Building an alternative to IDA

Radare2

History

- radare in 2006

History

- radare in 2006
- radare2 in 2009

History

- radare in 2006
- radare2 in 2009
- written in pure C

History

- radare in 2006
- radare2 in 2009
- written in pure C
- 350k LoC

History

- radare in 2006
- radare2 in 2009
- written in pure C
- 350k LoC
- several contributors

History

- radare in 2006
- radare2 in 2009
- written in pure C
- 350k LoC
- several contributors
- RSoC

Platforms

Runs on

- Windows
- Linux
- BSD
- OSX
- Android
- iOS
- Smartwatch
- Browser

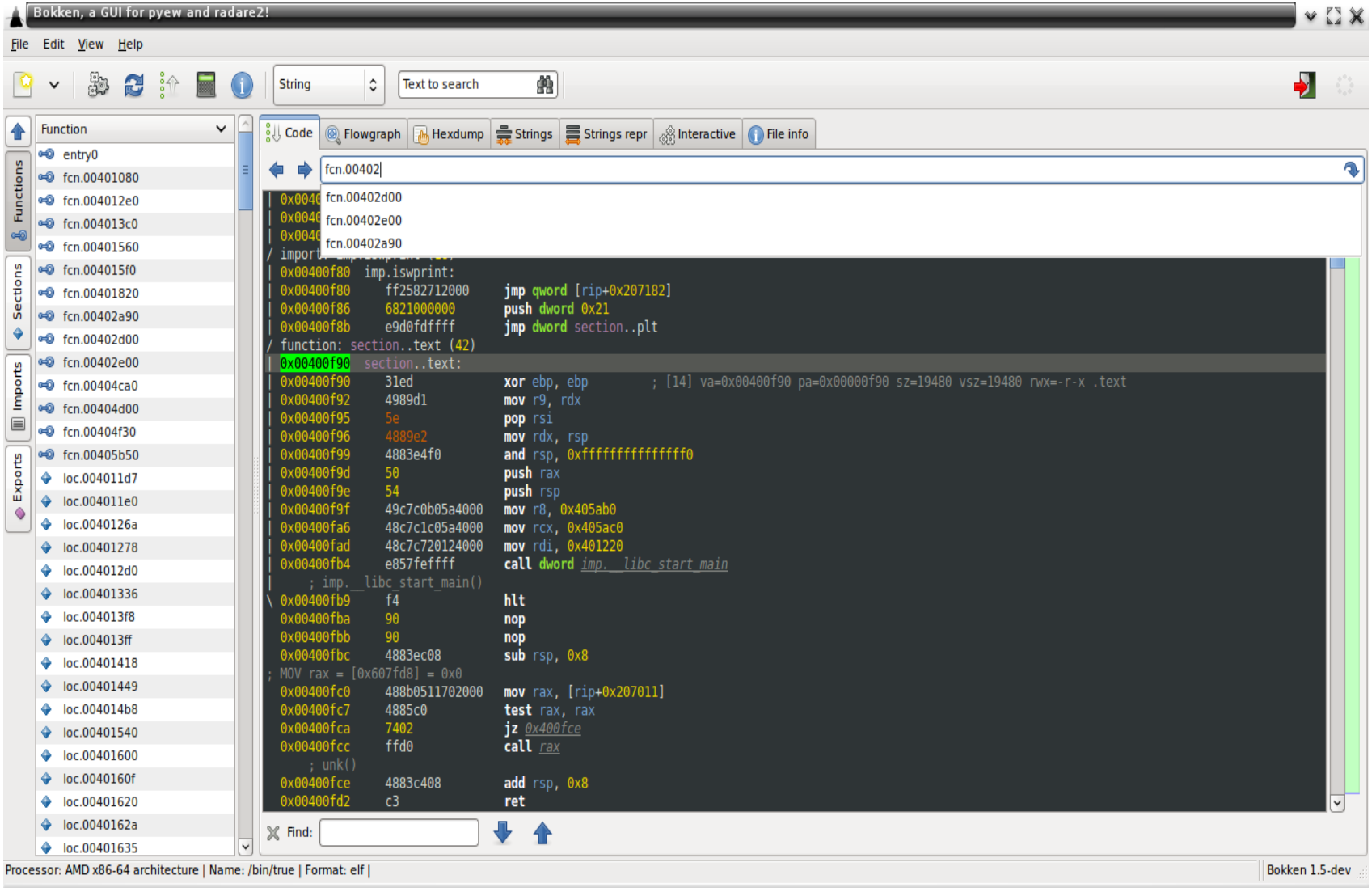
Handles

- PE/COFF
- ELF
- Mach0
- DEX/JAVA
- BIOS/TE
- GB/GBA/DS
- XBOX
- Plan9

