







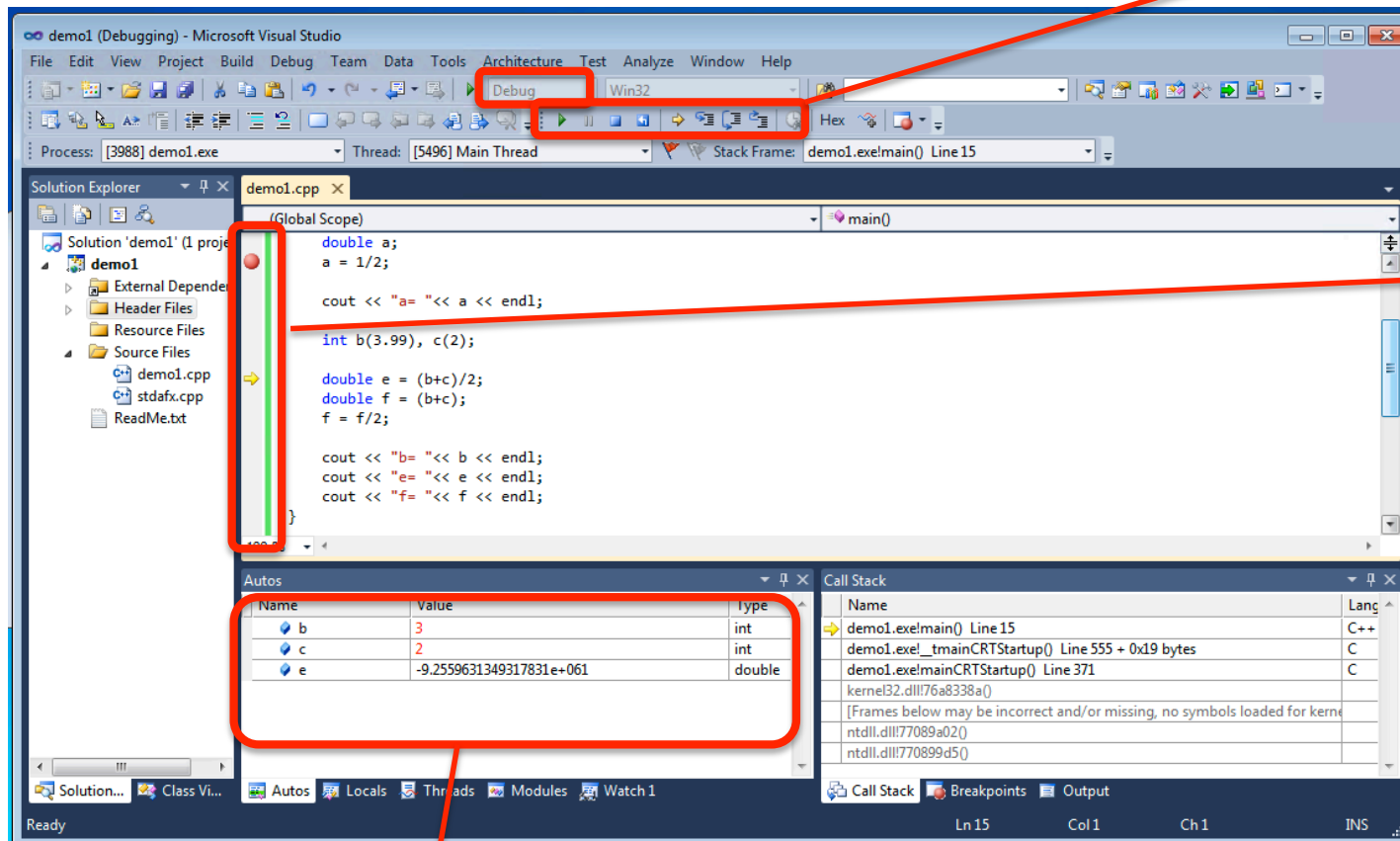
# Debuggers

v. March 2016

# Visual studio

-  Debug
-  Step over
-  Step in
-  Step out

-  Breakpoint
-  Current



Variables

# Xcode

Show/hide



Breakpoints  
