LLDB

NYC*BUG

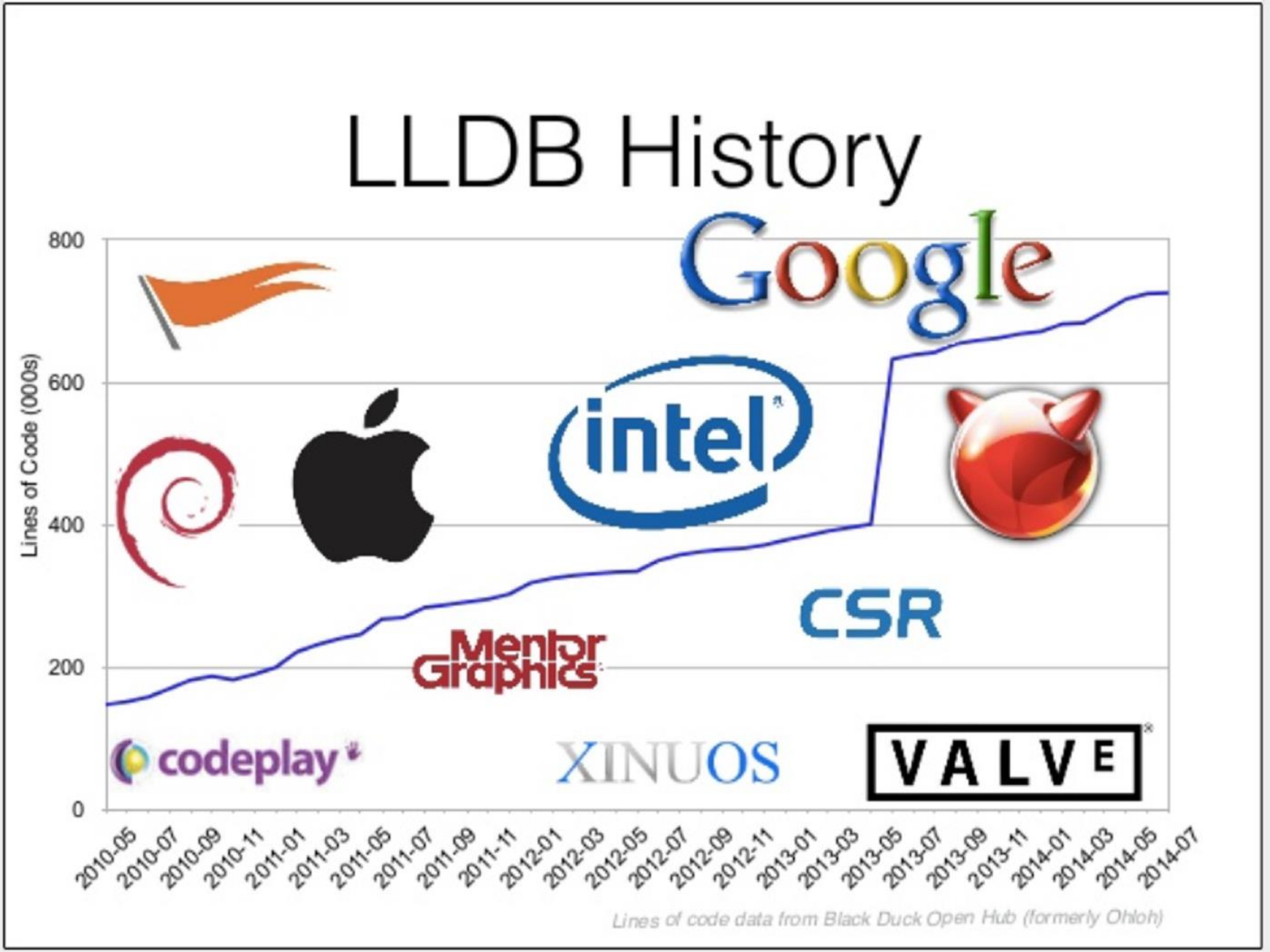
April 6,
2016

John
Wolfe

johnlwolfe@comcast.net

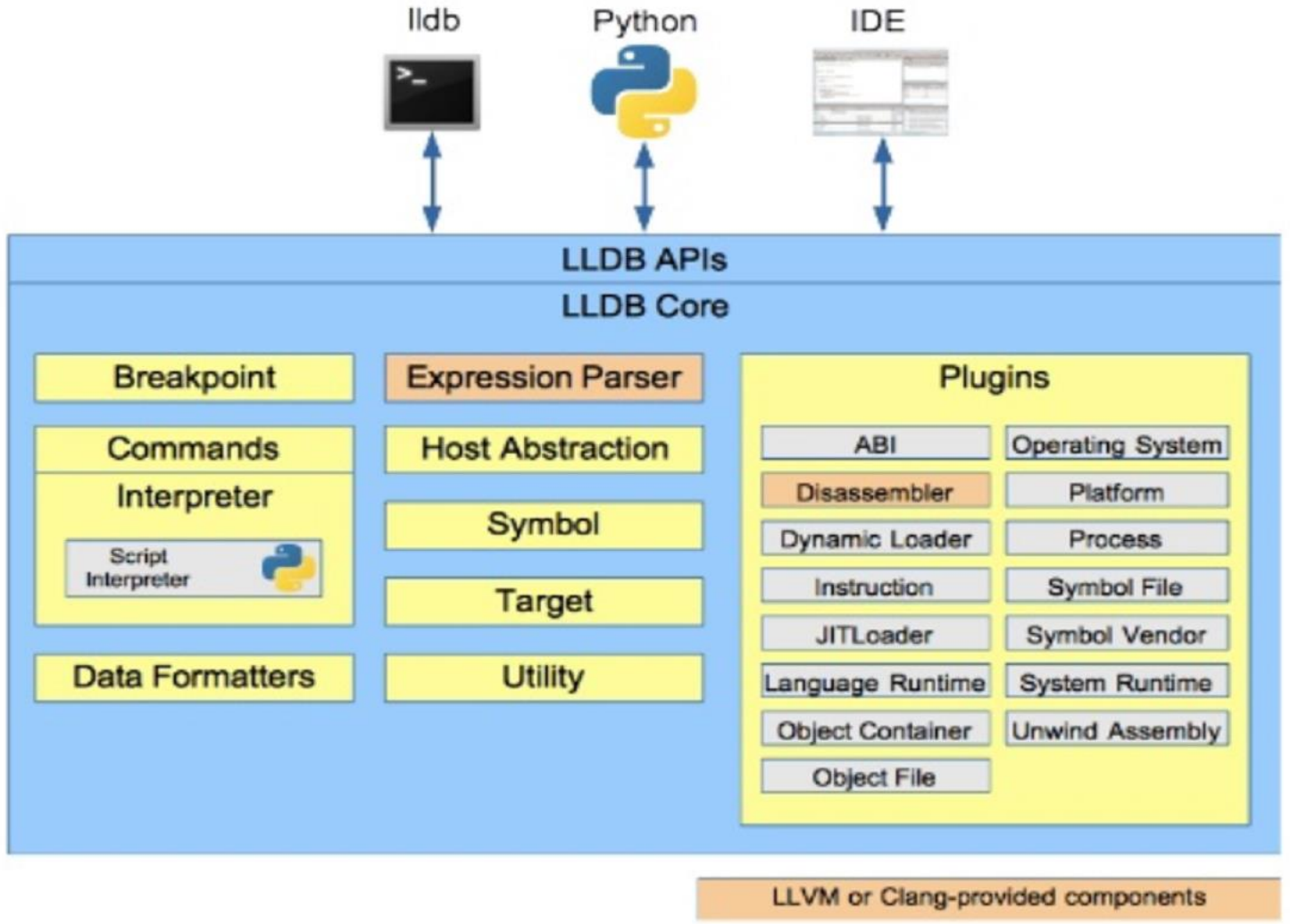
History - FreeBSD Slant

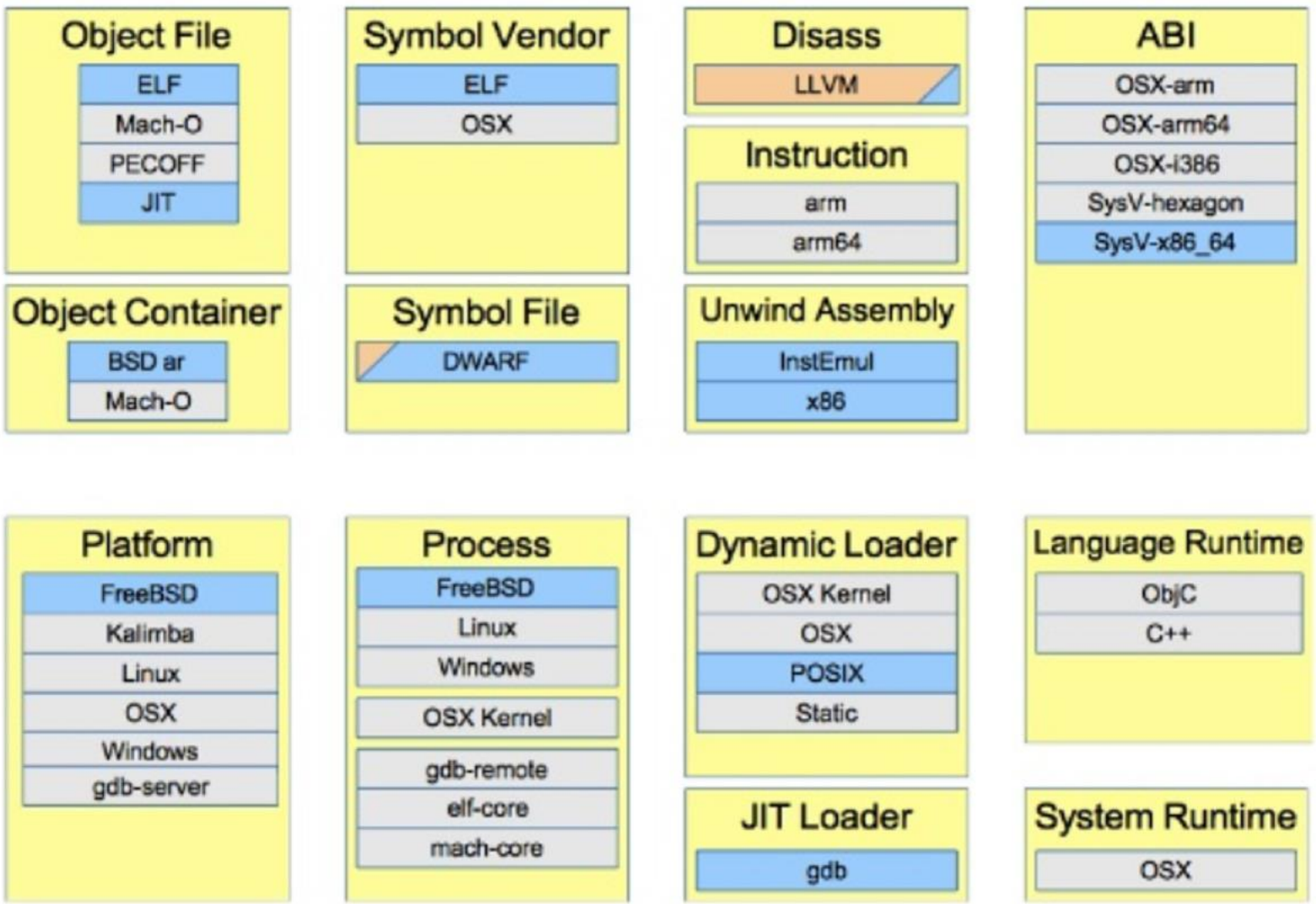
- “base system”
  - Kernel, libraries, system binaries & development tool chain
    - GCC and GDB
- 2004 – GDB version 6.1.1
- 2007 – GPLv3 in play
- /usr/ports contained later version of GCC & GDB


