

# Porting LTTng to Android for Kernel- space and User-space tracing

Charles Brière <c.briere@samsung.com>

Samsung Research America

# Why ?

- Unified tracing within
  - Kernel
  - Native
  - Java
- Consuming traces through network instead of storing locally on device

# What is different on Android

- Build system
  - Android.mk
- Shared memory
  - Android have ashmem (Anonymous SH MEMory)
- Pthread
  - Included within Bionic (Android's Libc)
  - Not full implementation

# Build system

- Android.mk works within
  - Android NDK (Native Development Kit)
  - Android source tree
- But setting up manually environment to build with autotools
  - Can use Androgenizer to generate Android.mk afterward
- NDK doesn't expose as much as complete source tree
- Building LLTng within AOSP

# Missing pthread functions

- pthread\_cancel
  - Used to kill consumer thread if it failed to initialize
  - Otherwise thread will be stopped with pipe
  - Now using pthread\_kill
  - Bad idea ? Not too bad
    - No cleanup methods anyway
- pthread\_cond\_timedwait
  - Every pthread\_condattr\_setclock uses MONOTONIC
  - Use pthread\_cond\_timedwait\_monotonic\_np

# Shared memory

- LTTng unlinks shm path
- Shares most memory regions by passing file descriptor through UNIX socket
- Makes using Android's ashmem instead of SYSV's shm a simple substitution

# Shared memory : Futex

- Daemon notification
  - Application can be started before daemon
  - Unix socket or pipes
  - Ashmem not suited as anonymous
  - Sharing Futex through file instead of shared memory
    - `shm_open` → `open`

# Missing definitions

- HOST\_NAME\_MAX <limits.h>
  - 64
- NAME\_MAX <limits.h>
  - 255
- PATH\_MAX <limits.h>
  - 4096
- SPLICE\_\* <fcntl.h>
  - Even if \_\_NR\_splice exists
- in\_port\_t <netinet/in.h>
  - uint16\_t



# Missing functions with equivalents

- `posix_fadvise`
  - `__NR_arm_fadvise64_64()`
- `splice`
  - `__NR_splice()`
- `bswap_{16,32,64}`
  - `__bswap_{16,32,64}`

# Missing functions w/o equivalents

- pthread\_cancel
  - pthread\_kill ( with SIGKILL )
- shm\_open
  - ashmem\_create – ftruncate
- pthread\_cond\_timedwait
  - pthread\_cond\_timedwait\_monotonic\_np
- getpwuid\_r
  - Set home to /sdcard
- sigwaitinfo
  - sigtimedwait ( arbitrary 10000 seconds timeout)

# Includes

- signal.h → sys/signal.h
- wait.h → sys/wait.h

# What's next

- Make patches upstreamable and submit
- Add JAVA tracing capabilities
- Integrate in AOSP source tree

# Questions ?