Fixing the leaking tap
--- power management enabling and tuning for IA Tizen phone

Austin Zhang
Yong Wang
Agenda

- Power management requirements to mobile devices
- Tizen mobile power management implementation
- The way of enabling new platform and tuning
- Cases study from IA phone
- Summary
What are the differences?
Additional Requirements

- **Longer battery life**
  - Only did needed jobs
  - Did job as fast as possible
  - Enter into low power status as much as possible
  - Avoid unnecessary wakeup as much as possible

- **Avoid race conditions**
How did Tizen mobile?
How does Tizen mobile implement PM – power states & transitions

Event including:
Software timeout
User interactive
Power APIs calling
Wakeup events
How does Tizen mobile implement PM – Kernel space

- Standard way of suspend/resume and enter into sleep state (STR for Tizen)
- Wakeup Events Framework from 2.6.36
  
  /sys/power/wakeup_count
  pm_stay_awake / pm_relax / pm_wakeup_event
How does Tizen mobile implement PM – User space

- power-manager: one daemon to implement and manage those states
- Device manager: provide platform specific callbacks
How does Tizen mobile implement PM – APIs (till to Tizen 2.1)

- **power**
  - power_lock_state / power_unlock_state / power_wakeup ……

- **libsIlp-pm**
  - pm_lock_state / pm_unlock_state / pm_change_state

- **power-manager**
  - Receive input from others

- **how it works**
Then how to enable new platform and tune?
Enabling NEW platform – Kernel space

- Implement traditional suspend/resume
- Implement runtime power management
- Using wakeup event framework interface in drivers if needed
Enabling NEW platform – Fill callback gaps for platform

• Platform specific callbacks implementation
  • Implement platform plugin of device-manager
device-manager-plugin-xxx
  • Implement those power related callbacks
    wakeup event, lcd, backlight
• Using power APIs in middleware/applications
Enabling NEW platform – User space tuning

- Most part of cases are from user space apps
- Power budget
  - Power consumption in key scenarios
  - Monitor regressions, no surprise in the last day
- IA platform specific tuning for various scenarios
  - System idle
  - Media playing
  - Browser
  - Telephony
Enabling NEW platform – Way and Tools for tuning

- Regular regression monitor
- Tizen commits logs and ‘bisect’
- Power/performance tools
  - PowerTOP: who is waking up the system frequently?
  - perf: are those hotspots reasonable?
Cases study from IA phone
Cases study -1

• Are you using Android implementation?
  • As to power management field, It is easy to transform to Tizen solution☺
Cases study -2

- Control the screen on/off may have different way for different platform
  - power-manager prefers to control this by itself
  - Driver implement of display operations in different platform with different way
Cases study -3

• Unexpected wake-up brings system back from idle to active
  • Unnecessary timer
More learning

- **Graphics**
  - Wrong status restore leads to low performance in GPU after resumed from suspend

- **Modem**
  - Always remember to put the modem into low power when not use
  - Disable non-critical modem events to reduce the wakeup from suspend

- **Multimedia**
  - HW acceleration features help a lot in power saving

- And …..
Summary

• **What?**
  For mobile devices: Additional requirements

• **How?**
  In Tizen
  • Standard Linux PM framework and Wakeup Event framework
  • User space PM daemon as decision maker

• **And then?**
  For enabling new platform
  • Implement platform specific callbacks in from kernel/driver to user space
  • Tuning is very important

• **Always keep Power Management in mind in full development cycle**
Credits to:

- Peng Li, Yan Zhang, Halley Zhao, Caiwen, Zhang, Vivian Zhang, Yan Yin, Yu Ma, Guobing Chen, Jie Chen, Dongsheng Sun, Seunghun Pi, MyungJoo Ham... sorry if I missed anyone.
Questions?