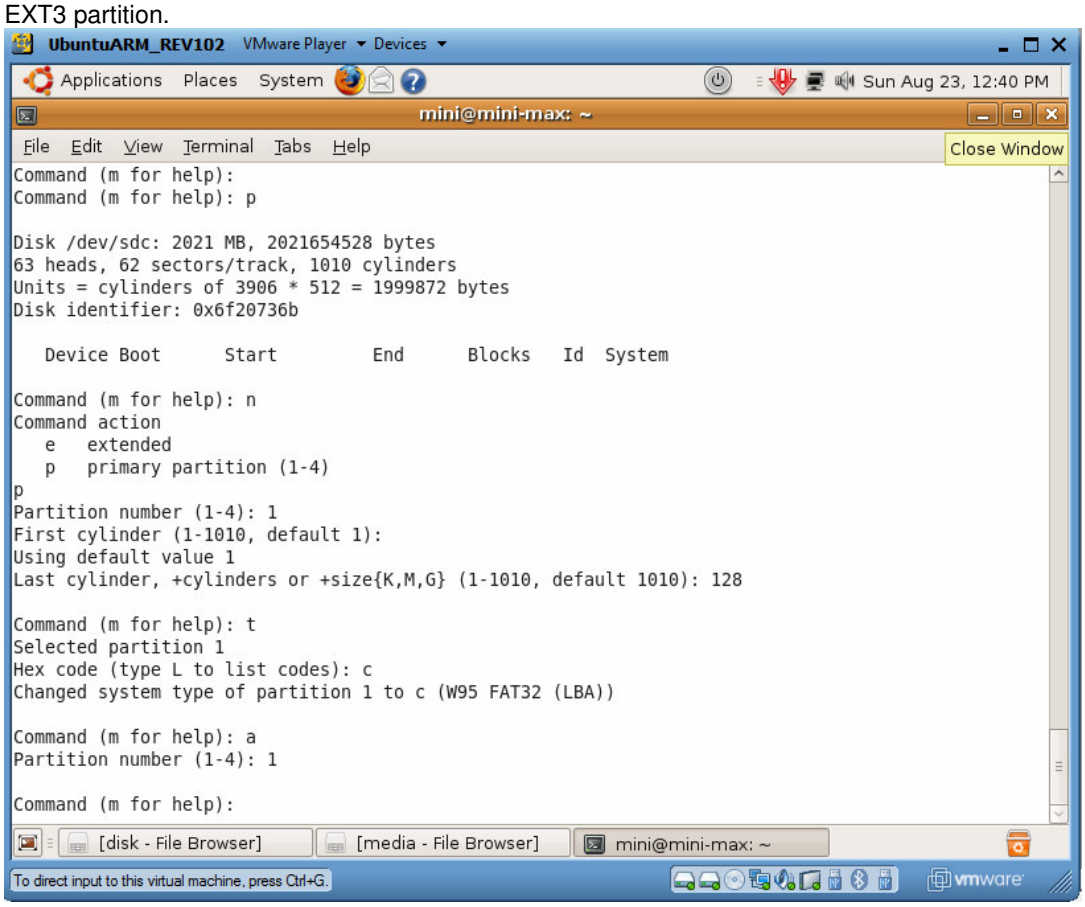
2.8. Run **fdisk** to create a dual partition.

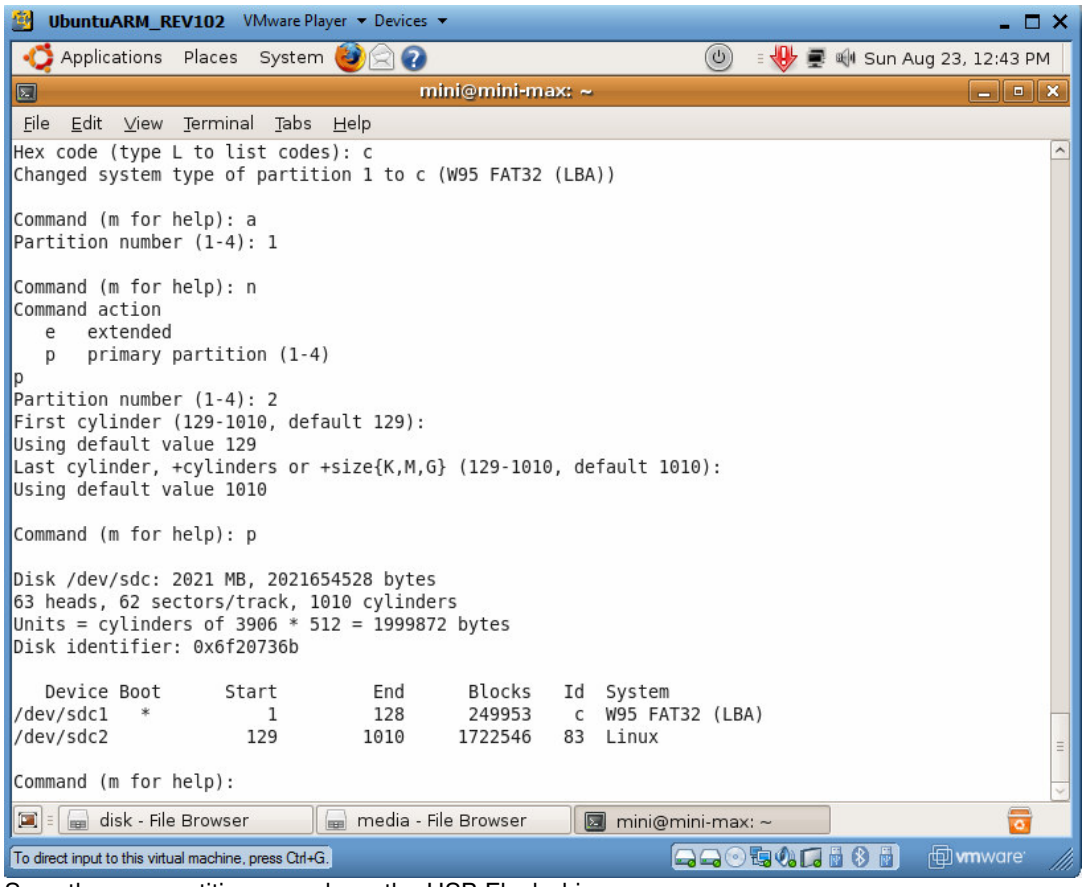


As the 1st step it is necessary to remove all the existing partitions using ‘d’ commands.

