Краткая справочная информация по функциям rizin, версия 0.1a (2025-01-20)

`rizin` - это набор утилит, предлагающий широкий набор возможностей:

1. Генерация кода на ассемблере для различных архитектур
   по тексту на языке Си (утилита rz-gg).

2. Генерация машинного кода по тексту на языке ассемблер
   и дизассемблирование (утилита rz-asm).

3. Анализ бинарных запускаемых файлов и определение их
   характеристик (утилита rz-bin с ключом -I и другими).

4. Преобразование чисел разных форм представления (текст,
   бинарный файл) и систем счисления (утилита rz-ax).

5. Сравнение и оценка схожести файлов (утилита rz-diff).

6. Поиск строк и подстрок в бинарных файлах (утилита rz-find).

7. Поблочное вычисление и проверка хэш-сумм файла (утилита rz-hash).

*) Поддерживается 105 различных процессорных архитектур
   (в т.ч. 8-bit Zilog Z80, 8/32-bit VAX, 32/64-bit RISC-V,
   32-bit WebAssembly 0.1.0, Intel 8080, Java bytecode, и т.д.)
   Однако часть из них - лишь в режиме дизассемблирования.

1. Утилита rz-gg (генерация кода на ассемблере по тексту на Си)

```
# With C file :
$ cat hi.c
main() {
  write(1, "Hello\n", 6);
  exit(0);
}

$ rz-gg -O -F hi.c

$ ./hi
Hello

# Linked into a tiny binary. This is 165 bytes
$ wc -c < hi
165
```

```
$ rz-asm -a x86 -b 64 -D 90
0x00000000   1                      90  nop

-L   List asm plugins (a=asm, d=disasm, A=analyze, e=ESIL)

$ rz-asm -a x86 -b 64 'mov rax, 0x77'
48c7c077000000

$ rz-asm -a x86 -b 64 'mov rax, 0x77; mov rbx, 0x99; not rax'
48c7c07700000048c7c39900000048f7d0

$ rz-asm -a x86 -b 32 'mov eax, 0x77; mov ebx, 0x99; not eax'
b877000000bb99000000f7d0

$ rz-asm -a x86 -b 16 'mov ax, 0x77; mov bx, 0x99; not ax'
b87700bb9900f7d0

$ rz-asm -a x86 -b 64 -D 48c7c077000000
0x00000000   7           48c7c077000000  mov rax, 0x77

$ rz-asm -a i8080 -b 8 -D 1f00070f0c
0x00000000   1                      1f  rar
0x00000001   1                      00  nop
0x00000002   1                      07  rlc
0x00000003   1                      0f  rrc
0x00000004   1                      0c  inr c

$ rz-asm -a z80 -D 90
0x00000000   1                      90  sub b

$ rz-asm -a z80 'ld b, c; nop'
4100

$ cat rz-asm.list | colrm 1 6 | colrm 9 | tr "[:space:]" "\n" | \
tr -s "\n" | wc -l
105

(105 вариантов процессорных архитектур)
```

3. Утилита rz-bin (преобразование чисел разных систем счисления)

А) Печать атрибутов бинарного файла (`rz-bin` с ключом `-I`):

```
$ rz-bin -I 4args
[Info]
arch     x86
cpu      N/A
baddr    0x00000000
binsz    0x000032ce
bintype  elf
bits     64
class    ELF64
compiler GCC: (GNU) 13.2.1 20240128 (ALT Sisyphus 13.2.1-alt3) GCC: (GNU) 14.2.1 20241028 (ALT Sisyphus 14.2.1-alt1)
dbg_file N/A
endian   LE
hdr.csum N/A
guid     N/A
intrp    /lib64/ld-linux-x86-64.so.2
laddr    0x00000000
lang     c
machine  AMD x86-64 architecture
maxopsz  16
minopsz  1
os       linux
cc       N/A
pcalign  0
relro    full
rpath    NONE
subsys   linux
stripped true
crypto   false
havecode true
va       true
sanitiz  false
static   false
linenum  false
lsyms    false
canary   false
PIE      true
RELROCS  false
NX       true
```

