Reverse, Multi-Process and Non-Stop Debugging come to the CDT

Or: *How I Learned to Stop Worrying and Love GDB (and DSF)*

Marc Khouzam, François Chouinard

Agenda

- Ericsson: What are we doing here?
- DSF Overview
  - Demo 1
    - What's New
    - Multi-threading
    - Multi-process
- Upcoming Debugging Features
  - Demo 2
    - Reverse Debugging
- Questions

Ericsson Context

- Not a tool vendor!
- 5 main proprietary platforms (sort of… ☺)
  - Some with proprietary OSes
- Each PF requires its own development environment
  - Heavy use of tools in all phases of SW development
  - Standard, everyday tools (e.g. editors, compilers, CM, …)
  - Custom tools for “advanced” needs

- Eclipse: our tool integration platform of choice
- Push for an Eclipse-based IDE for each platform
  - Open source solutions when available
  - Vendor or in-house solutions for the rest

- Everything was cool…

*Then the sh*t hit the fan…*

- ‘Difficulties’ at integrating a debugger for one of our proprietary platforms
- In-house CDI-GDB based solution (incomplete)
  - Worked for the emulator, not the real target
  - Support for freeze-mode only (a.k.a. all-stop)
  - Lacking support for:
    - Multiple targets
    - Multiple processes
    - Non-stop multi-threading
  - Awkward support for:
    - Execution model
    - Data model
- Internal study concluded that:
  - GDB could be enhanced to bridge the gap
  - CDI-based solution would eventually be problematic
  - DD/DSF offered a better alternative

GDB Improvements

- Support for non-stop multi-threading
  - Run mode selection (all- or non-stop)
  - Run control - select/suspend/resume/… individual threads or groups of threads
- Support for multi-process
  - Direct support for multiple processes
  - Handling of execution contexts and aggregation in GDB
  - Global breakpoints
  - Auto-attach
- Some proprietary extensions for our specific needs (not contributed back)

CDI-GDB Challenges

- No direct support for proprietary platform features
  - Non-standard execution and data models
  - Multiple targets
  - Multiple processes
  - Non-stop multi-threading
- Overall performance issue
  - Synchronous debugger communication
  - Slow data retrieval (for large data structures)
  - Stepping-in for large projects
  - Scalability ~
- CDI could have been extended, but:
  - Difficult to integrate our changes in the code base without breaking compatibility ☹
  - Would likely have had to maintain our patches outside of CDI ☹

DSF Features

- Aimed at embedded systems ☺
  - Support for non-standard execution/data models
  - Asynchronous *everything*
  - Command caching and coalescing (in theory…)
  - Fetch only what is required by the UI
  - Scalability !
- Built-in support for concurrency ☺
  - Asynchronous interfaces
  - Threading model
    - Thread pool of size 1 (*de facto* global semaphore)
    - Low overhead