LLDB History
Lines of Code (000s)
800
600
400
200
0
Google
intel
CSR
Mentor Graphics
codeplay
XINUOS
VALVE
2010-05
2010-07
2010-09
2010-11
2011-01
2011-03
2011-05
2011-07
2011-09
2011-11
2012-01
2012-03
2012-05
2012-07
2012-09
2012-11
2013-01
2013-03
2013-05
2013-07
2013-09
2013-11
2014-01
2014-03
2014-05
2014-07
Lines of code data from Black Duck Open Hub (formerly Ohloh)

LLDB Design

- Set of modular components on top of the LLVM project
  - Clang expression parser
  - LLVM disassembler
- Up to date language support for C, C++, Objective C
- Multi-line expressions with local variables & types
- JIT or evaluate expression IR as needed.


lldb
Python
IDE
LLDB APIs
LLDB Core
Breakpoint
Commands
Interpreter
Script Interpreter
Data Formatters
Expression Parser
Host Abstraction
Symbol
Target
Utility
Plugins
ABI
Disassembler
Dynamic Loader
Instruction
JITLoader
Language Runtime
Object Container
Object File
Operating System
Platform
Process
Symbol File
Symbol Vendor
System Runtime
Unwind Assembly
LLVM or Clang-provided components


Object File
ELF
Mach-O
PECOFF
JIT
Symbol Vendor
ELF
OSX
Disass
LLVM
Instruction
arm
arm64
ABI
OSX-arm
OSX-arm64
OSX-i386
SysV-hexagon
SysV-x86_64
Object Container
BSD ar
Mach-O
Symbol File
DWARF
Unwind Assembly
InstEmul
x86
Platform
FreeBSD
Kalimba
Linux
OSX
Windows
gdb-server
Process
FreeBSD
Linux
Windows
OSX Kernel
gdb-remote
elf-core
mach-core
Dynamic Loader
OSX Kernel
OSX
POSIX
Static
Language Runtime
ObjC
C++
JIT Loader
gdb
System Runtime
OSX

Using LLDB

- Command syntax is fairly structured

  <noun> <verb> [-options [option-value]] [argument [argument...]]

- breakpoint
- command
- expression
- frame
- memory
- platform
- process
- register
- script
- source
- target
- watchpoint

LLDB - Help

- help
  - Displays the “nouns” and gdb-like short cuts
- help <noun>
  - Displays the verbs for the “noun”
- help <noun> <verb>
  - Describes the format/options available

Loading a Program into LLDB

- Command line

lldb -f <filename> [-c <corefile>] [[--] <prog-arg> ...]

lldb -n <process-name>

lldb -p <pid>

-w

--wait-for

Loading a Program (cont.)

- From running LLDB session
  - **target create [-c <corefile>] filename**
    - **file [-c <corefile>] filename**
  - **process launch [run-args …]**
    - **run [run-args ...]**

Attaching to Running Process

- From running LLDB session
  - **process attach [-p <pid> | -n <proc-name>]**
  - **pr at [-p <pid> | -n <proc-name>]**
    - **attach [<pid> | <proc-name>]**
  - **process continue**
    - **continue**

LLDB Breakpoints

- breakpoint set
  - Function name (beginning or end of prologue)
  - Line number in specific file
  - Line number in current file
  - C++ method
  - C++ exception
  - Conditional stop
  - Skip count
  - Specific thread ID

Breakpoints (cont.)

- For GDB users
  - **b <filename>:<linenum>**
  - **b <linenum>**
  - **b <address>**
  - **b <function>** *// after prologue*
  - **b <&function>** *// first instruction*
  - **b <module>`<function>**

Breakpoints (cont.)

