

LAB 7

Wu Jiayao 517370910257

1.

1.

```
gcc - g
```

2. The GNU Project Debugger

```
gcc- c cthread.c -lpthread
```

2.

1. <https://www.gnu.org/software/gdb/>

- 2.
- Ada
 - Assembly
 - C
 - C++
 - D
 - Fortran
 - Go
 - Objective-C
 - OpenCL
 - Modula-2
 - Pascal
 - Rust

- 3.
- **backtrace:** a backtrace is a summary of how your program got where it is. It shows one line per frame, for many frames, starting with the currently executing frame (frame zero), followed by its caller (frame one), and on up the stack.
 - **where:** find out where the program is, i.e. where the segmentation fault occurred, using the where command. This gives a function call trace of how you got to this point and shows line numbers inside files.
 - **finish:** continue running until just after function in the selected stack frame returns. Print the returned value (if any).
 - **delete:** delete the breakpoints, watchpoints, or catchpoints of the breakpoint list specified as argument. If no argument is specified, delete all breakpoints.
 - **info breakpoints:** when the info breakpoints command is invoked without any arguments, it displays information about all breakpoints. When one or more numbers are specified,

4. `cond [breakpoint number][expression]`, e.g `cond 1 x==0`
5. **`gdb -tui`** enables the GDB Text User Interface (TUI), which is a terminal interface which uses the `curses` library to show the source file, the assembly output, the program registers and GDB commands in separate text windows. The TUI mode is supported only on platforms where a suitable version of the `curses` library is available
6. activate the instruction-recording target, by executing the command.

`record`

Step program backward until it reaches the beginning of a previous source line; then stop it, and return control to GDB.