

龙芯ljtag仿真器手册

龙芯中科技术股份有限公司

September 29, 2024

Contents

1	ljtag介绍	2
2	安装	2
2.1	安装包名字命名说明	2
2.2	linux下安装	2
2.3	windows下安装	2
3	loongson-debugger 执行	4
4	eclipse下用loongson-debugger调试	4
4.1	eclipse cdt里集成的attach和remote配置	4
4.2	eclipse 安装hardware debug插件来调试	6
5	目录结构	8
5.1	配置文件	8
5.1.1	主配置文件	8
5.1.2	configs目录	8
5.1.3	scripts目录	8
5.1.4	bin目录	8
5.1.5	tools目录	8
6	基础命令	8
6.1	set 寄存器具体用法	9
6.1.1	set [[reg cfg csr iocsr] [value]]	9
7	调试命令	9
7.1	设置断点命令	9
8	脚本命令	10
8.1	原生脚本	10
8.2	其他语言支持	10
8.3	loop 命令	10
9	flash烧写命令	11
10	扩展命令	11
11	core相关命令	11
12	linux kernel相关命令	11

1 ljtag介绍

loongarch处理器调试采用ljtag协议。管脚兼容jtag标准。处理器jtag_sel无效时，jtag管脚使用ljtag功能。ljtag调试具体细节参考龙芯架构参考手册卷一的监控点章节介绍。

2 安装

2.1 安装包名字命名说明

安装包名称	执行环境
loongson-debugger-20230723.tar.xz	ubuntu下执行
loongson-debugger-mingw-20230723.tar.xz	windows下执行
loongson-debugger-mips-20230723.tar.xz	黑盒子上执行
loongson-debugger-mips64-20230723.tar.xz	3a3000, 3a4000上执行
loongson-debugger-la64-20230723.tar.xz	3a5000机器上执行

2.2 linux下安装

linux下不需要安装驱动，直接安装包解压即可

2.3 windows下安装

windows下使用zdig-2.x.exe来安装。zadiag能自动找到usb，自动按照驱动。驱动安装完后，直接双击la_dbg_tool_usb.exe执行就可以了。

