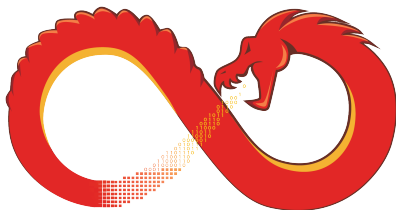


# Open Source Ghidra

## The First Few Months



**GHIDRA**

emteere  
ghidracadabra

Recon MTL 2019

# Outline

Ghidra Overview

New for 9.1: System Call Decompilation

New for 9.1: Sleight Development Tools

Community Interaction



## Ghidra Overview

- Full-featured SRE framework created by NSA Research.
- In development for ~20 years.
- Primarily written in Java.
  - ▶ Some C/C++.
  - ▶ Can write scripts in Python.
- Designed for customizability and extensibility.
- Ghidra 9.0 publicly released March 2019.
- Source code released on Github April 2019.
- [www.ghidra-sre.org](http://www.ghidra-sre.org)
- <https://github.com/NationalSecurityAgency/ghidra>



```

Listing: libc.so.6
libc.so.6 x

*                FUNCTION                *
*****
undefined fmemopen_write()
AL:1          <RETURN>
fmemopen_write

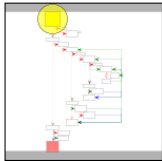
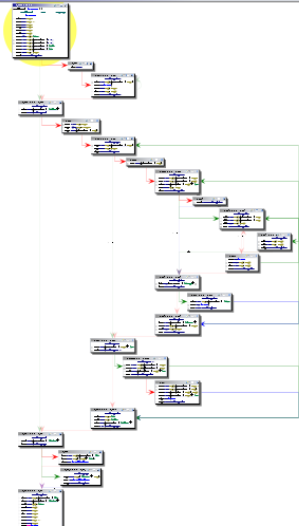
XREF[5]: fme
fme
fme
00:

00177150 41 55      PUSH    R13
00177152 41 54      PUSH    R12
00177154 55        PUSH    RBP
00177155 53        PUSH    RBX
00177156 48 89 d5   MOV     RBP,RDX
00177159 48 89 fb   MOV     RBX,RDI
0017715c 48 83     SUB     RSP,0x8
          ec 08
00177160 8b 4f 0c   MOV     ECX,dword ptr [RDI + 0xc]
00177163 85 c9     TEST   ECX,ECX
00177165 74 69     JZ     LAB_001771d0
00177167 48 8b     MOV     RDI,qword ptr [RDI + 0x20]
          7f 20

```

## Disassembler





# Function Graph



```
C:\Decompile: __fopen_internal - (libc.so.6)
1
2 FILE * __fopen_internal(char *param_1,char *param_2,int param_
3
4 {
5     void *__ptr;
6     long lVar1;
7     FILE *pFVar2;
8
9     __ptr = malloc(0x228);
10    if (__ptr != (void *)0x0) {
11        *(long *)((long)__ptr + 0x88) = (long)__ptr + 0xe0;
12        _IO_no_init(__ptr,0,0,(long)__ptr + 0xf0,_IO_wfile_jumps);
13        *(undefined8 *)((long)__ptr + 0xd8) = 0x4ad400;
14        _IO_new_file_init_internal(__ptr);
15        lVar1 = _IO_new_file_fopen(__ptr,param_1,param_2,(ulong)(u
16        if (lVar1 != 0) {
17            pFVar2 = (FILE *)__fopen_maybe_mmap();
18            return pFVar2;
19        }
20        _IO_un_link(__ptr);
21        free(__ptr);
22    }
```

## Decompiler



```

Listing: libc.so.6
libc.so.6 x
00177154 55      PUSH   RBP                STORE ram(RSP), $U2480
                                $U2480:8 = COPY RBP
                                RSP = INT_SUB RSP, 8:8
                                STORE ram(RSP), $U2480
00177155 53      PUSH   RBX                $U2480:8 = COPY RBX
                                RSP = INT_SUB RSP, 8:8
                                STORE ram(RSP), $U2480
00177156 48 89 d5    MOV    RBP,RDX            RBP = COPY RDX
00177159 48 89 fb    MOV    RBX,RDI            RBX = COPY RDI
0017715c 48 83      SUB    RSP,0x8            CF = INT_LESS RSP, 8:8
                                OF = INT_SBORROW RSP, 8:8
                                RSP = INT_SUB RSP, 8:8
                                SF = INT_SLESS RSP, 0:8
                                ZF = INT_EQUAL RSP, 0:8
                                00177160 8b 4f 0c    MOV    ECX,dword ptr [RDI + 0xc]
                                $U620:8 = INT_ADD RDI, 12:8
                                $U1f50:4 = LOAD ram($U620)

```

## P-code: Ghidra's IR Specified Using SLEIGH Language



The screenshot displays three windows from Immunity Debugger:

- Assembly Window (Top Left):** Shows assembly instructions for the 'free\_mem' function. Key instructions include:
  - 00262248: CMP RBX, RBP
  - 0026224b: MOV RDI, RBX
  - 0026224e: JNZ LAB\_00262240
  - 00262250: CHP qword ptr [0x900 + R13], 0x0
  - 00262258: JZ LAB\_002622ed
  - 00262268: MOV RBP, R13
  - 00262261: XOR R12D, R12D
  - 00262264: NOP dword ptr [RAX]
- Function Graph (Top Right):** Visualizes the control flow between the assembly instructions. A yellow highlight is placed on the instruction at address 00262261 (XOR R12D, R12D).
- Decompile Window (Bottom):** Shows the decompiled C code for 'free\_mem'. The code includes a loop and conditional logic corresponding to the assembly:
 

```

do {
    p1Var1 = (long *)*_ptr_00;
    free(_ptr_00);
    _ptr_00 = p1Var1;
} while (p1Var1 != p1Var4);
if (_DAT_004b6908 != 0) {
    p1Var4 = &_rtld_global;
    p1Var5 = 0;
    do {
        l1Var2 = *p1Var4;
        while (l1Var2 != 0) {
            *p1Var5 = *l1Var2;
            p1Var5 = *p1Var4;
        }
    } while (p1Var5 != 0);
}