Architectures

- 8051
- arc
- arm
- avr
- brainfuck
- cr16
- csr
- dalvik
- dcpu16
- ebc
- gb
- h8300
- i8080
- java
- m68k
- malbolge
- mips
- msil
- msp430
- nios2
- powerpc
- rar
- sh
- sparc
- spc700
- sysz
- tms320
- v850
- whitespace
- x86
- xcore
- z80
- propeller
- snes
- psosvm
- 6502

What about a GUI?



What about a GUI?

- We don't like GUI.

What about a GUI?

- We don't like GUI.
- We don't use GUI.

What about a GUI?

- We don't like GUI.
- We don't use GUI.
- There are unmaintained ones

What about a GUI?

- We don't like GUI.
- We don't use GUI.
- There are unmaintained ones
- We're busy.

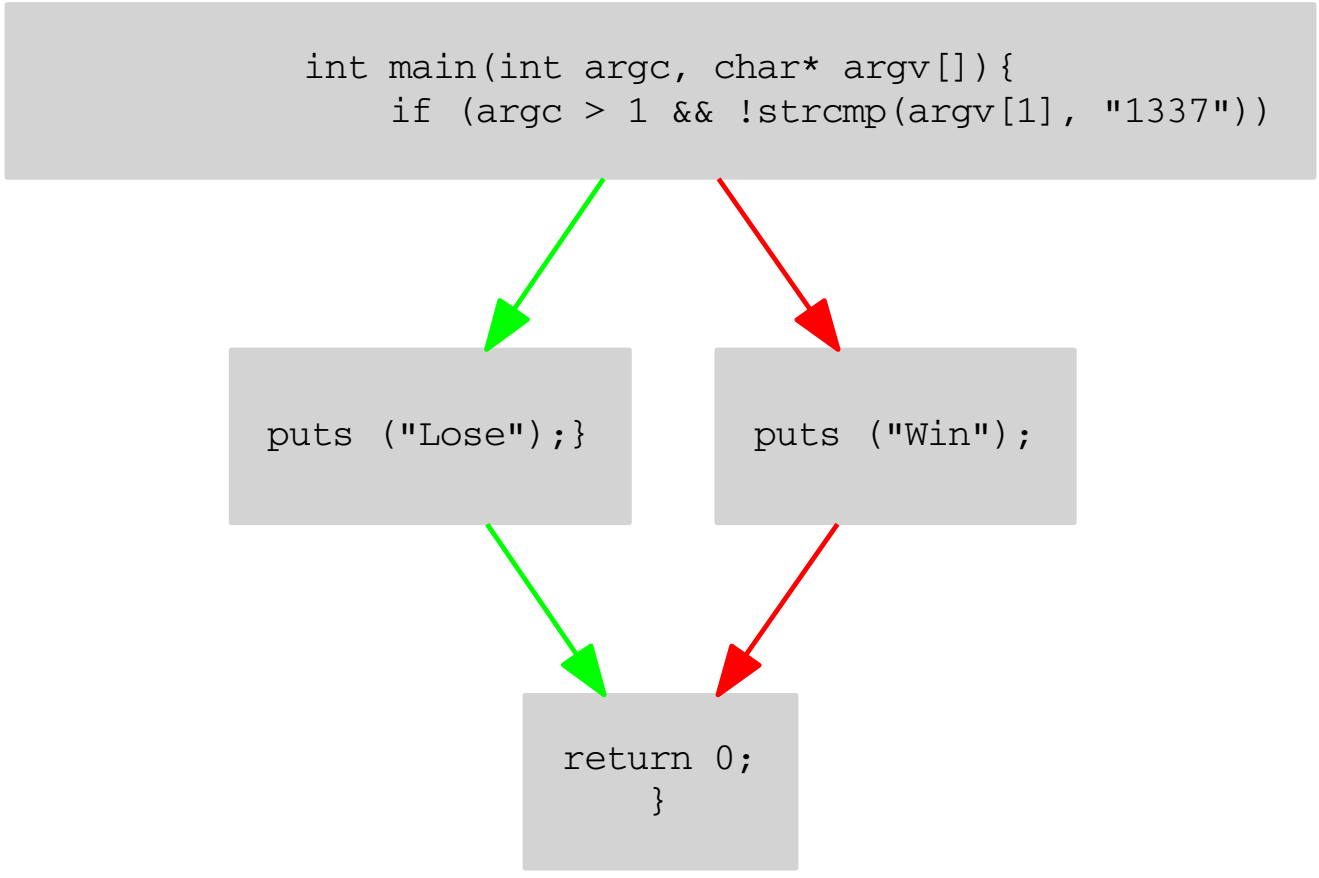
What about a GUI?

