Objectives - Tue 4/26/2022

- Endianness
- Disclaimer
- Stack
- Function calls
- Buffer overflow attacks

Endianness

16,909,060

= sixteen-million nine-hundred nine thousand sixty

$= 1 \times 2^{24} + 2 \times 2^{16} + 3 \times 2^{8} + 4 \times 2^{0}$

= 0 x 0 1 0 2 0 3 0 4

Every two hex digits represent a byte.

= 0 x 0 1 0 2 0 3 0 4

0 x 0 1 0 2 0 3 0 4

Leftmost byte is the "most significant".

0 x 0 1 0 2 0 3 0 4

↑

most significant

Rightmost byte is the "least significant".

0 x 0 1 0 2 0 3 0 4

↑

least significant

Left and right matter only if you are human.

Endianness

Humans write numbers starting with the most significant digit.

Most computers store integers in memory starting with the least significant digits.

0 x 0 1 0 2 0 3 0 4 would be stored in memory as 0x04 0x03 0x02 0x01.

Storing integers in this way is called…

little endian.

Endianness

Little endian byte order stores 0x01020304 in memory as 0x04 0x03 0x02 0x01.

Most computers (Windows, Mac, Linux) and mobile devices (Android, iOS) use little endian.

Big endian byte order stores 0x01020304 in memory as 0x01 0x02 0x03 0x04.

Only a few exotic computers use big endian.
Humans essentially use big endian, too.

How to remember…

- Big humans → big endian
- Microcomputer → little endian

Endianness only affects the byte order with which integers are stored in memory.

- ✓ Endianness affects int.
- ✓ Endianness affects long.
- ✓ Endianness affects addresses.
- ✗ Endianness does not affect bits within a byte.
- ✗ Endianness does not affect hex digits within a byte.
- ✗ Endianness does not affect arrays.
- ✗ Endianness does not affect strings.
- ✗ Endianness does not affect struct fields.

Disclaimer

Stack

Function calls

Buffer overflow attacks

Meet boa.c

```c
#include <stdio.h>

void greet_visitor() {
    int gt_top = 0xAAAAAAAA;
    char name[10]; // BAD!!!
    printf("Hello.  What is your name?\n");
    gets(name);    // VERY, VERY, VERY BAD!!!
    printf("Hello, %s.", name);
    int gt_btm = 0xAABBBBAA;
}

void scare_visitor() {
    int sv_top = 0xAACCCCAA;
    char message[10] = "BRAH!!!\n"; // OKAY
    printf(message);
    int sv_btm = 0xAADDDDAA;
}

int main(int argc, char *argv[]) {
    int mn_top = 0xAAEEEEAA;
    greet_visitor();
    int mn_btm = 0xAAFFFFAA;
    return 0;
}
```

main(…) » Disassembly

```
19      int main(int argc, char *argv[]) {
   0x00000000004005d8 <+0>:     push   %rbp
   0x00000000004005d9 <+1>:     mov    %rsp,%rbp
   0x00000000004005dc <+4>:     sub    $0x20,%rsp
   0x00000000004005e0 <+8>:     mov    %edi,-0x14(%rbp)
   0x00000000004005e3 <+11>:    mov    %rsi,-0x20(%rbp)

20              int mn_top = 0xAAEEEEAA;
   0x00000000004005e7 <+15>:    movl   $0xaaeeeeaa,-0x8(%rbp)

21              greet_visitor();
   0x00000000004005ee <+22>:    mov    $0x0,%eax
=> 0x00000000004005f3 <+27>:    callq  0x400554 <greet_visitor>

22              int mn_btm = 0xAAFFFFAA;
   0x00000000004005f8 <+32>:    movl   $0xaaffffaa,-0x4(%rbp)

23              return 0;
   0x00000000004005ff <+39>:    mov    $0x0,%eax

24      }
   0x0000000000400604 <+44>:    leaveq
   0x0000000000400605 <+45>:    retq
```