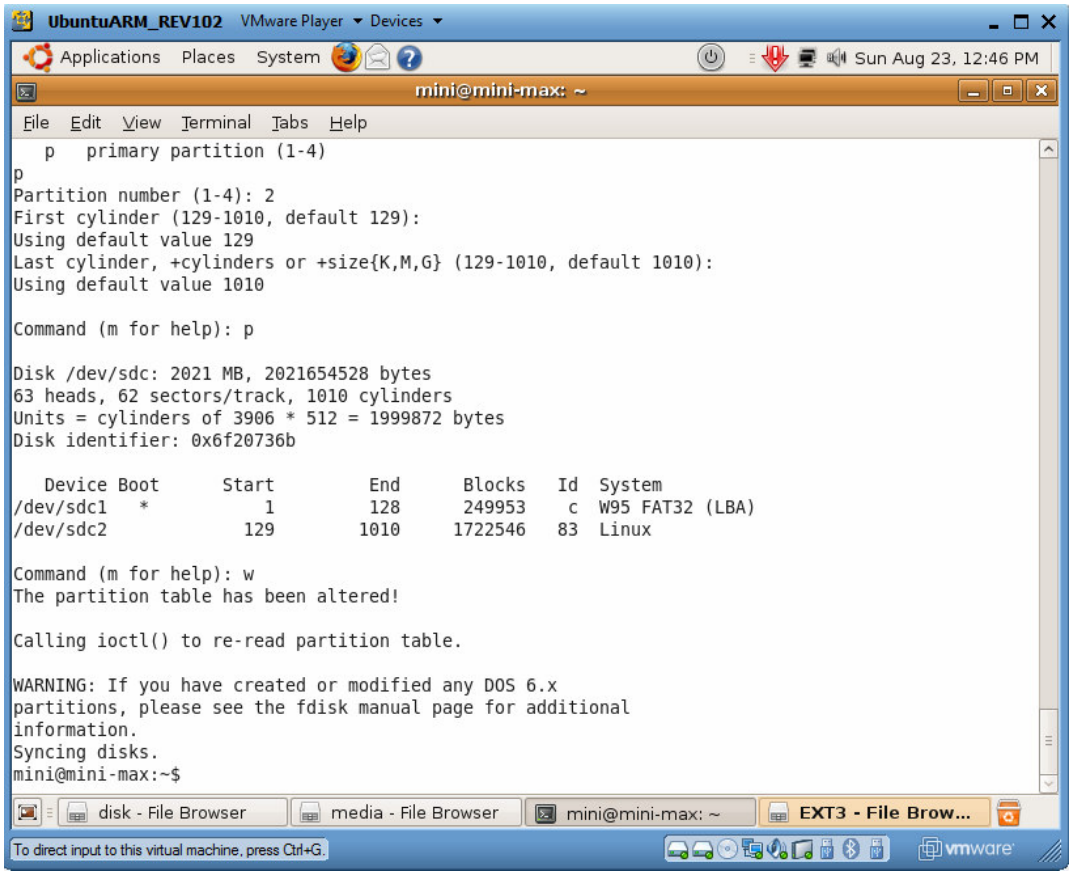


2.9. Create a dual-partition USB Flash drive to boot Linux from first FAT partition and have root file system at second EXT3 partition.

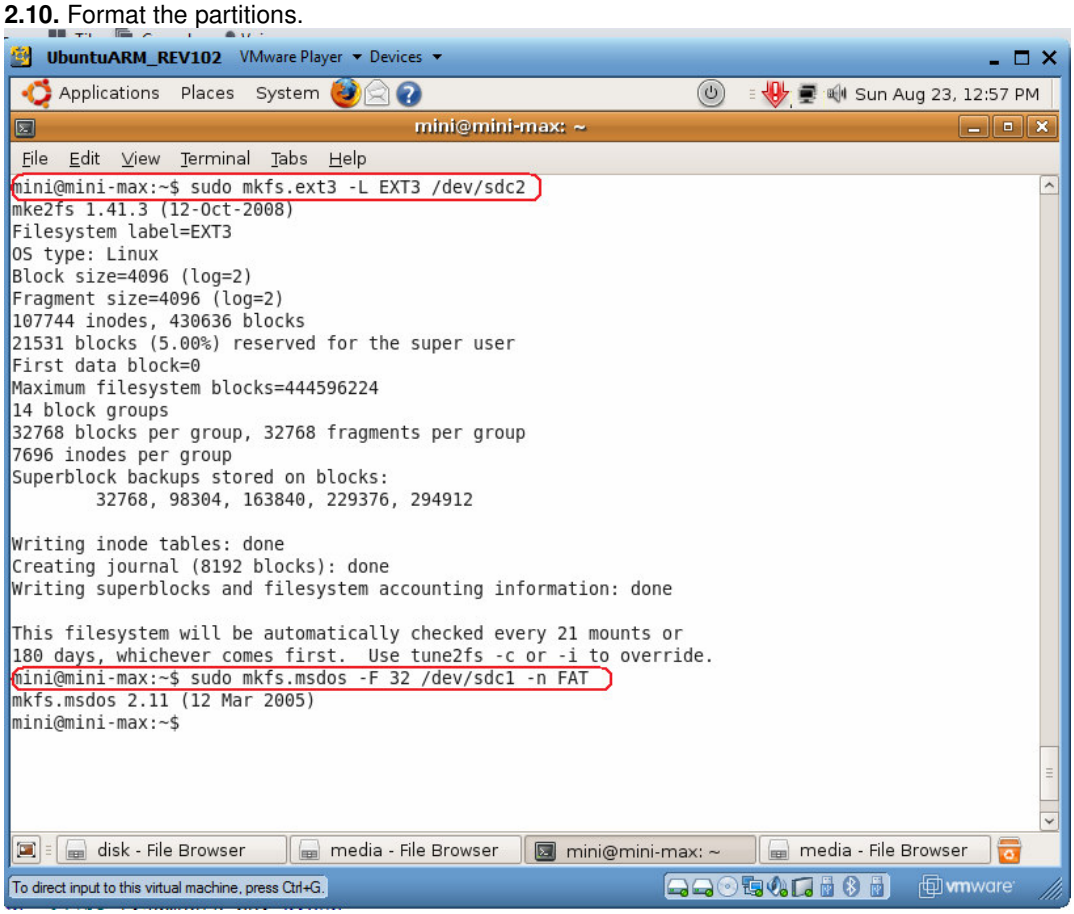




Save the new partition records on the USB Flash drive.