- We don't like GUI.
- We don't use GUI.
- There are unmaintained ones
- We're busy.
- R2 is open source, write one.

What about a GUI?

- We don't like GUI.
- We don't use GUI.
- There are unmaintained ones
- We're busy.
- R2 is open source, write one.
- Pay us to write one?

But what about graphs?



But what about graphs?

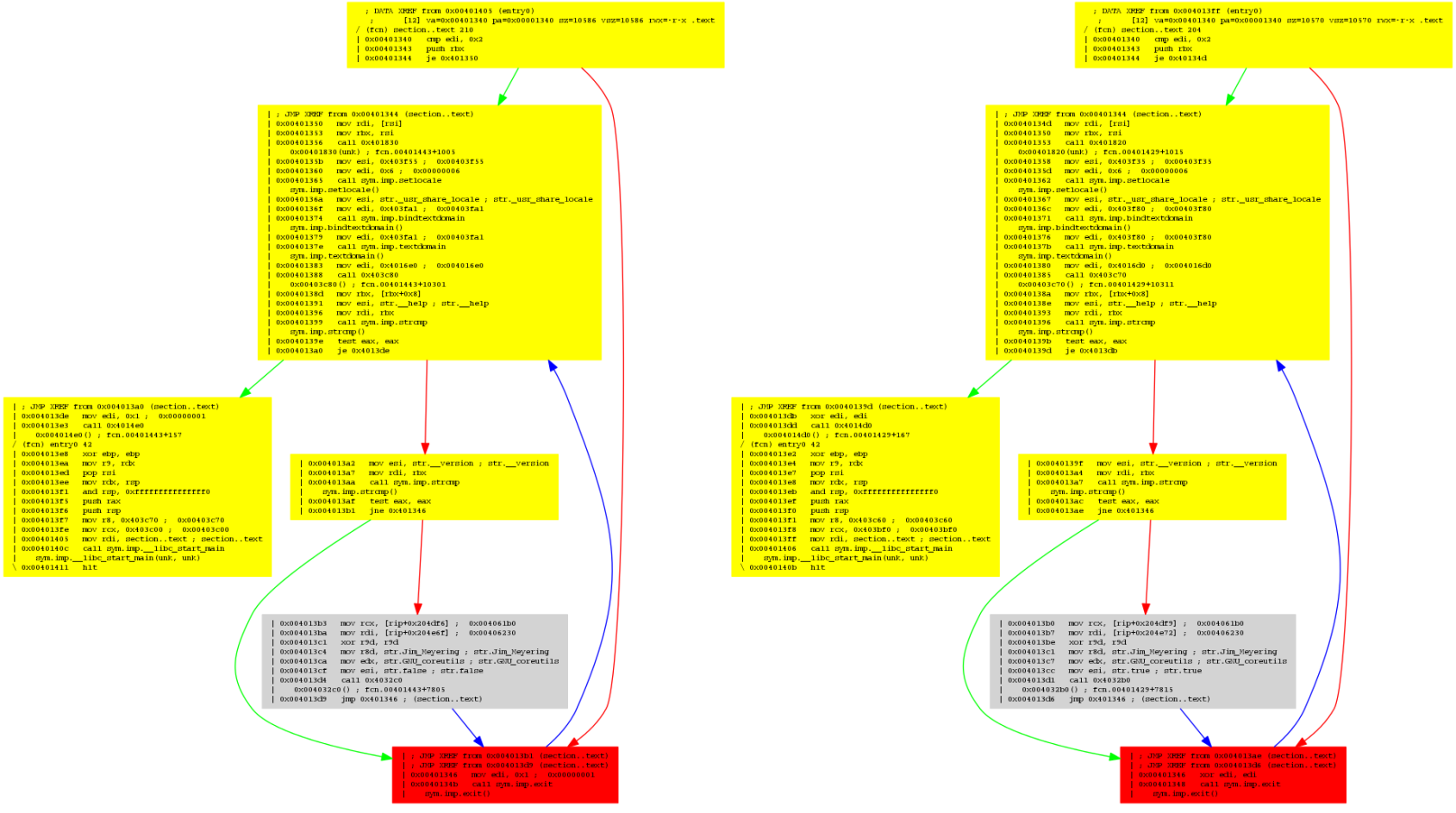
Analysis

- Functions detection
- Local var detection
- FLIRT integration
- (X)REF
- DWARF and PDB

Binary diffing with radiff2

/bin/true

/bin/false



Assembling and disassembling

```
jvoisin@kaa 2:30 ~ ragg2 -i exec -a x86 -b 32 -z
"\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x53\x89\xe1\x99\xb0\x0b\xcd\x80"
jvoisin@kaa 2:30 ~ ragg2 -i exec -a x86 -b 32 | rasm2 -d -
xor eax, eax
push eax
push 0x68732f2f
push 0x6e69622f
mov ebx, esp
push eax
push ebx
mov ecx, esp
cdq
mov al, 0xb
int 0x80
zsh: done       ragg2 -i exec -a x86 -b 32 |
zsh: exit 24    rasm2 -d -
jvoisin@kaa 2:30 ~ rasm2 -a z80 'ld hl,$1000;add a,(hl);inc hl'
218623
jvoisin@kaa 2:42 ~ rasm2 -L | wc -l
46
zsh: exit 1     rasm2 -L |
zsh: done       wc -l
jvoisin@kaa 2:43 ~
```

Structures

```
