

## 1 演習 1 (GDB)

この演習は ex1/に必要なリソースが入っています。ex1/に移動してください。

以下のことを GDB で実行してください。

1. デバッグ情報を付与して test.c をコンパイルしてください。
2. 1. で作成した実行ファイルを読み込んで GDB を起動してください。
3. 以下の場所にブレークポイントを設置してください。
  - 35 行目(ただし  $i \leq j$  のときのみ)
  - 関数 qsort
4. プログラムを開始してください。(22 行目で停止したことを確認)
5. 変数 data の中身を長さ 12 の配列として表示してください。
6. 次のブレークポイントまで進んでください。(35 行目で停止していることを確認)
7. 変数 i と j の値を確認してください。
8. 次のブレークポイントまで進んでください。(22 行目で停止していることを確認)
9. バックトレースを表示してください。
10. ブレークポイントをすべて削除してください。
11. qsort にブレークポイントを設置してください。ただし low>high の条件をつけること。
12. 次のブレークポイントまで進んでください。(22 行目で停止していることを確認)
13. 変数 data の中身を長さ 12 の配列として表示してください。
14. 一番最初の qsort を抜けるまで進んでください。
13. 変数 data の中身を長さ 12 の配列として表示してください。(ソースされていることを確認)
14. 最後までプロセスを実行してください。
15. GDB を終了してください。

回答 Section 4

## 2 演習 2 (LLDB)

この演習は ex2/に必要なリソースが入っています。ex2/に移動してください。

以下のことを LLDB で実行してください。

1. デバッグ情報を付与して test.c をコンパイルしてください。
2. 1. で作成した実行ファイルを読み込んで LLDB を起動してください。
3. 以下の場所にブレークポイントを設置してください。
  - 35 行目(ただし  $i \leq j$  のときのみ)
  - 関数 qsort
4. プログラムを開始してください。(22 行目で停止したことを確認)
5. 変数 data の中身を長さ 12 の配列として表示してください。
6. 次のブレークポイントまで進んでください。(35 行目で停止していることを確認)
7. 変数 i と j の値を確認してください。
8. 次のブレークポイントまで進んでください。(22 行目で停止していることを確認)
9. バックトレースを表示してください。
10. ブレークポイントをすべて削除してください。
11. qsort にブレークポイントを設置してください。ただし low>high の条件をつけること。
12. 次のブレークポイントまで進んでください。(22 行目で停止していることを確認)
13. 変数 data の中身を長さ 12 の配列として表示してください。
14. 一番最初の qsort を抜けるまで進んでください。

13. 変数 `data` の中身を長さ 12 の配列として表示してください。(ソースされていることを確認)
14. 最後までプロセスを実行してください。
15. LLDB を終了してください。

回答 Section 5

### 3 演習 3 (Perf)

この演習は ex3/に必要なリソースが入っています。ex3/に移動してください。

### 4 演習 1 回答 (GDB)

1. `$ gcc -g test.c -o a.out`
2. `$ gdb ./a.out`
3. 1. `(gdb) break 35 if i<=j`  
2. `(gdb)`
4. `(gdb)`
5. `(gdb)`
6. `(gdb)`
7. `(gdb)`
8. `(gdb)`
9. `(gdb)`
10. `(gdb)`
11. `(gdb)`
12. `(gdb)`
13. `(gdb)`
14. `(gdb)`

### 5 演習 2 回答 (LLDB)

1. `$ gcc -g test.c -o a.out`
2. `$ lldb ./a.out`
3. 1. `(lldb) breakpoint set -l 35 -c i<=j`  
2. `(lldb) breakpoint set -n qsort`
4. `(lldb) process launch`
5. `(lldb) frame variable -Z 12 data`
6. `(lldb) thread continue`
7. `(lldb) frame variable i j`
8. `(lldb) thread continue`
9. `(lldb) thread backtrace`
10. `(lldb) breakpoint delete`
11. `(lldb) breakpoint set -n qsort -c low>high`
12. `(lldb) thread continue`
13. `(lldb) frame variable -Z 12 data`
14. 色々方法あり。例えば  
`(lldb) breakpoint disable`  
`(lldb) thread until -f 64`  
`(lldb) breakpoint enable`
15. `(lldb) frame variable -Z 12 data`

16. 色々方法あり。例えば

```
(lldb) breakpoint disable  
(lldb) thread continue  
(lldb) breakpoint enable
```

17. (lldb) quit

## 6 演習 3 回答 (perf)