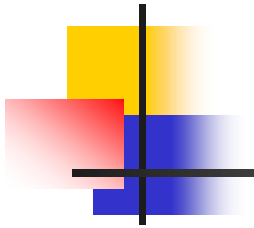
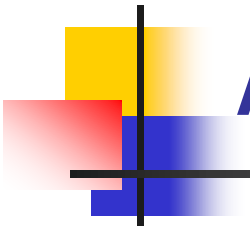


Maemo Activity in SUAI

•Evgeny Linsky
•Anton Dogadaev
•Vitaly Petrov
•Dmitry Rodionov

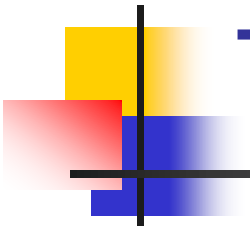


Agenda

- Finished project
  - **"Endurance testing of Maemo"**
- Ongoing project
  - **"Performance tools evaluation"**
- Proposed student project
  - **"Infrastructureless p2p network"**

Endurance Testing

- Regular actions, repeated many times
- Try to find «hard to find errors» in the default set of maemo application
  - Memory leakage
  - File system descriptors leakage
  - System fall
  - …

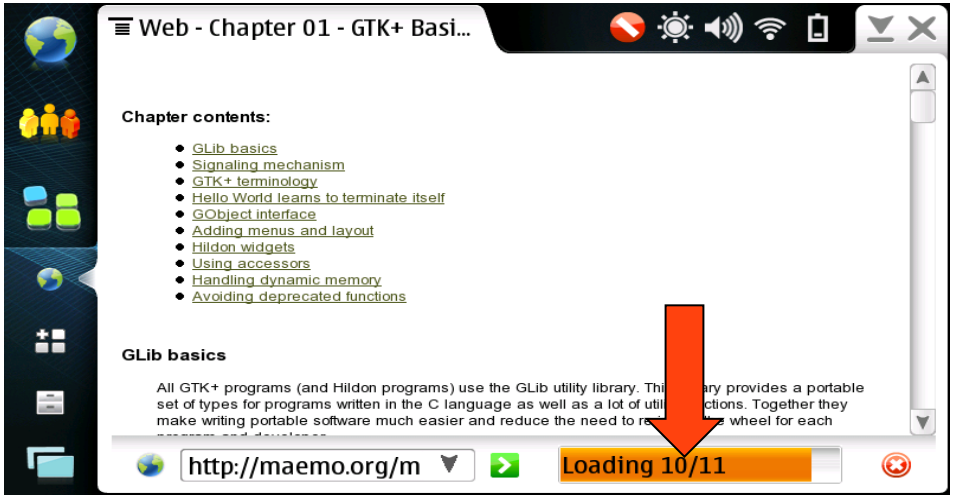


Technology

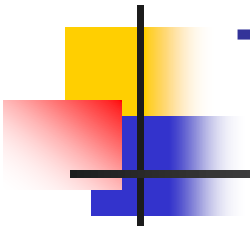
1. Run application in typical scenarios for long-time
   - Do the same simple actions in cycle
   - Example:
     - Open PDF Reader
     - 1. open PDF, 2. scroll through it, 3. go to 2.
2. Do system snapshots periodically
   - Snapshot: syslog, /proc
3. Compare snapshots with each other
4. Find difference in
   - Resource usage
   - Execution time of single steps
5. Analyze logs, try to understand the error reason

Tools and Apps

- Tools
  - Xnee , dbus
    - (repeat action sequence)
  - Xprop , xresponce
    - (check changes on screen)
  - Tool for gen. Report

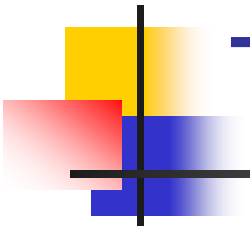


- Apps
  - Browser
  - Chat
  - Scetch
  - Pdf
  - Notes
  - Gtalk
  - Audio/video player
  - Image browser



Type of tests

- One application tests
  - Simple (repeat only)
  - Simple communications (wait till the end of previous action)
  - Advanced communications (2 devices work together)
- Many apps tests
  - Mixed
  - User day (simulate year of user activity)



