



Nemiver: Un Débogueur graphique intégré à GNOME

Dodji Seketeli
dodji@redhat.com

Plan de la présentation

- Présentation générale
- Historique
- Fonctionnalités
- Architecture
- Perspectives futures
- Questions

Présentation générale

- Débogueur pour programmes C/C++
- Volontaires sur leur temps libre
- Communauté GNOME
- Simple d'utilisation
 - S'intègre à GNOME
- Cas d'utilisations simples & courants
- Ressources requises raisonnables

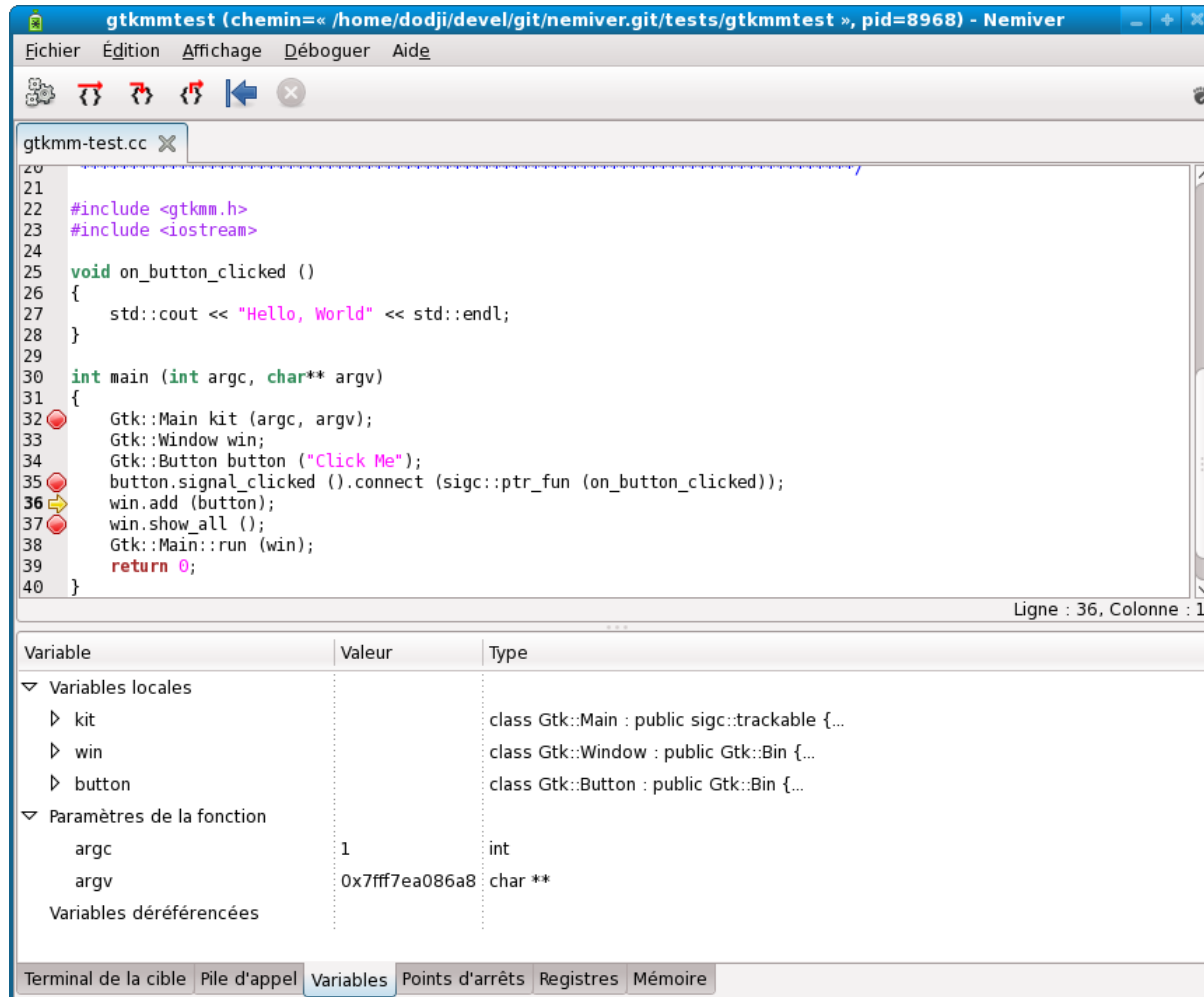
Historique

- En 2006 conscience du manque
 - Sur IRC, questions sans fin sur GDB
 - Ex Collègues râlant contre GDB
 - DDD pas beau, pas intuitif
 - Abus de langages interprétés
- Été 2006, début du code
- Histoire en cours ...

