Multimedia Technieken

2015 – 2016

Raspberry Pi 2 - Minibian

katholieke hogeschool
associatie KU Leuven
vives

Setting up an Embedded System

- To setup an embedded Linux system we have multiple choices:
  - Use a pre-built binary distribution such as Raspbian, Ubuntu or Fedora
    - Quick to set up, but not very flexible: support for only a few architectures, no flexibility on package configuration, no easy way to rebuild the entire system automatically.
  - Build all system components manually
    - Highly flexible, but painful and inefficient: need to handle complex cross-compilation issues, understand inter-package dependencies, not reproducible.
  - Use an automated build system, that builds the entire system from source
    - Automated, flexible, handle most cross-compilation issues
    - Examples: Buildroot, OpenWRT, PTXdist, OpenBricks, OpenEmbedded, Yocto, etc.

Using a Minimal Image

- For the sake of simplicity we start by using a pre-build image
  - Raspbian is a very good choice
    - Easy to setup
    - Debian based so familiar
  - Problem is classroom
    - Image read and write take too long
      - Too big
    - Solution
      - Minibian

Minibian

- Minimal Raspbian-based Linux image for Raspberry Pi.
- Main Focus
  - Small, updated and stable distribution that is fully compatible with official Raspbian “wheezy” image, without GUI and unneeded tools.
- Perfect for embedded projects
- Very small footprint, boots in some seconds and uses just few of precious RPi RAM.

- MINIBIAN has not been obtained purging unneeded packages from original image, neither recompiling the source code
  - It’s just a customized Raspbian installation obtained from the same repository used for official RPi wheezy image.
  - So kernel and binary files are exactely the same you will find on standard image, with the difference that MINIBIAN fit on 512MB SD Card, is fastest, and updated more often.

Raspberry Pi 2 - Minibian

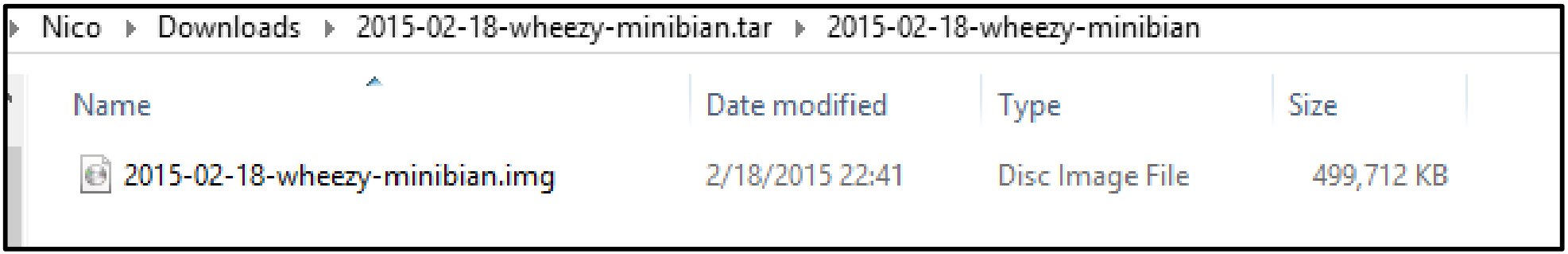
Booting Minibian

katholieke hogeschool
associatie KU Leuven
vives

Downloading the Image

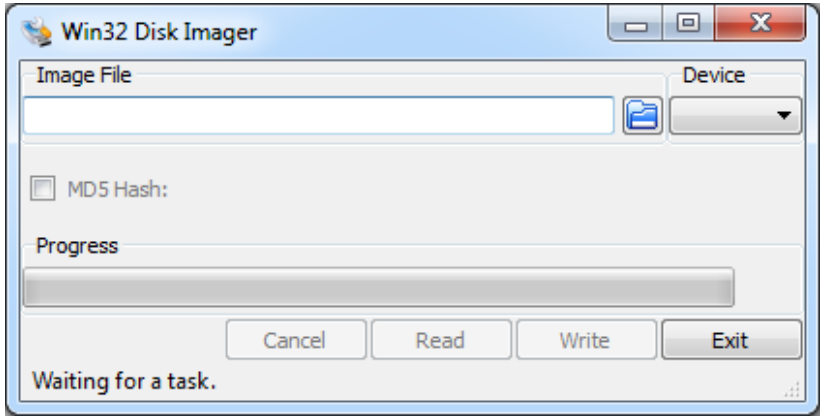
- Minibian can be downloaded here
  - https://minibianpi.wordpress.com/

- You will need a program like 7zip to extract the tarball



Creating an SD card - Windows

- Next the IMG file can be written to an SD card of at least 1GB using Win32 Disk Imager
  - Select the image file and the correct device
  - Next hit the write button



- To make a backup later on just select a non-existing image file (choose location and name yourself)
  - Select the correct device
  - Hit the read button

