Raspberry Pi as appliance

Open Source Days 2013, Copenhagen

Robert Scheck

fedora™

Robert Scheck

Fedora Package Maintainer and Provenpackager

Fedora Ambassador and Ambassador Mentor

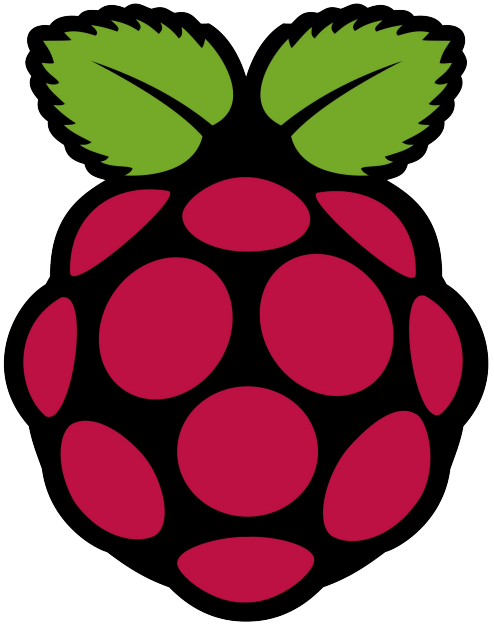
Part of Fedora Websites and Translation teams

Open Source Contributor and Software Developer

Mail: robert@fedoraproject.org
Web: http://fedoraproject.org/wiki/RobertScheck

What is Raspberry Pi?

- Credit-card-sized single-board computer
- Developed in the United Kingdom by a foundation
- Originally intended for teaching basic computer science in schools



- Price: 25-33 EUR
- Size: 85.60 mm × 53.98 mm × 17.00 mm
- Weight: 45 g

fedora

Raspberry Pi (model B)

Creative Commons Attribution-Share Alike 3.0 Unported

fedora

Hardware specifications

- ARMv6 CPU (ARM1176JZF-S) with 700 MHz
- 256 or 512 MB memory (shared with GPU)
- SD / MMC / SDIO card slot for storage
- Broadcom VideoCore IV GPU
- 10/100 Ethernet (optional), 1-2 USB 2.0 ports
- Composite RCA (PAL/NTSC), HDMI and DSI
- 3.5 mm jack, HDMI and I²S audio
- 500-700 mA (2.5-3.5 W)

Raspberry Pi as appliance?

- Relatively cheap
- Small device
- Still powerful
- Energy saving
- Fan-less
- Various connectors
- Linux-friendly

Operating systems

- AROS
- Haiku
- Linux (e.g. Debian, Fedora, Gentoo)
  - Still no (?) vendor with longer product life-cycle
  - Version upgrades vs. self-support and back-ports
- Plan 9 from Bell Labs
- RISC OS
- FreeBSD, NetBSD

Simple overclocking

- Overclocking of CPU up to 1200 MHz
  - Configuration via `/boot/config.txt`
  - Originally overvolting meant to void warranty
  - Settings and details at http://elinux.org/RPiconfig
  - Overclocking can impact system stability
- Overclock and overvoltage will be disabled at run-time when the SoC reaches 85 °C
  - Limit should not be hit even with max. settings at 25°C ambient temperature
  - Passive cooling via heat sink

Power supply & consumption

- Micro USB (type B) power connector
  - 5.0 V with 0.5 A (model A) or 0.7 A (model B)
- Lots of different user experiences known
  - http://elinux.org/RPi_VerifiedPeripherals
- Not every power adapter provides what it says
  - Partially working example: Samsung micro-USB power adapter ETA0U10EBECSTD has 5 V/0.7 A
  - Working example: Samsung micro-USB power adapter ETA0U80EBEGXEG has 5 V/1.0 A

Power usage and limits

- Originally limited 100 mA per USB port
  - Model B hardware revision 2.0 (since 08/2012) and revision 1.0 with ECN0001 have no limiting polyfuses
- Power consumption may increase due to
  - connected USB devices
  - high CPU or GPU usage
  - heavy SD or network I/O throughput
  - overclocking and overvolting
  - model/type of SD card

Memory vs. GPU

- 256 or 512 MB memory shared with GPU
- Originally memory splitting via different ELF files for 128/192/224/240 MB memory in `/boot`
  - Now via setting `gpu_mem` in `/boot/config.txt`
  - At least 16 MB memory must be still assigned to GPU in order to have a bootable system
- Trade-off between GPU performance (3D, HD) and regular CPU/system memory usage