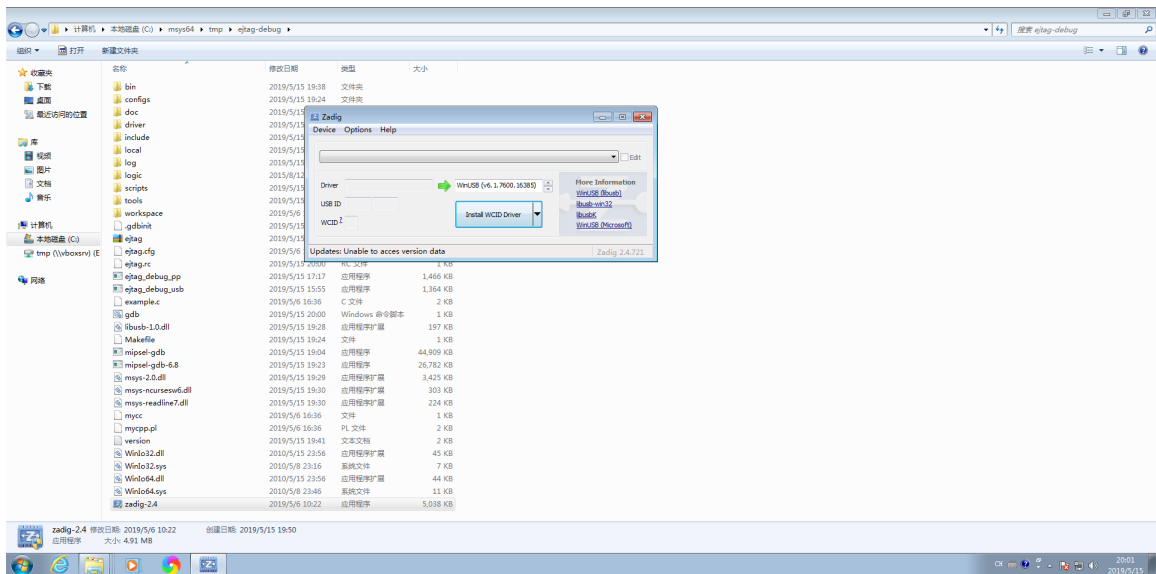


Figure 1: 打开zadiag

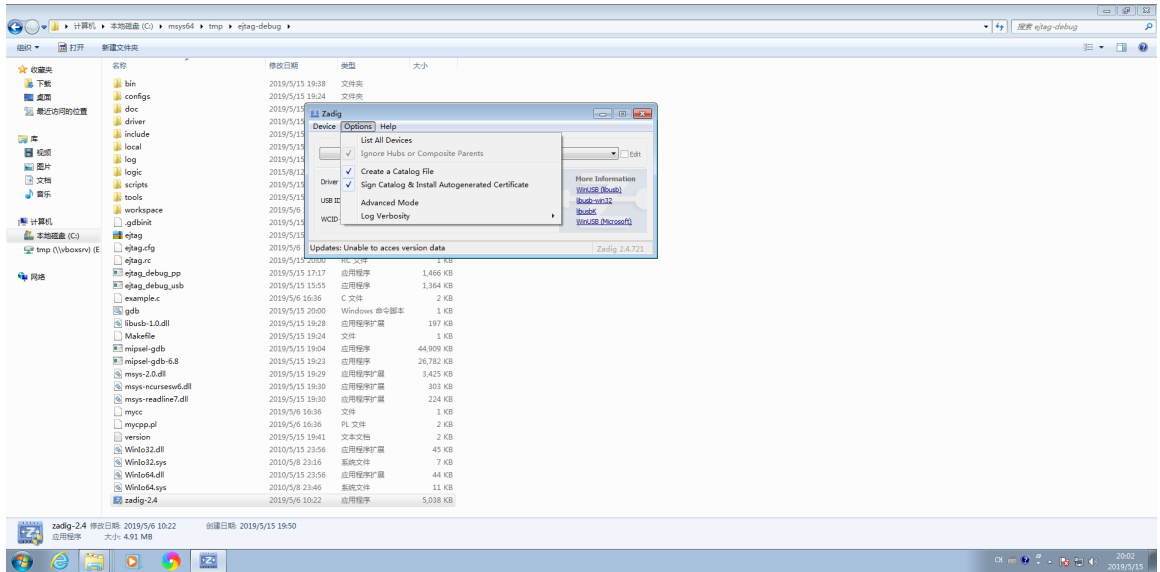


Figure 2: 选择列出设备

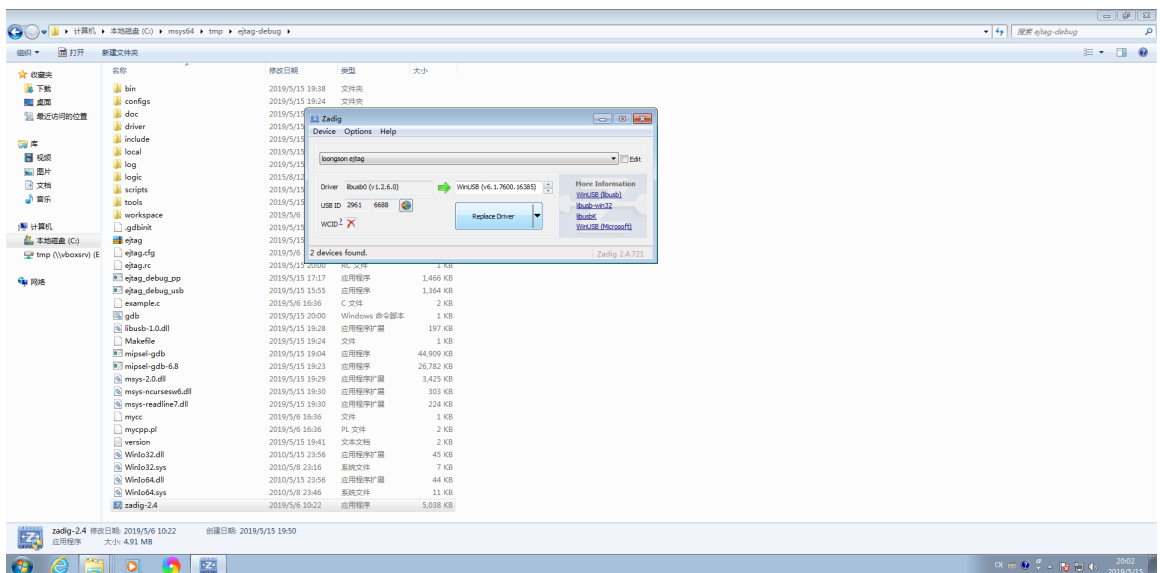


Figure 3: 选择驱动类型:mingw版本选winusb, cygwin版本选libusb, 安装即可

3 loongson-debugger 执行

```
sudo ./la_dbg_tool_usb -t  
source configs/config.ls2k  
cpus
```