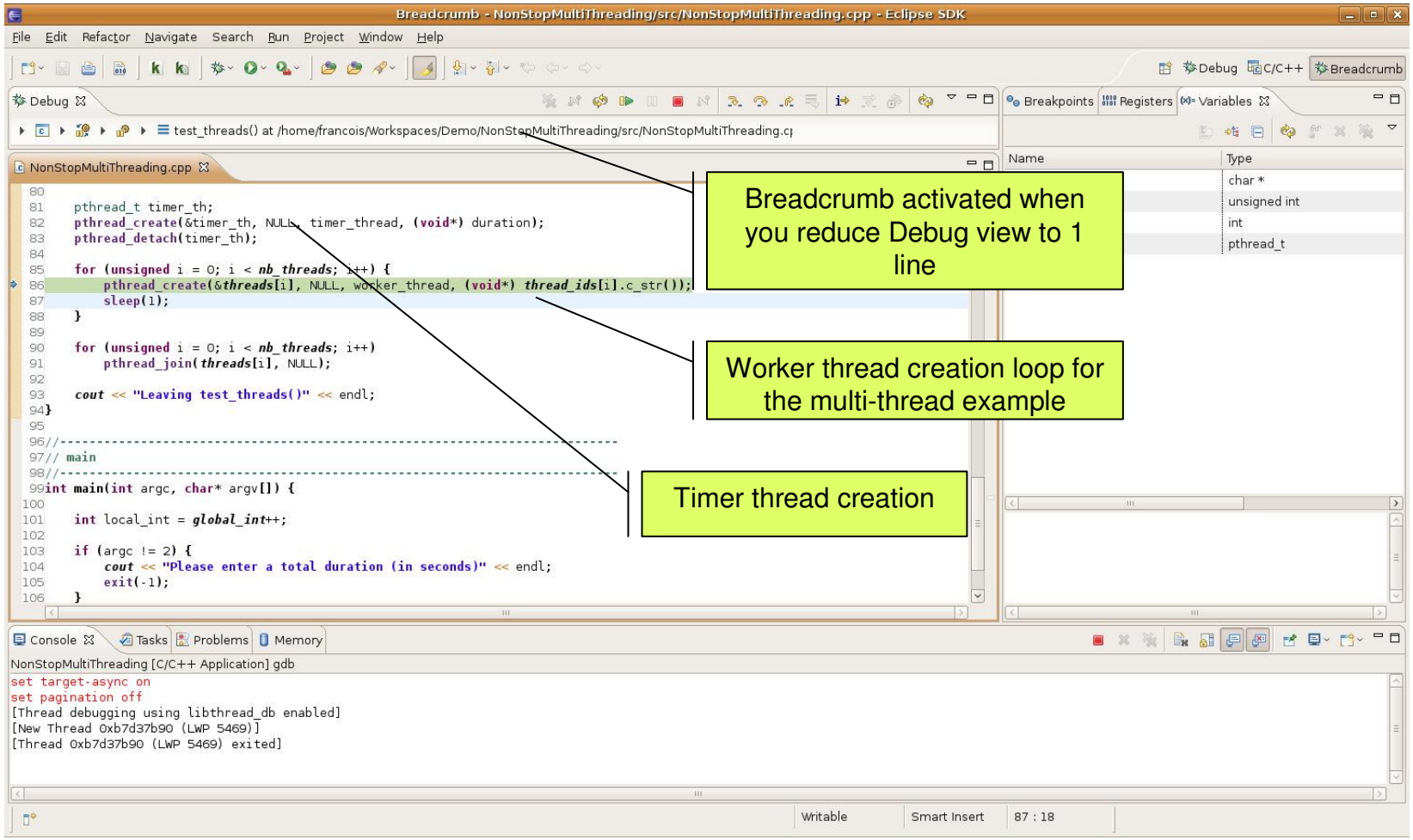
DSF Features

- Service oriented architecture ☺
  - OSGi-based
  - Can easily add/extend/substitute services
    - Run Control
    - Breakpoints
    - Variables, Registers, …
    - MyExcellentService
    - …
- Other nice UI features:
  - Update policies
  - Fast stepping
  - New memory rendering (ironically called “Traditional” ☺)
  - New disassembly view
  - Stack Frame partial display

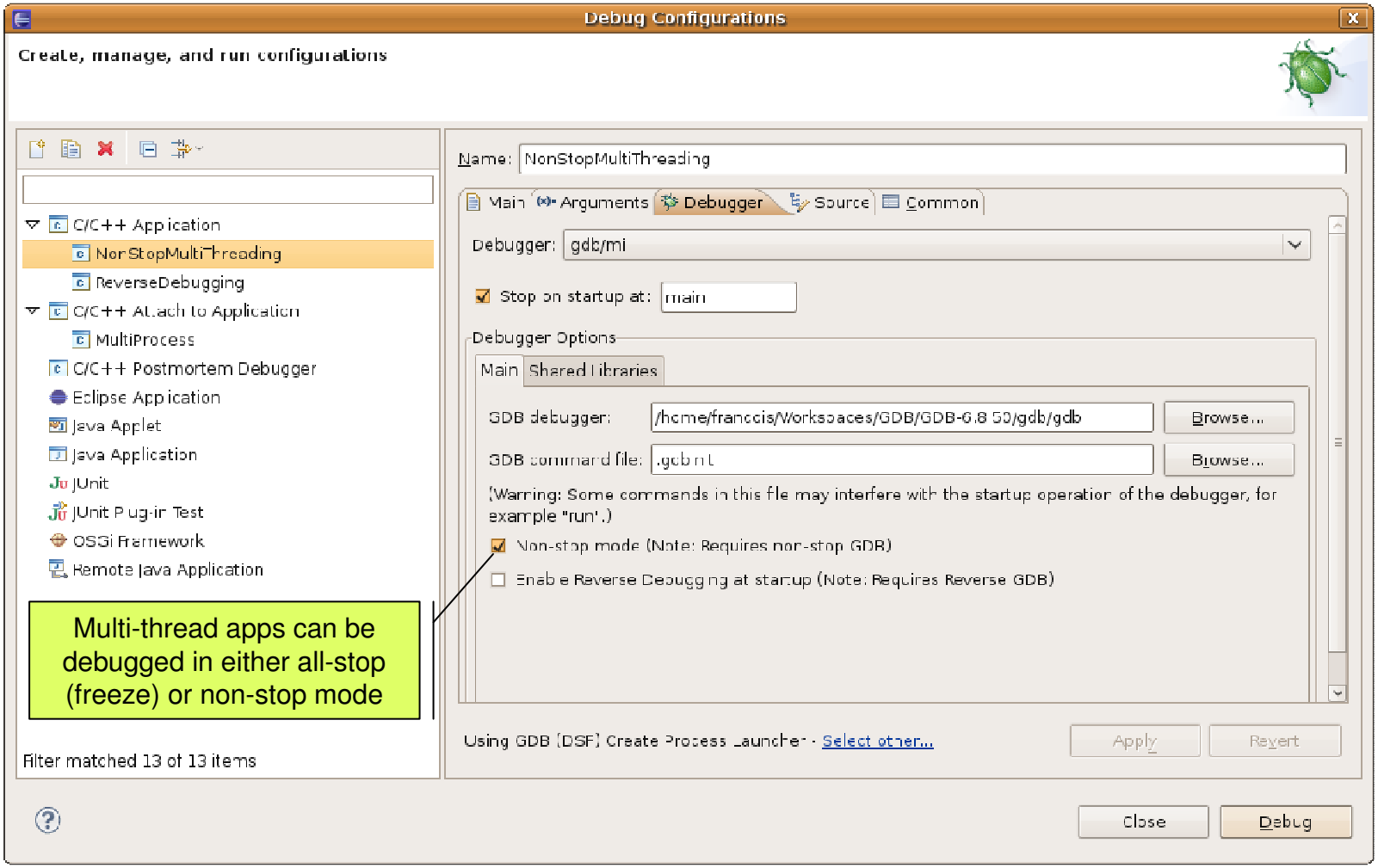
Demo 1

- What's New
  - Launch
  - Views
  - GDB traces
  - As-needed back-end requests
  - Caching (memory)
  - Breadcrumb
- Non-stop multi-threading
  - Thread selection/control
- Multi-process
  - Connecting to running processes