Б) Определение адреса, используемого dlopen (ключ `-Q`):

```
$ LD_LIBRARY_PATH=‘pwd‘ rz-bin -Q libgems.so
libgems.so is loaded at 0x558f6bd7aa60
```

В) Из каких компонентов состоит бинарный файл:

```
$ rz-bin -A 4args
[Archs]
offset     size  arch   machine
------------------------------------------------
0x00000000 14992 x86_64 AMD x86-64 architecture
```

Г) Точка входа исполняемого файла/библиотеки:

```
$ rz-bin -e 4args
[Entries]
vaddr      paddr      hvaddr     haddr      type
----------------------------------------------------
0x00001050 0x00001050 0x00000018 0x00000018 program

$ rz-bin -e /usr/lib/libc.so.6
[Entries]
vaddr      paddr      hvaddr     haddr      type
----------------------------------------------------
0x00023e30 0x00023e30 0x00000018 0x00000018 program
```

Д) Экспортируемые символы бинарного файла:

```
$ rz-bin -E 4args-func.o
[Exports]
nth paddr      vaddr      bind   type   size lib name
-------------------------------------------------
3   0x00000040 0x08000040 GLOBAL NOTYPE 0        L3

$ rz-bin -E /usr/lib/libc.so.6 | grep -we TLS
490  0x00000020 0x00000020 GLOBAL TLS  4         __libc_dlerror_result
2048 0x00000004 0x00000004 GLOBAL TLS  4         __resp
2274 0x00000044 0x00000044 GLOBAL TLS  4         __h_errno
2364 0x00000008 0x00000008 GLOBAL TLS  4         errno
```

Е) Точка входа конструктора/деструктора:

```
$ rz-bin -ee 4args
[Initfini]
vaddr      paddr      hvaddr     haddr      type
-----------------------------------------------
0x00001130 0x00001130 0x00003db0 0x00002db0 init
0x000010f0 0x000010f0 0x00003db8 0x00002db8 fini
```

Ж) Атрибуты заголовка:

```
$ rz-bin -H 4args.o
[Fields]
paddr      name       vaddr      comment
-------------------------------------------------------------------------------
0x00000000 MAGIC      0x00000000 7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
0x00000010 Type       0x00000010 0x0001
0x00000012 Machine    0x00000012 0x003e
0x00000014 Version    0x00000014 0x00000001
0x00000018 Entrypoint 0x00000018 0x00000000
0x00000020 PhOff      0x00000020 0x00000000
0x00000028 ShOff      0x00000028 0x00000308
0x00000030 Flags      0x00000030 0x0000
0x00000034 EhSize     0x00000034 64
0x00000036 PhentSize  0x00000036 0
0x00000038 PhNum      0x00000038 0
0x0000003a ShentSize  0x0000003a 64
0x0000003c ShNum      0x0000003c 14
0x0000003e ShStrndx   0x0000003e 13

$ rz-bin -H /usr/lib/libc.so.6
[Fields]
paddr      name       vaddr      comment
-------------------------------------------------------------------------------
0x00000000 MAGIC      0x00000000 7f 45 4c 46 01 01 01 03 00 00 00 00 00 00 00 00
0x00000010 Type       0x00000010 0x0003
0x00000012 Machine    0x00000012 0x0003
0x00000014 Version    0x00000014 0x00000001
0x00000018 Entrypoint 0x00000018 0x00023e30
0x0000001c PhOff      0x0000001c 0x00000034
0x00000020 ShOff      0x00000020 0x001f2b80
0x00000024 Flags      0x00000024 0x0000
0x00000028 EhSize     0x00000028 52
0x0000002a PhentSize  0x0000002a 32
0x0000002c PhNum      0x0000002c 13
0x0000002e ShentSize  0x0000002e 40
0x00000030 ShNum      0x00000030 60
0x00000032 ShStrndx   0x00000032 59
```

З) Импортируемые символы (из других библиотек):

```
$ rz-bin -i 4args.o
[Imports]
nth vaddr      bind   type   lib name
---------------------------------------
5   ---------- GLOBAL NOTYPE     L3
6   ---------- GLOBAL NOTYPE     printf
```