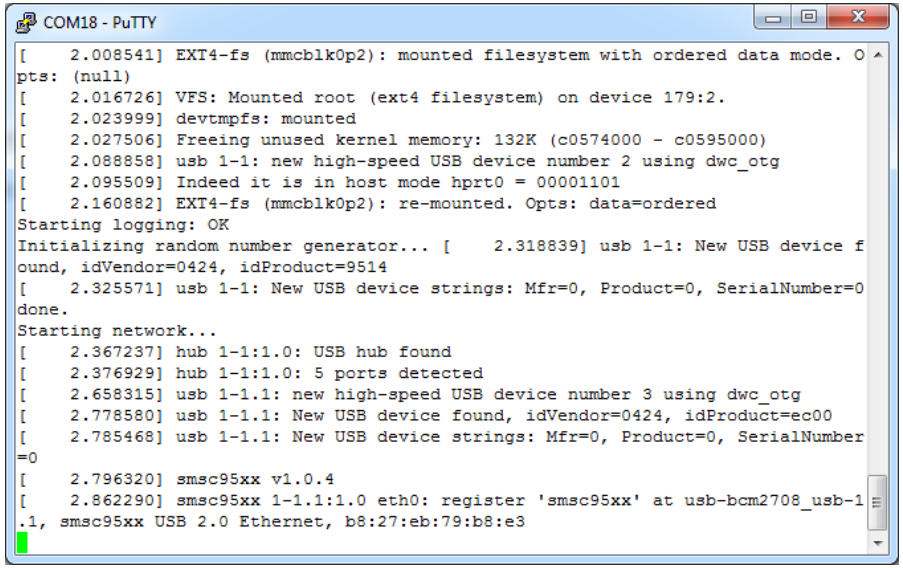
Creating an SD card - Linux

- Just use the ‘dd’ command to write the image to the SD card
  - This is not possible from a virtual machine
    - No direct (RAW) access to SD Card on host machine

```
bioboost@NDWMINT ~ $ sudo dd if=<image> of=/dev/mmcblk0 bs=1M
```

Success

- Insert the SD card into the Pi and connect the power supply
  - This perfectly boots.
  - Attaching a serial to USB would show you the kernel message



- However the actual login prompt is not displayed
  - It is actually available through HDMI

Raspberry Pi 2 - Minibian

Connecting Remotely to the Raspberry Pi

katholieke hogeschool
associatie KU Leuven
vives

Connecting Remotely to a Linux System

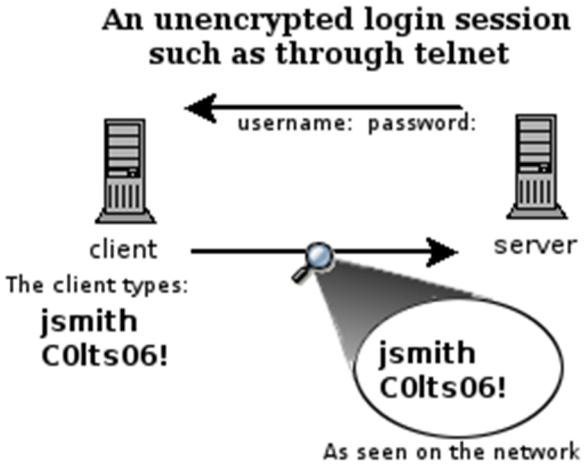
- There are a couple of ways that you can access a shell (command line) remotely on most Linux/Unix systems.
- One of the older ways is to use the telnet program, which is available on most network capable operating systems.
  - Poses a danger in that everything that you send or receive over that telnet session is visible in plain text on your local network, and the local network of the machine you are connecting to.
  - So anyone who can "sniff" the connection in-between can see your username, password, email that you read, and commands that you run.
  - For these reasons you need a more sophisticated program than telnet to connect to a remote host.

Secure SHell

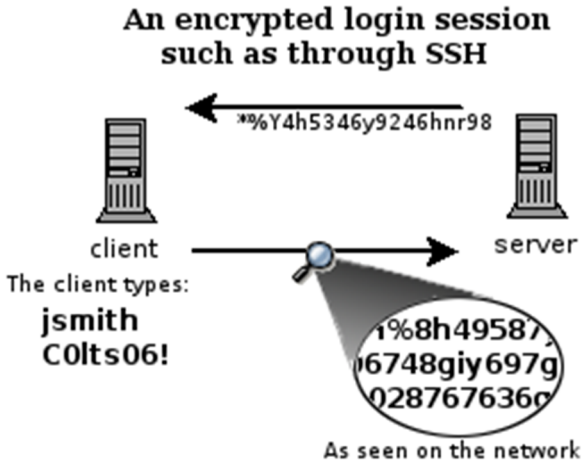
- SSH (Secure Shell), was designed and created to provide the best security when accessing another computer remotely.
- Not only does it encrypt the session, it also provides better authentication facilities, as well as features like
  - secure file transfer,
  - X session forwarding
  - port forwarding
  - …
- It can use different forms of encryption ranging anywhere from 512 bit on up to as high as 32768 bits and includes ciphers like AES (Advanced Encryption Scheme), Triple DES, Blowfish, CAST128 or Arcfour.
  - Of course, the higher the bits, the longer it will take to generate and use keys as well as the longer it will take to pass data over the connection.