Right-click to edit

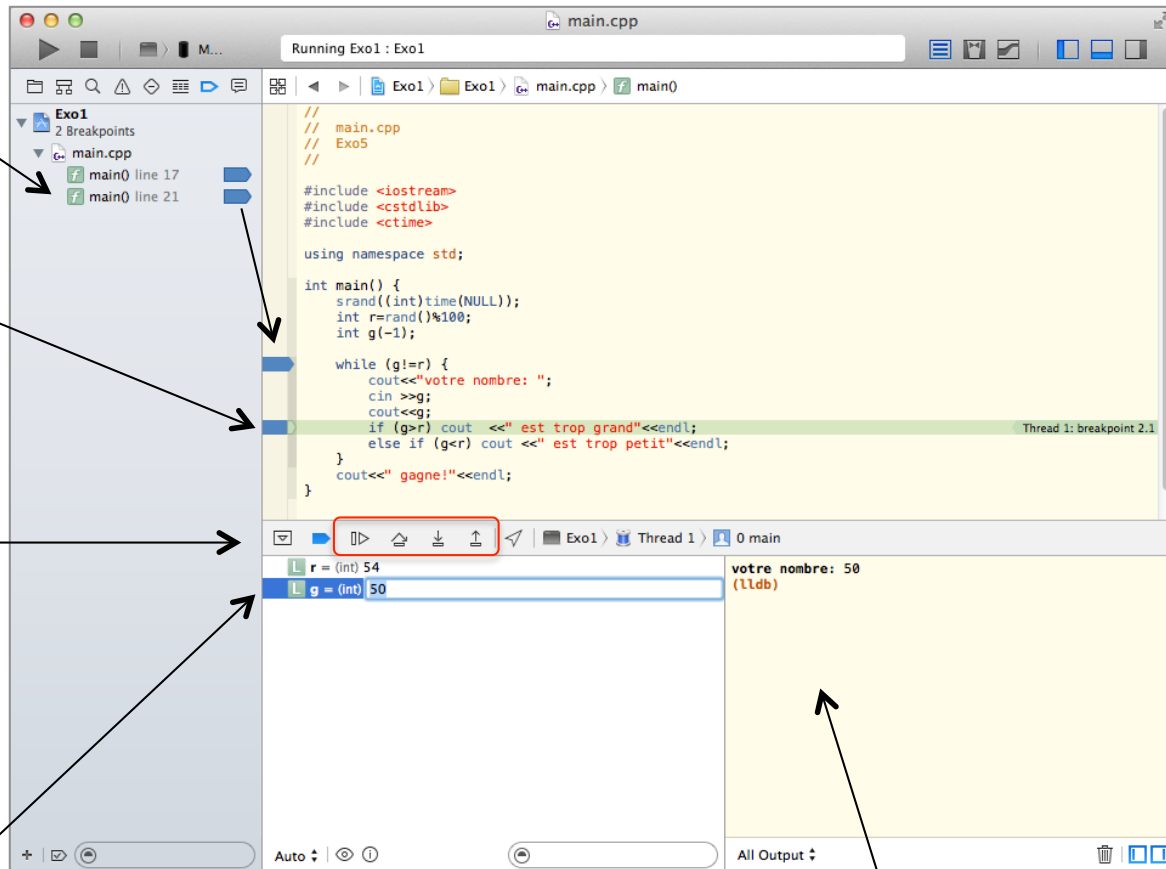
PC, position courante

Navigation

- de/activate breakpoint
- continue until next breakpoint
- next instruction
- jump into the function
- exit current function