И) С какими библиотеками слинкован бинарный файл:

```
$ rz-bin -l /usr/lib/libc.so.6
[Libs]
library
--------------
ld-linux.so.2
```

```
$ ldd /usr/lib/libc.so.6
  /lib/ld-linux.so.2 (0xf7eff000)
  linux-gate.so.1 (0xf7efd000)
```

```
*) Файл linux-gate.so.1 не существует в виде обычного файла в файловой системе.
   Чтобы определить местоположение linux-gate.so.1 в памяти, можно использовать
   команду cat /proc/self/maps:
   cat /proc/self/maps | grep -i linux-gate
   Если linux-gate.so.1 отсутствует, это не обязательно является проблемой.
   Важно понимать, что этот механизм предназначен для оптимизации системных
   вызовов и может не всегда отображаться в памяти.
```

К) Символы бинарных файлов:

```
$ rz-bin -s 4args
[Symbols]
nth paddr      vaddr      bind   type   size lib name
-------------------------------------------------------------------------------
1   ---------- ---------- GLOBAL FUNC   0        imp.__libc_start_main
2   ---------- ---------- WEAK   NOTYPE 0        imp._ITM_deregisterTMCloneTable
3   0x00001030 0x00001030 GLOBAL FUNC   16       imp.printf
4   ---------- ---------- WEAK   NOTYPE 0        imp.__gmon_start__
5   ---------- ---------- WEAK   NOTYPE 0        imp._ITM_registerTMCloneTable
6   0x00001040 0x00001040 WEAK   FUNC   16       imp.__cxa_finalize
```

```
$ rz-bin -s 4args-func.o
[Symbols]
nth paddr      vaddr      bind   type   size lib name
----------------------------------------------------------------
1   ---------- 0x00000000 LOCAL  FILE   0        4args-func.asm
2   0x00000040 0x08000040 LOCAL  SECT   0        .text
3   0x00000040 0x08000040 GLOBAL NOTYPE 0        L3
```

Для сравнения, утилита `nm` из GNU Development Tools:

```
$ nm -a 4args-func.o
0000000000000000 a 4args-func.asm
0000000000000000 T L3
0000000000000000 t .text
```

Л) Строки из бинарных файлов:

```
$ rz-bin -z 4args
[Strings]
nth paddr      vaddr      len size section type  string
-------------------------------------------------------------
0   0x00002004 0x00002004 14  15   .rodata ascii %d+%d+%d+%d=%d
```

```
$ rz-bin -zz 4args | grep -e '.comment'
019 0x00003004 0xffffffffffffffff 53 54 (.comment) ascii GCC: (GNU) \
    13.2.1 20240128 (ALT Sisyphus 13.2.1-alt3)
020 0x0000303a 0x00000035 53 54 (.comment) ascii GCC: (GNU) 14.2.1 \
    20241028 (ALT Sisyphus 14.2.1-alt1)
047 0x0000329d 0x00000104 8 9 (.shstrtab) ascii .comment
```

