25/01/2025 14:54 ASM : relecture de tentative

Accueil / Mes cours / ASM / Sections / Examen 1: 16/12/2024 / ASM Commencé le Monday 16 December 2024, 09:06 **État** Terminé Terminé le Monday 16 December 2024, 09:36 Temps mis 30 min 5 s **Points** 16,17/21,00 Note 15,40 sur 20,00 (76,98%) Question 1 Correct Note de 1,50 sur 1,50 What happens to the stack pointer (RSP) when a value is pushed onto the stack in x86-64 architecture? O b. Moves to a predefined address o. Increases by the size of the pushed value od. Stays unchanged Votre réponse est correcte. La réponse correcte est : Decreases by the size of the pushed value Ouestion 2 Incorrect Note de 0,00 sur 1,50 In one word, what is the difference between REPZ and REPE Condition Réponse : La réponse correcte est : None

	ASM : relecture de tentative
Question <b>3</b>	
Correct	
Note de 1,50 si	ur 1,50
Consider t	he following assembly code :
jge 0x4060	0d1
What is th	e name of the register on x86-64 that will be read by this instruction?
Réponse :	rflags
La réponse	e correcte est : rflags
Question <b>4</b>	
Correct	ur 1.50
	ur 1,50
Correct Note de 1,50 si	
Correct Note de 1,50 si	ister(s) are read during a REP instruction ?
Correct Note de 1,50 si	ister(s) are read during a REP instruction ?
Correct Note de 1,50 st Which reg	ister(s) are read during a REP instruction ?
Correct  Note de 1,50 so  Which reg  a. %	ister(s) are read during a REP instruction ? rdx r8
Correct  Note de 1,50 so  Which reg  a. % b. %  c. %	ister(s) are read during a REP instruction ? rdx r8 rcx❤️
Correct  Note de 1,50 so  Which reg  a. % b. %	ister(s) are read during a REP instruction ? rdx r8 rcx❤️
Which reg  a. % b. % c. % d. %	ister(s) are read during a REP instruction ?  rdx  r8  rcx  rflags
Which reg  a. % b. % c. % d. %	ister(s) are read during a REP instruction ? rdx r8 rcx❤️
Which reg  a. % b. % c. % d. %  Votre répo	ister(s) are read during a REP instruction ?  rdx  r8  rcx  rflags

```
Question 5
Correct
Note de 1,50 sur 1,50
```

Pick the most likely sample.

The next four questions pertain to the following four code samples.

```
f1
 f1:
            $8, %rsp
      subq
      call
            callfunc
      movl %eax, %edx
            1(%rax,%rax,2), %eax
      leal
      testb $1, %dl
             .L3
      jne
      movl
             %edx, %eax
            $31, %eax
       shrl
       addl %edx, %eax
       sarl
             %eax
 .L3:
       addq
             $8, %rsp
       ret
```

```
f2
 f2:
       pushq %rbx
      xorl %ebx, %ebx
 .L3:
       movl %ebx, %edi
       addl $1, %ebx
       call
            callfunc
            $10, %ebx
       cmpl
       jne
             .L3
              %rbx
       popq
       ret
```

```
f3
 f3:
                $8, %rsp
        subq
        call
                callfunc
        subl
                $97, %eax
        cmpb
                $4, %al
        ja
                .L2
                %al, %eax
        movzbl
        jmp
                 *.L4(,%rax,8)
 .L4:
                .L3
        .quad
                .L9
        .quad
        .quad
                .L6
        .quad
                .L7
        .quad
                .L8
 .L3:
        movl
                $42, %edx
        jmp
                .L5
 .L6:
                $4096, %edx
        movl
                .L5
        jmp
 .L7:
        movl
                $52, %edx
        jmp
                .L5
 .L8:
                $644<mark>0</mark>, %edx
        movl
        jmp
                .L5
 .L2:
        movl
                $0, %edx
                .L5
        jmp
 .L9:
                $61, %edx
        movl
 .L5:
                $.LC⊘, %esi
        movl
                $1, %edi
        movl
        movl
                $0, %eax
        call
                 __printf_chk
        addq
                $8, %rsp
        ret
 .LC0:
        .string "Sum = %d\n"
```

```
f4
f4:
       subq
               $40, %rsp
       movl
               $1, (%rsp)
       movl
               $0, 16(%rsp)
.L2:
              16(%rsp), %rsi
       leaq
       movq
              %rsp, %rdi
       call
              callfunc
       movl
              16(%rsp), %eax
              %eax, (%rsp)
       cmpl
       jne
               .L2
       addq
               $40, %rsp
       ret
```

```
Which sample contains a for loop?

Which sample contains a switch statement?

f3

w

Which sample contains a while loop?

The sample contains a while loop?

The sample contains a while loop?

The sample contains a while loop?
```