Listing 1: 调试器调试龙芯2k开始输入的命令

4 eclipse下用loongson-debugger调试

4.1 eclipse cdt里集成的attach和remote配置

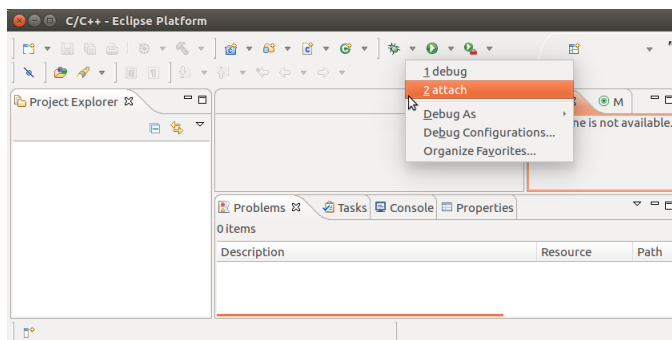


Figure 4: ejtag自动启动eclipse

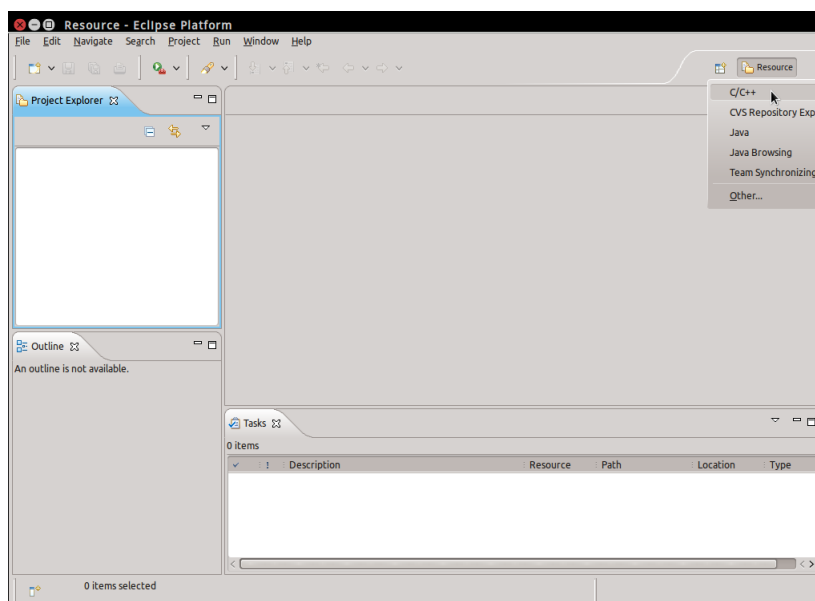


Figure 5: 选择调试c, c++

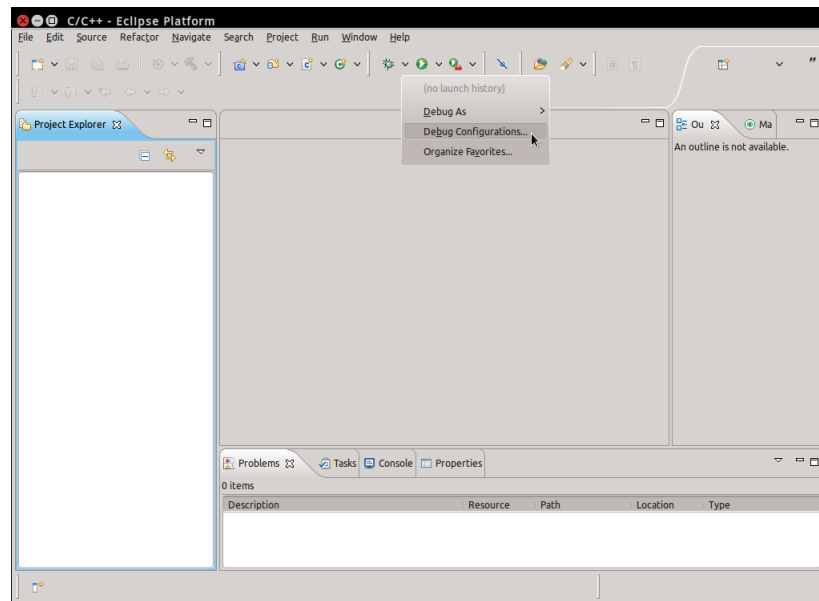


Figure 6: 选择调试c++ debug configure

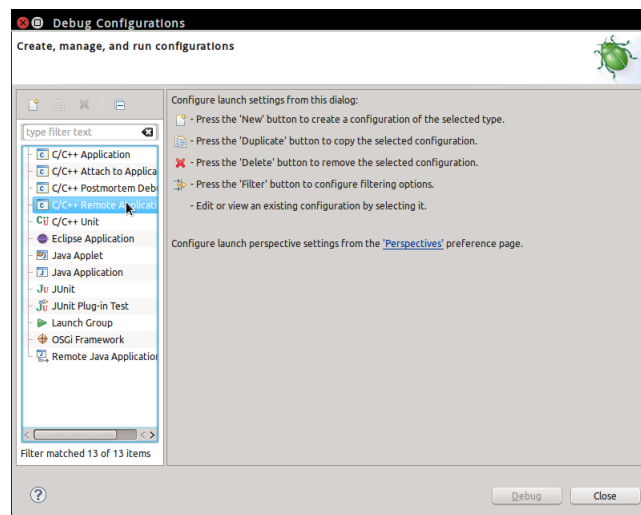


Figure 7: 选择调试c++ remote debug configure