Variables  
Right-click to edit

console

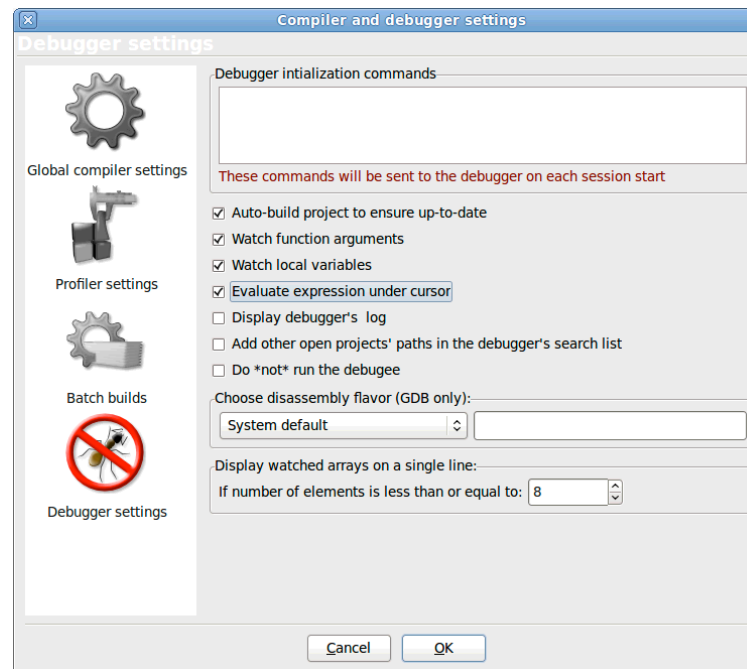


# Code::Blocks

Il est généralement possible d'observer l'exécution d'un programme (code machine) à l'aide d'un débogueur.

Le compilateur peut générer des informations supplémentaires (option `-g` dans `gcc`) afin de synchroniser le code machine et le code source. Il est alors possible de suivre l'exécution du programme pas à pas et d'observer la mise à jour des variables, par exemple avec le débogueur indépendant `ddd` (pas dans ce cours). Nous allons nous concentrer sur le débogueur intégré de Code::Blocks

La première étape est la configuration des paramètres du débogueur de Code::Blocks