```

## Connected Tools





Script Manager - 239 scripts

In T...	Stat...	Name	Description	Key	Category	Modified
<input type="checkbox"/>		CreateEmptyProgramScript.java	Creates an emp...		Program	04/03/2019
<input type="checkbox"/>		CreateExportFileForDLL.java	Causes a ,expor...		Windows	04/03/2019
<input type="checkbox"/>		CreateFunctionAfterTerminals.java	Create a functio...			04/03/2019
<input type="checkbox"/>		CreateFunctionsFromSelection.J...	Create Multiple f...		Functions	04/03/2019
<input type="checkbox"/>		CreateHelpTemplateScript.java	Creates a templ...		HELP	04/03/2019
<input type="checkbox"/>		CreateMultipleLibraries.java	Create multiple l...		FunctionID	04/03/2019
<input type="checkbox"/>		CreateOperandReferencesInSele...	This script creat...		Analysis	04/03/2019
<input type="checkbox"/>		CreatePdbXmlFilesScript.java				04/03/2019
<input type="checkbox"/>		CreatePICSwitch.java	This script work...			04/03/2019
<input type="checkbox"/>		CreateStringScript.java	finds and create...		Memory	04/03/2019
<input type="checkbox"/>		CreateStructure.java	Automatically cr...	F6	Data Types	04/03/2019
<input type="checkbox"/>		DebugSleighInstructionParse.java	Attempt to pars...		sleigh	04/03/2019
<input type="checkbox"/>		Decompile.java	Decompile an e...			04/03/2019
<input type="checkbox"/>		DeleteDeadDefaultPlatesScript.j...	Removes dead ...		Update	04/03/2019
<input type="checkbox"/>		DeleteEmptyPlateCommentsScri...	Removes EMPTY...		Update	04/03/2019
<input type="checkbox"/>		DeleteExitCommentsScript.java	Removes exit po...		Update	04/03/2019
<input type="checkbox"/>		DeleteFunctionDefaultPlates.java	Removes defaul...		Update	04/03/2019
<input type="checkbox"/>		DeleteSpacePropertyScript.java	Removes space ...		Update	04/03/2019
<input type="checkbox"/>		DemangleAllScript.java	Attempts to de...		Symbol	04/03/2019
<input type="checkbox"/>		DemangleElfWithOptionScript.java	An exemplar scri...		Examples->De...	04/03/2019
<input type="checkbox"/>		DemangleSymbolScript.java	Attempts to de...		Symbol	04/03/2019
<input type="checkbox"/>		DemangleSymbolScript.java	Attempts to de...		Symbol	04/03/2019

Filter: | Filter: |

## Scripting in Java and Python



the current byte in current memory block of

been functions

ual b

ed, u

a fu

creat

to the

ram

ngua

tion

on by

all of

lars c

'\n'

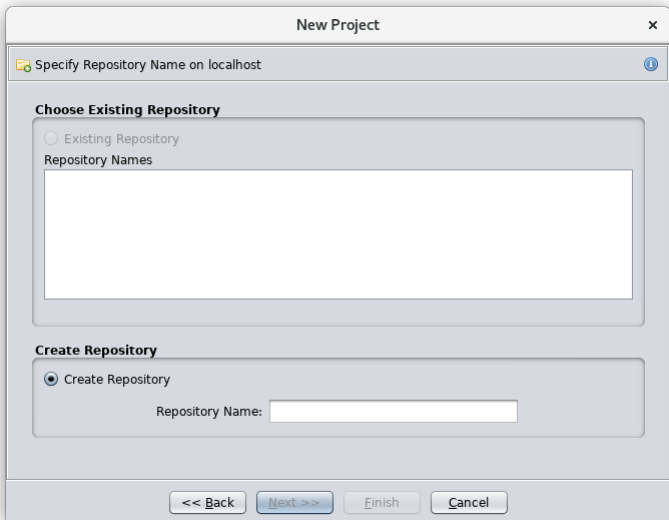
debug information and applies it to the progr

-----

- Assign Key Binding
- Delete
- Edit with Eclipse**
- Edit with basic editor
- Ghidra API Help
- New
- Refresh
- Rename
- Run
- Script Directories
- Copy
- Export

## Eclipse Integration





## Multi-user Server with Version Control



```
support]$ ./analyzeHeadless ghidra://localhost/repo  
-import /usr/bin/* -recursive  
-postScript MyScript.py
```

## Batch Processing with the Headless Analyzer



File Edit Window Help

Version Tracking Markups... [Session: test] - 118290 matches

Tag	Genus	Sub L	Type	Score	Confidenc	Votes	# Con	Multipl	Source Namespace	Source Label	Source A...	# 2	Multipl	Dest Namespace	Dest Label	Dest Address	Source...	Dest L...	Algorithm
2	Function			0.842	1.087	0	20	2	Global	off_swap_shdr_in	0504030e	2	Global	off_swap_shdr_in	00493410	228	313	Similar Symbol Name Mat...	
2	Function			0.844	1.125	0	20	2	Global	off_swap_shdr_out	0504030e	2	Global	off_swap_shdr_out	00493420	228	314	Similar Symbol Name Mat...	
2	Function			0.786	1.087	0	20	2	Global	off_swap_shdr_in	0504030e	2	Global	off_swap_shdr_in	00493410	228	313	Similar Symbol Name Mat...	
2	Function			0.816	1.162	0	20	2	Global	off_swap_shdr_in	0504030e	2	Global	off_swap_shdr_in	00493410	228	228	Similar Symbol Name Mat...	
2	Function			0.842	1.087	0	20	2	Global	off_swap_shdr_in	0504030e	2	Global	off_swap_shdr_in	00493410	228	313	Similar Symbol Name Mat...	
2	Function			0.529	1.048	0	19	2	Global	off_swap_shdr_in	0504030e	2	Global	off_swap_shdr_in	00493410	228	318	Similar Symbol Name Mat...	
2	Function			0.884	1.125	0	20	2	Global	off_swap_shdr_in	0504030e	2	Global	off_swap_shdr_out	00473760	228	175	Similar Symbol Name Mat...	
2	Function			1.000	1.192	0	20	2	Global	off_swap_shdr_in	0504030e	2	Global	off_swap_shdr_in	00493410	228	228	Similar Symbol Name Mat...	
2	Function			0.786	1.087	0	20	2	Global	off_swap_shdr_in	0504030e	2	Global	off_swap_shdr_in	00473950	228	305	Similar Symbol Name Mat...	
2	Function			0.806	1.192	0	20	2	Global	off_swap_shdr_in	0504030e	2	Global	off_swap_shdr_in	0048442a	228	249	Similar Symbol Name Mat...	
2	Function			0.816	1.162	0	20	2	Global	off_swap_shdr_in	0504030e	2	Global	off_swap_shdr_in	0048442a	228	249	Similar Symbol Name Mat...	
1	Function			1.000	1.000	0	0	0	Global	off_create_symlib	0504037a2	0	Global	off_create_symlib	004036a1	392	392	Exact Function Match...	
2	Function			0.811	1.170	0	0	0	Global	off_create_symlib	0504037a2	0	Global	off_create_symlib	004036a1	392	392	Exact Function Match...	
2	Function			1.000	1.000	0	4	4	Global	off_link_adjust_relocs	05040392a	0	Global	off_link_adjust_relocs	00440219	386	386	Exact Function Match...	
2	Function			0.572	1.125	0	8	8	Global	off_link_adjust_relocs	05040392a	0	Global	off_link_adjust_relocs	0048c690	386	386	Similar Symbol Name Mat...	
2	Function			1.000	1.147	0	4	4	Global	off_link_adjust_relocs	05040392a	0	Global	off_link_adjust_relocs	00440219	386	386	Similar Symbol Name Mat...	
2	Function			0.553	0.891	0	7	7	Global	read_slib128	050403ac	0	Global	read_slib128	00457c03	67	67	Similar Symbol Name Mat...	
2	Function			1.000	1.046	0	8	8	Global	read_slib128	050403ac	0	Global	read_slib128	004041ab	67	67	Similar Symbol Name Mat...	
2	Function			0.738	0.891	0	8	8	Global	read_slib128	050403ac	0	Global	read_slib128	004041ab	67	67	Exact Function Match...	
2	Function			0.738	0.891	0	8	8	Global	read_slib128	050403ac	2	Global	read_slib128	004041ab	67	67	Similar Symbol Name Mat...	
2	Function			0.596	0.849	0	7	7	Global	read_slib128	050403ac	0	Global	read_slib128	004041ab	67	67	Similar Symbol Name Mat...	
2	Function			1.000	1.000	0	0	0	Global	read_slib128	050403ac	0	Global	read_slib128	004041ab	67	67	Exact Function Match...	
2	Function			0.738	0.891	0	8	8	Global	read_slib128	050403ac	0	Global	read_slib128	004041ab	67	67	Similar Symbol Name Mat...	
2	Function			0.596	0.849	0	7	7	Global	read_slib128	050403ac	0	Global	read_slib128	004041ab	67	67	Similar Symbol Name Mat...	
2	Function			1.000	1.000	0	0	0	Global	skip_cfa_op	050403aef	0	Global	skip_cfa_op	004041aa	375	372	Similar Symbol Name Mat...	
2	Function			0.823	1.125	0	20	2	Global	off_swap_shdr_in	0504032fe	2	Global	off_swap_shdr_in	00473760	248	175	Similar Symbol Name Mat...	
1	Function			0.818	0.889	0	16	16	Global	off_swap_shdr_in	0504032fe	2	Global	off_swap_shdr_in	00473760	248	175	Similar Symbol Name Mat...	