Test Conditions

- Decisions:
  - Leak --- memory
  - Fall --- device reboots
  - Hang up --- application is not responding
  - OK --- all is ok
- Each test is executed for at least 3 hours
- Each test is executed on at least 3 different devices
  - The decision is made by majority
  - 4 Fall, 1 Ok = Fall

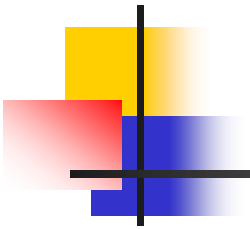
One Application Tests

- Simple
  - Open paint, draw something, ...
- Simple communications
  - Open rss, subscribing, geting news, unsubscribing ...
- Advanced communications
  - 2 devices (server, client) turn on chat , chat between each other , disconnect .....

One Application Tests Results

Each test is executed at least 3 times.
The duration of each run is at least 4 h.

NAME		Time/result/device		Full time	result
notes\chtext	4h10m, 0ms	4h10m, 0tr	4h, 0ad	12h	ok
video\fullvid	10h30m,+16.7ad	10H30m,+18.3el	2h, +2.5gm	23h	leak
image\rotate	5h, 0ms	45m, 0Ftr	1h15m, 0Ft1	7h	fault
email\test2	16h, +2.1ad	5h, +0.3tr	16h, +2.4t1	37h	Small leak
google-chat	6h20m, +3.0t1	6h10m, +1.4ad	8h, +17.4gm	20h	Small leak



Many Apps Tests

- Mixed
  - Tests run in sequence
  - Interaction between tests (e.g. write and read from same file)
- User day
  - Several tests work in parallel
  - Simulate user activity for year in months
  - Defined scenario of user activity

User Day

• Simulate activity of the one "user day" for specified amount of days

	Real time		Test time
Audio call	20 min	→	3 min
Media player	1 h 50 min	→	4 min
PDF	2 h	→	17 min
Browser	4 h	→	15 min
Gchat	2 h	→	5 min
...			

Time for real day: 12h 52 min **Time for test day:1h 16 min**

Year of user work can be simulated in 5 weeks!

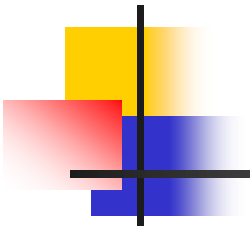
Many Apps Tests Results

- Mixed test

NAME	Time/result/device			Total time	result
Adv. comm.	6h30m, +3ad	6h30m, +3ms	6h30m, +5t1	19h 30 min	Small leak
Mixed simple	10h, Fad	20h, +6.9ms	20h, +6.4tr	50 h	leak

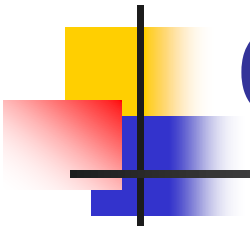
- User day test

NAME	Time/result/device			Total time	result
Uday	19h, +4.7t1	19h, +3.2tr	19h, +3.4ms	57 h	Small leak



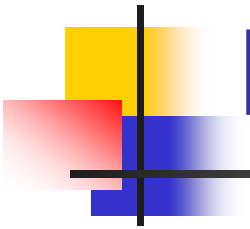
Result Summary

- One Application Tests Results
  - Simple: memory leak range 0..40 Mb, many faults
  - Simple communications: memory leak range 0..4 Mb
  - Advanced:memory leak range 0..6 Mb
- Many Apps Tests Results
  - Mixed: memory leak range 7..20 Mb
  - Uday: memory leak range ~4 Mb



Ongoing Project

- Name: “Performance tools evaluation”
- Development tools used for system programming on PC
  - **la-trace**: traces dynamic calls to libraries
  - **systemtap**: gathers different kernel statistics
  - **chronicle**: debugger
  - …
- Project goal: port these tools on Maemo