# Code::Blocks - debugger

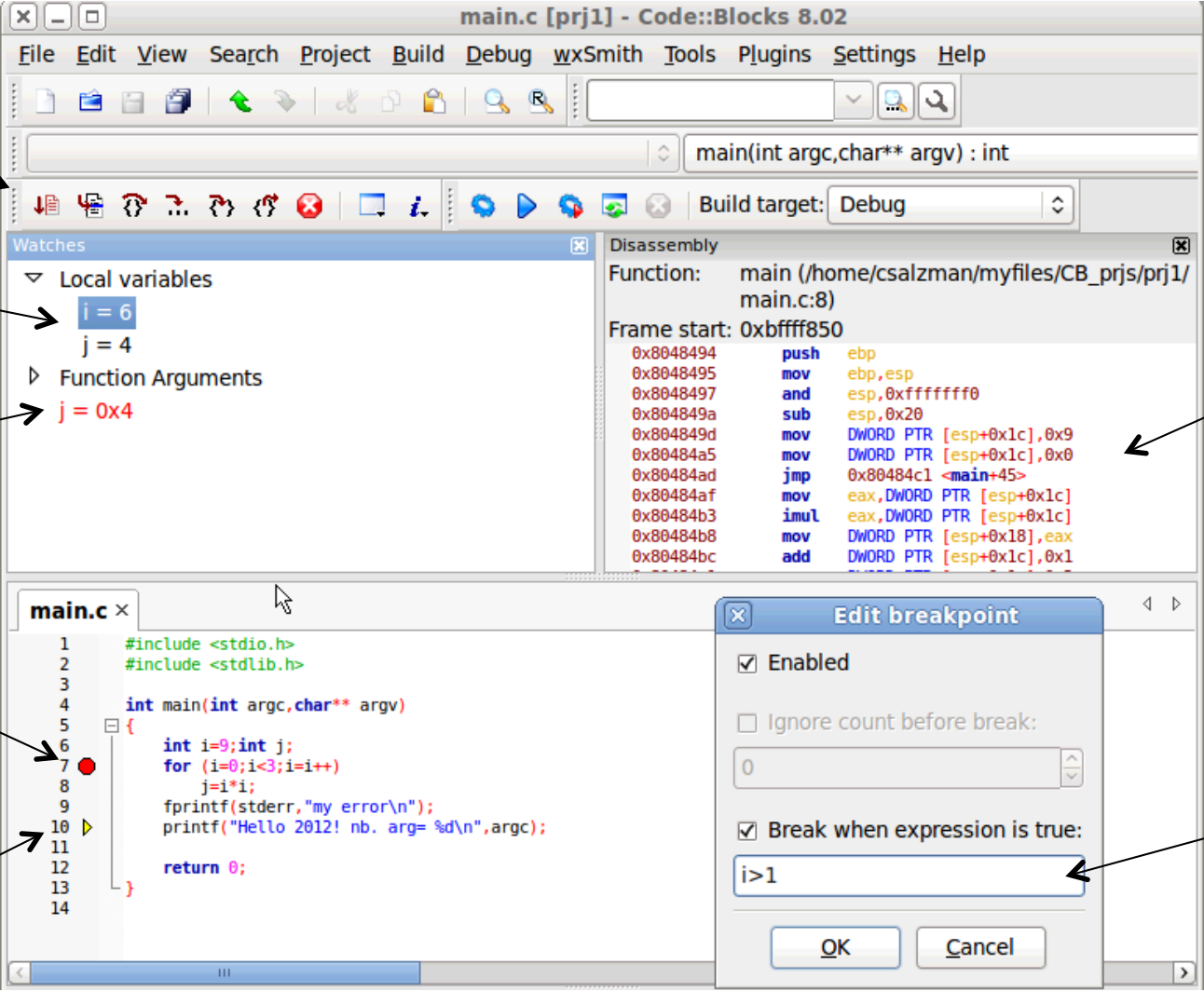
Navigation  
next line, etc

Variables  
possibilité de  
changer sa valeur

Watches

Breakpoint

PC



Code assembleur

Conditional  
breakpoint



# Debugger – démo 1

```
1 #include <iostream>
2 #define DivByZero -1
3
4 using namespace std;
5
6 double myDiv(double a, double b) throw (int) {
7     //if (b!=0.0)
8     return a/b;
9     // else throw DivByZero;
10 }
11
12 int main()
13 {
14     //try{
15     double r;
16     cout << "myDiv(4,2)=" ;
17     r= myDiv(4.0,2.0);
18     cout << r<<endl;
19     cout << "myDiv(3,0)=" ;
20     r= myDiv(3.0,0.0);
21     cout << r<<endl;
22     //}
23     // catch ( int e) {cout << "error ="<<endl;}
24     return 0;
25 }
```

Build target: Debug

Management

Projects Symbols

Workspace

Demo1

Sources

main.cpp

Logs & others

Build log Build messages Debugger

Registered new type: wxString  
Registered new type: STL String  
Registered new type: STL Vector  
Setting breakpoints  
Debugger name and version: GNU gdb (GDB) 7.1-ubuntu  
Program exited normally.  
Debugger finished with status 0

/home/csai default Line 20, Column 1 Insert Read/Write default

ION  
files" seront sauvegardés au logout !  
der will be saved at logout !



# Debugger - démo 2

```
1 #include <iostream>
2 #define DivByZero -1
3
4 using namespace std;
5
6 double myDiv(double a, double b) throw (int) {
7     //if (b!=0.0)
8         return a/b;
9     // else throw DivByZero;
10 }
11 int main()
12 {
13     //try{
14         double r;
15         cout << "myDiv(4,2)=" ;
16         r= myDiv(4.0,2.0);
17         cout << r<<endl;
18         cout << "myDiv(3,0)=" ;
19         r= myDiv(3.0,0.0);
20         cout << r<<endl;
21     //}
22     // catch ( int e) {cout << "error ="<<e<<endl;}
23     return 0;
24 }
25
```

Build log | Build messages | **Debugger**

At /home/csalmz/myfiles/CB\_prjs/Demo1/main.cpp:8  
At /home/csalmz/myfiles/CB\_prjs/Demo1/main.cpp:10  
At /home/csalmz/myfiles/CB\_prjs/Demo1/main.cpp:20  
At /home/csalmz/myfiles/CB\_prjs/Demo1/main.cpp:23  
At /home/csalmz/myfiles/CB\_prjs/Demo1/main.cpp:24  
Program exited normally.  
Debugger finished with status 0

ION  
files" seront sauvegardés au logout !  
der will be saved at logout !

Watches

- Local variables  
No locals.
- Function Arguments  
No arguments.