Filter: Score Filter: 0.000 | 1.000 | Confidence Filter: 0.999 | 1.000 | Length Filter: 0

Version Tracking Markup Items... [Session: test] - 2 markup items

Status	Source Address	Dest Address	Markup Type	Source Value	Current Dest Value	Original Dest Value
	0040392a	0048c690	Function Signature Function Name	void _bfd_elf_link_output_relocs bfd + off...	void _bfd_elf_link_output_relocs	void _bfd_elf_link_output_relocs
	0040392a	0048c690	Function Name	_bfd_elf_link_output_relocs	_bfd_elf_link_output_relocs	_bfd_elf_link_output_relocs

Filter:

Decompile View | Listing View

Source	off_elf_link_adjust_reloc(s) in ipcr_bvlib64_02_0bump	Destination	_bfd_elf_link_output_reloc(s) in ipcr_bvlib64_04_0bump
00403921 41 55 PUSH r11	0048c626 46 89 f5 MOV r11, r10	0048c626 46 89 f5 MOV r11, r10	0048c626 46 89 f5 MOV r11, r10
00403921 41 54 PUSH r10	0048c626 46 89 fc MOV r10, r11	0048c626 46 89 fc MOV r10, r11	0048c626 46 89 fc MOV r10, r11
00403925 33 PUSH r10	0048c626 46 89 fc MOV r10, r11	0048c626 46 89 fc MOV r10, r11	0048c626 46 89 fc MOV r10, r11
00403925 33 PUSH r10	0048c626 46 89 fc MOV r10, r11	0048c626 46 89 fc MOV r10, r11	0048c626 46 89 fc MOV r10, r11
00403925 33 PUSH r10	0048c626 46 89 fc MOV r10, r11	0048c626 46 89 fc MOV r10, r11	0048c626 46 89 fc MOV r10, r11
00403927 44 83 ec SCALEBQ rax, 2	0048c627 46 89 45 MOV rax, r10	0048c627 46 89 45 MOV rax, r10	0048c627 46 89 45 MOV rax, r10
00403928 44 80 47 10 MOV rax, qword ptr [r11 + abfd_vowc]	0048c628 53 PUSH r10	0048c628 53 PUSH r10	0048c628 53 PUSH r10
00403928 44 80 47 10 MOV rax, qword ptr [r11 + reldata_vndr]	0048c628 46 89 4c MOV rax, r10	0048c628 46 89 4c MOV rax, r10	0048c628 46 89 4c MOV rax, r10
00403942 44 80 56 10 MOV rax, qword ptr [r11 + reldata_hashes]	0048c627 46 8b 47 50 MOV rax, qword ptr [r11 + output_bfd_vowc]	0048c627 46 8b 47 50 MOV rax, qword ptr [r11 + output_bfd_vowc]	0048c627 46 8b 47 50 MOV rax, qword ptr [r11 + output_bfd_vowc]
00403942 44 80 56 10 MOV rax, qword ptr [r11 + reldata_hashes]	0048c627 46 8b 47 50 MOV rax, qword ptr [r11 + output_bfd_vowc]	0048c627 46 8b 47 50 MOV rax, qword ptr [r11 + output_bfd_vowc]	0048c627 46 8b 47 50 MOV rax, qword ptr [r11 + output_bfd_vowc]

Version Tracking Markups... | Version Tracking Markup...

# Version Tracking Tool



# New Features for 9.1



## Decompiling System Calls (syscalls)

- **System calls** are a way for a program to request a service from the operating system.
- Services include process control, file management, device management,...
- Typical implementation includes a native instruction and a register, which we'll call the **system call register**.
- When the instruction is executed, the value in the system call register determines which function is called.



```

Listing: libc-2.17.so_x64
libc-2.17.so_x64 x

*                FUNCTION                *
*****
int __stdcall chmod(char * __file, __mo...
int             EAX:4   <RETURN>
char *         RDI:8   __file
__mode_t      ESI:4   __mode

    __chmod
    __GI__chmod
    __GI__chmod
    chmod
XREF[4]: Entry Point(*),
grantpt:0023be06
00291410, 002ac8

001eebd0 b8 5a      MOV     EAX,0x5a
001eebd5 00 00 00
001eebd5 0f 05      SYSCALL
001eebd7 48 3d      CMP     RAX,0xffffffff01
001eebd8 01 f0
001eebd9 ff ff
001eebdd 73 01      JNC    LAB_001eebe0
001eebdf c3        RET

LAB_001eebe0
XREF[1]: 001eebdd(j)
= 00000000
001eebe0 48 8b      MOV     RCX,qword ptr [PTR_004c5e70]
001eebe1 0d 89
001eebe2 72 2d 00
001eebe7 f7 d8      NEG     EAX
001eebe9 64 89 01   MOV     dword ptr FS:[RCX],EAX
001eebec 48 83      OR     RAX,-0x1
001eebed c8 ff
001eebf0 c3        RET

```

## x64 Linux syscall





```
Decompile: chmod - (libc-2.17.so_x64)
1
2 /* WARNING: Removing unreachable block (ram,0x001eebe0) */
3
4 int chmod(char *__file, __mode_t __mode)
5
6 {
7     syscall_inject();
8     return 0x5a;
9 }
10
```

## System Calls as User-defined Operations

- In this example, the `syscall` instruction implemented with a `pcodeop/CALLOTHER`
- Such operators certainly have their uses, but not very satisfying in this case.