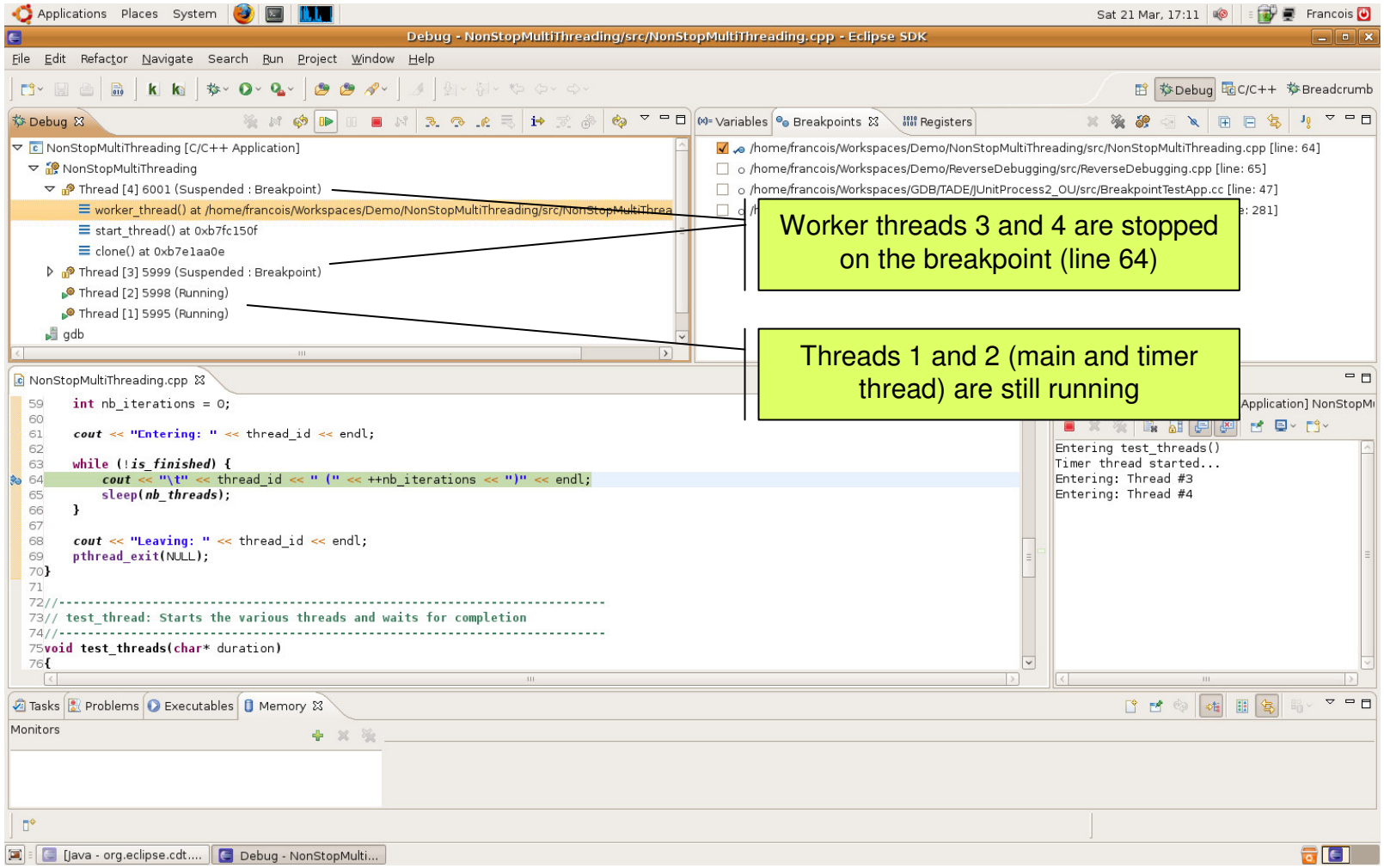
Debug Perspective



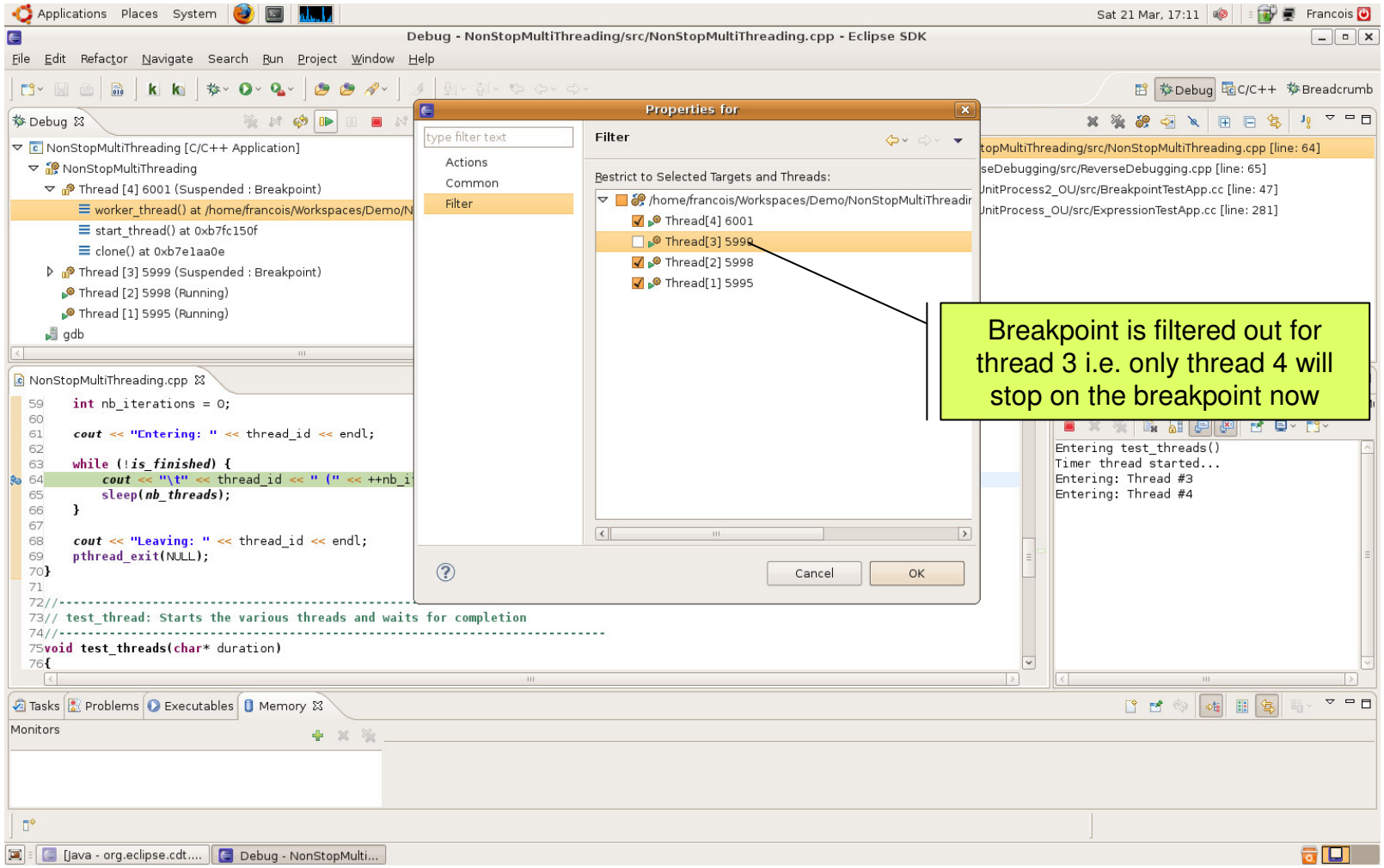
Non-Stop Multi-Threading



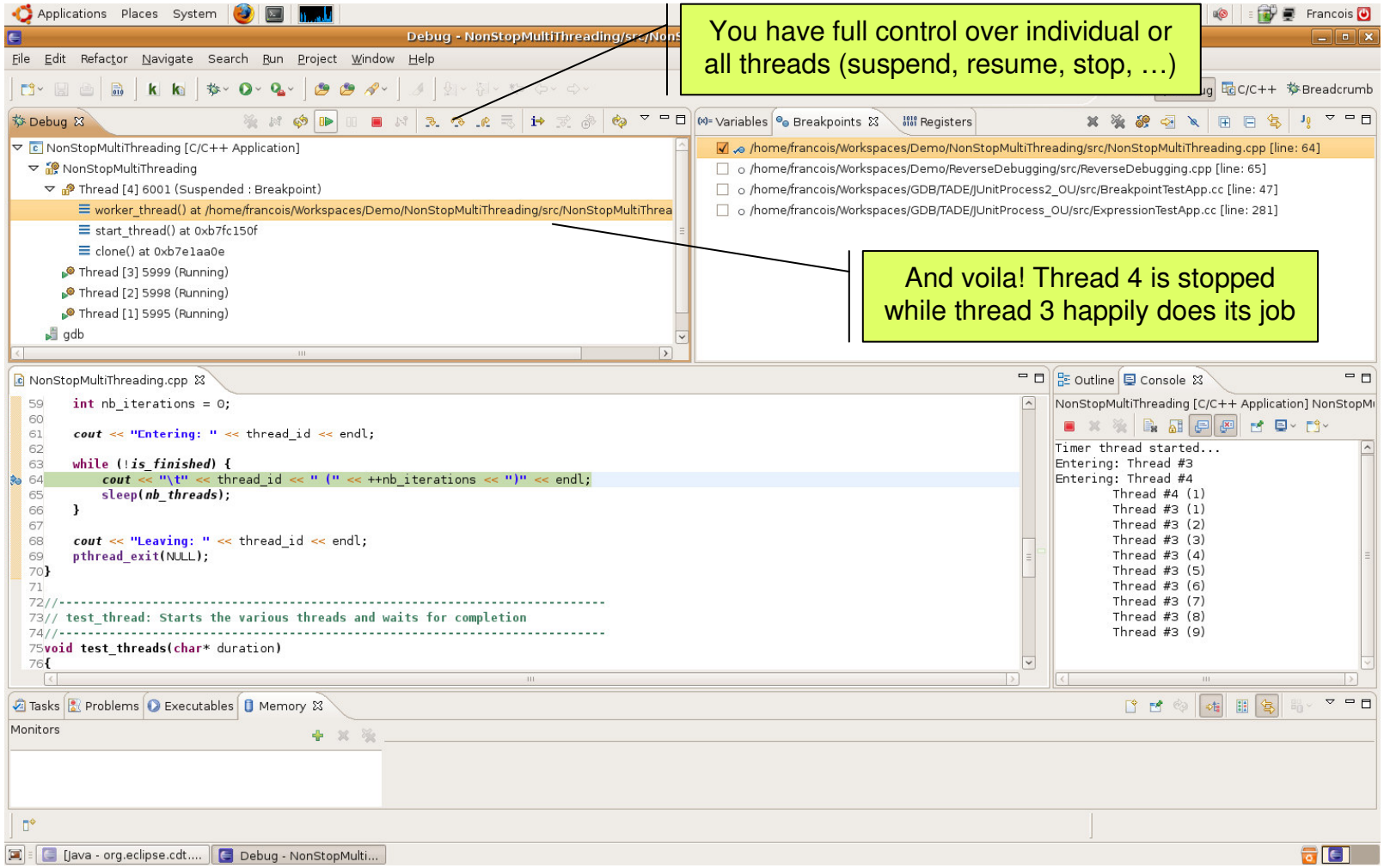
Non-Stop Multi-Threading



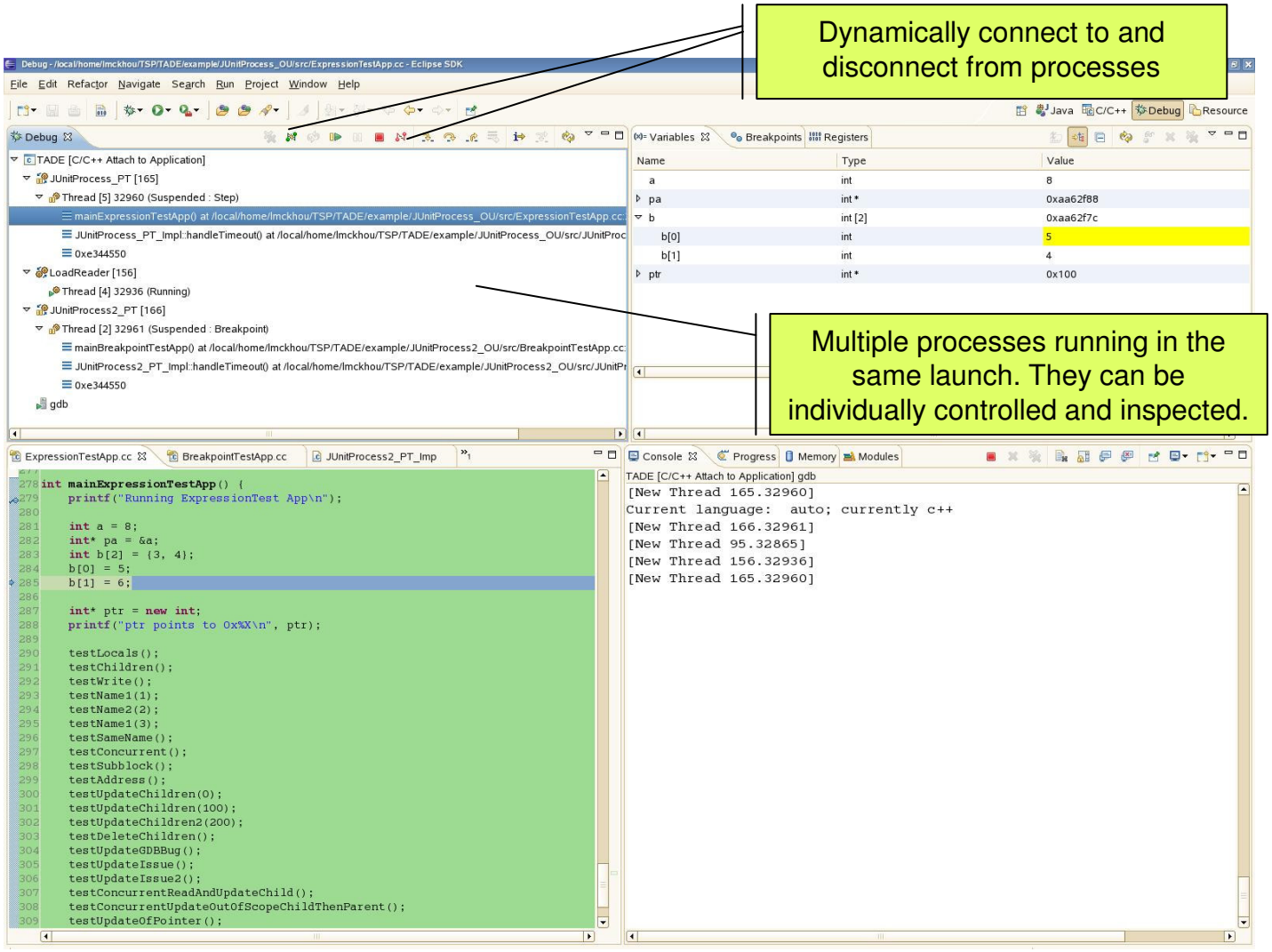
Non-Stop Multi-Threading



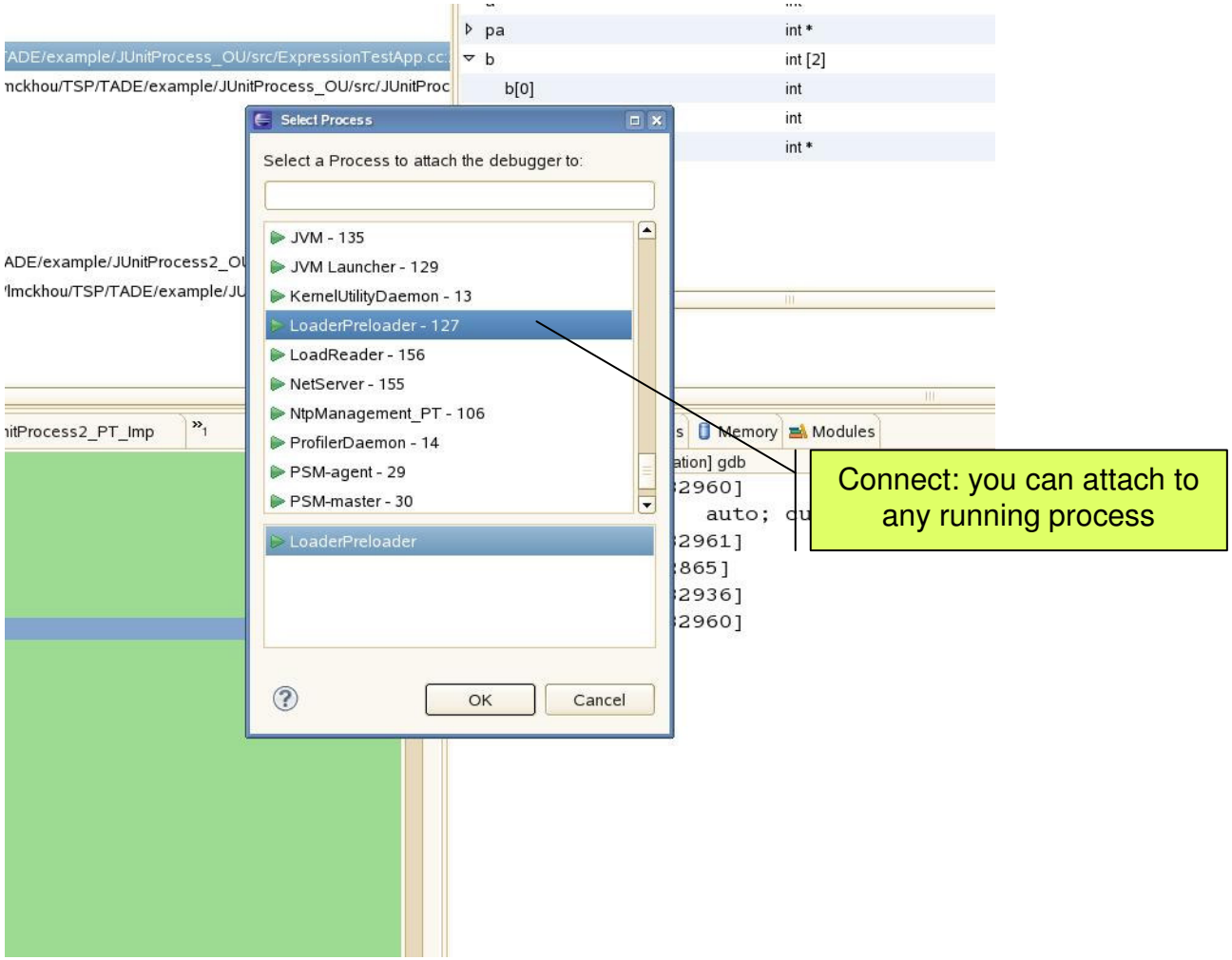