- Other **breakpoint** verbs
  - clear
  - command
  - delete
  - disable
  - enable
  - list
  - modify
  - name

LLDB - Watchpoints

- **watchpoint set variable <var-name>**
- **watchpoint set expr -- <addr_ptr>**

- **watchpoint command**
- **watchpoint delete**
- **watchpoint disable**
- **watchpoint enable**
- **watchpoint ignore**
- **watchpoint list**
- **watchpoint modify**

Controlling Your Process

- attach
- connect
- continue
- detach
- handle
- interrupt
- kill
- launch
- load
- plugin
- save-core
- signal
- status
- unload

LLDB - thread

- **thread backtrace [-c <count>]**
  - **bt [<count> | all]**
- **thread info**
- **thread list**
- **thread jump**
- **thread return**
  - Short-circuit execution – return with optional value
- **thread select**

LLDB – thread (cont.)

- **thread continue**
- **thread step-in** **step**
- **thread step-inst** **stepi**
- **thread step-inst-over** **nexti** or **ni**
- **thread step-over** **next** or **n**
- **thread step-out** **finish**

LLDB - frames

- **frame select [-r <offset>] [<frame-index>]**
  - **up <cnt>**
  - **down <cnt>**
- **frame variable <cmd-options> [<var-name> […]]**

LLDB – memory & registers

- memory
  - read / write / find / history
  - **x** - alias for **memory read** - not GDB
- expression / expr / print
- register
  - read / write

URLs of Interest

- http://lldb.llvm.org
  - http://lldb.llvm.org/tutorial.html
  - http://lldb.llvm.org/lldb-gdb.html
  - http://lldb.llvm.org/python-reference.html
  - http://lldb.llvm.org/scripting.html
- https://wiki.freebsd.org/lldb