## Desired Behavior

- We'd like to see the correct function call in the decompiler:
  - ▶ Correct name.
  - ▶ Correct signature.
  - ▶ Correct calling convention.
- We'd also like to get cross-references



- Need dataflow analysis to determine value in syscall register.

```

Listing: libc-2.17.so
libc-2.17.so x
001fc574 01 92 c2 SLTQ    DL
001fc577 29 d0    SUB    EAX,EDX
001fc579 0f be c0 MOVXSX EAX,AL
001fc57c 83 f8 01 CMP    EAX,0x1
001fc57f 83 d3 00 ADC    EBX,0x0

                                LAB_001fc582                                XREF[1]: 0
001fc582 48 8b    MOV    RSI,qword ptr [RBP + -0x68]
                                75 98
001fc586 4d 89 e0 MOV    R8,R12
001fc589 4c 89 f1 MOV    RCX,R14
001fc58c 4c 89 ea MOV    RDX,R13
001fc58f 44 89 ff MOV    EDI,R15D
001fc592 e8 d9    CALL  next_line
                                fa ff ff
001fc597 48 85 c0 TEST   RAX,RAX
001fc59a 75 c4    JNZ   LAB_001fc560
001fc59c 49 63 ff MOVXSQ RDI,R15D
001fc59f b0 03    MOV    AL,0x3
001fc5a1 0f 05    SYSCALL
001fc5a3 e9 9b    JMP   LAB_001fc443
                                fe ff ff
001fc5a8 0f    ??    0Fh
001fc5a9 1f    ??    1Fh
001fc5aa 84    ??    84h
001fc5ab 00    ??    00h

```



- Value in syscall register is not necessarily the syscall number defined in system header file.

```
Listing: x86-64-cpu0x3
libe-2.17.so  x86-64-cpu0x3
***** FUNCTION *****
                ssize_t __stdcall _write(int param_1, v...
                ssize_t  RAX:8    <RETURN>
                int      EDI:4    param_1
                void *   RSI:8    param_2
                size_t   RDX:8    param_3
                _write                                     XREF[3]: E
0001e6f0 b8 04    MOV     EAX,0x2000004
                00 00 02
0001e6f5 49 89 ca  MOV     R10,RCX
0001e6f8 0f 05    SYSCALL
0001e6fa 73 08    JNC    LAB_0001e704
0001e6fc 48 89 c7  MOV     param_1,RAX
0001e6ff e9 0a    JMP     _cerror
                54 ff ff
                -- Flow Override: CALL_RETURN (CALL_TERMIN...
                LAB_0001e704                                XREF[1]: 0
0001e704 c3      RET
0001e705 90      ??     90h
0001e706 90      ??     90h
0001e707 90      ??     90h
```



## Additional Issues

- The system call register can be an OS decision — not necessarily specified by ISA.
- System call numbers can change based on the OS version/service pack.
- System calls might have their own calling convention.
- There can be more than one native instruction used to make system calls (e.g., `syscall` and `int 2e`).
- Might not use a dedicated native system call instruction, e.g., system calls via `CALL GS: [0x10]`.



## Where to Put Them?

- In general, the code for system call targets is not in the program's address space.
- Where to put them in Ghidra?
- The OTHER space is used to store data from a binary that is not loaded into memory.
  - ▶ E.g., the `.comment` section of an ELF file.
- In 9.1, we've made the decompiler aware of the OTHER space.
- Recommendation for system calls:
  - ▶ System call target should be in overlay(s) of the OTHER space.
  - ▶ Use the system call number as the address in the overlay.



## How to Get There?

- OK, great, we have a place for system call targets.
- How do you get there?
- New feature: **Overriding References**.
- Basically, this allows you to intercept certain Pcode ops on their way to the decompiler and modify them.
  - ▶ Change CALLOTHER ops to CALL ops and set destination.
  - ▶ Change CALLIND to CALL ops and set destination.
  - ▶ (plus a few others)
- See `ResolveX86or64LinuxSystemCallsScript.java` for an example.



```

Listing: libc-2.17.so_x64
*libc-2.17.so_x64 x

                                *
                                FUNCTION
                                *
*****
int __stdcall chmod(char * __file, __mo...
char *
__mode_t

                                EAX:4  <RETURN>
                                RDI:8  __file
                                ESI:4  __mode
                                __chmod
                                __GI__chmod
                                __GI__chmod
                                chmod
                                XREF[4]: Entry Point(*),
                                grantpt:0023be00
                                00291410, 002ac8

001eebd0 b8 5a          MOV     EAX,0x5a
                                00 00 00

001eebd5 0f 05          SYSCALL
                                -- CALLOTHER(syscall inject) Call Override...
                                long chmod(char

001eebd7 48 3d          CMP     RAX,0xfffff001
                                01 f0
                                ff ff

001eebdd 73 01          JNC    LAB_001eebe0
001eebdf c3            RET

                                LAB_001eebe0
                                XREF[1]: 001eebdd(j)
                                = 00000000
001eebe0 48 8b          MOV     RCX,qword ptr [PTR_004c5e70]
                                0d 89
                                72 2d 00

001eebe7 f7 d8          NEG     EAX
001eebe9 64 89 01      MOV     dword ptr FS:[RCX],EAX
001eebec 48 83          OR     RAX,-0x1
                                c8 ff

```

## x64 Linux syscall with Overriding Reference





```

Listing: libc-2.17.so_x64
*libc-2.17.so_x86  *libc-2.17.so_x64 x

char *      RDI:8    __path
char *      RSI:8    __buf
size_t      RDX:8    __len
sys_readlink                                XREF[2]: readlink:001f06b5(c)
                                              __readlink_chk:00216

syscall::00000059    ??    ??

*****
*                               *
*                               *
*****
FUNCTION
*****
long syscall sys_chmod(char * __file, ...
long          RAX:8    <RETURN>
char *        RDI:8    __file
__mode_t      ESI:4    __mode
sys_chmod                                           XREF[1]: chmod:001eebd5(c)
syscall::0000005a    ??    ??

*****
*                               *
*                               *
*****
FUNCTION
*****
long syscall sys_fchmod(int __fd, __mod...
long          RAX:8    <RETURN>
int           EDI:4    __fd
__mode_t      ESI:4    __mode
sys_fchmod                                           XREF[1]: fchmod:001eec05(c)
syscall::0000005b    ??    ??

```

## Functions in an Overlay of the OTHER Space



```
C: Decompile: chmod - (libc-2.17.so_x64)
1
2 int chmod(char *__file, __mode_t __mode)
3
4 {
5     ulong uVar1;
6     int *in_FS_OFFSET;
7
8     uVar1 = sys_chmod(__file, __mode);
9     if (uVar1 < 0xfffffffffff001) {
10         return (int)uVar1;
11     }
12     *in_FS_OFFSET = -(int)uVar1;
13     return -1;
14 }
15
```

## x64 Linux syscall Decompile Ghidra 9.1 (after running script)



## Future Work

- We'd like an analyzer to be able to do this (mostly) automatically.
- Ghidra has a notion of per-processor configuration (.pspec files) and per-compiler configuration (.cspec files).
- System call data doesn't quite fit this model.
- Ideally all the system call related configuration would be in one place.
- Working on a notion of an OS/environment configuration.
- This will have other applications in Ghidra as well.