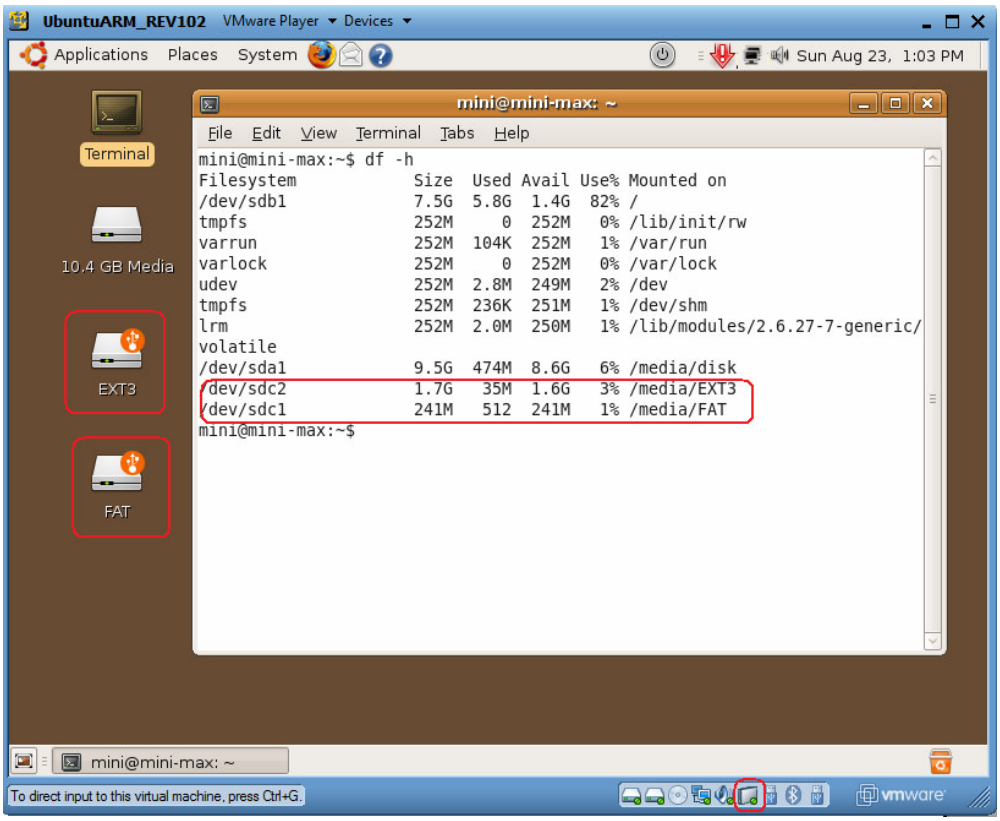


2.10. Format the partitions.

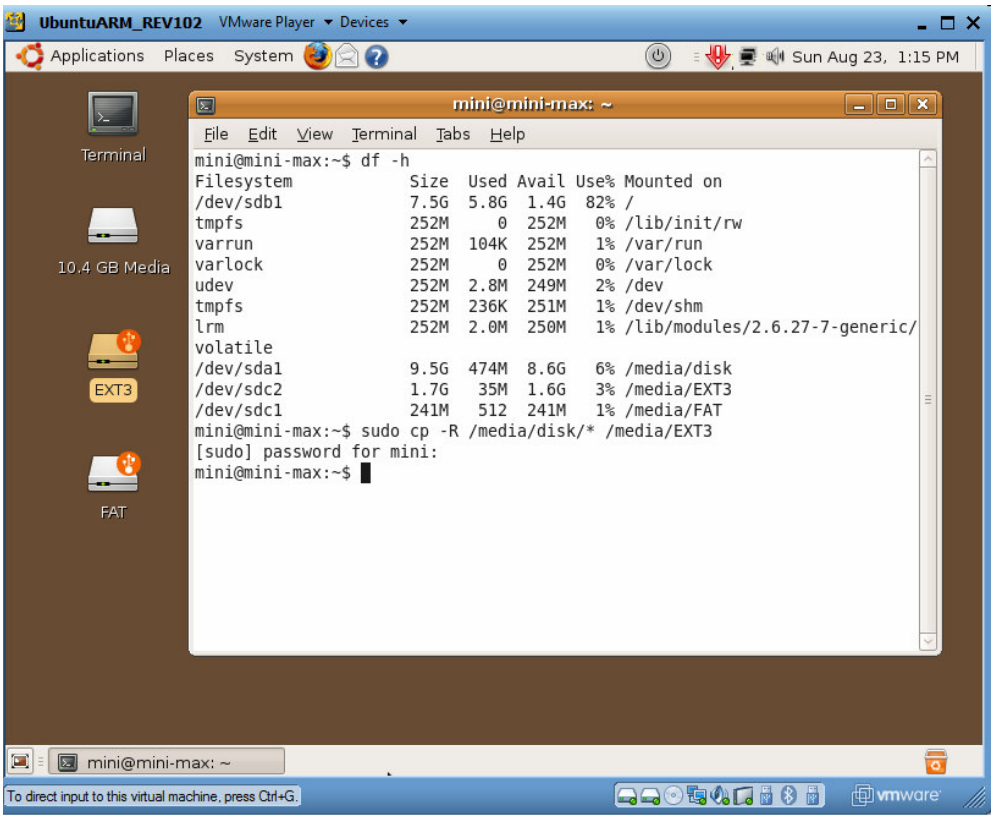


2.11. Disconnect the USB Flash drive from a virtual machine.

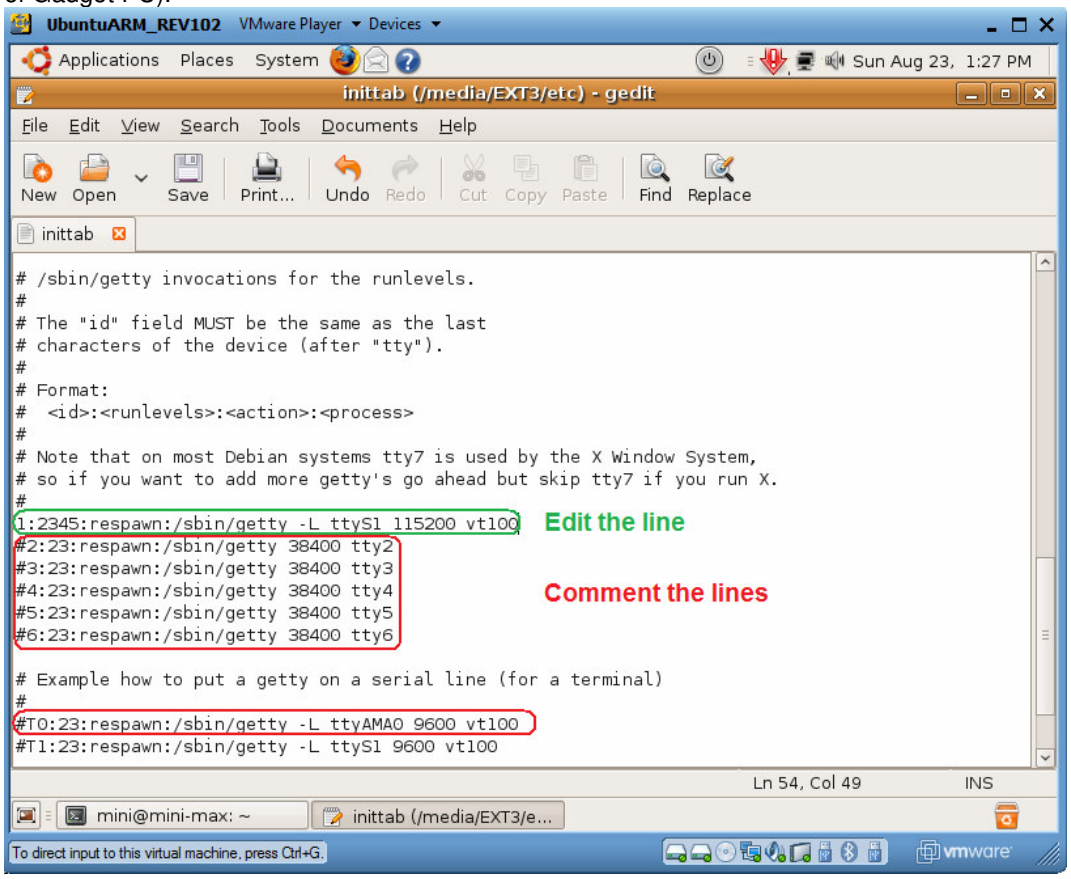
Re-connect the USB Flash drive to a virtual machine. A dual-partition USB will be mounted to /media/EXT3 and /media/FAT.



2.12. Copy the whole file structure from /media/disk (Debian RootFS) to /media/EXT3
It will take 2 minutes.



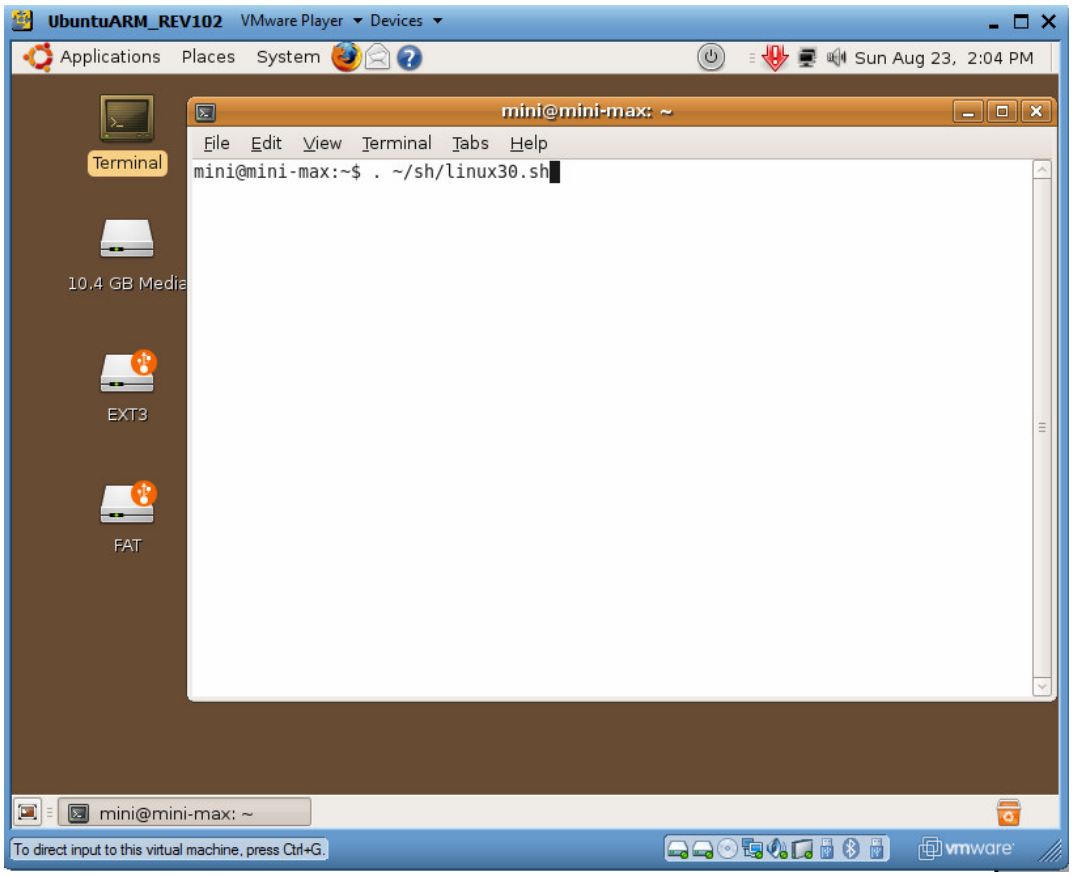
2.13. Run **sudo gedit** to edit **/media/EXT3/etc/inittab** to switch Linux console from a PC keyboard to ttyS1 (UART0 of Gadget PC).