M) Секции и сегменты:

```
$ rz-bin -S 4args
[Sections]
paddr      size  vaddr      vsize align perm name                  type       flags
-----------------------------------------------------------------------------------------
0x00000000 0x0   ---------- 0x0   0x0   ----                        NULL
0x00000318 0x1c  0x00000318 0x1c  0x0   -r-- .interp               PROGBITS   alloc
0x00000338 0x20  0x00000338 0x20  0x0   -r-- .note.gnu.property    NOTE       alloc
0x00000358 0x20  0x00000358 0x20  0x0   -r-- .note.ABI-tag         NOTE       alloc
0x00000378 0x24  0x00000378 0x24  0x0   -r-- .note.gnu.build-id    NOTE       alloc
0x000003a0 0x24  0x000003a0 0x24  0x0   -r-- .gnu.hash             GNU_HASH   alloc
0x000003c8 0xa8  0x000003c8 0xa8  0x0   -r-- .dynsym               DYNSYM     alloc
0x00000470 0x8f  0x00000470 0x8f  0x0   -r-- .dynstr               STRTAB     alloc
0x00000500 0xe   0x00000500 0xe   0x0   -r-- .gnu.version          VERSYM     alloc
0x00000510 0x30  0x00000510 0x30  0x0   -r-- .gnu.version_r        VERNEED    alloc
0x00000540 0xc0  0x00000540 0xc0  0x0   -r-- .rela.dyn             RELA       alloc
0x00000600 0x18  0x00000600 0x18  0x0   -r-- .rela.plt             RELA       alloc,info
0x00001000 0x17  0x00001000 0x17  0x0   -r-x .init                 PROGBITS   alloc,execute
0x00001020 0x20  0x00001020 0x20  0x0   -r-x .plt                  PROGBITS   alloc,execute
0x00001040 0x8   0x00001040 0x8   0x0   -r-x .plt.got              PROGBITS   alloc,execute
0x00001050 0x17b 0x00001050 0x17b 0x0   -r-x .text                 PROGBITS   alloc,execute
0x000011cc 0x9   0x000011cc 0x9   0x0   -r-x .fini                 PROGBITS   alloc,execute
0x00002000 0x14  0x00002000 0x14  0x0   -r-- .rodata               PROGBITS   alloc
0x00002014 0x2c  0x00002014 0x2c  0x0   -r-- .eh_frame_hdr         PROGBITS   alloc
0x00002040 0xac  0x00002040 0xac  0x0   -r-- .eh_frame             PROGBITS   alloc
0x00002db0 0x8   0x00003db0 0x8   0x0   -rw- .init_array           INIT_ARRAY write,alloc
0x00002db8 0x8   0x00003db8 0x8   0x0   -rw- .fini_array           FINI_ARRAY write,alloc
0x00002dc0 0x8   0x00003dc0 0x8   0x0   -rw- .data.rel.ro          PROGBITS   write,alloc
0x00002dc8 0x1f0 0x00003dc8 0x1f0 0x0   -rw- .dynamic              DYNAMIC    write,alloc
0x00002fb8 0x48  0x00003fb8 0x48  0x0   -rw- .got                  PROGBITS   write,alloc
0x00003000 0x4   0x00004000 0x4   0x0   -rw- .data                 PROGBITS   write,alloc
0x00003004 0x0   0x00004004 0x4   0x0   -rw- .bss                  NOBITS     write,alloc
0x00003004 0x6c  ---------- 0x6c  0x0   ---- .comment              PROGBITS   merge,strings
0x00003070 0x101 ---------- 0x101 0x0   ---- .GCC.command.line     PROGBITS   merge,strings
0x00003174 0x24  ---------- 0x24  0x0   ---- .gnu.build.attributes NOTE
0x00003198 0x136 ---------- 0x136 0x0   ---- .shstrtab             STRTAB
```

```
$ rz-bin -S 4args-func.o
[Sections]
paddr      size vaddr      vsize align perm name      type     flags
-------------------------------------------------------------------------
0x00000000 0x0  0x08000000 0x0   0x0   ----           NULL
0x0000004c 0x21 0x0800004c 0x21  0x0   ---- .shstrtab STRTAB
0x00000070 0x13 0x08000070 0x13  0x0   ---- .strtab   STRTAB
0x00000084 0x60 0x08000084 0x60  0x0   ---- .symtab   SYMTAB
0x00000040 0xb  0x08000040 0xb   0x0   -r-x .text     PROGBITS alloc,execute
```

```
$ rz-bin -SS 4args
[Segments]
paddr      size  vaddr      vsize align  perm name
-----------------------------------------------------------
0x00000040 0x2d8 0x00000040 0x2d8 0x8    -r-- PHDR
0x00000318 0x1c  0x00000318 0x1c  0x1    -r-- INTERP
0x00000000 0x618 0x00000000 0x618 0x1000 -r-- LOAD0
```

```
0x00001000 0x1d5 0x00001000 0x1d5 0x1000 -r-x LOAD1
0x00002000 0xec  0x00002000 0xec  0x1000 -r-- LOAD2
0x00002db0 0x254 0x00003db0 0x258 0x1000 -rw- LOAD3
0x00002dc8 0x1f0 0x00003dc8 0x1f0 0x8    -rw- DYNAMIC
0x00000338 0x20  0x00000338 0x20  0x8    -r-- NOTE
0x00000358 0x44  0x00000358 0x44  0x4    -r-- NOTE_0x358
0x00000338 0x20  0x00000338 0x20  0x8    -r-- UNKNOWN
0x00002014 0x2c  0x00002014 0x2c  0x4    -r-- GNU_EH_FRAME
0x00000000 0x0   0x00000000 0x0   0x10   -rw- GNU_STACK
0x00002db0 0x250 0x00003db0 0x250 0x1    -r-- GNU_RELRO
0x00000000 0x40  0x00000000 0x40  0x0    -rw- ehdr
```