## Sleigh Development Tools

- Sleigh
- SleighEditor
- Sleigh P-Code Tests
- Additional Techniques
- General Sleigh Development



# Sleigh Processor Models

- Memory model
- Registers
- Display (printpiece)
- Decode patterns
- Semantics (Pcode)

```
define alignment=2;

define space ram      type=ram_space      size=2;
define space register type=register_space size=2;
define space rom      type=ram_space      size=3 wordsize=2 default;

#TOKENS
define token opbyte (8)
{
  imm8 = (0,7)
  oplo = (0,3)
  ophi = (4,7)
  Areg = (0,2)
  rn   = (0,2)
};

# Registers
define register offset=0xFF00 size=8 [ C 0V Z X A ];
define register offset=0xFF00 size=4 [ d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12 d13 d14 d15 ];

attach variables [rn] [d0 d1 d2 d3 d4 d5 d6 d7];

# macros
macro addflags(op1, op2) { C = carry(op1,op2); }
macro resultflags(op1, op2) { Z = op1 == 0; }

# Sub constructors - addressing modes
OP1: *#imm8 is rn=2; imm8 { tmp:=1 & imm8; export tmp; }
OP1: (imm8,X) is rn=0 & X; imm8 { addr:=2 & zext(imm8 + X); tmp:=2 & *:2 addr; export *:=1 tmp; }

Rel8: relAddr is imm8 { relAddr:=inst_next+imm8; } { export *:=1 relAddr; }

# Base constructors
:ADD A,rn is ophi=2 & A & rn { addflags(A,rn); A = A + rn; resultflags(A); }
:JZ Rel8 is ophi=6 & oplo=0; Rel8 { if (A == 0) goto Rel8; }
```

**Build it and the tools just work**

*Disassembly, Assembler(patch), Decompiler, Analysis...*



## Sleigh Processors

- Currently Included - evolving list

*X86 16/32/64, ARM/AARCH64, PowerPC 32/64/VLE, MIPS 16/32/64/micro 68xxx, Java / DEX bytecode, PA-RISC, PIC 12/16/17/18/24, Sparc 32/64 CR16C, Z80, 6502, 8051, MSP430, AVR8, AVR32, and variants.*

- Full Processor Contributions

*Tricore, MCS-48*

- Extensions, Improvements, and Bugs

*ARM, PPC, 68xxx, AVR, PIC-16F, PPC, 6502, golang*

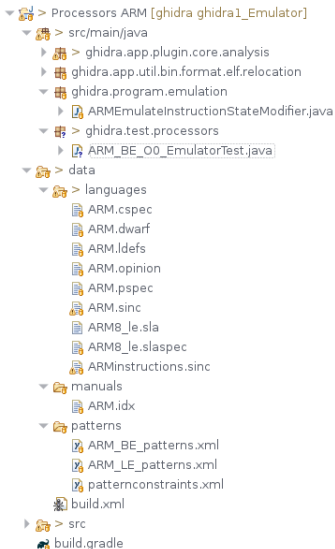
- Seen in Development

*SH-2, WebAssembly, Hexagon, Toshiba MeP-c4, Pic16F153xx, Arm4t-gba, NVIDIA Falcon, PowerPC 750CL/CXe, WDC-65816, RISC-V, TI TMS9900*



# Sleigh Files

- LDEF
- PSPEC
- CSPEC
- SLASPEC
- SLA
  
- Java Files
- Manual Index
- Pattern Files
  
- Emulator<sup>new</sup>
- Sleigh P-Code Tests<sup>new</sup>



# Sleigh Editor

- Syntax Coloring
- Hover
- Navigation
- Code Formatting
- Validation
- Quick Fixes
- Renaming
- Find References
- Content Assist
- Sleigh Compiler
- Error Navigation

```
# Base constructors
:ADD A,rn is ophi=2 & A & rn { addflags(A,rn); A = A + rn; resultflags(A); }
:J2 RetB is ophi=6 & oplo= held m 4 (6,2) attached to: do d1 d2 d3 d4 d5 d6 d7
```

```
26 # Sub constructors - addressing modes
27 OP1: "#imm8 is rn=2; imm8 {
28 Multiple markers at this line      tmp = imm8; export tmp;
29 - Couldn't resolve reference to exportvarmode 'tmp'.
   - Couldn't resolve reference to lhsvarmode 'tmp'.
```

```
587: cp RdFull,RrFull is phase=1 & ophi6=0x05 & RdFull & RrFull {
588: local x = RdFull - RrFull;
589: setSubCary(RdFull, RrFull);
590: setVfLagForSub(RdFull, RrFull);
591: setResultFlags(x); RrHi is RrHiLowSel=1 & RrHi
592: # but doesn't set RrHiLowSel=0 & RrLow
593: }
594: -----
```

Problems Console Search

Xtext References to RrFull (Processors:Atmel\data\languages\avr8.sinc)

- /Processors Atmel\data\languages\avr8.sinc
  - b2 = zero(RrFull);
  - doSubtract(RdFull,RrFull,RdFull);
  - doSubtractWithCary(RdFull,RrFull,RdFull);
  - doSubtractWithCary(RdFull,RrFull,res);
  - local res = RdFull + RrFull;
  - local res = RdFull + RrFull + \$(Cflag);
  - local x = RdFull - RrFull;
  - phase=1 & ophi6=0x05 & RdFull & RrFull
  - phase=1 & ophi6=0x1 & RdFull & RrFull
  - phase=1 & ophi6=0x2 & RdFull & RrFull

*Xtext - DSL Framework for Eclipse*

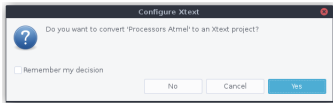
*Eclipse IDE for Java and DSL Developers - 2019-03*





## Setting up Sleigh Editor - Xtext project

- Eclipse Help: Install New Software
  - ▶ Add Archive: GhidraSleighEditor.zip
- Convert GhidraScript to Xtext project



- ▶ Allows for multi-file navigation
  - ▶ Good for casual browsing
  - ▶ Problem: all variables will be available (6502, PPC)  
quick-fixes will be slower
- Best: Import as new Java Project - Ghidra/Processors/6502
  - Large Sleigh projects can be slow - AARCH64 - 85K LOC
  - Use separate Eclipse



# Sleigh Editor

## Quick Demo

After Edit - ReloadSleigh Script

Only works for some changes

No Structural changes - register, memory, pcodeop, . . .



