



# セキュリティ & プログラミング キャンプ 2008 「デバッグの方法について」

ミラクル・リナックス株式会社  
よしおかひろたか

# Contents

- はじめに
- デバッグ入門
- デバッグの実際
- gdbの使い方
- 実習

# はじめに

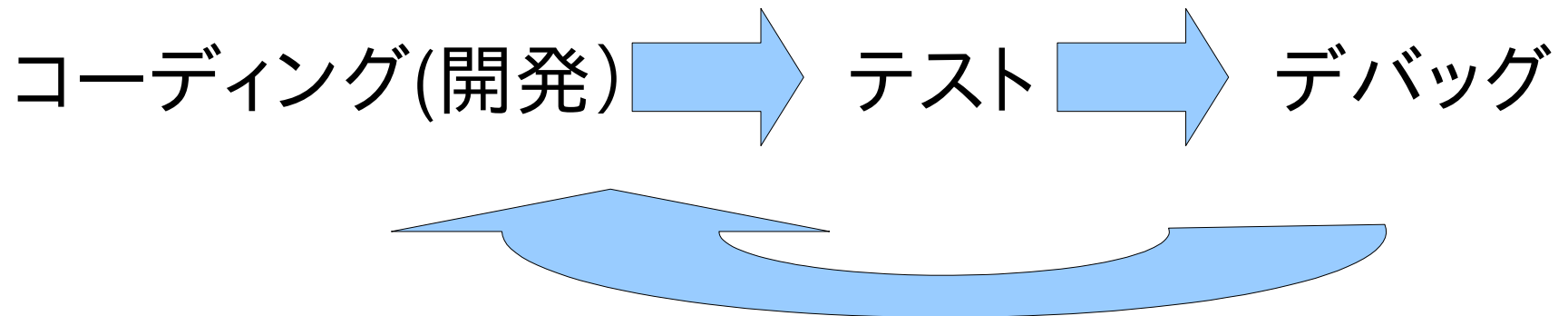
- バグとは
- デバッグとは
  - テストとデバッグ
- デバッグの基本的な手順
- 典型的なバグ

# バグ(不具合)とは

- バグ(不具合)とは、期待する振る舞い(値)と実際の振る舞い(値)との差
  - 期待する振る舞いが明確に定義されていないと実際の動作がバグかどうか判定できない。  
(それは仕様です…という言い訳)
- 狭義には仕様と実装(実際の振る舞い)との差

# デバッグとは

- デバッグ
  - 不具合を修正すること
  - 不具合とは、期待する振る舞い(仕様)と実際の振る舞いの差。
- テスト
  - 不具合を発見するプロセス

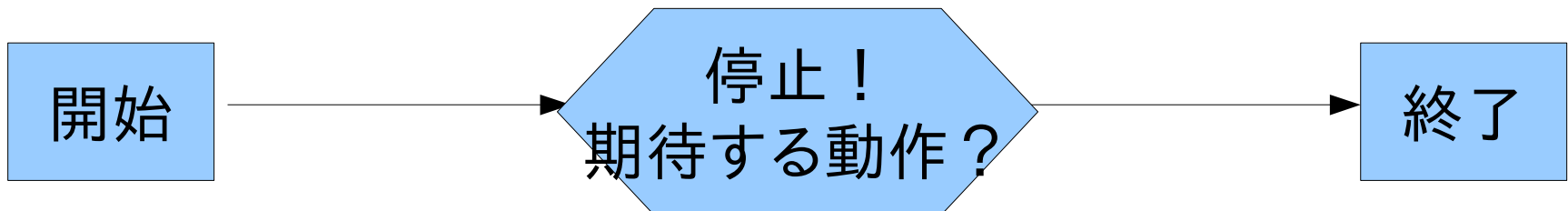


# デバッグ入門

- ツール: デバッガー
  - gdb
  - 高機能なソースコードデバッガー

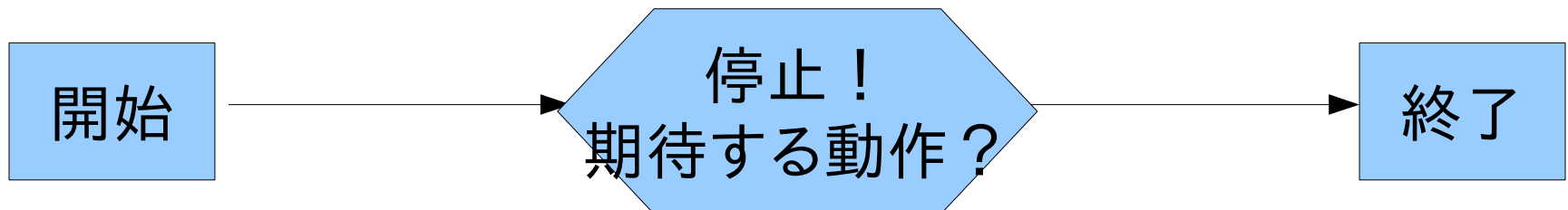
# プログラムの振る舞い(動作)