La réponse correcte est :					
Which sample contains a for loop? → f2,					
Which sample contains a switch statement? $\rightarrow$ f3,					
Which sample contains a while loop? → f4,					
Which sample contains only an if/else construct? → f1	Which sample contains only an if/else construct? → f1				
Question <b>6</b>					
Correct  Note de 1,50 sur 1,50					
A user assembly program pushes the syscall function address on to	the stack prior calling the syscall instruction.				
Veuillez choisir une réponse.					
○ Vrai ● Faux ✔					
● raux ▼					
La réponse correcte est « Faux ».					
Question <b>7</b>					
Correct					
Note de 1,00 sur 1,00					
Indicate the most likely type(s) of the data being accessed: mov1 -2	8(%rbp), %rdx.				
a. Array of int or unsigned int					
b. char *					
c. int					
d. We can not know					
e. unsigned short					
f. unsigned char					
☑ g. It does not compile ✓					
La vánanca covvacta act :					
La renonce correcte est.					
La réponse correcte est : It does not compile					

Indicate the most likely type(s) of the data being accessed: movb -8(%rbp), %d1.    a. We can not know     b. Array of int or unsigned int     c. unsigned char*     d. unsigned short     e. int     f. It does not compile     g. char*     La réponse correcte est : unsigned char     Which register is most likely to be used for the long data type?     a. rdx*     b. b. x     c. eax     d. the answer is not listed     e. cl     La réponse correcte est : rdx     c. cax     c. cax     d. the answer is not listed     e. cd     cax     cax	Question <b>8</b>	
Indicate the most likely type(s) of the data being accessed: nove -s(3rbp), 3d1.  a. We can not know b. Array of int or unsigned int c. unsigned char  d. unsigned short e. int f. It does not compile g. char*  La réponse correcte est : unsigned char  Which register is most likely to be used for the long data type?  a. rdx b. bx c. eax d. the answer is not listed e. c. d  La réponse correcte est : rdx  Which register is most likely to be used for the char data type?  a. rdx b. b. bx c. ax d. the answer is not listed e. c. d  Which register is most likely to be used for the char data type?  a. rdx b. b. bs c. c. ax d. d. the answer is not listed e. c. d  d. the answer is not listed e. c. d  d. the answer is not listed e. e. eax	Correct	00 1 00
a. We can not know b. Array of int or unsigned int c. unsigned char* d. unsigned short e. int f. It does not compile g. char*  La réponse correcte est : unsigned char  Which register is most likely to be used for the long data type? a. rok* b. bx c. c. eax d. the answer is not listed e. cl  La réponse correcte est : rdx  Which register is most likely to be used for the char data type?  Authority of the answer is not listed e. cl  Which register is most likely to be used for the char data type?  a. rok  La réponse correcte est : rdx  Which register is most likely to be used for the char data type?  d. c. cd d. the answer is not listed e. c. cd d. the answer is not listed e. c. dat e. dat e. c. dat	Note de 1,0	Ju sur 1,00
b. Array of int or unsigned int c. unsigned char* d. unsigned short e. int f. It does not compile g. char*  La réponse correcte est : unsigned char  Which register is most likely to be used for the long data type? a. a. rdx* b. b. bx c. eax d. the answer is not listed e. c. d  Which register is most likely to be used for the char data type?  Which register is most likely to be used for the char data type?  a. rdx  Descent 10  Which register is most likely to be used for the char data type?  a. rdx  Descent 10  Which register is most likely to be used for the char data type?  a. rdx b. bx c. c. dx d. the answer is not listed e. e. eax	Indicate	e the most likely type(s) of the data being accessed: movb -8(%rbp), %dl.
b. Array of int or unsigned int c. unsigned char* d. unsigned short e. int f. It does not compile g. char*  La réponse correcte est : unsigned char  Which register is most likely to be used for the long data type? a. a. rdx* b. b. bx c. eax d. the answer is not listed e. c. d  Which register is most likely to be used for the char data type?  Which register is most likely to be used for the char data type?  a. rdx  Descent 10  Which register is most likely to be used for the char data type?  a. rdx  Descent 10  Which register is most likely to be used for the char data type?  a. rdx b. bx c. c. dx d. the answer is not listed e. e. eax	Па	We can not know
