

Async Programming in Samba

Linuxkongress

Oktober 2009

Volker Lendecke

SerNet

Samba Team



Volker Lendecke

- Co-founder SerNet - Service Network GmbH
 - Free Software as a successful business model
 - Network Security for the industry and the public sector
 - Samba-Support/Development in Germany
- For almost 20 years concerned with Free Software
- First patches to Samba in 1994
- Consultant for industry in IT questions
- Co-founder emlix GmbH (Embedded Systems)



Motivation

- Samba has to do many things at once
 - Network calls can block indefinitely
 - While one server is blocked, other requests can be fulfilled
- One solution: pthreads
 - Samba does not use threading for many reasons
 - Threads look easy at first, but become very nasty to debug
- Samba uses event-driven programming



Event driven programming

- Some central event loop calls you back when something interesting happens
 - File descriptors read/writable, signals, timeouts
- Hairy from the beginning:
 - Memory handling is hard: When do you free memory associated with a network request
 - Sequences of multiple blocking requests require complex state handling
- C can not store the whole program state, call/cc is reserved to Scheme and Ruby...



Talloc

- Andrew Tridgell developed talloc, a hierarchical memory allocator
- Initially a standard memory pool allocator
- Every memory block is a pool itself
 - The talloc hierarchy is a tree structure
 - You delete the tree root, all children are also deleted
- Talloc provides destructors: You can close fd's before the file record is free'd



Tevent

- Samba provides its own event library
- We monitor fd's, signals and timeouts
- Why our own? NIH?
 - Tevent is talloc integrated, to remove an event you talloc_free its representation
 - Our signal handlers get the siginfo struct, which we need for Linux kernel oplocks
- Tevent has a flexible backend to adapt to other event loops like for example the gnome one



Tevent_req

- A standard event loop does not help with complex sequences of computation
- You have to split up the code at arbitrary places
 - The sequential structure of the code that makes threads attractive is destroyed
- Several Co-Routine libs for C exist, most are non-portable or have severe restrictions
 - GNU pth is the only exception
 - A smooth migration to GNU pth is not possible



Tevent_req

- Samba has tried for years to make async code readable and maintainable
- Bad examples
 - Winbind up to 3.4 (Look at current code! :-))
 - Parts of Samba 4 got it wrong
 - Ldb has yet another async model
- The current tevent_req model is our latest iteration
- Let's walk through an example....



Questions/comments?

Volker Lendecke, VL@SerNet.DE

SerNet - Service Network GmbH
Bahnhofsallee 1b
37081 Göttingen

Tel: +49 551 370000 0

Fax: +49 551 370000 9

<http://www.SerNet.DE>

<http://Samba.SerNet.DE>



Questions/comments?

Volker Lendecke, VL@SerNet.DE

SerNet - Service Network GmbH
Bahnhofsallee 1b
37081 Göttingen

Tel: +49 551 370000 0

Fax: +49 551 370000 9

<http://www.SerNet.DE>

<http://Samba.SerNet.DE>