Functionalités

- Points d'arrêts
- Inspection de variables
- Inspection de pile d'appel
- Inspection de la mémoire
- Sauvegarde du contexte
- Intégration dans GNOME

Une image de la bête



The screenshot shows a debugger window titled "gtkmmtest (chemin=< /home/dodji/devel/git/nemiver.git/tests/gtkmmtest », pid=8968) - Nemiver". The main window displays the source code for "gtkmm-test.cc". The code includes headers for gtkmm and iostream, defines a callback function "on_button_clicked", and a "main" function that creates a window with a button labeled "Click Me". The debugger is currently paused at line 36, which is "win.add (button);".

Below the code editor, the "Variables" window is open, showing the following table:

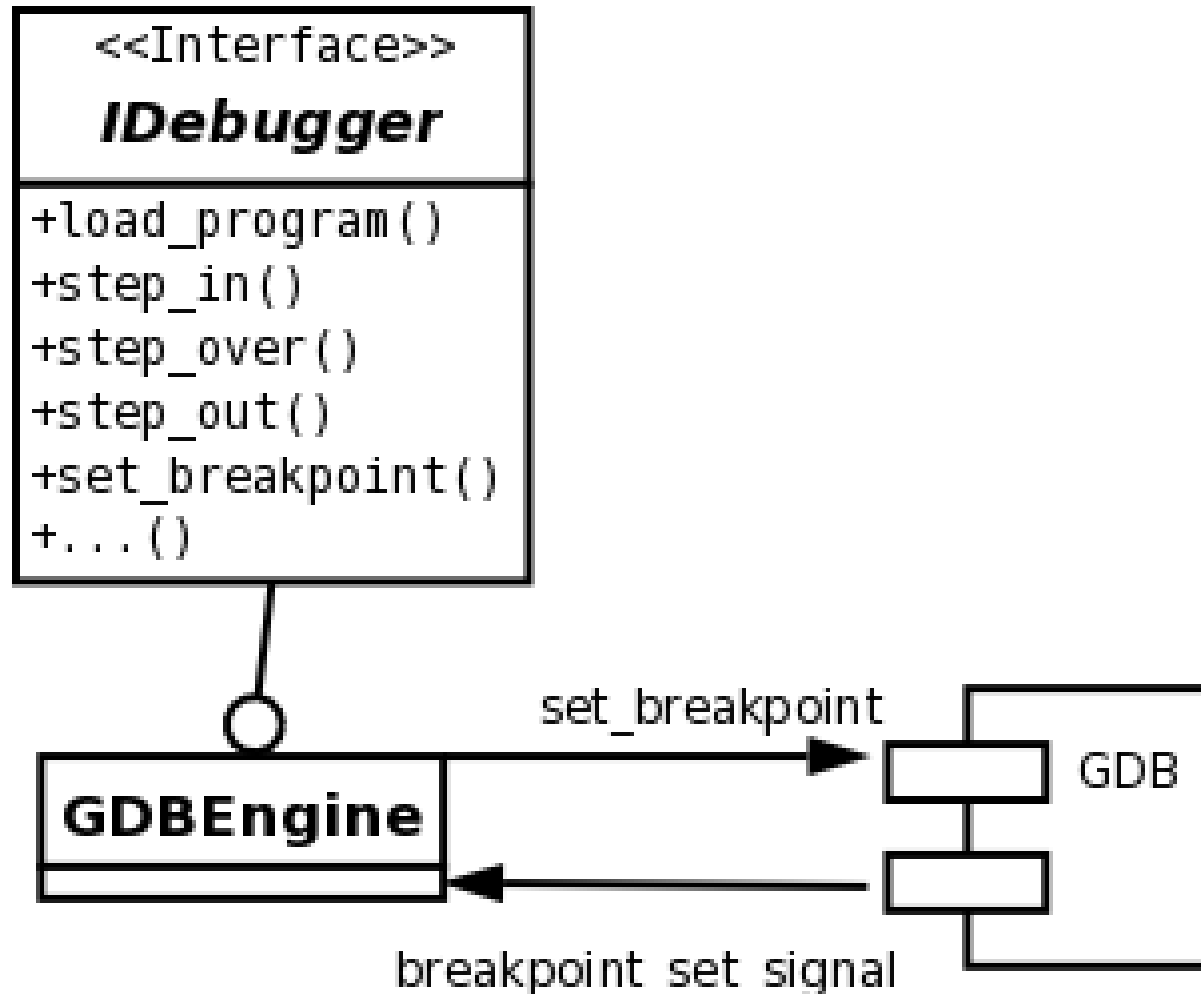
Variable	Valeur	Type
Variables locales		
kit		class Gtk::Main : public sigc::trackable {...
win		class Gtk::Window : public Gtk::Bin {...
button		class Gtk::Button : public Gtk::Bin {...
Paramètres de la fonction		
argc	1	int
argv	0x7fff7ea086a8	char **
Variables déréférencées		