## Sleigh Editor - Future Features

- Better project integration
- Code-Mining - auto-comment
- Navigation from Ghidra to SleighEditor in Eclipse
- Templates of common idioms
- More Hovers
- Conversion of number to different formats
- Syntax coloring in the printpiece
- Refactoring:  
Extract common patterns to sub-constructor
- Instruction Pattern Match Documentation



## Sleigh P-Code Tests - Sleigh Testing Framework

- C code compiled for processor
- Small tests with known result
- General coverage of instructions emitted by C compilers
- Verifies core constructs - Addressing Modes, Registers
- Pcode Emulator to Execute and Verify
- Repeatable - regression testing
- Extendable - needs more cowbell
- Special case code - Assembly



# Sleigh P-Code Tests - Tricore in Eclipse

```

23
24 public class TRICORE_BE_00_EmulatorTest extends ProcessorEmulatorTestAdapter {
25     private static final String LANGUAGE_ID = "tricore:LE:32:default";
26     private static final String COMPILER_SPEC_ID = "default";
27     private static final String[] REG_DUMP_SEF = new String[] {};
28
29     public TRICORE_BE_00_EmulatorTest(String name) throws Exception {
30         super(name, LANGUAGE_ID, COMPILER_SPEC_ID, REG_DUMP_SEF);
31     }
32
33     protected String getProcessorDesignator() { return "tricore_GCC_00"; }
34
35     protected void initializeState(EmulatorTestRunner testRunner, Program program) throws
36         testRunner.setRegister("A1D", 0x40000000L); // stack, unused location
37         testRunner.setRegister("PCX", 0x00020000L); // free context list start, unused
38         testRunner.setRegister("PCY", 0x00030000L); // free context list max
39         testRunner.setRegister("PCZ", 0x0L); // current thread context list
40
41
42     public static Test suite() {
43         return ProcessorEmulatorTestAdapter.builder(EmulatorTestSuite.TRICORE_BE_00_Emulo
44     }
45
46     protected void setAnalysisOptions(Options analysisOptions) {
47         super.setAnalysisOptions(analysisOptions);
48         analysisOptions.setBoolean("Reference", false); // too many bad disassemblies
49     }
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
    
```

## CUnit Test Results

Pass/Fail/CallOther	DATE/TIME	SUMMARY	BIOPS						None/Infinite
			biop4d11	biop4d22	biop4d44	biop4d11ut	biop4d11uc		
AARCH64_BE_00_EmulatorTest	06/15/2019 13:21	1553/+/	4/+/	0/+/	7/+/	4/+/	3/+/	0/+	
ARM_BE_00_EmulatorTest	06/15/2019 13:22	1117/+/	4/+/	0/+/	7/+/	4/+/	3/+/	0/+	
MIPS16_00_EmulatorTest	06/15/2019 13:24	1117/+/	4/+/	0/+/	7/+/	4/+/	3/+/	0/+	
MSP430X_00_EmulatorTest	06/15/2019 13:26	10792/4/ ERR: 1	4/+/	0/+/	7/+/	4/+/	3/+/	0/+	
TRICORE_BE_00_EmulatorTest	06/15/2019 13:38	10617/+/	4/+/	0/+/	7/+/	4/+/	3/+/	0/+	
TRICORE_BE_03_EmulatorTest	06/09/2019 20:09	10965/+/	4/+/	0/+/	7/+/	4/+/	3/+/	0/+	

## State of All Processors

All Passing



# Sleigh P-Code Tests - Example - Tricore

```
public class TRICOREmulateInstructionStateModifier extends EmulateInstructionStateModifier
Register FCX,PCXI,LCX,PSW,a10,a11,d8,a12,d12;

public TRICOREmulateInstructionStateModifier(Emulate emu) {
    super(emu);

    registerPcodeOpBehavior("saveCallerState", new tricore_SaveCallerState());
    registerPcodeOpBehavior("restoreCallerState", new tricore_RestoreCallerState());
    cacheRegisters(emu);
}

// Save Caller State, could be done in Pcode
//
private class tricore_SaveCallerState implements OpBehaviorOther {
    public void evaluate(Emulate emu, Varnode outputVarnode, Varnode[] inputs) {
        int numArgs = inputs.length - 1;
        if (numArgs != 3) throw new LowLevelError(this.getClass().getName() + ": require
MemoryState memoryState = emu.getMemoryState();

        BigInteger FCXvalue = memoryState.getBigInteger(FCX);
        // read the value at FCX, if get nothing, then assume just increment the FCX to
        long ea = FCXvalue.longValue();
        ea = ((ea & 0xffff0000) << 12) | ((ea & 0xffff) << 6);

        Address EA_addr = emu.getExecuteAddress().getNewAddress(ea);
        AddressSpace addressSpace = emu.getExecuteAddress().getAddressSpace();

        // new_FCX = H(EA, word);
        BigInteger new_FCXvalue = memoryState.getBigInteger(addressSpace, ea, 4, false)
        // if new_FCX == 0, or not-initialized, then just increment FCX again
        if (new_FCXvalue.equals(BigInteger.ZERO)) {
            new_FCXvalue = FCXvalue.add(BigInteger.ONE);
        }

        // H(EA, 16 * word) = {PCXI, PSW, A[10], A[11], D[8], D[9], D[10], D[11], A[12]}
        byte[] outBytes = new byte[4*16];
        int index = 0;
        index += copyRegisterToArray(PCXI, PCXI.getBitLength()/8, memoryState, outBytes,
index += copyRegisterToArray(PSW, PSW.getBitLength()/8, memoryState, outBytes,
index += copyRegisterToArray(a10, 2 * a10.getBitLength()/8, memoryState, outByte
index += copyRegisterToArray(d8, 4 * d8.getBitLength()/8, memoryState, outBytes
index += copyRegisterToArray(a12, 4 * a12.getBitLength()/8, memoryState, outByte
index += copyRegisterToArray(d12, 4 * d12.getBitLength()/8, memoryState, outByte
```

- Contribution - mumbel
- Surprisingly well written
- Call Context Save/Restore
- TRICORE\_O0\_EmulatorTest
- EmulateInstructionStateModifier



