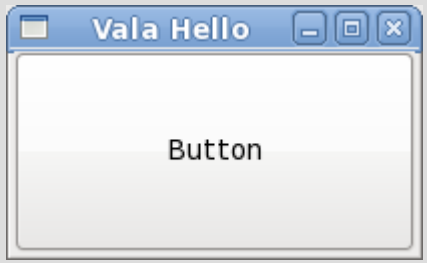
Momonga Linux

GUIプログラミング入門（4）
～2009忘年会議～

Momonga Project
西尾　太

今回は（も？）C#

- Linuxで使えるGUI開発用C#環境
  - Mono
    - http://www.mono-project.com/Main_Page
  - Vala
    - http://live.gnome.org/Vala
- 今回も簡単なアプリ



Mono

```
using Gtk;
using System;

class MainClass {
	public static void Main (string[] args) {
		Gtk.Application.Init ();

		Gtk.Window w = new Window (Gtk.WindowType.Toplevel);
		w.Title = "Mono Hello";
		w.SetDefaultSize (200, 100);
		Gtk.Button b = new Button ("Button");

		w.DeleteEvent += new DeleteEventHandler (Window_Delete);
		b.Clicked += new EventHandler (Button_Clicked);

		w.Add (b);
		w.ShowAll ();

		Gtk.Application.Run ();
	}

	static void Window_Delete (object o, DeleteEventArgs args) {
		Gtk.Application.Quit ();
		args.RetVal = true;
	}

	static void Button_Clicked (object o, EventArgs args) {
		System.Console.Write ("Hello, Mono World!\n");
	}
}
```

Vala

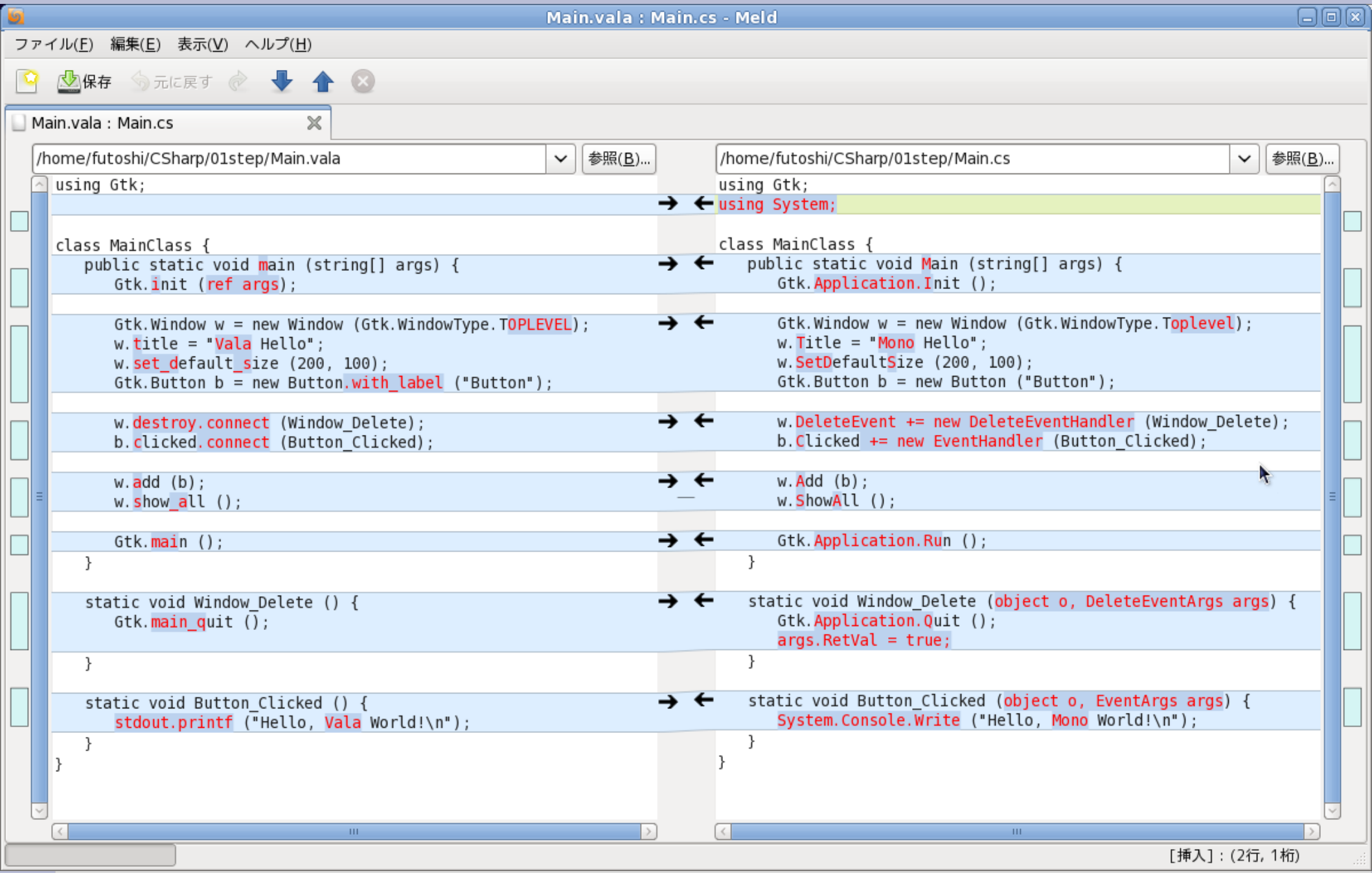
```
using Gtk;


class MainClass {
	public static void main (string[] args) {
		Gtk.init (ref args);

		Gtk.Window w = new Window (Gtk.WindowType.TOPLEVEL);
		w.title = "Vala Hello";
		w.set_default_size (200, 100);
		Gtk.Button b = new Button.with_label ("Button");

		w.destroy.connect (Window_Delete);
		b.clicked.connect (Button_Clicked);

		w.add (b);
		w.show_all ();

		Gtk.main ();
	}

	static void Window_Delete () {
		Gtk.main_quit ();

	}

	static void Button_Clicked () {
		stdout.printf ("Hello, Vala World!\n");
	}
}
```