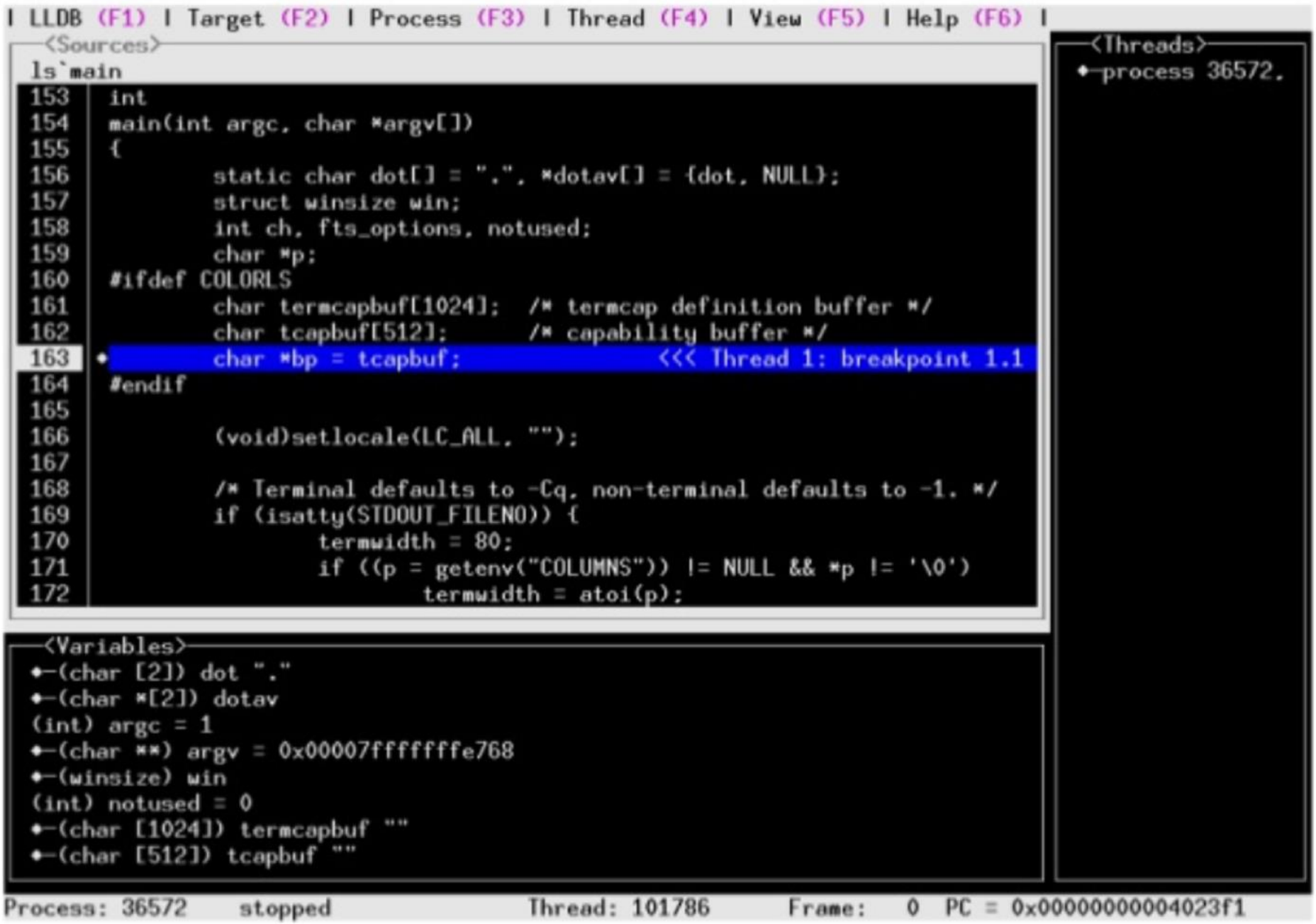
LLDB status on FreeBSD

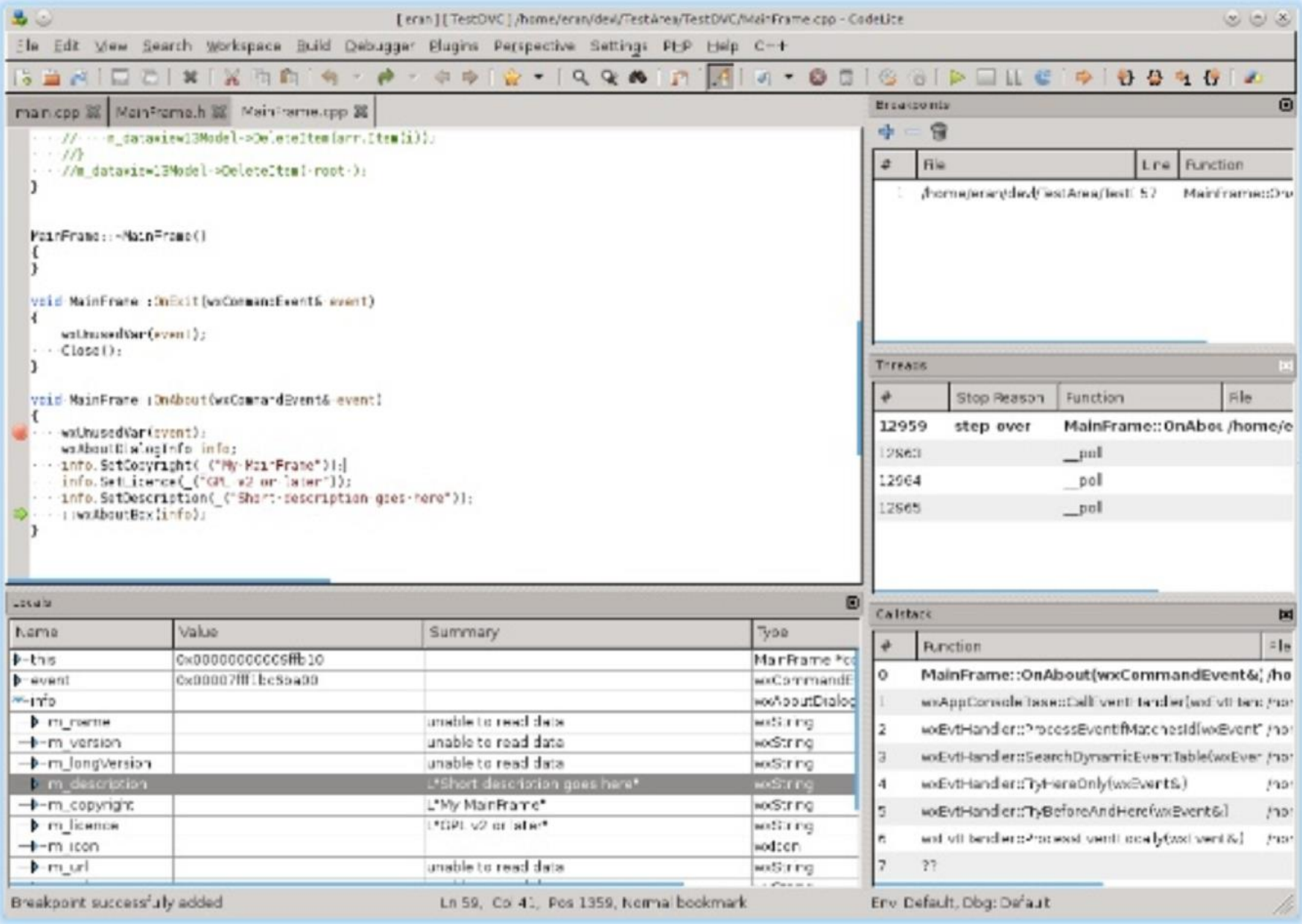
Feature	Status
process launch	Works
process attach (pid)	Works
process attach (name)	Works
userland core files	Works
Breakpoints	Works
Watchpoints	Works
Threads	Works
Testsuite	All non-passing tests have PRs
Remote debugging (gdbserver / debugserver)	In Progress
Kernel debugging	In progress (GSoC 2014)
Cross debugging	Untested

