Linux C gdb によるデバッグ

[Programming C][Linux]
以下の本からのメモ。良書。

準備

・gcc に、-g オプションを付けてプログラムをビルド

```
# gcc -Wall -g -o myhead myhead.c
```

・実行するとセグメンテーション違反となるバグ入りプログラム

```
# ./myhead -n 5
セグメンテーション違反です
```

デバッガの起動

・デバッグ対象のプログラム名とともに、gdb を起動

```
# gdb ./myhead
GNU gdb (GDB) Red Hat Enterprise Linux (7.0.1-23.el5_5.2)
Copyright (C) 2009 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 "i386-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /root/workspace/myhead/src/myhead...done.
(gdb)
```

デバッグ対象のプログラムを run コマンドで起動

・run コマンドにオプションを付けると、そのまま元のプログラムに引き渡す

```
(gdb) run -n 5
Starting program: /root/workspace/myhead/src/myhead -n 5

Program received signal SIGSEGV, Segmentation fault.
0x00bfe50b in ____strtol_l_internal () from /lib/libc.so.6
```

Segmentation fault が発生したので、その場所で停止し、問題の起きた場所を表示

スタックトレースを　backtrace コマンドで表示する

・実行した関数が呼出の逆順に列挙

```
(gdb) backtrace
#0  0x00bfe50b in ____strtol_l_internal () from /lib/libc.so.6
#1  0x00bfe26f in __strtol_internal () from /lib/libc.so.6
#2  0x00bfb5c9 in atol () from /lib/libc.so.6
#3  0x0804862b in main (argc=3, argv=0xbfffe864) at myhead.c:31
```

frame 、list コマンドでソースを参照

・上記スタックトレースで、問題が発生した原因は、main の 31 行目と推察される
・先頭に、#3 とあるので、以下のように frame コマンドを実行し、main に移動

```
(gdb) frame 3
#3  0x0804862b in main (argc=3, argv=0xbfffe864) at myhead.c:31
31                              nlines = atol(optarg);
```

実行中の関数が、main になり、実行中の行が表示される

・もう少し広い範囲を見るときには、list コマンドを使用する

```
(gdb) list
26              int opt;
27              long nlines = DEFAULT_T_N_LINES;
28              while((opt = getopt_long(argc,argv,"n",longopts,NULL)) != -1) {
29                      switch(opt) {
30                      case 'n':
31                              nlines = atol(optarg);
32                              break;
33                      case 'h':
34                              fprintf(stdout,"Usage: %s [-n LINES] [FILE...]\n", argv[0]);
35                              exit(0);
```

print コマンドで変数の中身を確認する

```
(gdb) print optarg
$1 = 0x0
```

中身が NULL。Null ポインタがセグメンテーション違反の原因と判明

continue 、quit で、デバッグを終了する

・continue で実行を再開し、quit でデバッガを終了する

```
(gdb) continue
Continuing.

Program terminated with signal SIGSEGV, Segmentation fault.
The program no longer exists.
(gdb) quit
```

まとめ

コマンド	省略形	内容
backtrace	bt	バックトレースを表示
frame N	f	フレーム N に移動する
list	l	現在の関数のソースを表示
print EXPR	p	式 EXPR の値を表示
continue	c	続きを実行
quit	q	gdb を終了