Makefile

- コマンドが違う
  - Mono
    - gmcs
  - Vala
    - valac
- pkg-configの指定
- 他は同じ

```
$ cat Makefile
all : Main.exe Main

Main.exe : Main.cs
    gmcs -pkg:gtk-sharp-2.0 $<

Main : Main.vala
    valac --pkg gtk+-2.0 $<

clean :
    rm Main.exe Main
```

meldを使う


Main.vala : Main.cs - Meld
ファイル(F) 編集(E) 表示(V) ヘルプ(H)
保存
元に戻す
Main.vala : Main.cs
/home/futoshi/CSharp/01step/Main.vala
参照(B)...
/home/futoshi/CSharp/01step/Main.cs
参照(B)...
using Gtk;
using System;
class MainClass {
public static void main (string[] args) {
Gtk.init (ref args);
Gtk.Window w = new Window (Gtk.WindowType.TOPLEVEL);
w.title = "Vala Hello";
w.set_default_size (200, 100);
Gtk.Button b = new Button.with_label ("Button");
w.destroy.connect (Window_Delete);
b.clicked.connect (Button_Clicked);
w.add (b);
w.show_all ();
Gtk.main ();
}
static void Window_Delete () {
Gtk.main_quit ();
}
static void Button_Clicked () {
stdout.printf ("Hello, Vala World!\n");
}
}
class MainClass {
public static void Main (string[] args) {
Gtk.Application.Init ();
Gtk.Window w = new Window (Gtk.WindowType.Toplevel);
w.Title = "Mono Hello";
w.SetDefaultSize (200, 100);
Gtk.Button b = new Button ("Button");
w.DeleteEvent += new DeleteEventHandler (Window_Delete);
b.Clicked += new EventHandler (Button_Clicked);
w.Add (b);
w.ShowAll ();
Gtk.Application.Run ();
}
static void Window_Delete (object o, DeleteEventArgs args) {
Gtk.Application.Quit ();
args.RetVal = true;
}
static void Button_Clicked (object o, EventArgs args) {
System.Console.Write ("Hello, Mono World!\n");
}
}
[挿入] : (2行, 1桁)