c unsigned char* d unsigned short e int f, It does not compile g, char*  La réponse correcte est : unsigned char  Which register is most likely to be used for the long data type? a rok* b bx c eax d the answer is not listed e c cl  La réponse correcte est : rdx  Which register is most likely to be used for the long data type? a rok* b bx c ax d the answer is not listed e c cl  Which register is most likely to be used for the char data type? a rok* b bx c c eax d the answer is not listed e c cl		
d. unsigned short e. int f. It does not compile g. char *  La réponse correcte est : unsigned char    Duestion 9		
e. int f. It does not compile g. char*  La réponse correcte est : unsigned char  Which register is most likely to be used for the long data type? a. rdx* b. bx c. e. eax d. the answer is not listed e. cl  La réponse correcte est : rdx  Multich register is most likely to be used for the char data type?  Which register is most likely to be used for the char data type?  a. rdx  b. bx c. c. eax  d. the answer is not listed e. cl		
f. It does not compile   g. char*  La réponse correcte est : unsigned char  Which register is most likely to be used for the long data type?  a. rdx ✓ b. bx c. c. eax d. d. the answer is not listed e. c. d  La réponse correcte est : rdx  Which register is most likely to be used for the char data type?  Which register is most likely to be used for the long data type?  a. rdx ✓ b. bx c. c. eax d. d. the answer is not listed e. c. d  Which register is most likely to be used for the char data type?  a. rdx b. bx c. c. d  d. the answer is not listed e. e. eax		
□ g. char*  La réponse correcte est : unsigned char  Decesion 9  Ornect  Late de 1,00 sur 1,00  Which register is most likely to be used for the long data type?  □ a. rdx ✓  □ b. bx  □ c. eax  □ d. the answer is not listed  □ e. cl  La réponse correcte est : rdx  Which register is most likely to be used for the char data type?  □ a. rdx  □ b. bx  □ c. ex  □ d. the answer is not likely to be used for the char data type?  □ a. rdx  □ b. bx  □ c. cl  □ d. the answer is not listed  □ d. the answer is not listed  □ e. ex		
La réponse correcte est : unsigned char  Desentor 9  Orient Interest is most likely to be used for the long data type?  a. rdx b. b. bx c. c. eax d. the answer is not listed e. cl  La réponse correcte est : rdx  Austron 10  Orient Interest is most likely to be used for the char data type?  a. rdx  Austron 10  Orient Interest is most likely to be used for the char data type?  a. rdx b. b. bx c. cl  d. the answer is not likely to be used for the char data type?  a. rdx b. b. bx c. cl  d. the answer is not listed e. e. eax		
Doesdoon 9  Ornect  Jose de 1,00 sur 1,00  Which register is most likely to be used for the long data type?  a. rdx*  b. b. bx  c. eax  d. the answer is not listed  e. cl  La réponse correcte est : rdx   Number of 10  Ornect  Jose de 1,00 sur 1,00  Which register is most likely to be used for the char data type?  a. rdx  b. bx  c. c. cl*  d. the answer is not listed  e. e. eax	_ g.	
Doesdoon 9  Ornect  Jose de 1,00 sur 1,00  Which register is most likely to be used for the long data type?  a. rdx*  b. b. bx  c. eax  d. the answer is not listed  e. cl  La réponse correcte est : rdx   Number of 10  Ornect  Jose de 1,00 sur 1,00  Which register is most likely to be used for the char data type?  a. rdx  b. bx  c. c. cl*  d. the answer is not listed  e. e. eax	. ,	
Which register is most likely to be used for the long data type?  a. rdx b. b. bx c. eax d. the answer is not listed e. cl  La réponse correcte est : rdx   Which register is most likely to be used for the char data type?  Which register is most likely to be used for the char data type?  a. rdx b. bx c. cl  d. the answer is not listed e. c. cl	La répo	onse correcte est : unsigned char
Which register is most likely to be used for the long data type?  a. rdx b. b. bx c. eax d. the answer is not listed e. cl  La réponse correcte est : rdx   Which register is most likely to be used for the char data type?  Which register is most likely to be used for the char data type?  a. rdx b. bx c. cl  d. the answer is not listed e. c. cl		