Non-Stop Multi-Threading



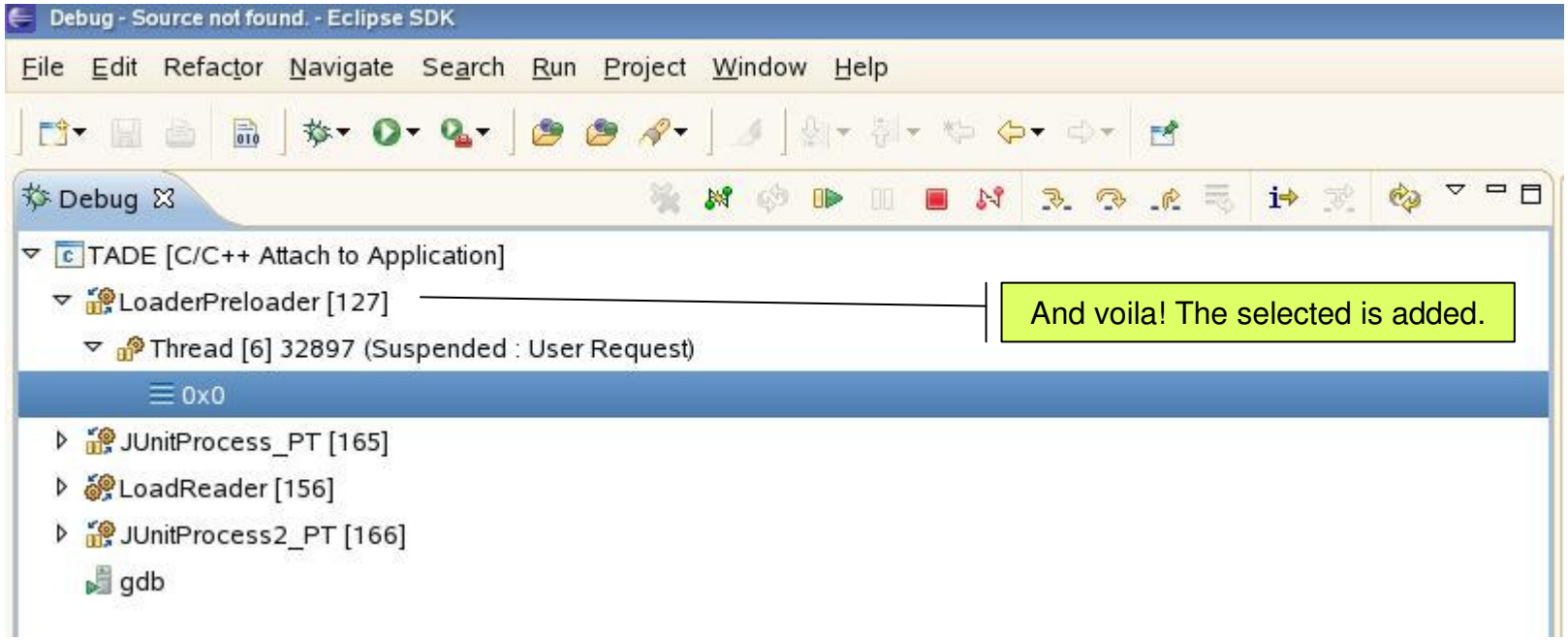
Multi-Process



Multi-Process



Multi-Process



Upcoming Debugging Features

- Reverse Debugging
- Tracepoints
- Multi-exec
- True GDB Server

Reverse Debugging

- Allows to step program backwards and forwards
  - Avoids repeating test over and over to gradually hone in on bug