MonoもValaも違いは少ない

- いや、違うだろ！！

- Mono
  - MicroSoftのメソッドに似てる
  - イベントに引数が必要？
  - 標準出力は System.Console.Write
- Valaは
  - GTK+のメソッドに似てる
  - イベントに引数は不要
  - 標準出力は stdout.printf

リファクタリング

- STEP 1
  - 宣言時の型指定
    - 推論できる場合はvarで十分
    - Gtk.Window Gtk.Button → var
- STEP 2
  - ラムダ関数（匿名関数）使うべし
    - メソッドの低減
    - () => {} ってカッコいい（カッコが多い）

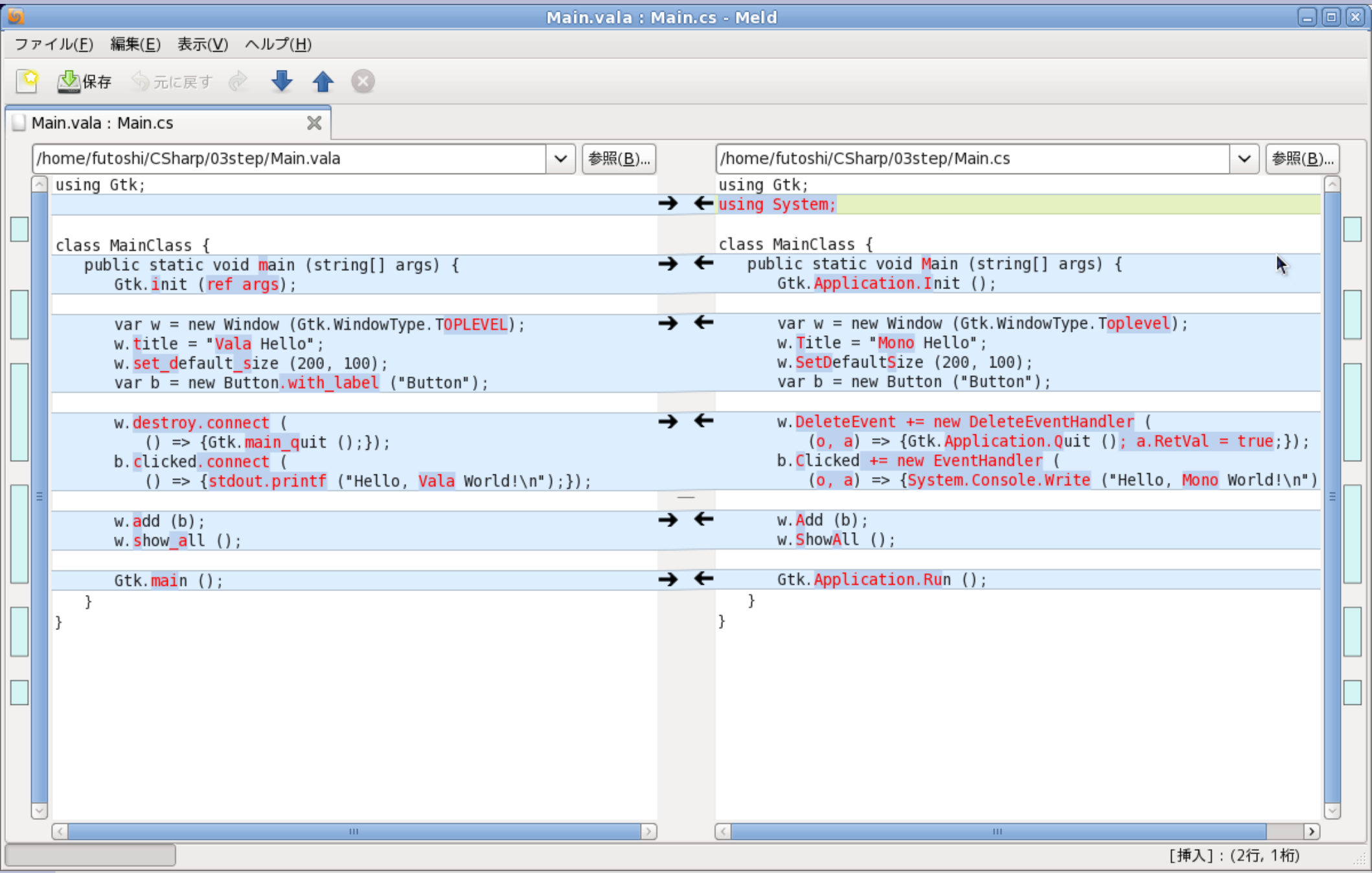
Mono

```
using Gtk;
using System;

class MainClass {
    public static void Main (string[] args) {
        Gtk.Application.Init ();

        var w = new Window (Gtk.WindowType.Toplevel);
        w.Title = "Mono Hello";
        w.SetDefaultSize (200, 100);
        var b = new Button ("Button");

        w.DeleteEvent += new DeleteEventHandler (
            (o, a) => {Gtk.Application.Quit (); a.RetVal = true;});
        b.Clicked += new EventHandler (
            (o, a) => {System.Console.WriteLine ("Hello, Mono World!");});

        w.Add (b);
        w.ShowAll ();

        Gtk.Application.Run ();
    }
}
```