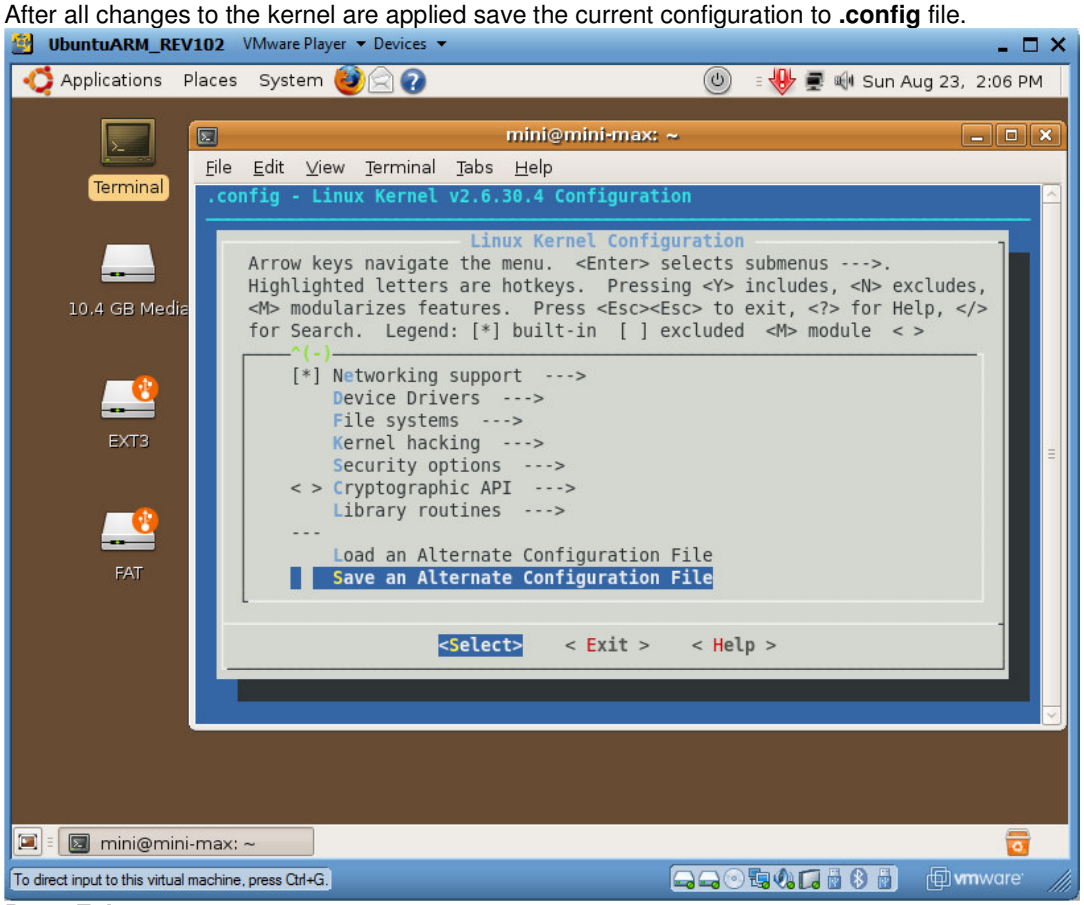
The RootFS is complete to run on GadgetPC.

3. Linux kernel

3.1. To configure the newest Linux kernel please run **Terminal** and execute **. ~/sh/linux30.sh**

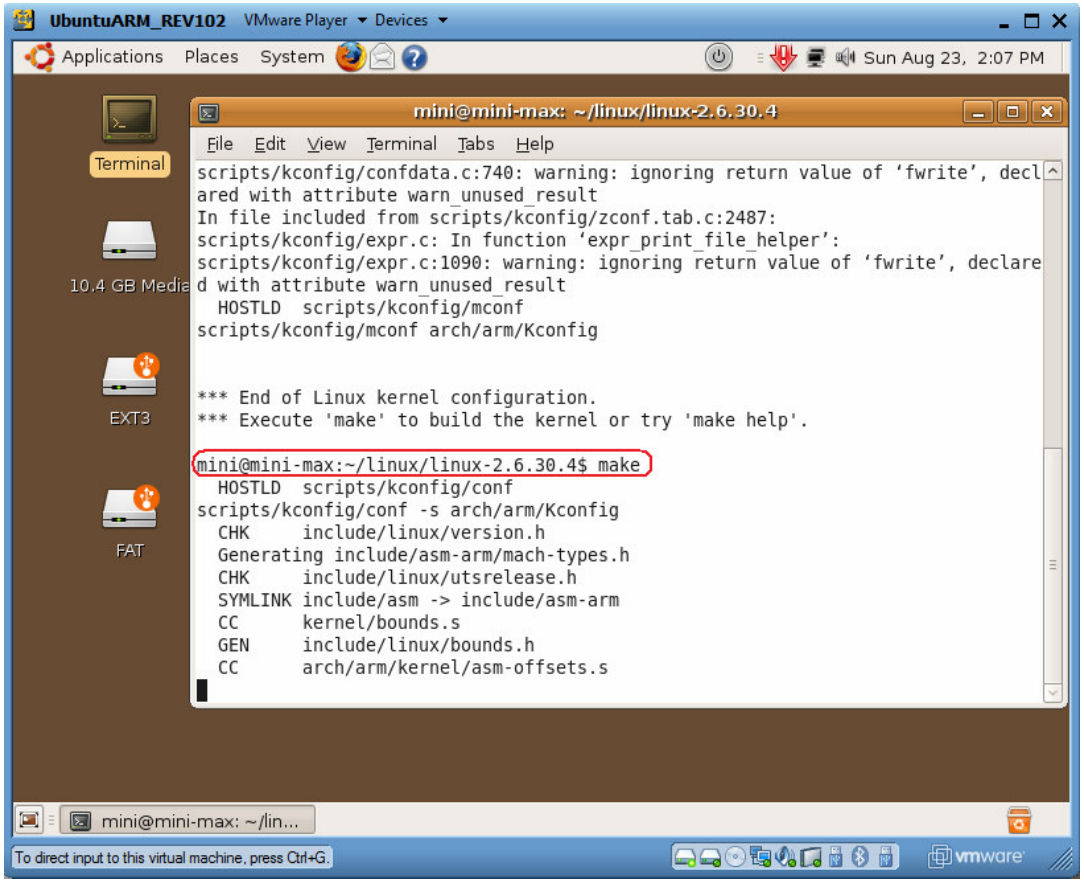


After all changes to the kernel are applied save the current configuration to **.config** file.

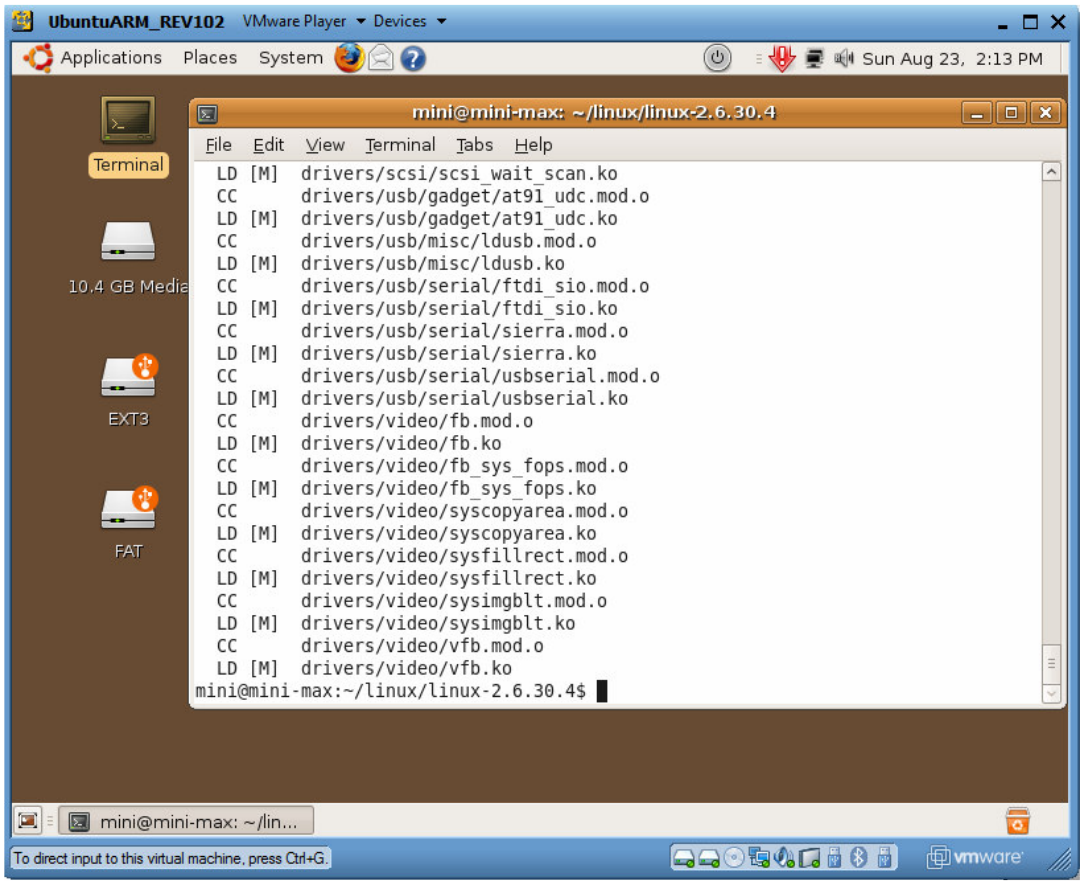


Press **Exit**.