  - Allows to replay a bug
  - Allows to go back and change program execution without having to recompile and repeat the test (some targets)
- Often available in simulators/emulators
  - VMWare
  - Simics

Reverse Debugging

- IDE front-end support now part of the CDT
  - Buttons, menus and key bindings for
    - Reverse Resume
    - Reverse StepIn
    - Reverse StepOver
    - Uncall
  - Perspective customization to show/hide these new UI features
- Availability for extension
  - Currently in DSF-GDB
  - Will eventually migrate to DSF
  - May even go to Debug Platform

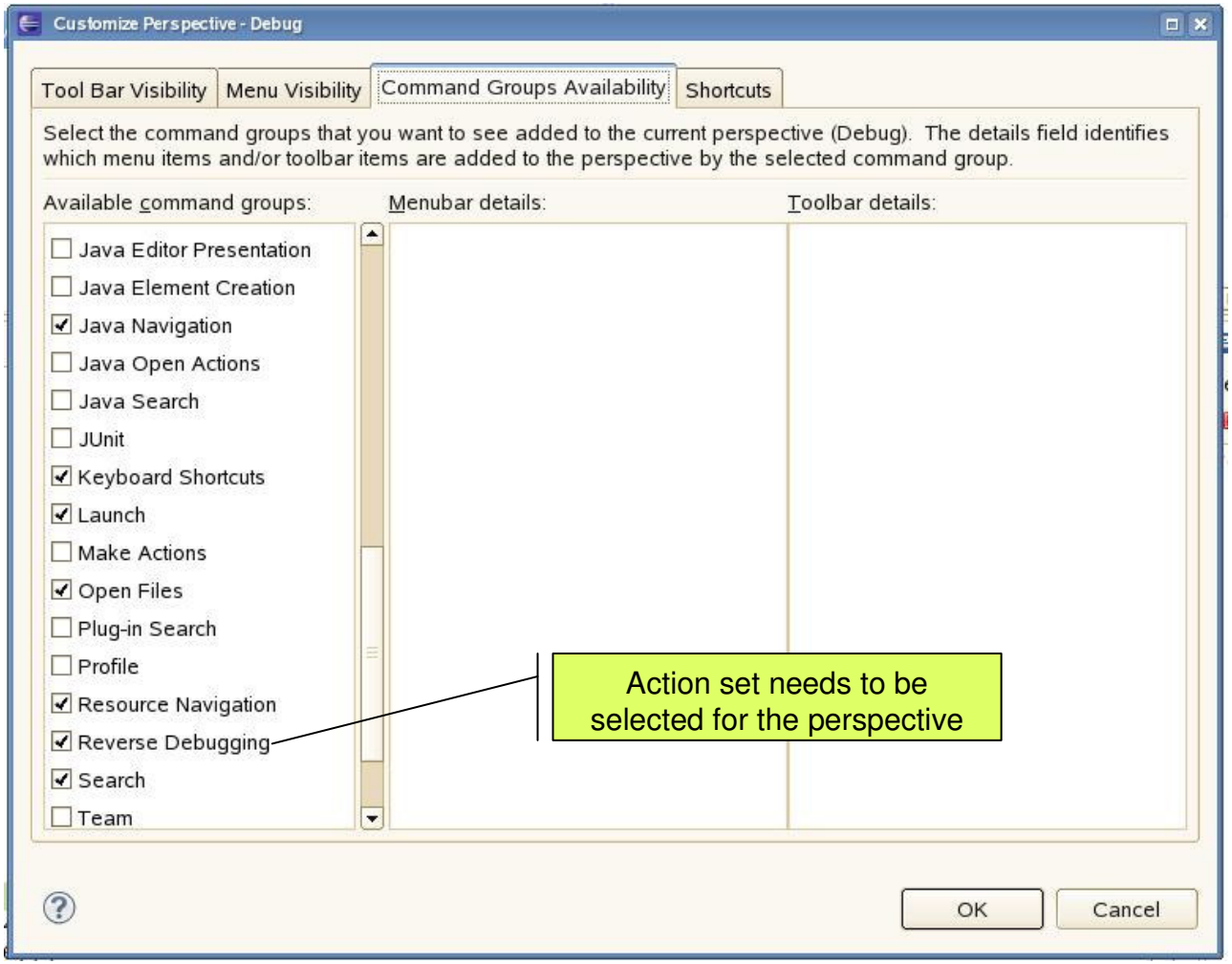
GDB and Reverse Debugging

- Reverse Debugging infrastructure in GDB HEAD
  - Allows to hook to target that support reverse
- Linux PRecord (Process Record and Replay) in GDB HEAD
  - By Hui Zhu (Teawater)
  - For Linux target
  - Records memory and register changes
  - 3 minor fixes are still awaiting approval