# Sleigh P-Code Tests - Debugging Sleight

```
Runs: 1/1 Errors: 0 Failures: 1
test_ParameterPassing Failure Trace
junit.framework.AssertionFailedError: ERROR One or more group tests failed ( Passed: 35 Failed: 4
at ghidra.test.processors.support.ProcessorEmulatorTestAdapter.runTest(ProcessorEmulatorTestAd
at ghidra.test.processors.support.ProcessorEmulatorTestAdapter.runTest(ProcessorEmulatorTest

tricorn.arc:22
8463
8464 # SHAS D[c], D[a], D[b] (RR)
8465: #shas Rd2831,Rd0811,Rd1215 is PCPMode=0 & Rd08
8466={
8467= local shift_count:4 = sext(Rd1215[0,5]);
8468 shift_count = (shift_count << (32 - 5)) s>> (32 -
8469 local res:4 = Rd0811;
8470 local shift_dir:1 = shift_count s< 0;
8471 res = (Rd0811 << shift_count) * zext(shift_dir ==
8472 overflowFlags(res);
8473 ssov(Rd2831, res, 32);
8474 }
8475
8476 # SHAS D[c], D[a], const9 (RC)
8477: #shas Rd2831,Rd0811,const12205 is PCPMode=0 & ( Rd
8478={
8479= local shift_count:4 = sext(const12205[0,6]);
8480 shift_count = (shift_count << (32 - 6)) s>> (32 -
8481 local res:4 = Rd0811;
8482 local shift_dir:1 = shift_count s< 0;
8483 res = (Rd0811 << shift_count) * zext(shift_dir ==
8484 overflowFlags(res);
8485 ssov(Rd2831, res, 32);
8486
8487 # local res:4 = Rd0811;
8488 # if (shift_count s> 0) goto <shift left>;
R4R9

Write PC=0x80001604
>> ram:80001604 shas d3,d2,d3
Read d3=0xffffffff
Write unique:000106f0:1=0xe1
Read unique:000106f0:1=0xe1
Write unique:00010700:1=0x01
Read unique:00010700:1=0x01
Write unique:00010720:4=0x00000001
Read unique:00010720:4=0x00000001
Write unique:00010730:4=0x0000001b
Read unique:00010740:4=0x08000000
Write unique:00010750:4=0x0000001b
Read unique:00010740:4=0x08000000
Write unique:00010750:4=0x0000001b
Read unique:00010740:4=0x08000000
Write unique:00010750:4=0x0000001b
Read unique:00010720:4=0x00000001
Read d2=0xffffffff9
Write unique:00010770:4=0xffffffff9
Read unique:00010720:4=0x00000001
Write unique:00010790:1=0x00
Read d2=0xffffffff9
Write unique:00010720:4=0x00000001
Write unique:000107a0:4=0xfffffffff2
Read unique:00010790:1=0x00
Write unique:000107b0:1=0x01
Read unique:000107b0:1=0x01
Write unique:000107c0:4=0x00000001
Read unique:000107a0:4=0xfffffffff2
Read unique:000107c0:4=0x00000001
Write unique:000107d0:4=0xfffffffff2
Read unique:00010720:4=0x00000001
Write unique:000107e0:4=0xfffffffff2
Read d2=0xffffffff9
Read unique:000107e0:4=0xfffffffff2
Write unique:000107f0:4=0xfffffffff2
```

**Debug One Failing test - lots of output**  
**Directory - test-output / cache, logs, results**







# InstructionInfo - Locating problems

InstructionInfo: Address 00400000 - (AARCH64\_BE\_GCC\_00\_cunitest.out)

Instruction Summary		
Mnemonic	: str	
Number of Operands	: 2	
Address	: ran:00400000	
Flow Type	: FALL_THROUGH	
Fallthrough	: 00400004	
Delay slot depth	: 0	
Hash	: af509b68	
Input Objects	: sp, x30, const:-0xb0	
Result Objects	: sp	
Constructor Line #'s:	Instruction(3830), str(5651), addrIndexed(2251), Rn_GPR64xsp(1910), Rt_GPR64(1906)	
Byte Length	: 4	
Instr Bytes	: 11111110 00001111 00010101 11111000	
Mask	: 11100000 00000000 00000000 11111111	
Masked Bytes	: 11100000 00000000 00000000 11111000	
Instr Context:	[Instruction context has not been set]	

	Operand-1	Operand-2
Operand	x30	[sp, #-0xb0]!
Labeled	x30	[sp, #-0xb0]!
Type	REG	REG
Scalar		
Address		
Register	x30	sp, const:-0xb0
Op-Objects	x30	sp, const:-0xb0
Operand Mask	00011111 00000000 00000000 00000000	00000000 00011111 11110000 00000000
Masked Value	00011110 00000000 00000000 00000000	00000000 00001111 00010000 00000000

```
2249  
2250 # pre indexed wback  
2251 # addrIndexed: "["Rn_GPR64xsp, "#^sim9^"]!"  
2252 is size.ldstr & b_2729=7 & b_2425=0 & b_2121=0 & Rn_GPR64xsp & sim9 & opc.indexmode=3  
2253 {  
2254     Rn_GPR64xsp = Rn_GPR64xsp + sim9;  
2255     export Rn_GPR64xsp;  
2256 }  
2257
```



## External Disassembly Field

- *binutils* wrapper *gdis*
  - ▶ Acts as a server
- Other Disassemblers
  - ▶ dump/scrape
  - ▶ code composer studio
- Verify, Debug, Mine

Mnemonic	Operands	PCode	External Disassembly
	Space	Post-Comment	
MOV	ECX,0x2f87428e	mov	ecx,0x2f87428e
MOV	DL,byte ptr [RBP + -0x21]	mov	dl,byte ptr [rbp-0x21]
TEST	DL,0x1	test	dl,0x1
CMOVNZ	EAX,ECX	cmovne	eax,ecx
MOV	dword ptr [RBP + -0x4c],EAX	mov	DWORD PTR [rbp-0x4c],eax
JMP	LAB_00401ac8	jmp	0x0000000000401ac8
LEA	RDX,[RBP + -0x40]	lea	rdx,[rbp-0x40]
MOV	RDI,qword ptr [RBP + -0x48]	mov	rdi,QWORD PTR [rbp-0x48]
MOV	RAX,qword ptr [RBP + -0x48]	mov	rax,QWORD PTR [rbp-0x48]
MOV	qword ptr [-0x128 + RBP],RDI=...	mov	QWORD PTR [rbp-0x128],rdi
MOV	RDI=>s_No_bruteforce_bro!!!_0...	mov	rdi,rax
MOV	qword ptr [-0x130 + RBP],RDX	mov	QWORD PTR [rbp-0x130],rdx
CALL	strlen	call	0x00000000004006c0
MOV	RDI=>s_No_bruteforce_bro!!!_0...	mov	rdi,QWORD PTR [rbp-0x128]
MOV	RSI,RAX	mov	rsi,rax
MOV	RDX,qword ptr [-0x130 + RBP]	mov	rdx,QWORD PTR [rbp-0x130]
CALL	SHA1	call	0x00000000004006d0
MOV	dword ptr [RBP + -0x4c],0xc7e...	mov	DWORD PTR [rbp-0x4c],0xc7eb99e7
MOV	qword ptr [-0x138 + RBP],RAX	mov	QWORD PTR [rbp-0x138],rax
JMP	LAB_00401ac8	jmp	0x0000000000401ac8
XOR	EAX,EAX	xor	eax,eax
MOV	ECX,dword ptr [RBP + -0x28]	mov	ecx,DWORD PTR [rbp-0x28]
MOV	...	...	...