Which register is most likely to be used for the long data type?  a. rdx b. bx c. eax d. the answer is not listed e. cl  La réponse correcte est : rdx  Which register is most likely to be used for the char data type?  a. rdx b. bx c. c. d  Which register is most likely to be used for the char data type?  a. rdx b. bx c. cl d. the answer is not listed e. e. eax		
Which register is most likely to be used for the long data type?  a. rdx  b. bx  c. eax  d. the answer is not listed  e. cl  La réponse correcte est : rdx   Duestion 10  Correct  Lorde de 1,00 sur 1,00   Which register is most likely to be used for the char data type?  a. rdx  b. bx  c. cl  d. the answer is not listed  e. c. cl  d. the answer is not listed  e. eax		00 sur 1.00
<ul> <li>a. rdx*</li> <li>b. bx</li> <li>c. eax</li> <li>d. the answer is not listed</li> <li>e. cl</li> </ul> La réponse correcte est: rdx  Question 10  Correct  Lote de 1,00 sur 1,00  Which register is most likely to be used for the char data type? <ul> <li>a. rdx</li> <li>b. bx</li> <li>c. cl*</li> <li>d. the answer is not listed</li> <li>e. eax</li> </ul>		
c. eax d. the answer is not listed e. cl  La réponse correcte est : rdx  Question 10 Correct Rote de 1,00 sur 1,00  Which register is most likely to be used for the char data type?  a. rdx b. bx c. cl d. the answer is not listed e. eax	a.	rdx <b>❤</b>
d. the answer is not listed e. cl  La réponse correcte est : rdx  Question 10  Correct  Lote de 1,00 sur 1,00  Which register is most likely to be used for the char data type?  a. rdx b. bx c. cl d. the answer is not listed e. eax	O b.	bx
e. cl  La réponse correcte est : rdx  Duestion 10  Correct  Lote de 1,00 sur 1,00  Which register is most likely to be used for the char data type?  a. rdx b. bx c. cl  d. the answer is not listed e. eax		
La réponse correcte est : rdx  Question 10 Correct  Note de 1,00 sur 1,00  Which register is most likely to be used for the char data type?  a. rdx b. bx c. cl* d. the answer is not listed e. eax		
Apuestion 10  Forrect  Note de 1,00 sur 1,00  Which register is most likely to be used for the char data type?  a. rdx b. bx c. cl d. the answer is not listed e. eax	О е.	cl
Apuestion 10  Forrect  Note de 1,00 sur 1,00  Which register is most likely to be used for the char data type?  a. rdx b. bx c. cl d. the answer is not listed e. eax		
Which register is most likely to be used for the char data type?  a. rdx b. bx c. cl d. the answer is not listed e. eax	La répo	onse correcte est : rdx
Which register is most likely to be used for the char data type?  a. rdx b. bx c. cl d. the answer is not listed e. eax	Ouestion 1	<u> </u>
Which register is most likely to be used for the char data type?  a. rdx b. bx c. cl d. the answer is not listed e. eax	Correct	
<ul> <li>a. rdx</li> <li>b. bx</li> <li>c. cl ✓</li> <li>d. the answer is not listed</li> <li>e. eax</li> </ul>	Note de 1,0	
<ul> <li>a. rdx</li> <li>b. bx</li> <li>c. cl ✓</li> <li>d. the answer is not listed</li> <li>e. eax</li> </ul>		20 sur 1,00
<ul> <li>b. bx</li> <li>c. cl ✓</li> <li>d. the answer is not listed</li> <li>e. eax</li> </ul>		00 sur 1,00
<ul> <li>○ c. cl</li> <li>○ d. the answer is not listed</li> <li>○ e. eax</li> </ul>	Which	
<ul> <li>○ c. cl</li> <li>○ d. the answer is not listed</li> <li>○ e. eax</li> </ul>		register is most likely to be used for the char data type?
<ul><li>d. the answer is not listed</li><li>e. eax</li></ul>	О а.	register is most likely to be used for the char data type? rdx
○ e. eax	<ul><li>a.</li><li>b.</li></ul>	register is most likely to be used for the char data type?  rdx  bx
	<ul><li>a.</li><li>b.</li><li>c.</li></ul>	register is most likely to be used for the char data type?  rdx  bx  cl
La réponse correcte est : cl	<ul><li>a.</li><li>b.</li><li>c.</li><li>d.</li></ul>	register is most likely to be used for the char data type?  rdx  bx  cl  the answer is not listed
La réponse correcte est : cl	<ul><li>a.</li><li>b.</li><li>c.</li><li>d.</li></ul>	register is most likely to be used for the char data type?  rdx  bx  cl  the answer is not listed
	<ul><li>a.</li><li>b.</li><li>c.</li><li>d.</li></ul>	register is most likely to be used for the char data type?  rdx  bx  cl  the answer is not listed