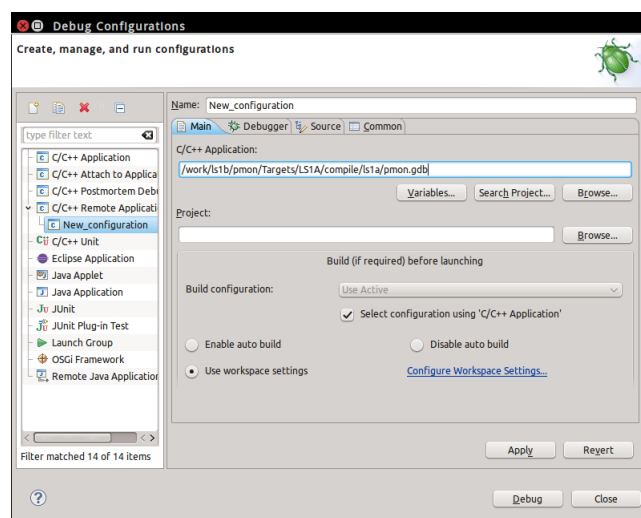


Figure 8: 选择调试c++ debug configure main

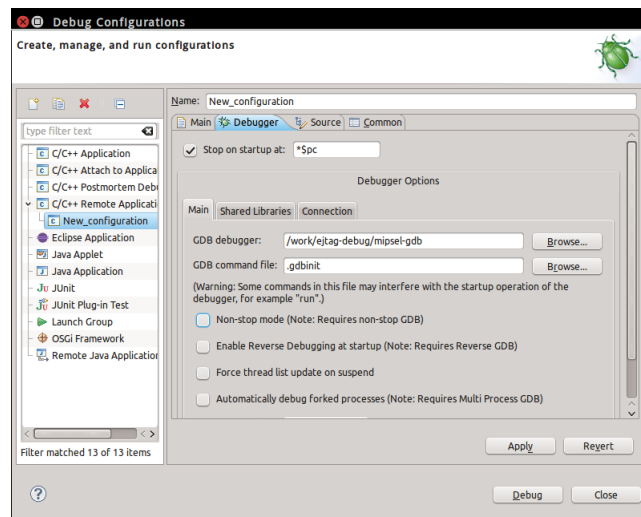


Figure 9: 选择调试c++ debugger为mipsel-gdb
break at 可以设置成*\$pc，这样可以停在调试开始的地方。

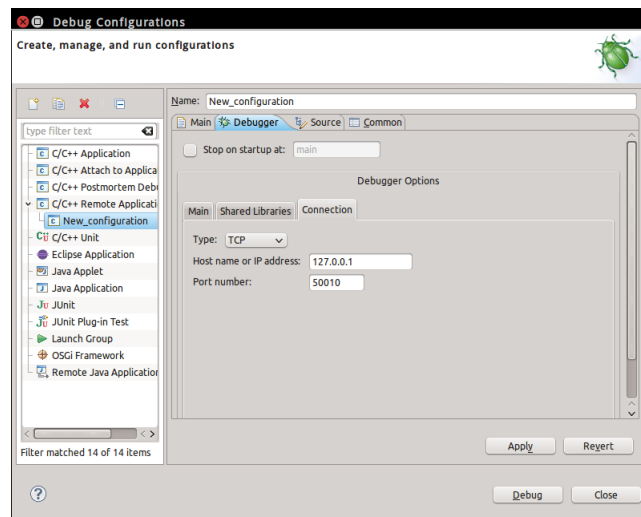


Figure 10: 选择调试c++ debug connection 127.0.0.1:50010

4.2 eclipse 安装hardware debug插件来调试

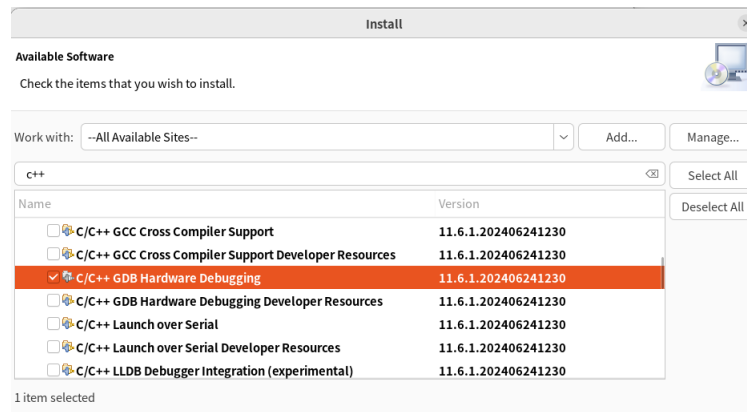


Figure 11: eclipse Help/Install new software安装插件