Soft vs. Hard Floating Point

- VFP (Vector Floating Point) technology
  - FPU co-processor extension to ARM architecture
  - Low-cost single-precision and double-precision floating-point computation
- ABI incompatibility between SFP and HFP
  - ARM1176JZF-S is HFP → performance benefit
  - Decision by Linux distribution, ARMv5 often without
  - CFLAGS: `-march=armv6z -mtune=arm1176jzf-s -mfpu=vfp -mfloat-abi=hard`
  - Do a benchmark for your case

SD / MMC / SDIO card

- SD card is mandatory for booting
  - Internal permanent storage up to 256 GB
  - Generally extendable via USB or network
- Lots of different user experiences known
  - http://elinux.org/RPi_SD_cards
- Example: SanDisk Extreme SDHC Class 10
  - 19-22 MB/s (via `dd`, speed also depends on batch)
  - 4 GB → 11 EUR, 128 GB → 120 EUR
  - Less benefit with “Extreme Pro”

Filesystem tuning

- Better I/O performance when using SD cards
- Consider mount options for ext4 in `/etc/fstab`
  - `noatime`: Do not update inode access times
  - `nouser_xattr`: Disable extended user attributes if you do not need them, see also: `man 5 attr`
  - `noacl`: Disable POSIX access control lists, if you do not need them, see also: `man 5 acl`
- Prefer ext4 over ext2 because of journaling
  - Trade-off: Speed vs. safety

Porting software

- Natively compile software to be run on ARM
  - Cross-compiling may introduces other issues
- Unclean or non-portable code will simply fail during compilation (or at latest while executing)
- Many standard software was already fixed and thus simply works on ARM as expected
- Your application might be not yet ARM-ready

Typical code example

```
void TraceMsg(char *lpMsg, int time, char *func,
              char *format, va_list va) {
  /* … */
  if(format && va) {
    va_copy(va_lentest, va);
    len += _vsnprintf(NULL, 0, format, va_lentest);
    va_end(va_lentest);
  }
  /* … */
}
```

- Compiling fails with “error: invalid operands of types”
- `va_list` is internally typed as integer on e.g. Intel, but not on ARM architectures
- Workaround: preprocessor macro

Another code example

```
void shmservinit(char **tables) {
  char **cpp;
  int i;
  timetab_t *tp;

  for(cpp = tables, i = 0; *cpp; ++cpp, ++i);
  shmsiz = ++i * sizeof *timetab + sizeof *lay;
  shmsiz /= PAGE_SIZE;
  ++shmsiz;
  shmsiz *= PAGE_SIZE;
  shminit();
  /* … */
}
```

- Kernel symbol `PAGE_SIZE` exists on e.g. Intel systems in code headers, but not on ARM
- Solution: `sysconf(_SC_PAGESIZE)`

Lack of real-time clock

- Ask user during system boot
  - Not very professional nor helpful for an appliance
- `nortc` by Open Source @ Seneca
  - Rough time by last access or mount date
- Use network time server (NTP client)
  - Requires Internet or at least network connectivity
- DS1307 board with battery via I²C interface
  - Separate hardware
  - Additional costs of 10 EUR

Lack of hardware buttons

- Raspberry Pi has no power or reset button
- Users maybe need to shutdown the appliance
  - Simply remove power cable
  - SSH login to execute `poweroff`
  - X or web interface
  - Integration into your application
- Trade-off: Fragile database vs. robust software

Ideas for backup concept

- Nobody wants a backup, but might be needed
  - Limited life-time of SD card
  - End user mistakes
  - Raspberry Pi gets easily lost (or stolen)
- Possible locations for the backup
  - USB memory stick (`mount` on attaching, copy backup, `umount` afterwards, LED for signalling)
  - Network or cloud (requires network access)
- Think about restore concept

Ideas for update/maintenance

- Software needs updates and maintenance
  - Bug fixes, security patches, new features
  - Affects operating system and your application
- Possibilities for updating the appliance
  - New SD card image → no (?) upgrade path
  - Network software repository for e.g. `yum` or `apt`
  - Simply nothing (less funny, but common)
- Think about product life-time if you do a commercial appliance

Raspberry Pi as appliance

- Media Center / Home Theater PC
  - XBMC on OpenELEC, Raspbmc, Xbian, Raspbian
- Groupware and e-mail server
  - Zarafa Collaboration Platform on Fedora
- Anti-spam for e-mail
  - eleven eXpurgate (proprietary experiment)
- Enterprise Resource Planning (ERP)
  - mercaware erp case (proprietary)

Some hardware add-ons

- “3G + GPS shield” by Cooking Hacks
- “Ultimate GPS Breakout” by Adafruit
- “Raspberry Pi Case” by ModMyPi
- “PiFace” and “Gertboard” by element14
- Arduino shield “Alamode” by Wyolum
- “Raspi-LCD” by emsystech engineering

Questions?

fedora™


Thank you!