4. Утилита rz-ax (преобразование чисел разных систем счисления)

```
USAGE
     Force output mode (numeric base)

       =f    floating point
       =2    binary
       =3    ternary
       =8    octal
       =10   decimal
       =16   hexadecimal

     Available variable types are:

       int     ->  hex     rz-ax 10
       hex     ->  int     rz-ax 0xa
       -int    ->  hex     rz-ax -77
       -hex    ->  int     rz-ax 0xffffffb3
       int     ->  bin     rz-ax b30
       int     ->  ternary rz-ax t42
       ternary ->  int     rz-ax 1010dt
       bin     ->  int     rz-ax 1010d
       float   ->  hex     rz-ax 3.33f
       hex     ->  float   rz-ax Fx40551ed8
       oct     ->  hex     rz-ax 35o
       hex     ->  oct     rz-ax Ox12 (O is a letter)
       bin     ->  hex     rz-ax 1100011b
       hex     ->  bin     rz-ax Bx63
       ternary ->  hex     rz-ax 212t
       hex     ->  ternary z-ax Tx23
       raw     ->  hex     rz-ax -S < /binfile
       hex     ->  raw     rz-ax -s 414141

$ rz-ax 0xc0ffee
12648430
```

5. Утилита rz-diff (сравнение и оценка схожести файлов)

```
$ rz-diff -A -d leven 4args 4args
similarity: 1.000
distance: 0

$ rz-diff -C -t lines rizin.tex~ rizin.tex
--- rizin.tex~
+++ rizin.tex
@@ -136,7 +136,7 @@
 \textbackslash{}end{verbatim}

 \textbackslash{}newpage
-\textbackslash{}noindent\textbackslash{}underline{7. Утилита rz-hash (вычисление и проверка хэш-сумм)}
+\textbackslash{}noindent\textbackslash{}underline{7. Утилита rz-hash (поблочное вычисление и проверка хэш-сумм файла)}

 \textbackslash{}begin{verbatim}
 man 1 rz-hash

===
```

6. Утилита rz-find (поиск строк и подстрок в бинарных файлах)

Идентификация типа файла:

```
$ rz-find -i 4args 2>/dev/null
0x00000000 1 ELF 64-bit LSB shared object, x86-64, version 1
```

Поиск всех строк в файле:

```
$ rz-find 4args | grep -e GNU
GCC: (GNU) 13.2.1 20240128 (ALT Sisyphus 13.2.1-alt3)
GCC: (GNU) 14.2.1 20241028 (ALT Sisyphus 14.2.1-alt1)
GNU C11 13.2.1 20240128 (ALT Sisyphus 13.2.1-alt3) -fstack-clash-protection -mtune=generic \
-march=x86-64 -g -O2 -std=gnu11 -fgnu89-inline -fmerge-all-constants -frounding-math \
-fstack-protector-strong -fno-common -fmath-errno -fPIE -ftls-model=initial-exec
```

Поиск подстроки в файле:

```
$ rz-find -s bug -Z /lib/libGL.so.1 | head -n5
ERROR: Cannot determine entrypoint, using 0x0002e110.
WARNING: unimplemented ELF/X86_32 reloc type 14
0x156cc bugMessageCallback
0x156e3 bugMessageCallbackAMD
0x156fd bugMessageCallbackARB
0x15717 bugMessageCallbackKHR
0x15731 bugMessageCallbackOES
```

Поиск по таблице символов:

```
$ rz-find -S unicode.Ps -Z /usr/bin/cover
paddr: 0x003f4be0 vaddr: 0x007f4be0 type: OBJ unicode.Ps

$ objdump -T /usr/bin/cargo | grep -we write
0000000000000000      DF *UND* 0000000000000000 (GLIBC_2.2.5) write

$ rz-find -Z -I write /usr/bin/cargo 2>&1 | grep -vwe ^WARNING
ordinal: 376 write
```

