Deep Dive into Xdebug

nsfisis (いまむら)

PHP カンファレンス小田原 2026

2026-04-11

いまむら

nsfisis

Xdebug のステップ実行は
どう実現されているのか

Xdebug

- PHP の拡張(extension)
- ステップ実行
- トレース
- プロファイリング
- コードカバレッジ計測

今回はステップ実行の話

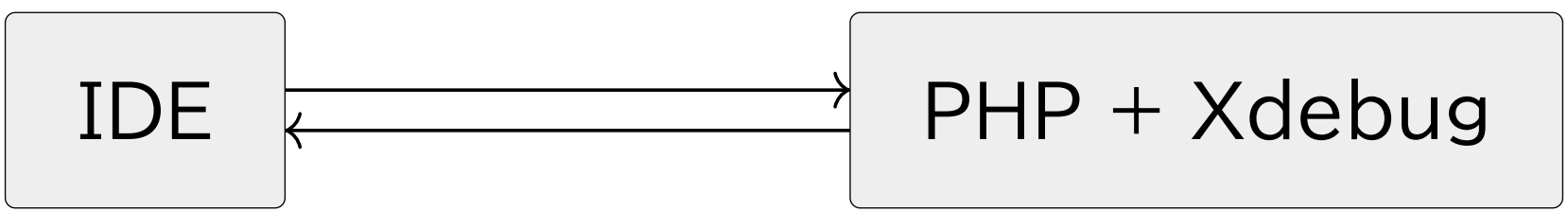
ステップ実行

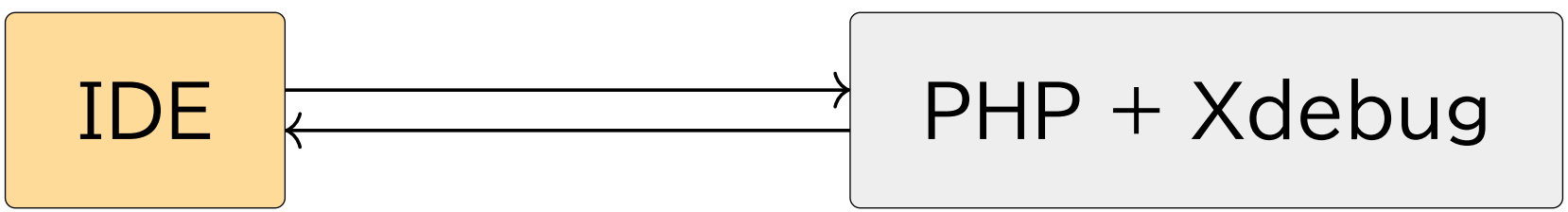
```
1 <?php
2
3 echo "Hello, ";
4 echo "Odawara!\n";
5
6
7
```

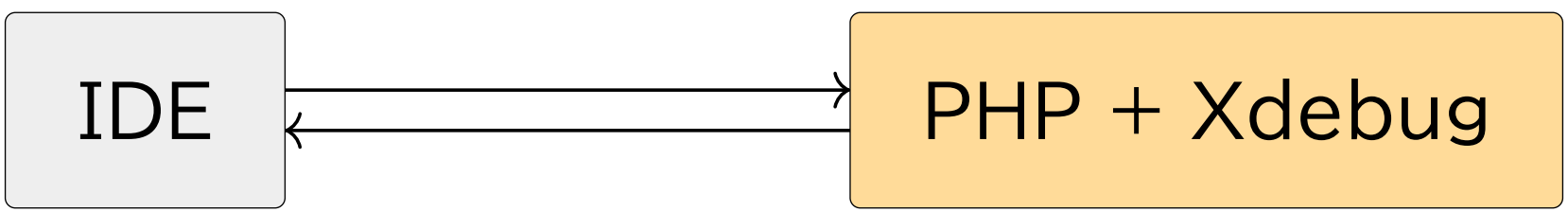
ステップ実行

```
1 <?php
2
3 echo "Hello, ";
4 echo "Odawara!\n";
5
6
7
```