[0x00000000]> pf.pe_nt_image_headers32 @ pe_nt_image_headers32
                          signature : 0x00000108 = PE
        (pe_image_file_header)fileHeader : <struct>
                  (pe_machine)machine : 0x0000010c =  machine (enum) = 0x14c ; IMAGE_FILE_MACHINE_I386
                     numberOfSections : 0x0000010e = 0x0005
                       timeDateStamp : 0x00000110 = 0x52d829f8
                pointerToSymbolTable : 0x00000114 = 0x00000000
                     numberOfSymbols : 0x00000118 = 0x00000000
                sizeOfOptionalHeader : 0x0000011c = 0x00e0
     (pe_characteristics)characteristics : 0x0000011e =  characteristics (bitfield) = 0x00000122 : IMAGE_FILE_EXECUTABLE_IMAGE |
FILE_LARGE_ADDRESS_AWARE | IMAGE_FILE_32BIT_MACHINE
(pe_image_optional_header32)optionalHeader : <struct>
                         (pe_magic)magic : 0x00000120 =  magic (enum) = 0x10b ; IMAGE_NT_OPTIONAL_HDR32_MAGIC
                    majorLinkerVersion : 0x00000122 = 10 ; 0x0a ; ''
                    minorLinkerVersion : 0x00000123 = 0 ; 0x00 ; ''
                           sizeOfCode : 0x00000124 = 0x001bce00
```

Visual mode

- Ncurses-like
- Static
- Dynamic
- Analysis
- Try it, really.