Figure 12: eclipse hardware debug配置

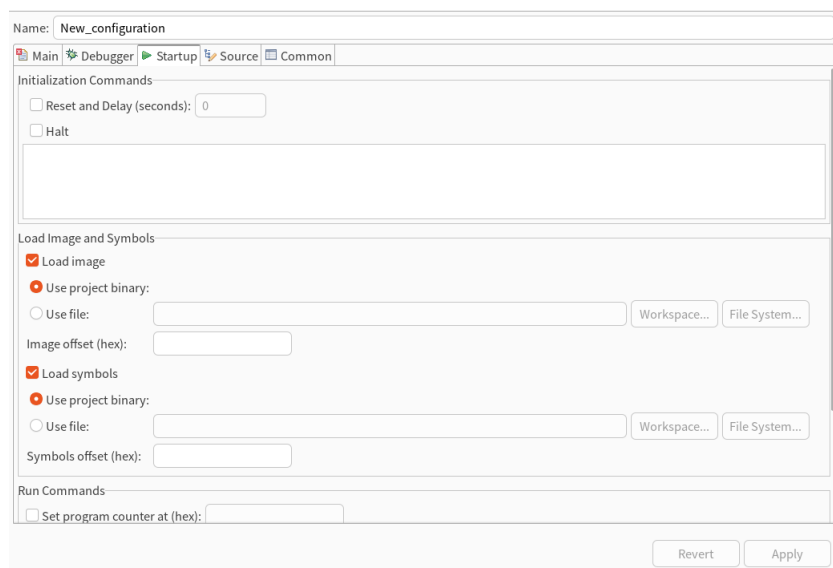


Figure 13: eclipse hardware debug配置

5 目录结构

5.1 配置文件

5.1.1 主配置文件

debugger.cfg是la

5.1.2 configs目录

configs目录下是各种处理器的配置文件

config.ls2k	龙芯2k的配置文件
config.ls3a5000	3a5000的配置文件

5.1.3 scripts目录

scripts目录下是一个方便调试用的脚本

i2c.cmd	
spi.cm	
linux.py	gdb脚本帮助分析linux内核进程中断等
vxworks.py	gdb脚本帮助分析vxworks smp模式
vxwork.py	gdb脚本帮助分析vxwork非smp模式

