**19-linux工具**

**Gdb**

gdb attach/detach 是GDB（GNU调试器）的一个功能，它允许你将GDB附加到正在运行的进程上进行调试，或者在调试完成后从进程中分离出来。

解决方案1：

在命令行中使用gdb，attach，detach。

例如，如果你想调试一个名为my\_program的正在运行的进程，你可以使用以下命令：

1. 设置断点：

* break 或 b：在指定的行或函数处设置断点。例如，b main.c:10 会在main.c文件的第10行设置一个断点。
* tbreak：设置一个临时断点，程序在第一次停在该断点后会自动删除。

1. 单步执行：

* step 或 s：单步执行，如果当前行有函数调用，则会进入该函数。
* next 或 n：单步执行，如果当前行有函数调用，则不会进入该函数。
* reverse-step 和 reverse-next：反向单步执行。

1. 继续执行：

* continue 或 c：从当前位置继续执行程序，直到遇到下一个断点或程序结束。

1. 查看变量和表达式：

* print 或 p：打印表达式的值。例如，p variable\_name 会打印变量variable\_name的值。
* display：设置一个表达式，每次程序停止时自动打印该表达式的值。例如，display variable\_name 会每次程序停止时打印变量variable\_name的值。

1. 查看内存：

* x：查看内存地址的内容。例如，x/10x 0x12345678 会以16进制格式查看地址0x12345678开始的10个内存单元。

1. 查看调用栈：

* backtrace 或 bt：显示当前的函数调用栈。

1. 退出GDB：

* quit 或 q：退出GDB调试器。

1. 其他命令：

* info breakpoints：列出所有断点。
* delete：删除一个或所有断点。
* disable/enable：禁用或启用一个或所有断点。
* run 或 r：启动被调试的程序。

**多线程调试**

1. info threads：显示当前可调试的所有线程。每个线程会有一个gdb为其分配的ID，后续操作会用到这个ID。

|  |
| --- |
| Bash (gdb) info threads  Id Target Id Frame Thread 0x7ffff7ff6700 (LWP 28750) "a.out" 0x00007ffff773e655 in select() Thread 0x7ffff67f5700 (LWP 28751) "a.out" 0x00007ffff773e655 in select() Thread 0x7ffff5ff4700 (LWP 28752) "a.out" 0x00007ffff773e655 in select() |

1. thread ID：切换当前调试的线程为指定ID的线程。

|  |
| --- |
| Bash (gdb) thread 1 [Switching to thread 1 (Thread 0x7ffff7ff6700 (LWP 28750))]*#0 0x00007ffff773e655 in select()* |

1. break thread\_function：在指定的线程函数处设置断点。

|  |
| --- |
| Bash (gdb) break my\_thread\_function thread 1 Breakpoint 1 at 0x40094a: file thread.c, line 12. |

1. set scheduler-locking off|on|step：在多线程调试中，设置调度器锁定。

* off：不锁定任何线程，所有线程都会运行。
* on：只有当前被调试的线程会运行
* step：在单步执行时，只有当前线程会因为单步而运行。

|  |
| --- |
| Bash (gdb) set scheduler-locking off |

1. set follow-fork-mode parent|child：在多线程调试中，设置跟踪模式。

* parent：调试父进程。
* child：调试子进程。

|  |
| --- |
| Bash (gdb) set follow-fork-mode parent |

1. detach：分离被调试的线程。

|  |
| --- |
| Bash (gdb) detach |

1. thread apply ID1 ID2 ... all：在指定的线程上应用命令。

|  |
| --- |
| Bash (gdb) thread apply 1 2 3 bt |

1. thread info：显示线程信息。

|  |
| --- |
| Bash (gdb) thread info  Id Target Id Status Blocked Thread 1 (LWP 28750) 0x00007ffff773e655 (select() at ../sysdeps/unix/syscall-template.S:84)  Thread 2 (LWP 28751) 0x00007ffff773e655 (select() at ../sysdeps/unix/syscall-template.S:84)  Thread 3 (LWP 28752) 0x00007ffff773e655 (select() at ../sysdeps/unix/syscall-template.S:84) |

1. thread switch：在多线程间切换。

|  |
| --- |
| Bash (gdb) thread switch 1 |

1. set scheduler-locking on：在调试多线程程序时，只有当前被调试的线程会运行。

|  |
| --- |
| Bash (gdb) set scheduler-locking on |

这些是GDB多线程调试的常用命令，可以帮助开发者在调试多线程程序时更高

**netstat**

可以查看 IP、TCP、UDP和ICMP协议

查看网络连接数有多少条 netstat -an

**Top**

查看资源消耗

**火焰图工具**

火焰图的宽度代表资源占用，高度代表栈的深度

Perf cpu火焰图

使用jemalloc 可以打印内存火焰图