

目次

1 デバッガ	1
2 GDB	1
2.1 GDB の起動	1
2.2 GDB の終了	1
2.3 シェルコマンド	1
2.4 ロギング出力	2
3 GDB コマンド	2

1 デバッガ

プログラムのバグ(bug)を取り除く(de-)ことをデバッグといいます。デバッグを行う手法はいくつかあり、例えばプログラム中に標準出力を行う命令を追加してデバッグを行う print デバッガと呼ばれる方法があります。デバッガはデバッグを支援するツールで、プログラムの任意箇所での停止や、変数の値の表示や変更、スタックトレースやメモリ内容の監視など高度な機能によりデバッグを支援します。

C 言語で書かれたプログラムに対応するデバッガはいくつか存在しており、有名なものに GDB と LLDB が存在します。このドキュメントではこの二つのデバッガについて基本的な使用方法の解説を行います。

2 GDB

GDB は Gnu Project のデバッガです。

2.1 GDB の起動

GDB を起動するには以下のいずれかのコマンドを使用します。起動後はコマンドを受け付けます。

```
gdb [options] [executable-file [core-file or process-id]]
gdb [options] --args <executable-file> [inferior-arguments ...]
```

--args を指定する場合、実行可能ファイルの後の引数(inferior-arguments) が実行時に渡されます。例えば `gdb --args gcc -O2 -c foo.c` は `gcc -O2 -c foo.c` の実行にデバッガをアタッチします。

options に指定できるオプションは `gdb -h` で確認できます。

2.2 GDB の終了

GDB を終了するには `quit [expression]`, `exit [expression]` または `q` で終了できます。expression に指定した値は終了コードとして帰ります。

2.3 シェルコマンド

GDB 起動中にシェルコマンドを使用することができます。

```
shell <command-string>
!<command-string>
```

pipe 命令を使用して gdb の出力を他のプログラムに繋ぐことができます。

```
pipe [command] | <shell_command>
```

```
| [command] | <shell_command>  
pipe -d <delim> <command> <delim> <shell_command>  
| -d <delim> <command> <delim> <shell_command>
```

command が | を含むときには -d で別の記号(列)を指定します。

2.4 ロギング出力

GDB の出力をファイルに行うことができます。GDB にはロギングを制御するコマンドがいくつか用意されています。

set logging enabled [on|off] ロギングのオンオフ切り替え

set logging file <file> 現在のログファイルの名前を変更。デフォルト値は gdb.txt

set logging overwrite [on|off] 上書きか書き足しか(on で上書き)。デフォルト値は off

set logging redirect [on|off] on にすると GDB の出力がログファイルにのみ行われる。デフォルト値は off

set logging debugredirect [on|off] on にすると GDB デバッグの出力がログファイルにのみ行われる。デフォルト値は off

show logging ロギングの設定を表示する

3 GDB コマンド