Telnet

- Telnet
  - Unsecure



- SSH
  - Secure



katholieke hogeschool
associatie KU Leuven
vives

Connect to Raspberry Pi

- You can use the ‘ssh” command

```
bioboost@NDWMINT ~ $ ssh <username>@<ip-address>
```

- Or you can use the putty tool
  - To use putty first install it using apt-get

```
bioboost@NDWMINT ~ $ sudo apt-get install putty
```

Connect to Raspberry Pi

- We do however still need the IP address of the Pi
  - Use windows netscan or use the ‘nmap’ tool on linux

```
bioboost@NDWMINT ~ $ nmap -sn <network>/<mask_size>
Starting Nmap 6.00 ( http://nmap.org ) at 2015-09-27 13:10 BST
Nmap scan report for 10.0.0.2
Host is up (0.00064s latency).
MAC Address: D4:BE:D9:5E:38:FF (Dell)
Nmap scan report for 10.0.0.3
MAC Address: B8:27:EB:DD:96:A1 (Raspberry Pi foundation)
Nmap scan report for 10.0.0.5
Host is up (0.00078s latency).
MAC Address: 00:22:68:64:37:60 (Hon Hai Precision Ind. Co.)
Nmap scan report for 10.0.0.64
Host is up (0.00040s latency).
MAC Address: 18:59:33:14:51:47 (Cisco Spvtg)
...
Nmap done: 256 IP addresses (5 hosts up) scanned in 3.06 seconds
```

Connect to Raspberry Pi

- So once we know the IP address we can connect to the Pi
  - Default username of Minibian: root
  - Default password of Minibian: raspberry

```
bioboost@NDWMINT ~ $ ssh root@<ip_address>
login as: root
root@<ip_address>'s password:
Linux raspberrypi 3.18.7-v7+ #755 SMP PREEMPT Thu Feb 12 17:20:48 GMT 2015
armv7l

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.
Last login: Sun Sep 27 13:04:01 2015 from 10.0.0.5
root@raspberrypi:~#
```

Raspberry Pi 2 - Minibian

Manually resizing the SD card on the Raspberry Pi

katholieke hogeschool
associatie KU Leuven
vives

Manually resizing the SD card on the Raspberry Pi

- You can actually resize the partitions of the SD card that your Pi is running on
- First you need to change the partition table with fdisk.
- You need to remove the existing partition entry and create a new one
  - This will only change the partition table, not the partitions data on disk.
  - The start of the new partition needs to be aligned with the old partition!

```
root@raspberrypi:~# fdisk /dev/mmcblk0
```

- Then delete partitions with *d* and create a new with *n*.
- You can view the existing table with *p*.

Manually resizing the SD card on the Raspberry Pi

- p to see the current start of the main partition
  - d, 2 to delete the main partition
  - n p 2 to create a new primary partition
    - next you need to enter the start of the old main partition and then the size.
    - The main partition on the Minibian image from 2015-02-18 starts at 97728
      - But the start of your partition might be different. Check the p output!

- w write the new partition table

Manually resizing the SD card on the Raspberry Pi

```
root@raspberrypi:~# fdisk /dev/mmcblk0

Command (m for help): p

Disk /dev/mmcblk0: 7969 MB, 7969177600 bytes
4 heads, 16 sectors/track, 243200 cylinders, total 15564800 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0004a452

        Device Boot      Start         End      Blocks   Id  System
/dev/mmcblk0p1              16       97727       48856    b  W95 FAT32
/dev/mmcblk0p2           97728      999423      450848   83  Linux
```

Manually resizing the SD card on the Raspberry Pi

```
Command (m for help): d
Partition number (1-4): 2

Command (m for help): n
Partition type:
   p   primary (1 primary, 0 extended, 3 free)
   e   extended
Select (default p): p
Partition number (1-4, default 2): 2
First sector (97728-15564799, default 97728):
Using default value 97728
Last sector, +sectors or +size{K,M,G} (97728-15564799, default 15564799): +750M

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or
resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
```

Manually resizing the SD card on the Raspberry Pi

- Next just restart the Raspberry Pi

```
root@raspberrypi:~# reboot
```

- After the reboot you need to resize the filesystem on the partition.
  - The 'resize2fs' command will resize your filesystem to the new size from the changed partition table.

```
root@raspberrypi:~# resize2fs /dev/mmcblk0p2
resize2fs 1.42.5 (29-Jul-2012)
Filesystem at /dev/mmcblk0p2 is mounted on /; on-line resizing
required
old_desc_blocks = 2, new_desc_blocks = 3
The filesystem on /dev/mmcblk0p2 is now 768000 blocks long.
```

Manually resizing the SD card on the Raspberry Pi

- You can check the new available space using the disk-free command

```
root@raspberrypi:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
rootfs          719M  334M  347M  50% /
/dev/root       719M  334M  347M  50% /
devtmpfs        484M     0  484M   0% /dev
tmpfs            98M  188K   98M   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           195M     0  195M   0% /run/shm
/dev/mmcblk0p1   48M   15M   34M  30% /boot
tmpfs           195M     0  195M   0% /tmp
```