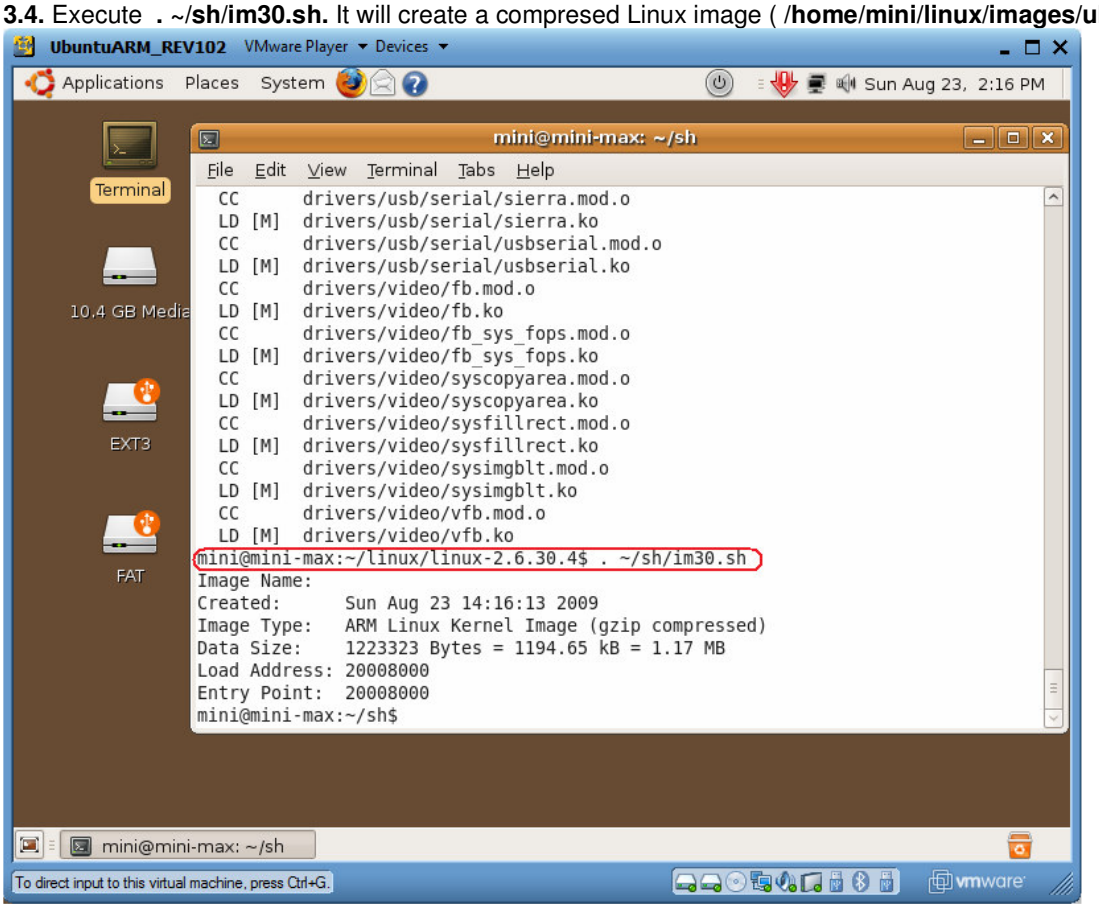
3.2. Execute **make** to build the Linux kernel



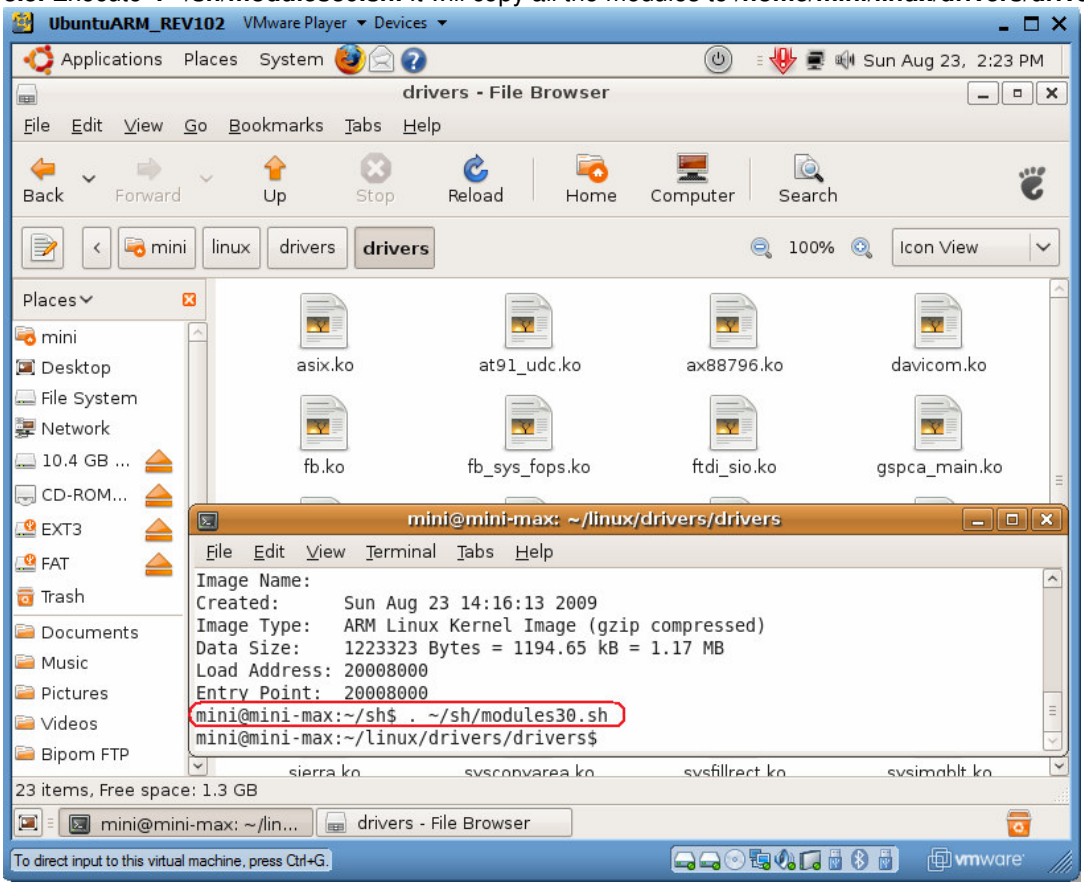
3.3. The newest kernel will be built in 6 minutes.