5.1.4 bin目录

bin目录下是，集成了一个小系统，可以直接调用c库函数，printf会直接在ejtag界面打印信息。
bin目录下labin是loongarch执行的bin文件，mipsbin是mips执行的bin文件

5.1.5 tools目录

tools目录下是一些工具，不常用。

6 基础命令

h	查看帮助
cpus [count[,cpubitmap]]	读当前pc,asid
sample [n]	读当前pc
cpu n	切换到第n个处理器
stop [mask]	停住处理器
go [mask]	处理器继续执行
cont	继续执行当前处理器
set	读写处理器寄存器
dl addr [count]	1字节读addr内存count个
ml addr [val [...]]	1字节写addr内存 val ...
dwait	
Dwait	

6.1 set 寄存器具体用法

6.1.1 set [[reg|cfg|csr|iocsr] [value]]

set	不接参数时候是读通用寄存器数值
set regname	读通用寄存器regname
set cfgn	读cpucfg n,n是16进制, 如set cfg10
set csr n	读csr n,n是16进制, 如set csr6
set iocsrn	读iocsrn,n是16进制, 如set iocsr0
set iocsrn:1	
set iocsrn:2	
set iocsrn:4	
set iocsrn:8	

读通用寄存器数值

7 调试命令

7.1 设置断点命令

b addr [addr2...]	设置软件断点
unb addr [addr2...]	删除软件断点
bls	列出软件断点
watch addr [dbm] [size] [value]	设置数据断点
rwatch :rwatch addr [dbm] [size]	设置读数据断点
awatch addr [dbm] [size]	设置查读写数据断点
Watch addr [dbm] [size] [value]	设置所有核数据断点
unWatch addr	删除所有核数据断点
si	单步
si.s	
si.h	
nextpc	
disas [addr] [count]	
server	
gdbserver	
gdbmap	

8 脚本命令

8.1 原生脚本

命令	说明
<code>expr [%”fmt”] [expression] [...]</code>	
<code>expr1 match string regexp substr pos len</code>	<code>expr cmd in shell</code>
<code>math</code>	<code>shell \$()</code>
<code>test</code>	<code>test cmd in shell</code>
<code>let</code>	
<code>let1</code>	
<code>letg</code>	
<code>setenv</code>	
<code>cat [+startno] [-linecount] files [...]</code>	
<code>try seconds cmd1 [\; cmd2 [\; ...]]</code>	
<code>alarm seconds cmd1 [\; cmd2 [\; ...]]</code>	
<code>loop count ” cmd1;cmd2;...” cmd1 \; cmd2 \; ...</code>	
<code>timer</code>	
<code>newcmd</code>	
<code>renamecmd</code>	
<code>delcmd</code>	
<code>function</code>	
<code>newfunc</code>	
<code>call</code>	
<code>ret</code>	
<code>source file</code>	包含文件到当前位置
<code>. file</code>	包含文件到临时位置, 可以嵌套
<code>label:[f]</code>	定义一个标号
<code>goto label:[b f]</code>	跳转到最近标号[后面, 前面]
<code>dellabel</code>	
<code>dellabels</code>	
<code>renamelabel</code>	

8.2 其他语言支持

debugger利用server和client模式, 类似于busybox, toybox方式

<code>shell python scripts/testpython.py</code>	底层ejtagio.py实现
<code>shell perl scripts/can.pl</code>	底层io.pm实现
<code>shell bash scripts/testshell.sh</code>	<code>\$EJTAGEXE cmd</code> , 如 <code>\$EJTAGEXE set pc</code>

8.3 loop 命令

- `loop count ” cmds”`
- `loop count cmd1 \; cmd2 \; ...`

9 flash烧写命令

put file addr [size [offset]]	上传文件到内存
put64 file addr [size [offset]]	
sput file addr [size [offset]]	
get file addr size	从内存下载文件
get64	
memsetl addr c size	
memcpy1 src dst size	
call program_cachelock_spi	锁cache烧flash
call erase_spi	
call program	

10 扩展命令

callbin binfile args	调用bin文件, args是mymain的参数
scallbin binfile args	
callelf elffile args	
scallelf elffile args	

11 core相关命令

```
. scripts/corefiles.sh
```

12 linux kernel相关命令

python scripts/dumpkallsyms.py vmlinux
python scripts/dumpkallsyms-la.py vmlinux
python scripts/getgz.py vmlinux