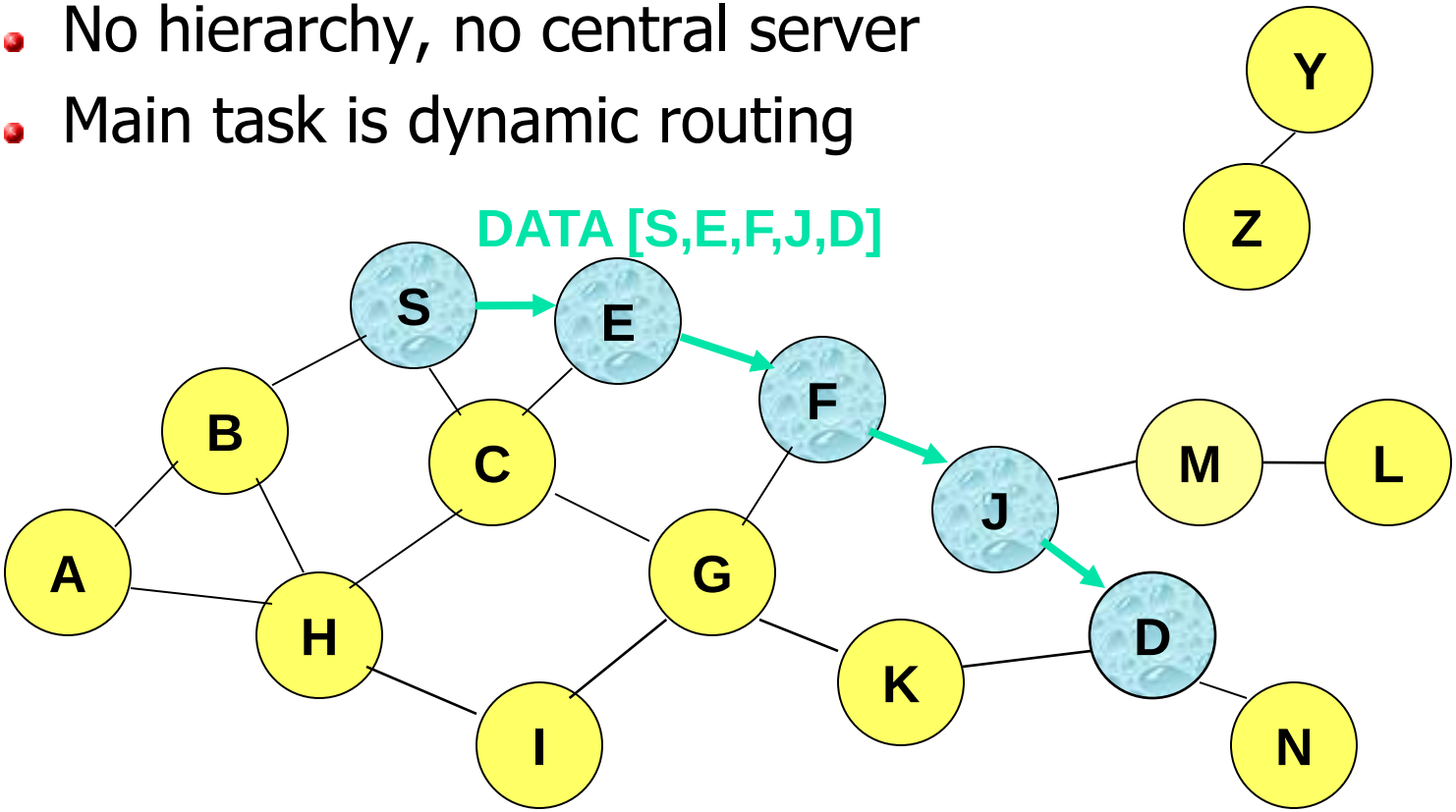


Project Challenges

- Usually such tools include system-dependent code, e.g.
  - Differences in hardware (x86 vs ARM)
  - Differences in environment (versions of libraries, kernel features)
- Porting requires communication with tool developers

Proposed Student Project

- Name: “Infrastructureless p2p network”
- P2P file sharing over mobile ad-hoc network
  - Nodes are mobile (users with N810): arbitrary join/leave
  - No hierarchy, no central server
  - Main task is dynamic routing



Initial Plan

- We plan
  - Implement ad-hoc routing protocol for Maemo
  - Implement gnutella-like p2p file sharing network on top of routing

Routing candidates:

- DSR: RFC4728
- AODV: RFC3561