prologue of main(…)

epilogue of main(…)

main(…) » Assembly language instructions

```
19      int main(int argc, char *argv[]) {
   0x0000000004005d8 <+0>:     push   %rbp
   0x0000000004005d9 <+1>:     mov    %rsp,%rbp
   0x0000000004005dc <+4>:     sub    $0x20,%rsp
   0x0000000004005e0 <+8>:     mov    %edi,-0x14(%rbp)
   0x0000000004005e3 <+11>:    mov    %rsi,-0x20(%rbp)

20              int mn_top = 0xAAEEEEAA;
   0x0000000004005e7 <+15>:    movl   $0xaaeeeeaa,-0x8(%rbp)

21              greet_visitor();
   0x0000000004005ee <+22>:    mov    $0x0,%eax
=> 0x0000000004005f3 <+27>:    callq  0x400554 <greet_visitor>

22              int mn_btm = 0xAAFFFFAA;
   0x0000000004005f8 <+32>:    movl   $0xaaffffaa,-0x4(%rbp)

23              return 0;
   0x0000000004005ff <+39>:    mov    $0x0,%eax

24      }
   0x0000000000400604 <+44>:    leaveq
   0x0000000000400605 <+45>:    retq
```

prologue of main(…)

epilogue of main(…)

main(…) » Address of each instruction (in text segment)

```
19      int main(int argc, char *argv[]) {
   0x0000000004005d8 <+0>:     push   %rbp
   0x0000000004005d9 <+1>:     mov    %rsp,%rbp
   0x0000000004005dc <+4>:     sub    $0x20,%rsp
   0x0000000004005e0 <+8>:     mov    %edi,-0x14(%rbp)
   0x0000000004005e3 <+11>:    mov    %rsi,-0x20(%rbp)

20              int mn_top = 0xAAEEEEAA;
   0x0000000004005e7 <+15>:    movl   $0xaaeeeeaa,-0x8(%rbp)

21              greet_visitor();
   0x0000000004005ee <+22>:    mov    $0x0,%eax
=> 0x0000000004005f3 <+27>:    callq  0x400554 <greet_visitor>

22              int mn_btm = 0xAAFFFFAA;
   0x0000000004005f8 <+32>:    movl   $0xaaffffaa,-0x4(%rbp)

23              return 0;
   0x0000000004005ff <+39>:    mov    $0x0,%eax

24      }
   0x0000000000400604 <+44>:    leaveq
   0x0000000000400605 <+45>:    retq
```

prologue of main(…)

epilogue of main(…)

main(…) » Memory offset from beginning of function

```
19      int main(int argc, char *argv[]) {
   0x00000000004005d8 <+0>:     push   %rbp
   0x00000000004005d9 <+1>:     mov    %rsp,%rbp
   0x00000000004005dc <+4>:     sub    $0x20,%rsp
   0x00000000004005e0 <+8>:     mov    %edi,-0x14(%rbp)
   0x00000000004005e3 <+11>:    mov    %rsi,-0x20(%rbp)

20              int mn_top = 0xAAEEEEAA;
   0x00000000004005e7 <+15>:    movl   $0xaaeeeeaa,-0x8(%rbp)

21              greet_visitor();
   0x00000000004005ee <+22>:    mov    $0x0,%eax
=> 0x00000000004005f3 <+27>:    callq  0x400554 <greet_visitor>

22              int mn_btm = 0xAAFFFFAA;
   0x00000000004005f8 <+32>:    movl   $0xaaffffaa,-0x4(%rbp)

23              return 0;
   0x00000000004005ff <+39>:    mov    $0x0,%eax

24      }
   0x0000000000400604 <+44>:    leaveq
   0x0000000000400605 <+45>:    retq
```

prologue of main(…)

epilogue of main(…)