Vala

```
using Gtk;


class MainClass {
    public static void main (string[] args) {
        Gtk.init (ref args);

        var w = new Window (Gtk.WindowType.TOPLEVEL);
        w.title = "Vala Hello";
        w.set_default_size (200, 100);
        var b = new Button.with_label ("Button");

        w.destroy.connect (
            () => {Gtk.main_quit ();});
        b.clicked.connect (
            () => {stdout.printf ("Hello, Vala World!\n");});

        w.add (b);
        w.show_all ();

        Gtk.main ();
    }
}
```

meldを使う（STEP2）



Mono vs Vala

$ stat Main Main.exe
$ cat /proc/<PID>/status | grep Vm

- Mono
  - stat
    - Size: 4096
    - Blocks: 8
    - IO Block: 4096
  - proc
    - VmPeak: 110532 kB
    - VmSize: 110288 kB
    - VmLck: 0 kB
    - VmHWM: 13752 kB
    - VmRSS: 13752 kB
    - VmData: 77852 kB
    - VmStk: 84 kB
    - VmExe: 2308 kB
    - VmLib: 25984 kB
    - VmPTE: 152 kB
- Vala
  - stat
    - Size: 12841
    - Blocks: 32
    - IO Block: 4096
  - proc
    - VmPeak: 22604 kB
    - VmSize: 22600 kB
    - VmLck: 0 kB
    - VmHWM: 8936 kB
    - VmRSS: 8936 kB
    - VmData: 2036 kB
    - VmStk: 84 kB
    - VmExe: 8 kB
    - VmLib: 14756 kB
    - VmPTE: 92 kB

Mono vs Vala

- ファイルサイズ
  - Mono ≦ Vala
  - stripすれば、あまり変わらない？
- メモリ使用量
  - Mono ≧ Vala
  - アセンブリ依存が大きい？

まぁ、もう少し大きなプログラムで試すべし

Mono vs Vala

- Monoの目標
  - .NET互換
- Valaの目標
  - GNOMEアプリをC#で書く

まぁ、Microsoftの仕事もあるなら、
Monoを使った方が良いんだろうな～

Momonga Linux

GUIプログラミング入門（4）
～2009忘年会議～

おわり