ステップ実行は
どのように実装されているのか？


IDE
PHP + Xdebug


IDE
PHP + Xdebug


IDE
PHP + Xdebug

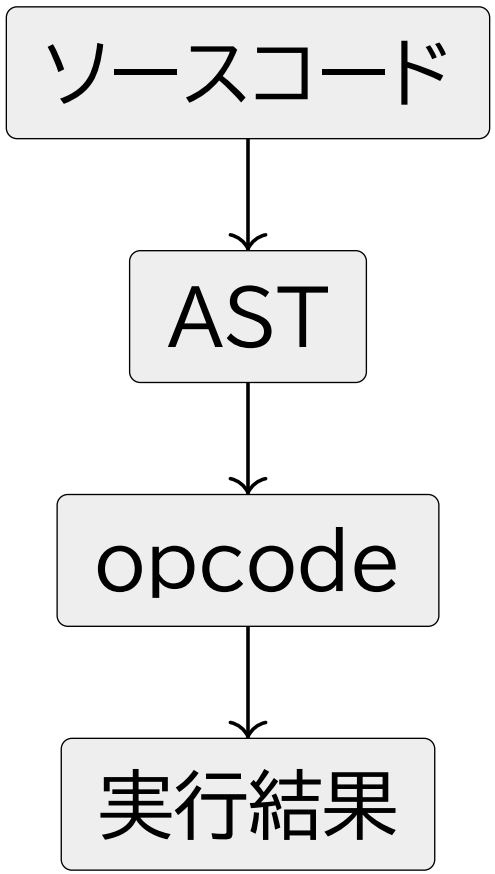
PHPerKaigi 2026

Xdebug と IDE によるデバッグ実行の仕組みを見る

by 新原雅司 / @shin1x1

PHPerKaigi 2026

fortee


ソースコード
AST
opcode
実行結果

```
echo "Hello, ";
echo "Odawara!\n";
```

↓

```
ECHO string("Hello, ")
ECHO string("Odawara!\n")
```

```
echo "Hello, ";
echo "Odawara!\n";
```

↓

```
EXT_STMT
ECHO string("Hello, ")
EXT_STMT
ECHO string("Odawara!\n")
```

```
// php-src: Zend/zend_vm_def.h
ZEND_VM_COLD_HANDLER(ZEND_EXT_STMT) {
    // 予め登録されているハンドラをすべて呼び出す
    zend_llist_apply_with_argument(
        &zend_extensions,
        zend_extension_statement_handler,
        execute_data
    );
}
```

```
// xdebug: xdebug.c
void xdebug_statement_call(zend_execute_data *frame) {
    if (ステップ実行が有効になっている？) {
        xdebug_debugger_statement_call(...);
    }
}
```

```
// xdebug: src/debugger/debugger.c
void xdebug_debugger_statement_call(...) {
    if (step out している？) { ... }
    if (step over している？) { ... }
    if (step into している？) { ... }
    if (この行にブレークポイントが設定されている？) { ... }
}
```

```
// xdebug: src/debugger/handler_dbgp.c
int xdebug_dbgp_breakpoint(...) {
    // レスポンスを生成して IDE へ送信
    send_message(context, response);
    // IDE からの命令を待ち受けるループへ入る
    xdebug_dbgp_cmdloop(...);
}
```

```
// xdebug: src/debugger/handler_dbgp.c
static int xdebug_dbgp_cmdloop(...) {
    do {
        // IDE からの命令を待ち受け
        option = xdebug_fd_read_line_delim(...);
        // IDE からの命令を実行
        // ret: 1=プログラムの実行再開 / 0=その他の命令
        ret = xdebug_dbgp_parse_option(...);
        if (ret != 1) {
            send_message(context, response);
        }
    } while (ret != 1);
}
```

ここまでのまとめ

- コンパイル時、各ステートメントの前に EXT_STMT が挿入される
- EXT_STMT 実行時に Xdebug のハンドラが呼ばれる
- IDE へレスポンスを送信し、命令待ちループへ
- IDE から実行再開命令が来ると、ループを抜けて実行を再開

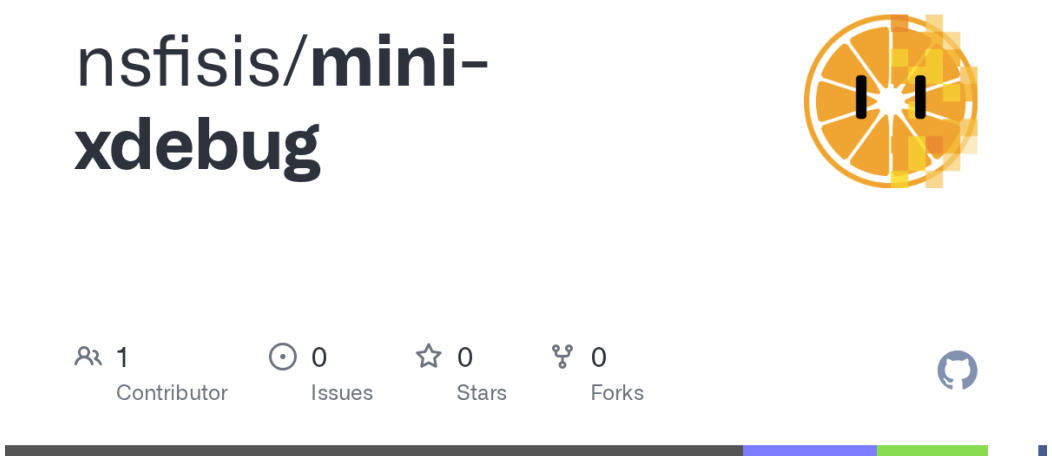
Xdebug はあくまで
PHP の一拡張として
実現されている

Xdebug はあくまで
PHP の一拡張として
実現されている

→ 自分でも作れるはず

ミニ Xdebug を作ろう！

https://github.com/nsfisis/mini-xdebug



ご清聴
ありがとうございました

参考文献:

- https://github.com/php/php-src
- https://github.com/xdebug/xdebug
- https://xdebug.org/docs/step_debug
- https://xdebug.org/docs/dbgp
- https://speakerdeck.com/shin1x1/exploring-how-debugging-works-with-xdebug-and-an-ide