main(…) » Prologue

```
19      int main(int argc, char *argv[]) {
   0x0000000004005d8 <+0>:     push   %rbp
   0x0000000004005d9 <+1>:     mov    %rsp,%rbp
   0x0000000004005dc <+4>:     sub    $0x20,%rsp
   0x0000000004005e0 <+8>:     mov    %edi,-0x14(%rbp)
   0x0000000004005e3 <+11>:    mov    %rsi,-0x20(%rbp)

20              int mn_top = 0xAAEEEEAA;
   0x0000000004005e7 <+15>:    movl   $0xaaeeeeaa,-0x8(%rbp)

21              greet_visitor();
   0x0000000004005ee <+22>:    mov    $0x0,%eax
=> 0x0000000004005f3 <+27>:    callq  0x400554 <greet_visitor>

22              int mn_btm = 0xAAFFFFAA;
   0x0000000004005f8 <+32>:    movl   $0xaaffffaa,-0x4(%rbp)

23              return 0;
   0x0000000004005ff <+39>:    mov    $0x0,%eax

24      }
   0x0000000000400604 <+44>:    leaveq
   0x0000000000400605 <+45>:    retq
```

prologue of main(…)

epilogue of main(…)

main(…) » Initialize local variable mn_top to 0xaaeeeeaa

```
19      int main(int argc, char *argv[]) {
   0x00000000004005d8 <+0>:     push   %rbp
   0x00000000004005d9 <+1>:     mov    %rsp,%rbp
   0x00000000004005dc <+4>:     sub    $0x20,%rsp
   0x00000000004005e0 <+8>:     mov    %edi,-0x14(%rbp)
   0x00000000004005e3 <+11>:    mov    %rsi,-0x20(%rbp)
```

} prologue of main(…)

```
20              int mn_top = 0xAAEEEEAA;
   0x00000000004005e7 <+15>:    movl   $0xaaeeeeaa,-0x8(%rbp)

21              greet_visitor();
   0x00000000004005ee <+22>:    mov    $0x0,%eax
=> 0x00000000004005f3 <+27>:    callq  0x400554 <greet_visitor>

22              int mn_btm = 0xAAFFFFAA;
   0x00000000004005f8 <+32>:    movl   $0xaaffffaa,-0x4(%rbp)

23              return 0;
   0x00000000004005ff <+39>:    mov    $0x0,%eax

24      }
   0x0000000000400604 <+44>:    leaveq
   0x0000000000400605 <+45>:    retq
```

} epilogue of main(…)

main(…) » call function greet_visitor(…)

```
19      int main(int argc, char *argv[]) {
   0x00000000004005d8 <+0>:     push   %rbp
   0x00000000004005d9 <+1>:     mov    %rsp,%rbp
   0x00000000004005dc <+4>:     sub    $0x20,%rsp
   0x00000000004005e0 <+8>:     mov    %edi,-0x14(%rbp)
   0x00000000004005e3 <+11>:    mov    %rsi,-0x20(%rbp)

20              int mn_top = 0xAAEEEEAA;
   0x00000000004005e7 <+15>:    movl   $0xaaeeeeaa,-0x8(%rbp)

21              greet_visitor();
   0x00000000004005ee <+22>:    mov    $0x0,%eax
=> 0x00000000004005f3 <+27>:    callq  0x400554 <greet_visitor>

22              int mn_btm = 0xAAFFFFAA;
   0x00000000004005f8 <+32>:    movl   $0xaaffffaa,-0x4(%rbp)

23              return 0;
   0x00000000004005ff <+39>:    mov    $0x0,%eax

24      }
   0x0000000000400604 <+44>:    leaveq
   0x0000000000400605 <+45>:    retq
```

prologue of main(…)

epilogue of main(…)