ROP

- Efficient
- Multi-arch
- Filterable output
- No ROP-chain builder

```
[0x004013e2]> /R mov eax,pop,pop,pop,ret
0x0040364a ret
  0x00403644         89d8  mov eax, ebx
  0x00403646           5b  pop rbx
  0x00403647           5d  pop rbp
  0x00403648         415c  pop r12
  0x0040364a           c3  ret

0x0040366a ret
  0x00403664         89d8  mov eax, ebx
  0x00403666           5b  pop rbx
  0x00403667           5d  pop rbp
  0x00403668         415c  pop r12
  0x0040366a           c3  ret

0x00403685 ret
  0x0040367f         89d8  mov eax, ebx
  0x00403681           5b  pop rbx
  0x00403682           5d  pop rbp
  0x00403683         415c  pop r12
  0x00403685           c3  ret

0x00403b0d ret
  0x00403b04   b8ffffffff  mov eax, 0xffffffff
  0x00403b09           5b  pop rbx
  0x00403b0a           5d  pop rbp
  0x00403b0b         415c  pop r12
  0x00403b0d           c3  ret

[0x004013e2]>
```

Shellcode compiling

```
jvoisin@kaa 1:11 ~ cat /tmp/meh.c
int main() {
  write (1,"Hello!\n",7);
  exit(0);
}
jvoisin@kaa 1:12 ~ ragg2-cc -x /tmp/meh.c 2>/dev/null | rasm2 -d - 2>/dev/null
jmp 0x2
dec eax
lea esi, [0x1d]
mov edi, 0x1
mov edx, 0x7
mov eax, 0x1
syscall
xor edi, edi
mov eax, 0x3c
syscall
xor eax, eax
ret
dec eax
gs insb
insb
outsd
and [edx], ecx
jvoisin@kaa 1:12 ~ ragg2-cc /tmp/meh.c 2>/dev/null
/tmp/meh.c.bin
jvoisin@kaa 1:12 ~ /tmp/meh.c.bin
Hello!
jvoisin@kaa 1:12 ~
```

