



MD5 Checksums in LLDB

Jonas Devlieghere

LLVM Dev Meeting 2024

Source Level Debugging

```
* thread #1, stop reason = EXC_BAD_ACCESS (code=1, address=0x0)
  frame #0: 0x000000102400354 test.out`main() at test.c:9:10
  6
  7   int main(int argc, char** argv) {
  8       int* i = foo();
-> 9       return *i;
 10   }
```

Source Level Debugging

```
* thread #1, stop reason = EXC_BAD_ACCESS (code=1, address=0x0)
  frame #0: 0x0000000102400354 test.out`main() at test.c:9:10
  6
  7   int main(int argc, char** argv) {
  8       int* i = foo();
-> 9       return *i;
 10   }
```

Address	File	Line	Column
...
0x0000000100000350	/tmp/test.c	9	11
0x0000000100000354	/tmp/test.c	9	10
0x0000000100000358	/tmp/test.c	9	3

What if the source file changed between compiling & debugging?

What if the source file changed between compiling & debugging?

```
* thread #1, stop reason = EXC_BAD_ACCESS (code=1, address=0x0)
  frame #0: 0x0000000102400354 test.out`main() at test.c:9:10
  6
  7   int main(int argc, char** argv) {
  8       int* i = bar();
-> 9       return *i;
 10   }
```

What if the source file changed between compiling & debugging?

DWARF 4

- Modification time
- Size of the file

What if the source file changed between compiling & debugging?

DWARF 4

- Modification time (version control, reproducible builds)
- Size of the file (doesn't catch `s/foo/bar/`)

What if the source file changed between compiling & debugging?

DWARF 4

- Modification time (version control, reproducible builds)
- Size of the file (doesn't catch `s/foo/bar/`)

DWARF 5

- MD5 checksum

Checksums in LLDB

Option 1: Reuse `FileSpec`

- Represents path (that doesn't necessarily exist)
- Used extensively throughout LLDB
- Adding checksum Increases the size of the class while mostly unused

Checksums in LLDB

Option 1: Reuse `FileSpec`

- Represents path (that doesn't necessarily exist)
- Used extensively throughout LLDB
- Adding checksum increases the size of the class while mostly unused

Option 2: Create new `SupportFile`

- Consists of a `FileSpec` + Checksum
- Requires distinguishing between `FileSpec` and `SupportFile`
- Also needed for DWARF 6 source embedding[†]

Checksum Mismatch

```
* thread #1, stop reason = EXC_BAD_ACCESS (code=1, address=0x0)
   frame #0: 0x0000000102400354 test.out`main() at test.c:9:10
   6
   7     int main(int argc, char** argv) {
   8         int* i = bar();
->  9         return *i;
   10    }
```

warning: test.c: source file checksum mismatch between line table
(8f3ef3297464849993e5da6f2d5f1dd0) and file on disk
(6dd9958b3547fe0475a5a3697e11290e)

NEW

