GDB工具實習

2016/10/21

Liang-Cheng Hu

簡介

- GNU Debugger (GDB)
- GNU 軟體系統中的標準除錯器
- 支援C、C++、FORTRAN、Pascal

準備

- Linux / Unix
- GDB
- Pietty / putty
  - http://ntu.csie.org/~piaip/pietty/
  - Linux.cs.ccu.edu.tw / 22
- Source code(s)
  - http://www.cs.ccu.edu.tw/~cyj102m/test.c

執行環境

- 程式的參數
  - (gdb) set args aa bb cc
  - (gdb) show args
- 工作目錄
  - (gdb) cd
  - (gdb) pwd

程式執行

Start

- % gcc –g –o test test.c
- (gdb) file test

- run (r) -- 開始執行程式
- next (n) -- 單步執行 (step over)
- step (s) -- 進入函式 (step into)
- finish (f) -- 退出該函數返回到它的調用函數中
- continue (c) -- 離開中斷點，繼續執行程式
- quit(q) -- 離開GDB

觀看程式碼

list

- (gdb) l …

參數	說明
list linenum	列出第幾行
list filename:linenum	列出某檔案的第幾行，檔案可省略。
list [function]	列出某函數的程式碼
list	繼續印出程式碼
list -	印出上一次list的程式碼的前一段程式碼(類似向上翻動)
show listsize	顯示現在一次印出幾行
set listsize	設定一次印出幾行

List(cont.)

- (gdb) l
- (gdb) l -
- (gdb) l test.c:20
- (gdb) l FuncA
- (gdb) show listsize
- (gdb) set list size 15

設定中斷點 (breakpoint)

Break

- (gdb) b …
- (gdb) info break / (gdb) i b

參數	說明
break linenum	指定行號設中斷點
break filename:linenum	在某source file的第幾行或指定函式設定中斷點
break [function]	在某函數的進入點設中斷點
break +offset | -offset	當程式停止時，在停止位置的前/後第offset行設中斷點
break	在下一個要執行的指令設中斷點
break [args] if [cond]	當[cond]這個運算式為真，設定中斷點。args可能是上列的任一種情形。
tbreak args	只會生效一次的中斷點

Break (cont.)

- (gdb) b 20
- (gdb) b test.c:111
- (gdb) b main
- (gdb) b +5 /(gdb) b -5
- (gdb) b test.c:main
- (gdb) b 12 if(i==30)

Break (cont.)

- (gdb) clear 10
  - 刪除第10行的break point
- (gdb) delete 10
  - 刪除第10個的break point

✧ 搭配info break!

Break (cont.)

- (gdb) enable 5
  - 關閉第5個 break point
- (gdb) disable 5
  - 重啟第5個 break point

✧ 搭配info break!

設定 watch point

watchpoint

- watch *varname* -- 當變數被寫入時，中斷程式
- rwatch *varname* -- 當變數被讀取時，中斷程式
- awatch *varname* -- 當變數被讀取或寫入時，中斷程式
- watch **(int *)0x12345678* -- 監看記憶體位址，監看範圍由變數型別決定，當此記憶體位址被寫入時，中斷程式
- info watch -- 列出所有的 watchpoint

觀察變數資料

print

- (gdb) p var
- (gdb) p var[1]@5
- (gdb) p /x var
- (gdb) p var*5
- (gdb) p var=100 (set var=100)

/x	十六進位
/d	有號整數
/u	無號整數
/o	八進位
/t	二進位
/a	位址
/c	字元
/f	浮點數

display

- (gdb) display AAA
- (gdb) info display
- (gdb) enable display 1
- (gdb) disable display 2
- (gdb) delete display 1

Ptype & whatis

- (gdb) whatis var -- 顯示變數型別
- (gdb) ptype Struct[tmp] -- 顯示 class, struct 的定義內容

Reference

1. http://www.gnu.org/software/gdb/
2. http://www.gnu.org/software/gdb/documentation/
3. http://www.study-area.org/cyril/opentools/opentools/debug.html
4. http://www.cs.cmu.edu/~gilpin/tutorial/
5. http://loda.hala01.com/2012/04/gdb%E7%AD%86%E8%A8%98/

Lab Gdb Homework

Lab Gdb

- 繳交格式 docx 上傳至CyberCCU2作業區
- 繳交期限 2016/11/11 23:59

- 下載測試檔
  - Lab-GDB_testfile.zip(cyber ccu2 > 課程教材 > 工具實習教材)

Lab Gdb

- 截圖並說明如何使用的gdb中的指令

1. callandcall.c
   - 列出在main()結束前X、Y的值。
   - 列出在FuncA()結束前X、Y的值。
   - 列出在FuncB()結束前X、Y的值。

   (若傳入值為記憶體位址，請列出此記憶體位址裡的值。)
2. qsort.c
   - 請自訂欲排序的數字。(大於三個).
     - Ex: ./a.out 59 74 22 99 13 101
   - 列印出array[] 排序過程中資料的移動情況。
     - Hints ：watch 、display、print x[]@6

Any Question?