Question 11	
Correct Note de 0,50 sur 0	,50
How many by	yte(s) can be stored in the register %ebx?
Réponse :	<b>√</b>
La réponse co	orrecte est : 4
Question <b>12</b>	
Correct Note de 0,50 sur 0	,50
How many by	yte(s) can be stored in the register %r9d?
Réponse :	4
La réponse co	prrecte est : 4
Question 13	
Correct	
Note de 0,50 sur 0	.50
How many by	yte(s) can be stored in the register %eax?
Réponse :	4
La réponse co	orrecte est : 4
Question <b>14</b> Correct	
Note de 0,50 sur 0	,50
How many by	yte(s) can be stored in the register %edx?
Réponse :	4
La réponse co	orrecte est · 4
La reponse C	

25/0

2025 14:54		ASM : relecture de tentative
Question <b>1</b>	5	
Incorrect		
Note de 0,0	0 sur 1,00	
What is	the idiomatic way to get the quotient of	f an unsigned integer division by two in x86 assembly ?
<ul><li>a.</li></ul>	shr \$1, %rax	
O b.	movq \$2, %rbx idivq %rbx	
O c.	sal \$1, %rax	
O d.	shl \$1, %rax	
e.	sar \$1, %rax 🗙	
shr \$1,		
Correct		
Note de 1,0	0 sur 1,00	
	function call, the stack is <b>always</b> unaligned choisir une réponse.	ed.
Fau:		
● rau	(♥	
La répo	nse correcte est « Faux ».	

Question 17	
Correct	
Note de 1,00 sur 1,00	

Is the following code valid according to the ABI?

```
.global main
.text
main:
   push %rbp
   mov %rsp, %rbp
   mov str, %rdi
   call puts
   mov $0, %eax
   pop %rbp
   ret
   .section .data
str:
   .asciz "Hello!"
```

Veuillez choisir une réponse.

Vrai

■ Faux

The issue is:

mov str, %rdi

It copies the string's bytes in to %rdi, but puts expect a pointer.\

Correct code is:

mov \$str, %rdi

In this case the source will as well be an immediate value but with the address of the label str which is a pointer.

La réponse correcte est « Faux ».

Question 18
Incorrect
Note de 0,00 sur 1,00

Is the following stack pointer aligned?

rsp: 0x7ffffff8884

Veuillez choisir une réponse.

Vrai X

Faux

La réponse correcte est « Faux ».

```
Question 19
Partiellement correct
Note de 0,67 sur 2,00
```

In the following questions, we give you C code and a portion of the assembly generated by some compiler for that code. (Sometimes we blank out a part of the assembly.) The C code contains a variable, constant, or function called waldo, and a point in the assembly is marked with asterisks \*\*\*. Your job is to find Waldo: write an **assembly expression or constant** that holds the value of waldo at the marked point. We've done the first one for you.

## NON-QUESTION: Where's Waldo?

```
int identity(int waldo) {
   return waldo;
}
```

```
00000000004007f6 <identity>:
 4007f6:
                                 push %rbp
 4007f7:
           48 89 e5
                                 mov
                                       %rsp,%rbp
 4007fa:
           89 7d fc
                                  mov
                                      %edi,-0x4(%rbp)
 4007fd:
             8b 45 fc
                                       -0x4(%rbp),%eax
                                  mov
 400800:
             5d
                                  pop
 400801:
             с3
                                  retq
```