7. Утилита rz-hash (поблочное вычисление и проверка хэш-сумм файла)

```
man 1 rz-hash

rz-hash allows you to calculate, check and show the hash values of each block of a target file.


$ rz-hash -B 4args
4args: 0x00000000-0x00001000 sha256: a318c954ff1fa7b6bf2359c9ccdc099235d39a8f09b7750cadbb9d00e82344e3
4args: 0x00001000-0x00002000 sha256: 8455d94c3a6f70cece4fc777f8da998da881c072ad4fee75f4ccb1b16eaaf863
4args: 0x00002000-0x00003000 sha256: fa1609dee4ed165cc4c60a50ee34a64e57263becd33e7f0a0ae3676cbf071f43
4args: 0x00003000-0x00004000 sha256: 7b02a982de0c28c60e3eb2ad614d31f95ad97383e9ea983df66b9ae431cdcc82

$ rz-hash -L
flags  algorithm      license    author
____hm md2            LGPL3      swedenspy
____hm md4            Apache 2.0 OpenSSL Team
____hm md5            Apache 2.0 OpenSSL Team
____hm sha1           Apache 2.0 OpenSSL Team
____hm sha256         Apache 2.0 OpenSSL Team
____hm sha384         Apache 2.0 OpenSSL Team
____hm sha512         Apache 2.0 OpenSSL Team
____hm sm3            Apache 2.0 OpenSSL Team
____h_ blake3         CC0        Samuel Neves,Jack O'Connor

[ ... ]

ED____ serpent-ecb    LGPL-3     NicsTr
ED____ xor            LGPL-3     pancake
ED____ sm4-ecb        LGPL-3     0xSh4dy

flags legenda:
    E = encryption, D = decryption
    e = encoding, d = encoding
    h = hash, m = hmac
```

При сравнении бинарных файлов, размер их ограничен 5 МБ, rizin.git/rizin/librz/main/rz-diff.c

```
444          if (dio->filesize > MEGABYTE(5)) {
445                  rz_diff_error("cannot open file '%s' because its size is above 5Mb\n", file);
446                  goto rz_diff_slurp_file_end;
447          }
```

```
$ RZ_BIN_MAXSTRBUF=10000000 rz-diff -v -d leven "Агни Парфене.mp3" "orig/Агни Парфене.mp3"
ERROR: rz-diff: error, cannot open file 'Агни Парфене.mp3' because its size is above 5Mb
```