# Script - CompareSleighExternal

Address Break | Plate | Function | Variable | Instruction/Data | Open Data | Array

Register Transition

Pre-Comment

Label

Address	Mnemonic	Operands	External Disassembly
0045baa3	ADD	RI4, 0x1	add r14, 0x1
0045baa7	ADD	RDX, 0x1	add rdx, 0x1
0045baab	MOV	byte ptr [R14 + -0x1], CL	mov BYTE PTR [r14-0x1], cl
0045baaf	CMP	RSI, R14	cmp rsi, r14
0045bab2	JNZ	LAB_0045baa0	jne 0x000000000045baa0
0045bab4	ADD	R13, RAX	add r13, rax
0045bab7	MOVZX	EAX, byte ptr [R13]	movzx eax, BYTE PTR [r13+0x0]
0045babc	TEST	AL, AL	test al, al
0045babe	JNZ	LAB_0045ba24	jne 0x000000000045ba24

Post-Comment

Space

Bookmarks - (70137 bookmarks)

Type	Category	Description	Location	Label	Code Unit
Warning	Mnemonic Disagreement	jae 0x00000...	0045ba84	JNC LAB_0045bbf0	
Warning	Missing characters	+0x0	0045ba98	?? 0Fh	
Error	Bad Instruction	(bad)	0045ba99	?? 1Fh	
Warning	Missing String Markup	+	0045baab	MOV byte ptr [R14 + ...	
Warning	Missing characters	-0x1	0045baab	MOV byte ptr [R14 + ...	
Warning	Mnemonic Disagreement	jne 0x00000...	0045bab2	JNZ LAB_0045baa0	
Warning	Missing characters	+0x0	0045bab7	MOVZX EAX, byte ptr ...	
Warning	Mnemonic Disagreement	jne 0x00000...	0045babe	JNZ LAB_0045ba24	
Warning	Missing characters	+0x0	0045bae4	NOP dword ptr [RAX]	



# Script - DebugSleighInstructionParse

```
: {line# 4522} XOR <spec_rm64>, <usimm8_64>
decide on instruction bits: byte-offset=1, bitrange=(4,7), value=0x9, bytes=01001000.1000(1001)
decendent constructors for decision node (complete tree dump ordered by line number):
: {line# 3610} MOV <rm16>, <Reg16>
: {line# 3612} MOV <rm32>, <Reg32>
: {line# 3614} MOV <rm64>, <Reg64>
decide on context bits: bitrange=(2,3), value=0x2, context=10(10)0000.00100010.00000000.00100000
decendent constructors for decision node (complete tree dump ordered by line number):
: {line# 3614} MOV <rm64>, <Reg64>
check pattern[1 of 1] instruction: {line# 3614} MOV <rm64>, <Reg64>
( byte pattern: mask=11111111.00000000.00000000.00000000
  bytes[1-4]=10001001.11000111.01001000.10001001
  match-value=10001001.00000000.00000000.00000000 Matched
) -and- (
  context pattern: mask=00110000.00000001.00000000.00000000
  context(0..31)=10100000.00100010.00000000.00100000
  match-value=00100000.00000000.00000000.00000000 Matched
  vexMode(15.15) == 0x0 Match
  opsize(2,3) == 0x2 Match
) Matched
rm64: resolving...
check pattern[1 of 2] rm64: {line# 1365} <Rmr64>
byte pattern: mask=11000000.00000000.00000000.00000000
bytes[2-5]=11000111.01001000.10001001.10010101
match-value=11000000.00000000.00000000.00000000 Matched
Rmr64: resolving...
decide on context bits: bitrange=(13,13), value=0x0, context=10100000.00100(0)10.00000000.00100000
decendent constructors for decision node (complete tree dump ordered by line number):
  Rmr64: {line# 910} <r64>
  check pattern[1 of 1] Rmr64: {line# 910} <r64>
  context pattern: mask=00000100.00000000.00000000.00000000
  context(8..39)=00100010.00000000.00100000.00000000
  match-value=00000000.00000000.00000000.00000000 Matched
  rexPrefix(13,13) == 0x0 Match
  r64: resolving...
  r64: register RDI (size:8)
Reg64: resolving...
decide on context bits: bitrange=(11,11), value=0x0, context=10100000.001(0)0010.00000000.00100000
decendent constructors for decision node (complete tree dump ordered by line number):
  Reg64: {line# 893} <reg64>
  check pattern[1 of 1] Reg64: {line# 893} <reg64>
  context pattern: mask=00010000.00000000.00000000.00000000
  context(8..39)=00100010.00000000.00100000.00000000
  match-value=00000000.00000000.00000000.00000000 Matched
  rexPrefix(11,11) == 0x0 Match
  reg64: resolving...
  reg64: register RAX (size:8)

Prototype parse successful: MOV RDI,RAX
Instruction length = 3 bytes
Instr Mask: 11111000.11111111.11000000
Instr Value: 01001000.10001001.11000000
Op-0 Mask: 00000000.00000000.00000111
Op-0 Value: 00000000.00000000.00000111
Op-1 Mask: 00000000.00000000.00111000
Op-1 Value: 00000000.00000000.00000000 (DebugSleighInstructionParse.java:54)
```



## Developing a Sleigh Module - What's Good Enough?

- Disassembly
  - ▶ Decode, Display, Flow instructions
- References
  - ▶ Addressing modes
- Decompilation
  - ▶ All Data Flow, pseudoOp In/Out, Logic, Math
- Emulation - EmulateInstructionStateModifier
- Theorem Proving - Detailed effects
- Partial languages - OK
  - ▶ Use unimpl, BadInstruction(), pseudoOp
- Speed up the Process - Automate it
  - ▶ Scraping disassembly / PDF
  - ▶ Parse disassembly tables, XML descriptions



## Developing a Sleight Module - Now What?

- Tune for decompilation - calling convention
- Load format
  - ▶ ELF, .opinion for magic machinelD
- Tune for emulation - Sleight P-Code Tests
- Analyzers
  - ▶ Stock constant reference propagation can work well
  - ▶ Write specialized register propagation - Page register
- Pattern Files - recognize common patterns or key functions
- Variants - Pointer checking, Control Flow Guard
  - ▶ Decompiler Pcode UserOp injection
  - ▶ Use context, Define, variants with Slaspec
- FID Files - Static library pattern matching



## Contacting Us

- The Ghidra team is on Github.
- @NSAGov on Twitter announces new releases.
- The Ghidra team is **not** on Twitter, reddit, Slashdot, VKontakte,...



## Reporting Bugs

- Please report bugs!
- The perfect bug report includes:
  1. Source code.
  2. Relevant bytes from the binary.
  3. XML Debug Function Decompilation from decompiler.
  4. Stack trace if there is one.
- Often we need an entire function and surrounding instructions.
- Pictures work, but can limit triage.
- We reserve the right to ignore sketchy binaries :)





## www.ghidra-sre.org Stats (June 25)

- 9.0.0: 302k downloads
- 9.0.1: 36k downloads
- 9.0.2: 100k downloads
- 9.0.4: 42k downloads
- Site views: 10.6M
- Video hits: 751k

## Github Stats (June 25)

- 16145 stars
- 2019 forks
- 718 watching
- 608 issues, 272 open
- 111 pull requests, 35 open



## References

- Xtext - itemis.com, <https://www.eclipse.org/Xtext/>
- mumbel - <https://github.com/mumbel/ghidra/tree/tricore>
- SleighEditor\_README.html, build\_README.txt



Questions?