**ANSWER**: %edi, -0x4(%rbp), %eax, and %rax all hold the value of waldo at the marked point, so any of them is a valid answer. If the asterisks came before the *first* instruction, only %edi would work.

```
int int_array_get(int* a, int waldo) {
   int x = a[waldo];
   return x;
}
```

```
00000000004007d9 <int_array_get>:
INSTRUCTIONS OMITTED

***

4007dc: 8b 04 b7 mov (%rdi,%rsi,4),%eax
4007df: c3 retq
```

```
int matrix_get(int** matrix, int row, int col) {
   int* waldo = matrix[row];
   return waldo[col];
}
```

```
00000000004007e0 <matrix get>:
4007e0: 48 63 f6
                                   movslq %esi,%rsi
                                                              %rsi
4007e3:
                                   movslq %edx,%rdx
             48 63 d2
4007e6:
             33 33 33 33
                                          ??,%rax
                                   mov
             8b 04 90
                                          (%rax,%rdx,4),%eax
4007ea:
                                   mov
4007ed:
             с3
                                   retq
```

```
int f1(int a, int b, int waldo, int d) {
   if (a > b) {
      return waldo;
   } else {
      return d;
   }
}
```

```
0000000000400802 <f1>:
 400802:
                                    push %rbp
 400803:
              48 89 e5
                                           %rsp,%rbp
                                    mov
 400806:
              89 7d fc
                                    mov
                                           %edi,-0x4(%rbp)
                                                              %edx
 400809:
             89 75 f8
                                         %esi,-0x8(%rbp)
                                    mov
 40080c:
             89 55 f4
                                           %edx,-0xc(%rbp)
                                    mov
 40080f:
             89 4d f0
                                           %ecx,-0x10(%rbp)
                                    mov
 400812:
             8b 45 fc
                                           -0x4(%rbp),%eax
             3b 45 f8
 400815:
                                    cmp
                                           -0x8(%rbp),%eax
             7e 05
                                         40081f <f1+0x1d>
 400818:
                                    jle
 40081a:
            8b 45 f4
                                          -0xc(%rbp),%eax
                                    mov
 40081d:
             eb 03
                                    qmr
                                         400822 <f1+0x20>
 40081f:
            8b 45 f0
                                    mov
                                           -0x10(%rbp),%eax
 400822:
             5d
                                    pop
                                           %rbp
 400823:
            c3
                                    reta
```

La réponse correcte est :

```
int int_array_get(int* a, int waldo) {
  int x = a[waldo];
  return x;
}
```

→ %rsi,

```
int matrix_get(int** matrix, int row, int col) {
   int* waldo = matrix[row];
   return waldo[col];
}
```

```
00000000004007e0 <matrix_get>:
4007e0:
              48 63 f6
                                     movslq %esi,%rsi
4007e3:
              48 63 d2
                                      movslq %edx,%rdx
4007e6:
              33 33 33 33
                                             ??,%rax
                                      mov
                                             (%rax,%rdx,4),%eax
4007ea:
              8b 04 90
                                      mov
4007ed:
                                      retq
```

→ (%rdi,%rsi,8),

```
int f1(int a, int b, int waldo, int d) {
   if (a > b) {
      return waldo;
   } else {
      return d;
   }
}
```

```
0000000000400802 <f1>:
 400802:
                                  push %rbp
 400803: 48 89 e5
                                  mov %rsp,%rbp
 400806: 89 7d fc
                                  mov %edi,-0x4(%rbp)
 400809: 89 75 f8
40080c: 89 55 f4
40080f: 89 4d f0
                                  mov %esi,-0x8(%rbp)
                                         %edx,-0xc(%rbp)
                                   mov
                                   mov
                                         %ecx,-0x10(%rbp)
 400812:
             8b 45 fc
                                   mov
                                         -0x4(%rbp),%eax
             3b 45 f8
                                         -0x8(%rbp),%eax
 400815:
                                   cmp
                                        40081f <f1+0x1d>
 400818:
            7e 05
                                   jle
 40081a:
           8b 45 f4
                                         -0xc(%rbp),%eax
                                   mov
 40081d:
           eb 03
                                  jmp
                                        400822 <f1+0x20>
 40081f:
            8b 45 f0
                                          -0x10(%rbp),%eax
                                   mov
 400822:
             5d
                                          %rbp
                                   pop
 400823:
           c3
                                   reta
```

→ %edx

## ■ Linux System V Syscall Table

Aller à...