- プログラムの動作を次のように分類
  - 1.プログラムは期待する値を出力して終了する。
  - 2.プログラムは誤った値(期待と違う値)を出力して終了する。
  - 3.プログラムは何も終了しないで停止する。
  - 4.プログラムは終了しない。
- 2~4をバグと呼ぶことにする。バグを直すプロセスをデバッグと呼ぶ。



# デバッグの戦略

- 動作の確認
  - 期待する動作と実際の動作を確認
  - 差分は何かを分析
- 実際の実行
  - 途中で実行を停止する
  - 期待する動作(値)か？
    - 期待する動作なら、さらに実行継続し、繰り返す
  - 期待する動作でないなら、実行開始してから、ここまでの間に不具合があるはず。



# デバッグ

- デバッグの4大機能
  - プログラムを動作させる
  - 指定した条件でプログラムを停止させる
  - プログラムが止まった時に、何が起こったか調べる
  - プログラムを変更する

# gdb : デバッガ

- 起動  
\$ gdb file名
- 実行  
(gdb) run 引数
- 停止する場所の指定  
(gdb) break 関数名  
(gdb) break 行番号
- 変数の値の表示  
(gdb) print 変数名  
(gdb) x アドレス

# gdb:デバツガ

- 変数の値を変更する  
(gdb) set 変数=式

# デモ

- coreutilsのソースコードを読む
- 準備。
- ソースコードの入手  
\$ apt-get source coreutils  
\$ cd coreutils-6.10
- ビルド  
\$ dpkg-buildpackage -rfakeroot
- ソースコードは以下にある  
\$ cd build-tree/coreutils-6.10/src

# デモ

- 読みたいコマンドを実行してみる  
コマンドレベルの場合  
\$ gdb pwd  
emacsからの起動の場合  
M-x gdb pwd
- 停止点(break point)を設定してみる  
(gdb) b main
- 実行してみる  
(gdb) run

p p\*           

```
Current directory is ~/work/coreutils-6.10/build-tree/coreutils-6.10/src/
GNU gdb 6.8-debian
Copyright (C) 2008 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i486-linux-gnu"...
(gdb) b main
Breakpoint 1 at 0x80491d7: file pwd.c, line 288. (2 locations)
(gdb) run
Starting program: /home/hyoshiok/work/coreutils-6.10/build-tree/coreutils-6.10/s
src/pwd

Breakpoint 1, main (argc=1, argv=0xbfd34224) at pwd.c:288
(gdb) █
```

```
--u:** *gud-pwd* All (15,6) (Debugger:run [stopped])-----
```

```
    }
    /* See if a leading slash is needed; file_name_prepend adds one. */
    if (file_name->start[0] == '\\0')
        file_name_prepend (file_name, "", 0);
}

int
main (int argc, char **argv)
{
    char *wd;

    initialize_main (&argc, &argv);
    program_name = argv[0];
    setlocale (LC_ALL, "");
    bindtextdomain (PACKAGE, LOCALEDIR);
    textdomain (PACKAGE);

    atexit (close_stdout);
```

```
--SKK:--:-- pwd.c 90% (288,0) (C/1 Abbrev Fill)-----
```

```
Loading cc-mode...done
```

p p\*           

```

Current directory is ~/work/coreutils-6.10/build-tree/coreutils-6.10/src/
GNU gdb 6.8-debian
Copyright (C) 2008 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i486-linux-gnu"...
(gdb) b main
Breakpoint 1 at 0x80491d7: file pwd.c, line 288. (2 locations)
(gdb) run
Starting program: /home/hyoshiok/work/coreutils-6.10/build-tree/coreutils-6.10/s
src/pwd

Breakpoint 1, main (argc=1, argv=0xbfd34224) at pwd.c:288
(gdb)

```

-u:\*\* \*gud-pwd\* All (15,6) (Debugger:run [stopped])-----

```

if (getopt_long (argc, argv, "", NULL, NULL) != -1)
  usage (EXIT_FAILURE);

if (optind < argc)
  error (0, 0, _("ignoring non-option arguments"));

wd = xgetcwd ();
if (wd != NULL)
{
  puts (wd);
  free (wd);
}
else
{
  struct file_name *file_name = file_name_init ();
  robust_getcwd (file_name);
  puts (file_name->start);
  file_name_free (file_name);
}

```

--SKK:--:-- pwd.c 96% (310,0) (C/l Abbrev Fill)-----

Command: next 1