

差出人: Naruaki TOMA <tnal@ie.u-ryukyu.ac.jp>
件名: [Slab34-ec:00337] gdbの補足
日時: 2012年11月15日 17:06:08 JST
宛先: Evolutionary Computing <slab34-ec@ie.u-ryukyu.ac.jp>
返信先: Evolutionary Computing <slab34-ec@ie.u-ryukyu.ac.jp>

當間@情報工学科です。

全グループJavaだったのでEclipseでのデバッグ演習をした方が良かったかもしれませんが、また機会があれば。以下の説明では触っていないコマンドもありますが、下に列挙しているものくらいで大抵は事足りるんじゃないかと。

サンプルの場所: shark:-info3/HG/2012/share2/gdb/

=====
gcc 時に -g オプション付けてコンパイル。

prompt> gdb 実行ファイル名

注意: ()の中身は引数ではなく、未省略時のコマンド名です。

r (run) #プログラム実行
l (list) #ソースリスト
f (frame) フレーム番号#指定したフレーム番号(呼び出し元)に戻る。
b (breakpoint) 行数or関数名#実行を一時中断する行数を指定。関数名も指定可能。
追加で、一時中断した際に自動実行して欲しい (printさせるとか) 等のコマンド設定も可能[1]。
info b (breakpoint) 設定したブレークポイント一覧を確認。
condition [bnum] [condition]
ブレークポイントに条件を付与する際の書式。
bnum = 条件を付与したいブレークポイント番号
condition = 付与したい条件
例えば「condition 1 x==1」と書くと、
ブレークポイント1番が「x==1」の時だけブレークする
となる。

n (next) 中断箇所から1行実行して停止する。
s (step) 中断箇所から1行実行して停止するが、
該当行が関数呼び出しの場合にはその関数の中に入る(step in)。
c (continue) 中断箇所から続行。
p (print) 変数名 プログラム内の変数が実施に持っている値を確認。
d (display) 変数名 毎回自動的にprintコマンドを実行する。
dd (delete display) display設定を消去。
w (watch) 変数名 指定した変数の値が変更された時に停止。
bt (backtrace) 関数呼び出し順の流れ(フレーム)を確認。

[1] gcc+gdbによるプログラムのデバッグ 第1回 ステップ実行、変数の操作、ブレークポイント

<http://rat.cis.k.hosei.ac.jp/article/devel/debugongccgdb1.html>

[2] 第2回 変数の監視、バックトレース、その他のコマンド

<http://rat.cis.k.hosei.ac.jp/article/devel/debugongccgdb2.html>

[3] 第3回 gdbの便利な機能、デバッグの例

<http://rat.cis.k.hosei.ac.jp/article/devel/debugongccgdb3.html>

[4] gdb を用いたデバッグ方法

<http://uguisu.skr.jp/Windows/gdb.html>

[5] ポインタ虎の巻

<http://www.nurs.or.jp/~sug/soft/tora/index.htm>

=====
>アクセサ

基本的にはOOPにおけるカプセル化に関する概念の一つ。

オブジェクトの操作(データ参照/編集)、特にクラスメンバ変数の操作を原則としてメソッドを介して行うようにすること。データ参照するアクセサをgetter、編集するアクセサをsetterと呼び、例えば Class A のメンバ変数 hoge があるとき、

A.setHoge(1); // setterの例
fuga = A.getHoge(); // getterの例

のように、メンバ変数名にset/getを組み合わせたメソッドでアクセスする事が多い。

[6] カプセル化とアクセサの関係

<http://ja.wikipedia.org/wiki/カプセル化>

<http://ja.wikipedia.org/wiki/アクセサ#.E3.82.A2.E3.82.AF.E3.82.BB.E3.82.B5>

[7] オブジェクト指向プログラムでgetter/setterメソッドを使わなければならない10の理由

http://fukumori.org/oo/why_use_getters_and_setters_j.html

[8] O4_encapsulation (C言語での疑似カプセル化の例)

http://www.geocities.jp/cxh2x/c_like_cxx_web/encapsulation/O4_encapsulation/index.html

*ここまでやる必要があるかは別として、こういう書き方もあるよという参考例。

Naruaki Toma

E-mail: tnal@ie.u-ryukyu.ac.jp, Tel: 098-895-8830

<http://www.eva.ie.u-ryukyu.ac.jp/~tnal/>

Slab34-ec mailing list

Slab34-ec@ie.u-ryukyu.ac.jp

<https://ginowan.ie.u-ryukyu.ac.jp/mailman/listinfo/slab34-ec>