Linux on cellphones

Pavel Machek

Phones are everywhere

- everyone has their cellphone
  - and carries it whereever they go
- cellphones are not just phones any more
  - they browse web
  - can read mail
  - play mp3s and videos
  - play radio
  - they show maps, and you can use them for navigation

Phones are sensitive

- They contain your contacts
  - ...your passwords
  - ...your emails
  - ...can eavesdrop on you
  - ...can steal your money and transfer them to attacker
- Backups are important because they break down
  - non-smart phones do not have adequate ways to backup more than contacts

Phones are working against their owner

- Cellphone operators have „interesting“ requirements before they'll sell a cellphone
- Branded phones are actively evil here
  - right button takes you right into provider's portal, and you pay for it
    - without confirmation
    - without chance to change that
  - branding is non-removable, so you are stuck with looking at red wallpaper
  - you can't use it with other operator
- MMS / push to talk are designed to be expensive
- Voice-over-IP is a big no-no for a phone

Phones are working against their owner

- You can only transfer pictures out of a phone using MMS
- You can only download applications using GPRS
- You can't transfer pictures/apps/songs between phones
- Have to confirm actions even of your own apps

Phones are limited

- (but maybe that's a good thing?)
- Java applications work everywhere
  - but they can't do interresting stuff
  - usually can't access microphone, camera
  - can't go background
  - can't interact with one another
- Symbian / Windows Mobile are slightly better here

Phones are powerful

- 0.4-1.5GHz CPUs, often dualcore
- 128MB-512MB RAM
- 128MB-32GB flash
- GPRS connection ~5KB/sec, EDGE ~25KB/sec, UMTS ~40KB/sec, HSDPA ~100KB/sec
- WIFI

Sharp Zaurus

- 2001 Sharp SL-5500 (aka collie)
- 2004 Sharp SL-3000 (aka spitz)
- Qtopia
  - Linux system with Qtembedded
- Then OpenZaurus
- ...and Angstrom

...powerful enough for Linux

- Siemens SX1-- low end symbian cellphone from 2003
  - 116g
  - ARM cpu @120MHz
  - 16MB RAM
  - 24..32MB flash depending on model
  - MMC slot
  - 176x220 color display
  - USB client, bluetooth, GPRS
  - misdesigned keyboard, misdesigned radio parts

Greenphone

- Trolltech's qtopia based phone
  - 0-9*# keyboard
- Expensive
- Evil EULA
- Important parts are non-free
  - but at least it is not locked down by DRM

Neo 1973

- 2007
- ARM s3c2410 @ 266MHz, 128MB RAM, 64MB flash
- 2.8“ VGA screen
- 1.2 Ah battery, microSD slot, bluetooth
- resistive display

OpenMoko on Neo

- basic functionality (display, touchscreen, audio, GSM) works
- MicroSD support is flakey
- X/gtk+ works
  - but it is not clear if gtk+ is suitable for finger-controlled applications
- ipkg packaging system works, allows installing things like python
- qemu based cross-development environment
- charger is funny

OpenMoko

- pretty much normal system
  - busybox for size, but full system is possible
- PDA components
  - ipkg system
  - battery meter
  - on-screen keyboard
- Phone components
  - gsmd

UI

- Stylus is not a mouse
- Finger is not a stylus
  - bigger buttons are needed
  - feedback outside area that is pressed is needed

GSM functionality

- GUI code is needed
- Is there good standard for contacts? vcard?
- ...for calendar? vcalendar?
- Should be useful for desktops, too.

Power management

- On desktop, hibernation is nice
- On laptop, suspend or hibernation is very useful to have
- On PDA, suspend is mandatory
- On cellphone, suspend is mandatory, but you have to pretend you are not suspended
    - what is right interface for that?
    - should select() wake the system when timeout is done?

Nokia n900

- 2009
- X/gtk
- Maemo 5
- Resistive touchscreen
- TI OMAP @600MHz, C64x DSP
- 256MB RAM
- 800x480 display

..and others

- Palm PRE
- Motorola A1200

T-Mobile G1

- aka HTC Dream
- 2008
- Qualcomm MSM @528MHz
- 192MB RAM, 256MB flash

Androids

- Army of Androids is huge
- Different sizes
- Keyboards or not
- Cheap
  - From cca 3500CZK up (Vodafone 845)
- Usable
- So our Dream is here, right?
  - Right?

Android system

- Linux kernel
  - + lot of non-standard patches
  - MSM is difficult to support
- Bionic libc
- Custom Java interpretter
  - Not even standard glibc
  - Neither X nor dbus etc
- Applications terminated by OOM killer

Android security

- Each application has separate user
  - Applications are separated from each other
    - You can use closed-source applications with
  - Each application has separate permissions
    - Even network access needs permission
- You typically do not get root
  - Sometimes even operator “enhancements” are present

+/-

- + cheap, usable cellphone
- + Linux
- + open source user land
- - closed source Google apps
- - can't do system updates
- - can't do advanced stuff like tethering
- - can't run standard Linux apps

So you want custom apps?

- You can use Java SDK
- Terminal does not need root permissions
  - So you can run command line applications
  - But you'd better have keyboard
    - No ctrl/alt keys, no arrows
- Can do Java frontend for commandline app
  - navit
- SI4a
  - Python, Perl, Lua, Ruby, Tcl...

sl4a

- Apache License 2.0
- It is possible to hack directly on phone
- Access to Andoid APIs
  - GPS, sensors, backlight, charger,camera
  - Reverse geolocation
  - Limited UI interface
  - Text-to-speech
  - Control silent mode, airplane mode, wifi, backlight

rooting

- Hacking your own phone
- Not necessary on ADP1
- Sometimes you can use manufacturer's backdoor
- Kernel hole works every time
  - They are common enough :-)

Inside android

- Bootloader
- Recovery system (also Linux)
- Full system

What to do with root

- Full system update
  - Cyanogen ROMs
- Tethering
- sshd
- Debian in chroot
  - offlineimap
  - Can do Java X server + X apps
    - But you don't want to...

Questions?

- ?