

Summary of GDB commands for x86-64 Systems

X86-64 系统的 GDB 命令摘要

Command 命令	Effect 影响
Starting: 开始:	
gdb	
Gdb	
gdb <file>	
Gdb < file >	
Running and stopping 跑步 停下来	
quit 退出	Exit gdb 退出 gdb
run 快跑	Run program 运行程序
run 1 2 3 运行 1 23	Run program with command-line arguments 1 2 3 用命令行参数 123 运行程序
kill 杀	Stop the program 停止这个程序
quit 退出	Exit gdb 退出 gdb
Ctrl-d Ctrl-d	Exit gdb 退出 gdb
Note: Ctrl-C does not exit from gdb, but halts the current 注意: Ctrl-C 不会退出 gdb, 但会暂停当前 gdb command Gdb 命令	
Breakpoints 断点	
break sum 断点 总额	Set breakpoint at the entry to function sum 在函数求和条目处设置断点
break *0x80483c3 断点 * 0x80483c3	Set breakpoint at address 0x80483c3 将断点设置为地址 0x80483c3
delete 1 删除 1	Delete breakpoint 1 删除断点 1
disable 1 禁用 1 号	Disable the breakpoint 1 禁用断点 1 (gdb numbers each breakpoint you create) (gdb 为您创建的每个断点编号)
enable 1 启用 1	Enable breakpoint 1 启用断点 1
delete 删除	Delete all breakpoints 删除所有断点
clear sum 清除 总额	Clear any breakpoints at the entry to function sum 清除函数 sum 入口处的所有断点
Execution 处决	
stepi 步骤	Execute one instruction 执行一条指令
stepi 4 步骤 4	Execute four instructions 执行四条指令
nexti 下一步	Like stepi, but proceed 就像 stepi 一样, 但是继续 through function calls without stopping 通过函数调用而不停止
step 台阶	Execute one C statement 执行一条 c 语句
continue 继续	Resume execution until the next breakpoint 继续执行, 直到下一个断点
until 3 直到 3	Continue executing until program hits breakpoint 3 继续执行, 直到程序到达断点 3
finish finish	Resume execution until current function returns

结束了
call sum(1, 2)
调用和(1,2)

Examining code

检查代码

```
disas
Disas
disas sum
Disas 总额
disas 0x80483b7
Disas 0x80483b7
disas 0x80483b7 0x80483c7
Disas 0x80483b70x80483c7
```

```
print /x $rip
打印/x $rip
print /d $rip
打印/d $rip
print /t $rip
打印/撕裂
```

Examining data

检查数据

```
print /d $rax
print /x $rax
print /t $rax
print /d (int)$rax
印刷/d $rax print/x
$rax print/t $rax
print/d (int) $rax
```

继续执行，直到当前函数返回
Call sum(1,2) and print return value
调用 sum (1,2)并打印返回值

Disassemble current function

拆卸电流功能

Disassemble function sum

反汇编函数求和

Disassemble function around 0x80483b7

在 0x80483b7 附近拆卸功能

Disassemble code within specified address range

在指定地址范围内反汇编代码

Print program counter in hex

十六进制打印程序计数器

Print program counter in decimal

十进制打印程序计数器

Print program counter in binary

二进制打印程序计数器

Print contents of %rax in decimal

Print contents of% rax in decimal 以小数为单位打印内容

Print contents of %rax in hex

打印以十六进制表示的% rax 的内容

Print contents of %rax in binary

Print content of% rax in binary 二进制文件

Print contents of %rax in decimal after

打印后以十进制表示的% rax 的内容

sign-extending lower 32-bits.

下 32 位的符号扩展。

You need this to print 32-bit, negative numbers stored in the lower 32 bits of %rax. For example, if the lower 32-bits of %rax store 0xffffffff, you will see