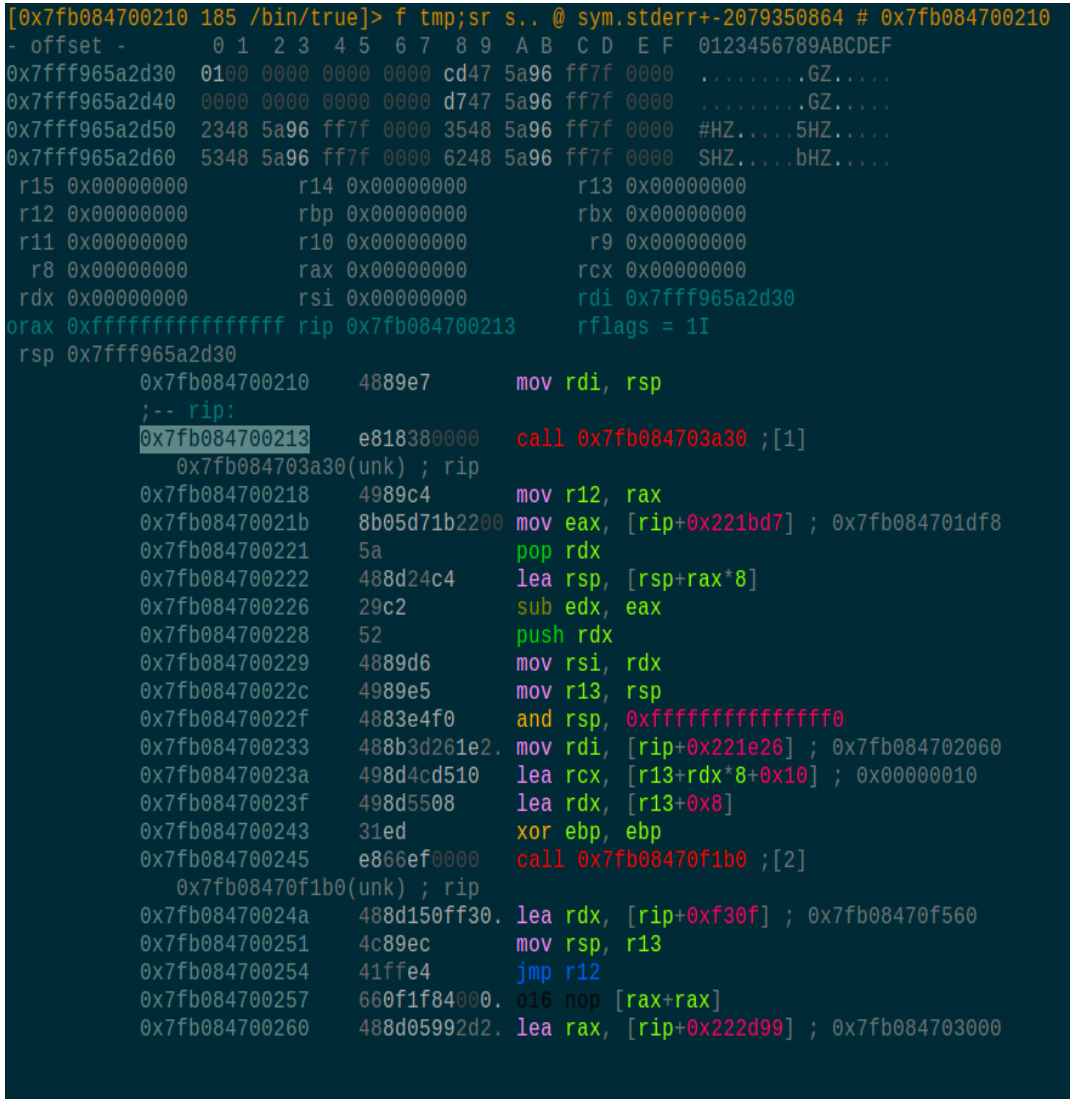
Binary patching

- Assemblers
- Cracker's toolbag
- Cacheable
- Patchfiles

```
[0x00000000]> wo?
|Usage: wo[asmdxoArl24] [hexpairs] @ addr[:bsize]
| wow [val]             ==  write looped value (alias for 'wb')
| woa [val]             +=  addition (f.ex: woa 0102)
| wos [val]             -=  substraction
| wom [val]             *=  multiply
| wod [val]             /=  divide
| woe [from-to] [step]  ..  create sequence
| wox [val]             ^=  xor  (f.ex: wox 0x90)
| woo [val]             |=  or
| woA [val]             &=  and
| woR                   random bytes (alias for 'wr $b')
| wor [val]             >>= shift right
| wol [val]             <<= shift left
| wo2 [val]             2=  2 byte endian swap
| wo4 [val]              4=  4 byte endian swap
| woD [len]             De Bruijn Pattern (syntax woD length @ addr)
| woO [len]             De Bruijn Pattern Offset (syntax: woO value)
[0x00000000]>
```

Debugger

- Works in visual mode
- Several backends
- Classic features
- Tracing



Conclusion

- Open
- Attracting contributors
- Full of features
- Steep learning curve

Conclusion

Radare2 is nice.

You should use it.

Conclusion

We're doing a workshop a 14h o'clock in Fischbach!