Architecture	Host Status	Target Status
amd64	Works	Works
arm	Unknown	Unknown
arm64 (aarch64)	Works	Works
i386	In progress	In progress
mips	Unknown	In progress
mips64	Unknown	In progress
powerpc	In progress	In progress
powerpc64	In progress	In progress

User Interfaces

- Built-in text CLI
- Prototype built-in curses GUI
- Codelite
- XCode
- GDB/MI compatible interface
  - Eclipse
  - Others
  - Affinic (commercial)




[eran] [TestDVC] /home/eran/dev/TestArea/TestDVC/MainFrame.cpp - CodeLite
File Edit View Search Workspace Build Debugger Plugins Perspective Settings PHP Help C++
main.cpp
MainFrame.h
MainFrame.cpp
//m_dataview13Model->DeleteItem(arr.Item(i));
//}
//m_dataview13Model->DeleteItem(root);
}
MainFrame::~MainFrame()
{
}
void MainFrame::OnExit(wxCommandEvent& event)
{
wxUnusedVar(event);
Close();
}
void MainFrame::OnAbout(wxCommandEvent& event)
{
wxUnusedVar(event);
wxAboutDialogInfo info;
info.SetCopyright(_("My MainFrame"));
info.SetLicence(_("GPL v2 or later"));
info.SetDescription(_("Short description goes here"));
::wxAboutBox(info);
}
Breakpoints
File
Line
Function
Threads
Stop Reason
Function
File
12959
step over
MainFrame::OnAbou
/home/e
12963
__pol
12964
__pol
12965
__pol
Locals
Name
Value
Summary
Type
this
0x00000000005ffb10
MainFrame *
event
0x00007fff:bc5ba00
wxCommandE
info
wxAboutDialog
m_name
unable to read data
wxString
m_version
unable to read data
wxString
m_longVersion
unable to read data
wxString
m_description
L"Short description goes here"
wxString
m_copyright
L"My MainFrame"
wxString
m_licence
L"GPL v2 or later"
wxString
m_icon
wxIcon
m_url
unable to read data
wxString
Callstack
Function
File
0
MainFrame::OnAbout(wxCommandEvent&)
/ho
2
wxEvtHandler::ProcessEventIfMatchesId(wxEvent
3
wxEvtHandler::SearchDynamicEventTable(wxEver
4
wxEvtHandler::TryHereOnly(wxEvent&)
5
wxEvtHandler::TryBeforeAndHere(wxEvent&)
7
??
Breakpoint successfully added
Ln 59, Col 41, Pos 1359, Normal bookmark
Env: Default, Dbg: Default