您需要它来打印存储在% rax 下 32 位中的 32 位负数。

例如，如果下面的 32 位% rax 存储 0xffffffff，你会

看到

		(gdb) print \$rax
		(gdb)打印 \$rax
		\$1 = 4294967295
		\$1 = 4294967295
		(gdb) print (int)\$rax
		(gdb) print (int) \$rax
		\$2 = -1
		\$2 = -1
		(gdb)
		(gdb)
print	0x100	Print decimal representation of 0x100
打印	0x100	打印 0x100 的小数表示形式
print	/x 555	Print hex representation of 555
打印	/x555	打印十六进制代表 555
print	/x (\$rsp+8)	Print (contents of %rsp) + 8 in hex
打印	价格: \$rsp + 8	Print (% rsp 的内容) + 8(十六进制)
print	*(int *) 0xbffff890	Print integer at address 0xbffff890
打印	*(int *)0xbffff890	在地址 0xbffff890 打印整数
print	*(int *) (\$rsp+8)	Print integer at address %rsp + 8
打印	*(int *)(\$rsp + 8)	在地址% rsp + 8 处打印整数
print	(char *) 0xbffff890	Examine a string stored at 0xbffff890
打印	(char *)0xbffff890	检查存储在 0xbffff890 中的字符串
x/w	0xbffff890	Examine (4-byte) word starting at address
X/w	0xbffff890	从地址开始检查(4 字节)单词
		0xbffff890
		0xbffff890
x/w	\$rsp	Examine (4-byte) word starting at address in \$rsp
X/w	\$rsp	Examine (4 字节)单词, 从 \$rsp 中的地址开始
x/wd	\$rsp	Examine (4-byte) word starting at address in \$rsp.
X/wd	\$rsp	Examine (4 字节) word 起始于 \$rsp 中的地址。
		Print in decimal
		十进制打印
x/2w	\$rsp	Examine two (4-byte) words starting at address
X/2w	\$rsp	检查从地址开始的两个(4 字节)单词
		in \$rsp
		以 \$rsp 计
x/2wd	\$rsp	Examine two (4-byte) words starting at address
X/2wd	\$rsp	检查从地址开始的两个(4 字节)单词
		in \$rsp. Print in decimal
		以十进制打印
x/g	\$rsp	Examine (8-byte) word starting at address in \$rsp.
X/g	\$rsp	Examine (8 字节)以 \$rsp 中的地址开始的单词。
x/gd	\$rsp	Examine (8-byte) word starting at address in \$rsp.
X/gd	\$rsp	Examine (8 字节)以 \$rsp 中的地址开始的单词。
		Print in decimal
		十进制打印
x/a	\$rsp	Examine address in \$rsp. Print as offset from
X/a	\$rsp	检查 \$rsp 中的地址
		previous global symbol.
		上一个全局符号。
x/s	0xbffff890	Examine a string stored at 0xbffff890
X/s	0xbffff890	检查存储在 0xbffff890 中的字符串
x/20b	sum	Examine first 20 opcode bytes of function sum
X/20b	总额	检查函数和的前 20 个操作码字节
x/10i	sum	Examine first 10 instructions of function sum
X/10i	总额	检查函数和的前 10 个指令

(Note: the format string for the 'x' command has the general form x/[NUM][SIZE][FORMAT] where

(注意: 'x' 命令的格式字符串通常是 x/[NUM][SIZE][FORMAT]

NUM = number of objects to display
 要显示的对象数
 NUM = size of each object (b=byte, h=half-word, w=word,
 SIZE = 每个对象的大小(b = 字节, H = 半字 w = 单词,
 尺寸 = g=giant (quad-word))
 g = 巨人 (四个字)
 FORM (d=decimal, x=hex, o=octal, etc.)
 AT = how to display each object (d = 十进制, x = 十六进制, o = 八进制
 格式 = 如何显示每个对象 等)

If you don't specify SIZE or FORMAT, either a default value, or the last value you specified in a previous 'print' or 'x' command is used.
 如果没有指定 SIZE 或 FORMAT, 则使用默认值, 或者在前面的" print"或" x"命令中指定的最后一个值。

)

Useful information

有用 资料

backtrace	Print the current address and stack backtrace
反向追踪	打印当前地址和堆栈回溯
where	Print the current address and stack backtrace
在哪	打印当前地址和堆栈回溯
里	Print current status of the program)
info program	打印程序的当前状态)
信息 计划	Print functions in program
info functions	程序中的 Print 函数
信息 功能	Print backtrace of the stack)
info stack	打印堆栈的反向跟踪)
信息 堆栈	Print information about the current stack frame
info frame	打印有关当前堆栈帧的信息
信息 框架	Print registers and their contents
info registers	打印寄存器及其内容
信息 登记册	Print status of user-settable breakpoints
info breakpoints	用户可设置断点的打印状态
信息 断点	Print expression EXPR using format FMT
display /FMT EXPR	使用 FMT 格式打印表达式 EXPR
显示/FMT EXPR	every time GDB stops
	每次 GDB 停止
undisplay	Turn off display mode
取消展示	关闭显示模式
help	Get information about gdb
救命	获取关于 gdb 的信息