Если урезать файлы до 4 МБ, успешно находим расстояние Левенштейна:

```
$ rz-diff -v -d leven "Agni Parfene.mp3" "orig/Agni Parfene.mp3"
similarity: 1.000
distance: 121
```

```
Plugins: a=asm, d=disasm, A=analyze, e=ESIL

_dAe  8 16       6502        LGPL3   6502/NES/C64/Tamagotchi/T-1000 CPU
adAe  8          8051        PD      8051 Intel CPU
_dA_  32         amd29k      LGPL3   AMD 29k RISC CPU (by deroad)
a___  16 32 64   arm.as      LGPL3   as ARM Assembler (use RZ_ARM{32,64}_AS environment) (by pancake)
adAe  16 32 64   arm         BSD     Capstone ARM disassembler
adAe  8 16       avr         LGPL3   AVR Atmel
adA_  16 32 64   bf          LGPL3   Brainfuck (by pancake, nibble) v4.0.0
_dA_  32         chip8       LGPL3   Chip8 disassembler
_dA_  16 32 64   cil         LGPL3   .NET Common Intermediate Language
_dA_  16         cr16        LGPL3   cr16 disassembly plugin
adA_  32 64      dalvik      LGPL3   AndroidVM Dalvik
ad__  16         dcpu16      PD      Mojang's DCPU-16
_dA_  32 64      ebc         LGPL3   EFI Bytecode (by Fedor Sakharov)
adAe  16         gb          LGPL3   GameBoy(TM) (z80-like) (by condret)
_dAe  16         h8300       LGPL3   H8/300 disassembly plugin
_dA_  32         hexagon     LGPL3   Qualcomm Hexagon (QDSP6) V6 (by Rot127)
_dA_  4          i4004       LGPL3   Intel 4004 microprocessor
_dA_  8          i8080       BSD     Intel 8080 CPU
adA_  32         java        LGPL-3  Java bytecode disassembler (by deroad)
_d__  8          lh5801      LGPL3   SHARP LH5801 disassembler
_d__  32         lm32        BSD     disassembly plugin for Lattice Micro 32 ISA (by Felix Held)
adA_  8          luac        LGPL3   luac disassemble plugin
_dA_  32         m68k        BSD     Capstone M68K disassembler
_dA_  8 32       m680x       BSD     Capstone M680X Disassembler
_dA_  32         malbolge    LGPL3   Malbolge Ternary VM (by condret)
_dA_  32         mcore       LGPL3   Motorola MCORE disassembler
_d__  16         mcs96       LGPL3   condrets car
adAe  16 32 64   mips        BSD     Capstone MIPS disassembler
_dA_  16         msp430      LGPL3   msp430 disassembly plugin
adA_  16 32 64   null        MIT     no disassemble (by pancake) v1.0.0
_dA_  32         or1k        LGPL3   OpenRISC 1000
_dAe  8          pic         LGPL3   PIC disassembler
a___  32 64      ppc.as      LGPL3   as PPC Assembler (use RZ_PPC_AS environment) (by eagleoflqj)
_dAe  32 64      ppc         BSD     Capstone PowerPC disassembler (by pancake)
_dA_  32         propeller   LGPL3   propeller disassembly plugin
_dA_  8 16       pyc         LGPL3   PYC disassemble plugin
adA_  32         rl78        LGPL3   Renesas RL78 disassembler (by Bastian Engel)
_dA_  32         rsp         LGPL3   Reality Signal Processor
_dA_  32         rx          LGPL3   Renesas RX Family disassembler (by Heersin)
adAe  32         sh          LGPL3   SuperH-4 CPU (by DMaroo)
_dA_  8 16       snes        LGPL3   SuperNES CPU
_dA_  32 64      sparc       BSD     Capstone SPARC disassembler
_dA_  16         spc700      LGPL3   spc700, snes' sound-chip
_dA_  32 64      sysz        BSD     SystemZ CPU disassembler
_dA_  32         tms320      LGPLv3  TMS320 DSP family (c54x,c55x,c55x+,c64x)
_d__  32         tms320c64x  BSD     Capstone TMS320c64x disassembler
_dAe  32         v810        LGPL3   v810 disassembly plugin (by pancake)
_dAe  32         v850        LGPL3   v850 disassembly plugin
adA_  32         wasm        MIT     WebAssembly (by cgvwzq) v0.1.0
a___  16 32 64   x86.as      LGPL3   Intel X86 GNU Assembler (Use RZ_X86_AS env)
_dAe  16 32 64   x86         BSD     Capstone X86 disassembler
a___  16 32 64   x86.nasm    LGPL3   X86 nasm assembler
a___  16 32 64   x86.nz      LGPL3   x86 handmade assembler
_dA_  16         xap         PD      XAP4 RISC (CSR)
_dA_  32         xcore       BSD     Capstone XCore disassembler (by pancake)
_dAe  32 64      riscv.cs    BSD     Capstone RISCV disassembler
_dA_  32         tricore     BSD     Siemens TriCore CPU (by billow)
_dA_  16 32      arc         GPL3    Argonaut RISC Core
_dA_  32         cris        GPL3    Axis Communications 32-bit embedded processor (by pancake)
_d__  32         hppa        GPL3    HP PA-RISC
_d__  32         lanai       GPL3    LANAI
adAe  32 64      mips.gnu    GPL3    MIPS CPU
_dA_  32         nios2       GPL3    NIOS II Embedded Processor
_dAe  32 64      riscv       GPL3    RISC-V
_dA_  32 64      sparc.gnu   GPL3    Scalable Processor Architecture
_dA_  8 32       vax         GPL3    VAX
_dAe  32         xtensa      GPL3    XTensa CPU
adA_  8          z80         GPL3    Zilog Z80 (by condret)
```