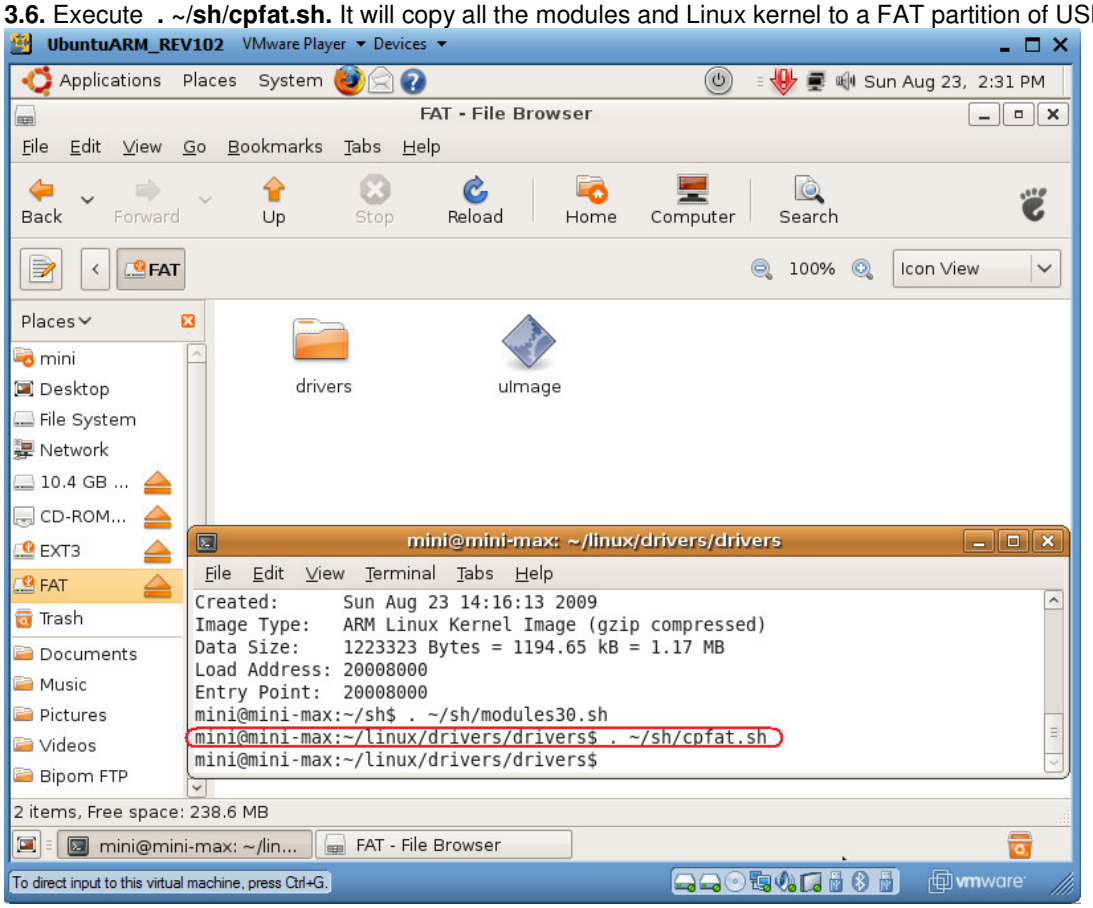
3.4. Execute **. ~/sh/im30.sh.** It will create a compresed Linux image (**/home/mini/linux/images/uImage**).



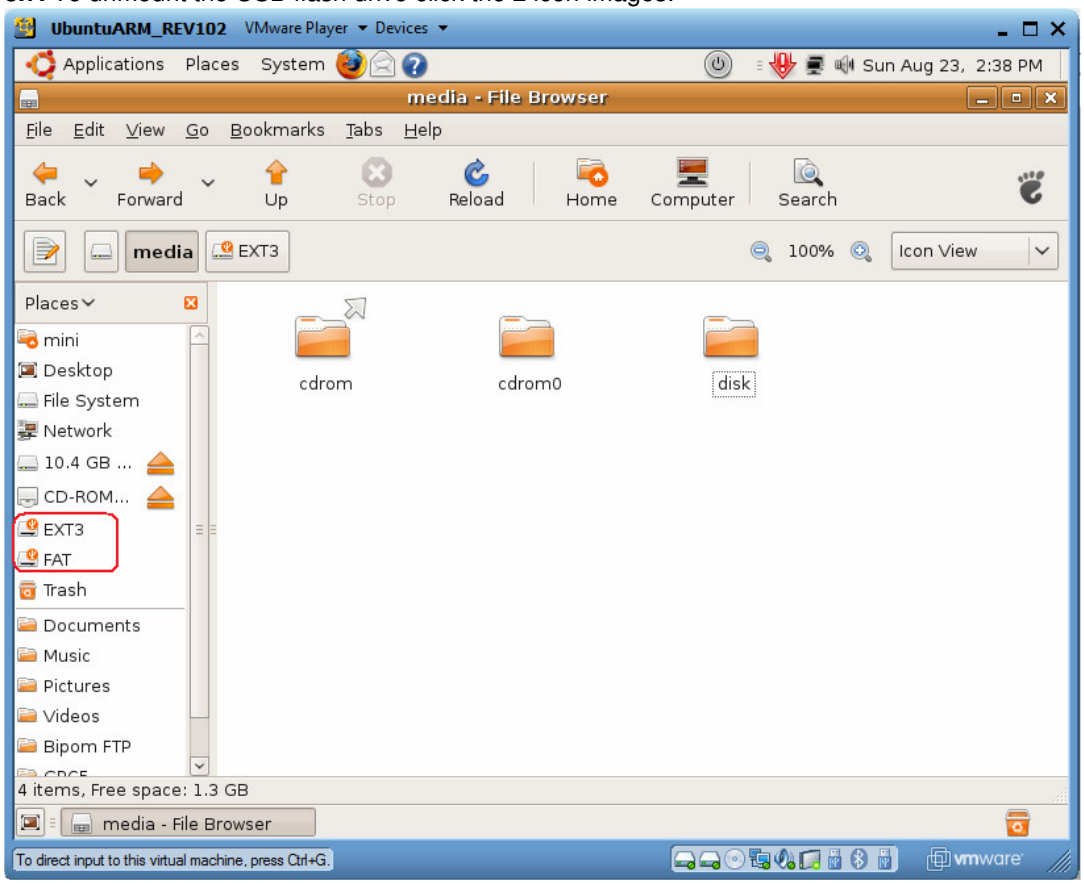
3.5. Execute **. ~/sh/modules30.sh.** It will copy all the modules to **/home/mini/linux/drivers/drivers** folder.



3.6. Execute **. ~/sh/cpfat.sh.** It will copy all the modules and Linux kernel to a FAT partition of USB Flash drive.



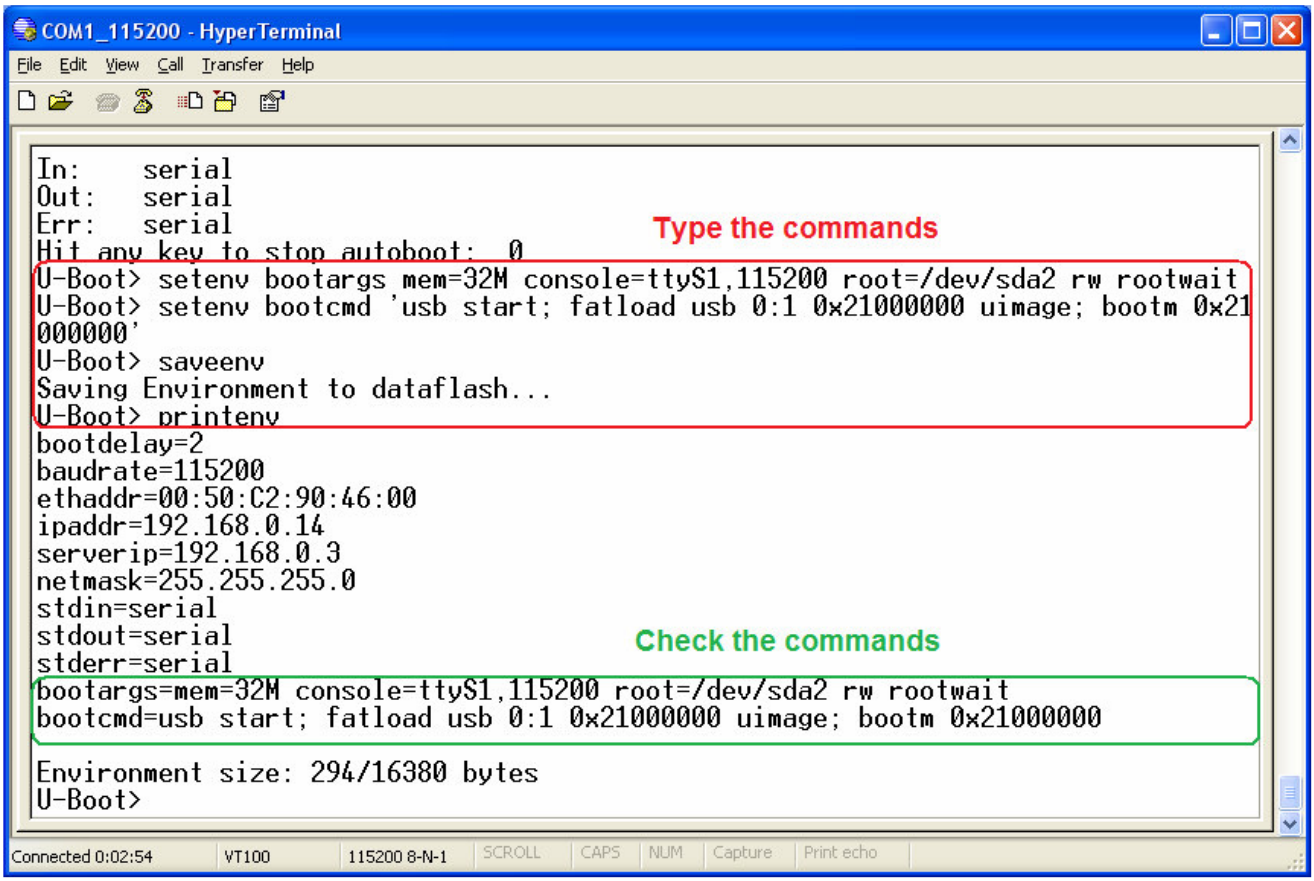
3.7. To unmount the USB flash drive click the 2 icon images.