At the bottom of the debugger window, there are tabs for "Terminal de la cible", "Pile d'appel", "Variables", "Points d'arrêts", "Registres", and "Mémoire". The "Variables" tab is currently selected.

Architecture (I): Maître GDB sur un arbre perché ...

- Nemiver utilise GDB
 - Mais, comment piloter GDB ?
 - Pas de librairie disponible
 - GDB/MI
 - Nouvelle Interface: IDebugger
 - Modèle événementiel
 - Implémentation: Fork/Exec/Pipes
 - Boucle d'événements Glib

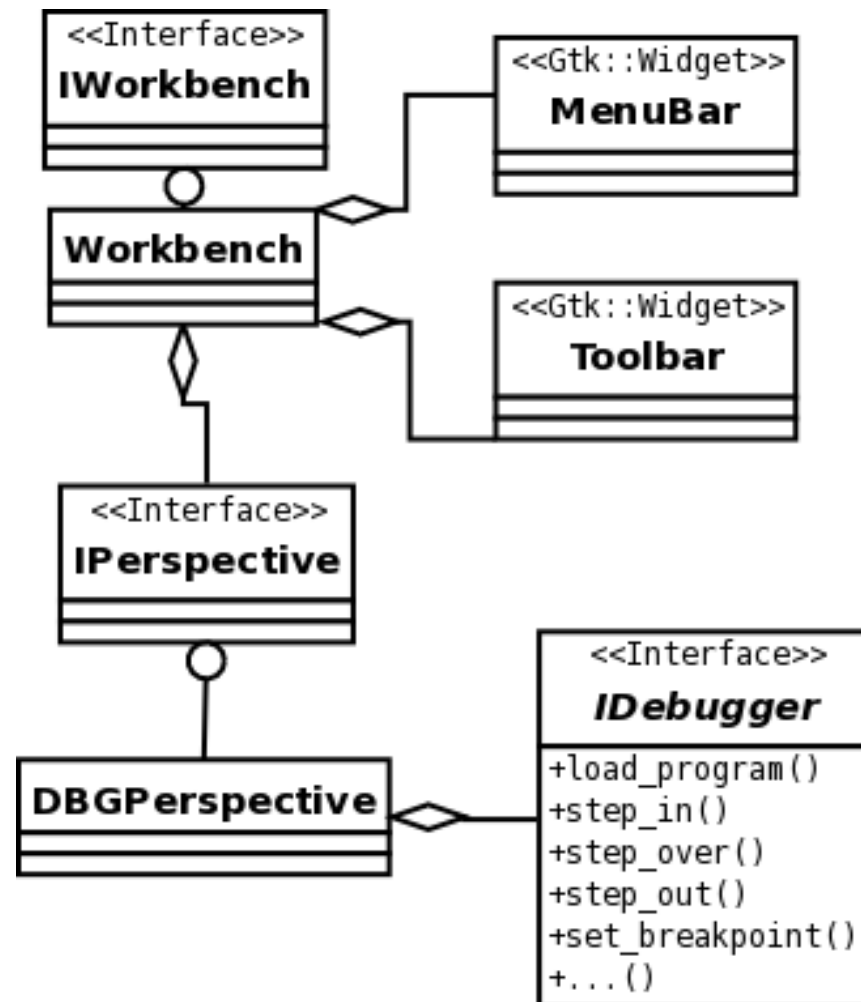
Architecture (I): version courte



Architecture (II): Et ce que l'on voit ?

- Séparation interfaces/implémentations
- IWorkbench
 - interface graphique générale
 - barre de menu et d'outil
 - perspective
- IPerspective
 - Ensemble des widgets de débogage
 - Utilise IDebugger

Architecture (II): diagramme général



Perspectives

- Couvrir plus de cas d'utilisation simples
- Scripting (Javascript ou Lua)
 - Cas d'utilisations complexes
 - Interface ligne de commande intégrée
- Accès direct aux binaires ELF
 - Problèmes de portabilité ?
 - Puissance accrue
- Librairie externe de débogage ?



Questions?

dodji@redhat.com