

gdb调试入门

使用gdb调试需要在gcc编译时加上-g选项，产生调试信息

`gdb + <程序名>` 进入调试，或 `gdb` 直接进入调试环境，但需用 `file` 命令指定调试程序

- 命令总览

命令	含义
<code>file (fil)</code>	<code>file + 程序名</code> ，指定调试程序
<code>run(r)</code>	运行调试程序
<code>quit(r)</code>	退出调试程序
<code>break(b)</code>	设置断点，接<文件名:行号>，或函数名
<code>list(l)</code>	接行号或函数名显示周围源代码
<code>continue(n)</code>	继续执行调试程序
<code>next(n)</code>	单步执行调试程序（不进入函数内部）
<code>step(s)</code>	单步执行调试程序（进入函数内部）
<code>finish(fin)</code>	执行函数内部剩余部分并返回
<code>print (p)</code>	接变量名或函数名打印变量值或函数地址，数组支持索引
<code>info breakpoints(i b)</code>	查看设置了哪些断点
<code>info registers(i r)</code>	查看寄存器信息
<code>info variables(i va)</code>	查看全局或全局静态变量
<code>info locals(i lo)</code>	查看当前局部或局部静态变量
<code>info args(i ar)</code>	查看当前函数参数
<code>delete (d)</code>	接序号删除断点，不接删除所有
<code>help(h)</code>	输入help查询命令用法
<code>kill(k)</code>	终止调试
<code>watch(wa)</code>	接变量名，监控变量值变化，监控时变量必须存在
<code>backtrace(bt)</code>	列出函数调用栈

- 注

1. 执行语句后显示的语句是接下来要执行的，不是当前已经执行的
2. 按下Enter键可以重复上一次指令
3. tab可以补全，空行按下两次tab可查询所有命令
4. 按上下方向键查询历史命令