4. U-Boot loader

4.1. Connect a Gadget PC board to a PC COM port (baudrate is 115200).
Please read **System Installation Guide** document to obtain extra details
http://www.bipom.com/documents/boards/gadgetpc/GadgetPC System Installation Guide.pdf
from http://www.bipom.com/web_documents/2899678.html

4.2. Run the terminal program (for example, hyperterminal), press any key to enter **u-boot** commad mode, type the following commands (see the picture below).



5. Linux Boot sequence
5.1. Install the prepared USB Flash drive to any port of GadgetPC.
5.1. Power the board. The Linux boot log has to appear.
Root password is **root**.
User password is **user**.

```
GadgetPC rev 1.06 08/06/2009
BiPOM Electronics, Inc.
www.bipom.com
AT91BootStrap...OK


U-Boot 2008.10 (Feb 24 2009 - 11:45:00)

DRAM:  32 MB
DataFlash:AT45DB642
Nb pages:   8192
Page Size:   1056
Size= 8650752 bytes
Logical address: 0xD0000000
Area 0: D0000000 to D0003FFF (RO) Bootstrap
Area 1: D0004000 to D0007FFF      Environment
Area 2: D0008000 to D002FFFF (RO) U-Boot
Area 3: D0030000 to D042FFFF      Kernel
Area 4: D0430000 to D083FFFF      FS
In:    serial
Out:   serial
Err:   serial
Hit any key to stop autoboot:  2  1  0
(Re)start USB...
USB:   scanning bus for devices... 3 USB Device(s) found
       scanning bus for storage devices... 1 Storage Device(s) found
reading uimage
.....................................................................................................

1223323 bytes read
## Booting kernel from Legacy Image at 21000000 ...
   Image Name:
   Image Type:   ARM Linux Kernel Image (gzip compressed)
   Data Size:    1223259 Bytes =  1.2 MB
   Load Address: 20008000
   Entry Point:  20008000
   Verifying Checksum ... OK
   Uncompressing Kernel Image ... OK

Starting kernel ...

Linux version 2.6.30.4 (mini@mini-max) (gcc version 4.3.2 (Sourcery G++ Lite 2008q3-41) ) #46 Sun Aug 23
15:32:00 EEST 2009
CPU: ARM926EJ-S [41069265] revision 5 (ARMv5TEJ), cr=00053177
CPU: VIVT data cache, VIVT instruction cache
Machine: Mini-Max/ARM9260
Memory policy: ECC disabled, Data cache writeback
Clocks: CPU 198 MHz, master 99 MHz, main 18.432 MHz
Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 8128
Kernel command line: mem=32M console=ttyS1,115200 root=/dev/sda2 rw rootwait
NR_IRQS:192
AT91: 96 gpio irqs in 3 banks
PID hash table entries: 128 (order: 7, 512 bytes)
Console: colour dummy device 80x30
Dentry cache hash table entries: 4096 (order: 2, 16384 bytes)
Inode-cache hash table entries: 2048 (order: 1, 8192 bytes)
Memory: 32MB = 32MB total
Memory: 29940KB available (2128K code, 193K data, 96K init, 0K highmem)
Calibrating delay loop... 98.91 BogoMIPS (lpj=494592)
Mount-cache hash table entries: 512
```

```
CPU: Testing write buffer coherency: ok
net_namespace: 296 bytes
NET: Registered protocol family 16
bio: create slab <bio-0> at 0
SCSI subsystem initialized
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
NET: Registered protocol family 2
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 1024 (order: 1, 8192 bytes)
TCP bind hash table entries: 1024 (order: 0, 4096 bytes)
TCP: Hash tables configured (established 1024 bind 1024)
TCP reno registered
NET: Registered protocol family 1
ROMFS MTD (C) 2007 Red Hat, Inc.
msgmni has been set to 58
io scheduler noop registered
io scheduler anticipatory registered (default)
atmel_usart.0: ttyS0 at MMIO 0xfefff200 (irq = 1) is a ATMEL_SERIAL
atmel_usart.1: ttyS1 at MMIO 0xfffb0000 (irq = 6) is a ATMEL_SERIAL
console [ttyS1] enabled
atmel_usart.2: ttyS2 at MMIO 0xfffb4000 (irq = 7) is a ATMEL_SERIAL
brd: module loaded
ssc ssc.0: Atmel SSC device at 0xc2828000 (irq 14)
Driver 'sd' needs updating - please use bus_type methods
ohci_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver
at91_ohci at91_ohci: AT91 OHCI
at91_ohci at91_ohci: new USB bus registered, assigned bus number 1
at91_ohci at91_ohci: irq 20, io mem 0x00500000
usb usb1: configuration #1 chosen from 1 choice
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 2 ports detected
Initializing USB Mass Storage driver...
usbcore: registered new interface driver usb-storage
USB Mass Storage support registered.
mice: PS/2 mouse device common for all mice
rtc-at91sam9 at91_rtt.0: rtc core: registered at91_rtt as rtc0
IRQ 1/rtc0: IRQF_DISABLED is not guaranteed on shared IRQs
rtc-at91sam9 at91_rtt.0: rtc0: SET TIME!
i2c-gpio i2c-gpio: using pins 55 (SDA) and 56 (SCL)
Linux video capture interface: v2.00
AT91SAM9 Watchdog: sorry, watchdog is disabled
at91_wdt: probe of at91_wdt failed with error -5
TCP cubic registered
NET: Registered protocol family 17
VFP support v0.3: not present
rtc-at91sam9 at91_rtt.0: hctosys: unable to read the hardware clock
Waiting for root device /dev/sda2...
usb 1-1: new full speed USB device using at91_ohci and address 2
usb 1-1: configuration #1 chosen from 1 choice
hub 1-1:1.0: USB hub found
hub 1-1:1.0: 4 ports detected
usb 1-1.2: new full speed USB device using at91_ohci and address 3
usb 1-1.2: configuration #1 chosen from 1 choice
scsi0 : SCSI emulation for USB Mass Storage devices
scsi 0:0:0:0: Direct-Access     Generic  USB Flash Disk   0.00 PQ: 0 ANSI: 2
sd 0:0:0:0: [sda] 3948544 512-byte hardware sectors: (2.02 GB/1.88 GiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] Assuming drive cache: write through
 sda: sda1 sda2
sd 0:0:0:0: [sda] Attached SCSI removable disk
kjournald starting.  Commit interval 5 seconds
EXT3 FS on sda2, internal journal
EXT3-fs: mounted filesystem with ordered data mode.
VFS: Mounted root (ext3 filesystem) on device 8:2.
```

```
Freeing init memory: 96K
modprobe: FATAL: Could not load /lib/modules/2.6.30.4/modules.dep: No such file or directory


INIT: version 2.86 booting

Starting the hotplug events dispatcher: udevd.
Synthesizing the initial hotplug events...done.
Waiting for /dev to be fully populated...done.
Setting the system clock.
RTC_RD_TIME: Invalid or incomplete multibyte or wide character
ioctl() to /dev/rtc to read the time failed.
Unable to set System Clock to: Thu Jan 1 00:00:38 UTC 1970 □[33m(warning).□[39;49m
Activating swap...done.