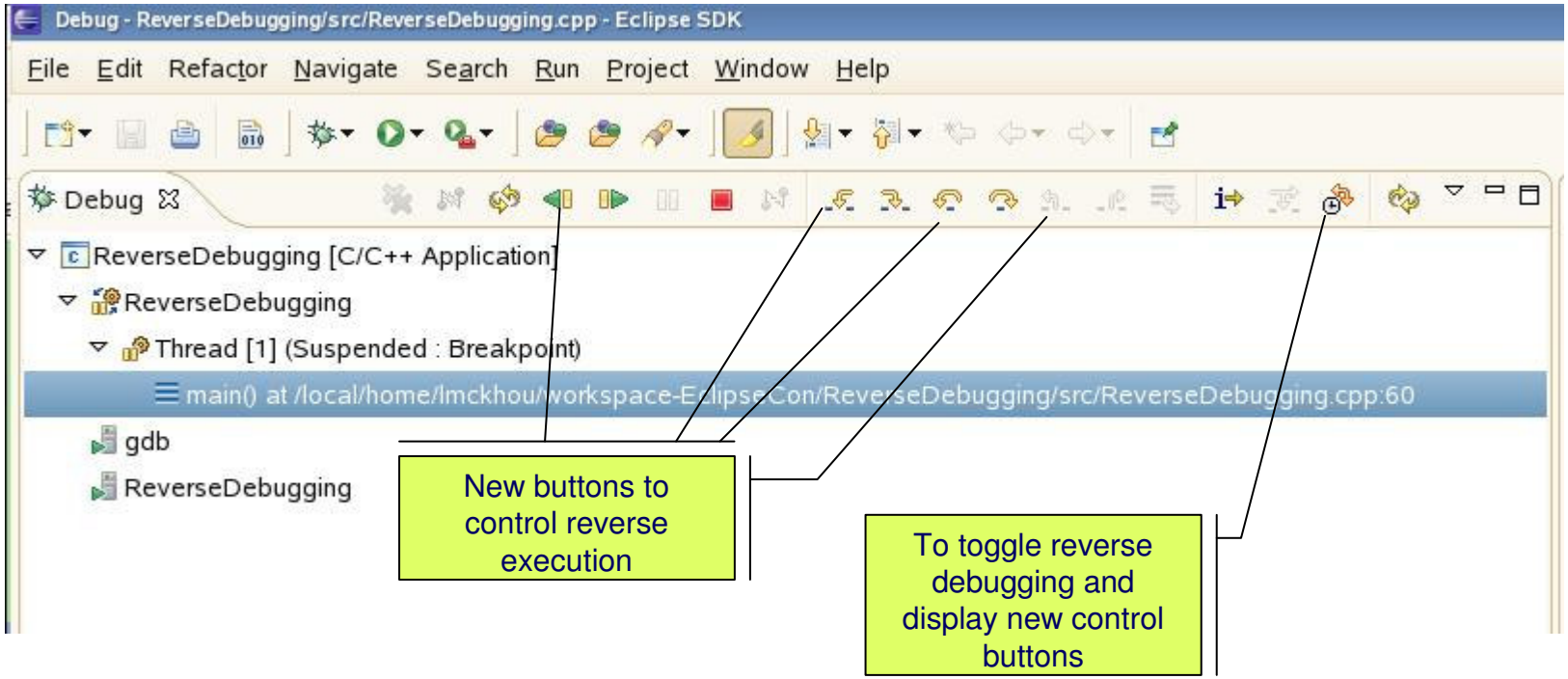
Demo 2

- Reverse Debugging
  - New reverse debugging action set
  - Reverse step-in, reverse step-over, resume, uncall
  - Buttons, key bindings and menus
  - Launch option
  - Views in reverse
  - Change execution path

Reverse Debugging



Reverse Debugging



Tracepoints

- Dynamic tracepoints
  - Add instrumentation to running code
  - Low-overhead
  - Enable/Disable dynamically
  - Trigger on user-defined condition
  - Off-line tracing
  - Trace collection from target

Multi-Exec

- In CDT 6.0
  - Multi-process support in DSF
  - Attaching to multiple processes in DSF-GDB
- Next steps:
  - DSF-GDB support for launching multiple processes in the same debug session
  - GDB support for Multi-process on Linux (should be part of the next release of GDB)

True GDB Server

- GDB provides gdbserver
  - Allows to debug remote program on Linux
  - Can be used as a basis to write a new debug server for your own OS
  - Accepts a single GDB connection
  - Usually started manually before beginning debugging session
- True gdbserver
  - Daemon
  - Accepts multiple GDB connections
  - Ready for debugging at any time

Additional Information

- Contacts
  - Marc Khouzam, marc.khouzam@ericsson.com
  - François Chouinard, francois.chouinard@ericsson.com

Questions?