Checking root file system...fsck 1.41.3 (12-Oct-2008)
EXT3 has filesystem last checked time in the future, check forced.
□EXT3: |===                                    |  5.0%
□□EXT3: |======                                 / 10.0%
□□EXT3: |========                               - 15.0%
□□EXT3: |============                           \ 20.0%
□□EXT3: |===============                        | 25.0%
□□EXT3: |==================                     / 30.0%
□□EXT3: |====================                   - 35.0%
□□EXT3: |======================                 \ 40.0%
□□EXT3: |=========================              | 45.0%
□□EXT3: |============================           / 50.0%
□□EXT3: |================================       - 55.0%
□□EXT3: |==================================     \ 60.0%
□□EXT3: |====================================   | 65.0%
.....
□□EXT3: |=====================================================   - 95.0%
....
□□EXT3: |=========================================================/ 99.5%
□□EXT3: |==========================================================| 100.0%
□□
□EXT3: 19186/107744 files (0.1% non-contiguous), 99168/430636 blocks
done.
EXT3 FS on sda2, internal journal
Setting the system clock.
RTC_RD_TIME: Invalid or incomplete multibyte or wide character
ioctl() to /dev/rtc to read the time failed.
Unable to set System Clock to: Thu Jan 1 00:01:47 UTC 1970 □[33m(warning).□[39;49m
Cleaning up ifupdown....
Loading kernel modules...FATAL: Could not load /lib/modules/2.6.30.4/modules.dep: No such file or directory
Checking file systems...fsck 1.41.3 (12-Oct-2008)
done.
Setting kernel variables (/etc/sysctl.conf)...done.
Mounting local filesystems...done.
Activating swapfile swap...done.
Setting up networking....
Configuring network interfaces...done.
Starting portmap daemon....
Starting NFS common utilities: statd.
Setting console screen modes and fonts.

INIT: Entering runlevel: 2

Starting enhanced syslogd: rsyslogd.
Starting MTA: exim4.
Starting NFS common utilities: statd.
Not starting internet superserver: no services enabled.
Starting deferred execution scheduler: atdStarting periodic command scheduler: crond.

Debian GNU/Linux 5.0 debian-armel ttyS1

debian-armel login: root
```

```
Password:
Linux debian-armel 2.6.30.4 #46 Sun Aug 23 15:32:00 EEST 2009 armv5tejl

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
debian-armel:~# cd /
debian-armel:/# ls -l
total 76
drwxr-xr-x  2 root root  4096 2009-08-23 11:12 bin
drwxr-xr-x  2 root root  4096 2009-08-23 11:12 boot
drwxr-xr-x  9 root root 12920 1970-01-01 01:02 dev
drwxr-xr-x 55 root root  4096 1970-01-01 01:01 etc
drwxr-xr-x  3 root root  4096 2009-08-23 11:12 home
drwxr-xr-x 11 root root  4096 2009-08-23 11:12 lib
drwx------  2 root root 16384 2009-08-23 10:55 lost+found
drwxr-xr-x  2 root root  4096 2009-08-23 11:12 media
drwxr-xr-x  2 root root  4096 2009-08-23 11:12 mnt
drwxr-xr-x  2 root root  4096 2009-08-23 11:12 opt
dr-xr-xr-x 36 root root     0 1970-01-01 01:00 proc
drwxr-xr-x  4 root root  4096 2009-08-23 11:12 root
drwxr-xr-x  2 root root  4096 2009-08-23 11:12 sbin
drwxr-xr-x  2 root root  4096 2009-08-23 11:12 selinux
drwxr-xr-x  2 root root  4096 2009-08-23 11:12 srv
drwxr-xr-x 11 root root     0 1970-01-01 01:00 sys
drwxr-xr-t  2 root root  4096 1970-01-01 01:02 tmp
drwxr-xr-x 10 root root  4096 2009-08-23 11:14 usr
drwxr-xr-x 13 root root  4096 2009-08-23 11:14 var
debian-armel:/# cd /mnt
debian-armel:/mnt# mkdir -p fat
debian-armel:/mnt# mount /dev/sda1 /mnt/fat
debian-armel:/mnt# cd fat
debian-armel:/mnt/fat# ls -l
total 1197
drwxrwxrwx 2 root root    2048 2009-08-23 15:30 drivers
-rwxr-xr-x 1 root root 1223323 2009-08-23 16:46 uImage
debian-armel:/mnt/fat# cd drivers
debian-armel:/mnt/fat/drivers# ls -l
total 520
-rwxr-xr-x 1 root root 20763 2009-08-23 16:46 asix.ko
-rwxr-xr-x 1 root root 15050 2009-08-23 16:46 at91_udc.ko
-rwxr-xr-x 1 root root 19103 2009-08-23 16:46 ax88796.ko
-rwxr-xr-x 1 root root  4690 2009-08-23 16:46 davicom.ko
-rwxr-xr-x 1 root root 41172 2009-08-23 16:46 fb.ko
-rwxr-xr-x 1 root root  3156 2009-08-23 16:46 fb_sys_fops.ko
-rwxr-xr-x 1 root root 76771 2009-08-23 16:46 ftdi_sio.ko
-rwxr-xr-x 1 root root 27444 2009-08-23 16:46 gspca_main.ko
-rwxr-xr-x 1 root root 14142 2009-08-23 16:46 gspca_pac7311.ko
-rwxr-xr-x 1 root root 54266 2009-08-23 16:46 gspca_zc3xx.ko
-rwxr-xr-x 1 root root 42047 2009-08-23 16:46 hid.ko
-rwxr-xr-x 1 root root  6997 2009-08-23 16:46 ili9325.ko
-rwxr-xr-x 1 root root 16368 2009-08-23 16:46 ldusb.kp
-rwxr-xr-x 1 root root 14367 2009-08-23 16:46 macb.ko
-rwxr-xr-x 1 root root  6526 2009-08-23 16:46 mii.ko
-rwxr-xr-x 1 root root 11979 2009-08-23 16:46 pca9698.ko
-rwxr-xr-x 1 root root 15085 2009-08-23 16:46 sierra.ko
-rwxr-xr-x 1 root root  4490 2009-08-23 16:46 syscopyarea.ko
-rwxr-xr-x 1 root root  5114 2009-08-23 16:46 sysfillrect.ko
-rwxr-xr-x 1 root root  3907 2009-08-23 16:46 sysimgblt.ko
-rwxr-xr-x 1 root root 19226 2009-08-23 16:46 usbnet.ko
-rwxr-xr-x 1 root root 36135 2009-08-23 16:46 usbserial.ko
-rwxr-xr-x 1 root root 68494 2009-08-23 16:46 uvcvideo.ko
debian-armel:/mnt/fat/drivers#
```

6. Alternative releases for advanced Linux users.
6.1. Bipom provides the prepared archives under FTP server:
debian_rootfs_xxxxxxxxx.tar.bz2
linux2.60.30.4_xxxxxxxxx.tar.bz2
where xxxxxxxxx is date.

host=www.bipom.com
port=21
user=bipomftp
pass=guest123!

6.2. Download the packages to a native Linux machine from BiPOM FTP server.
For example,
debian_rootfs_23aug2009.tar.bz2
linux2.60.30.4_23aug2009.tar.bz2

Note. The packages are available under /**home**/**mini**/**fs**/**debian** of Ubuntu virtual machine as well.

6.3. Use the following command to extract files from archive
tar –xvjf debian_rootfs_23aug2009.tar.bz2
tar –xvjf linux2.6.30.4_23aug2009.tar.bz2

6.4. Create a dual partition USB flash drive (see **2.6-2.12**).

6.5. Copy all the files/folders from **linux2.6.30.4_23aug2009** folder **to** /**media**/**FAT**
cp -R linux2.6.30.4_23aug2009/* /media/FAT

6.6. Copy all the files/foders from **debian_rootfs_23aug2009** folder **to** /**media**/**EXT3**
cp -R debian_rootfs_23aug2009/* /media/EXT3