main(…) » Initialize local variable mn_btm to 0xaaffffaa

```
19      int main(int argc, char *argv[]) {
   0x00000000004005d8 <+0>:     push   %rbp
   0x00000000004005d9 <+1>:     mov    %rsp,%rbp
   0x00000000004005dc <+4>:     sub    $0x20,%rsp
   0x00000000004005e0 <+8>:     mov    %edi,-0x14(%rbp)
   0x00000000004005e3 <+11>:    mov    %rsi,-0x20(%rbp)

20              int mn_top = 0xAAEEEEAA;
   0x00000000004005e7 <+15>:    movl   $0xaaeeeeaa,-0x8(%rbp)

21              greet_visitor();
   0x00000000004005ee <+22>:    mov    $0x0,%eax
=> 0x00000000004005f3 <+27>:    callq  0x400554 <greet_visitor>

22              int mn_btm = 0xAAFFFFAA;
   0x00000000004005f8 <+32>:    movl   $0xaaffffaa,-0x4(%rbp)

23              return 0;
   0x00000000004005ff <+39>:    mov    $0x0,%eax

24      }
   0x0000000000400604 <+44>:    leaveq
   0x0000000000400605 <+45>:    retq
```

prologue of main(…)

epilogue of main(…)

main(…) » Place return value in %eax register

```
19      int main(int argc, char *argv[]) {
   0x00000000004005d8 <+0>:     push   %rbp
   0x00000000004005d9 <+1>:     mov    %rsp,%rbp
   0x00000000004005dc <+4>:     sub    $0x20,%rsp
   0x00000000004005e0 <+8>:     mov    %edi,-0x14(%rbp)
   0x00000000004005e3 <+11>:    mov    %rsi,-0x20(%rbp)

20              int mn_top = 0xAAEEEEAA;
   0x00000000004005e7 <+15>:    movl   $0xaaeeeeaa,-0x8(%rbp)

21              greet_visitor();
   0x00000000004005ee <+22>:    mov    $0x0,%eax
=> 0x00000000004005f3 <+27>:    callq  0x400554 <greet_visitor>

22              int mn_btm = 0xAAFFFFAA;
   0x00000000004005f8 <+32>:    movl   $0xaaffffaa,-0x4(%rbp)

23              return 0;
   0x00000000004005ff <+39>:    mov    $0x0,%eax

24      }
   0x0000000000400604 <+44>:    leaveq
   0x0000000000400605 <+45>:    retq
```

prologue of main(…)

epilogue of main(…)

main(…) » Epilogue

```
19      int main(int argc, char *argv[]) {
   0x0000000004005d8 <+0>:     push   %rbp
   0x0000000004005d9 <+1>:     mov    %rsp,%rbp
   0x0000000004005dc <+4>:     sub    $0x20,%rsp
   0x0000000004005e0 <+8>:     mov    %edi,-0x14(%rbp)
   0x0000000004005e3 <+11>:    mov    %rsi,-0x20(%rbp)

20              int mn_top = 0xAAEEEEAA;
   0x0000000004005e7 <+15>:    movl   $0xaaeeeeaa,-0x8(%rbp)

21              greet_visitor();
   0x0000000004005ee <+22>:    mov    $0x0,%eax
=> 0x0000000004005f3 <+27>:    callq  0x400554 <greet_visitor>

22              int mn_btm = 0xAAFFFFAA;
   0x0000000004005f8 <+32>:    movl   $0xaaffffaa,-0x4(%rbp)

23              return 0;
   0x0000000004005ff <+39>:    mov    $0x0,%eax

24      }
   0x0000000000400604 <+44>:    leaveq
   0x0000000000400605 <+45>:    retq
```

prologue of main(…)

epilogue of main(…)

main(…) » Before calling greet_visitor(…)

```
19      int main(int argc, char *argv[]) {
   0x00000000004005d8 <+0>:     push   %rbp
   0x00000000004005d9 <+1>:     mov    %rsp,%rbp
   0x00000000004005dc <+4>:     sub    $0x20,%rsp
   0x00000000004005e0 <+8>:     mov    %edi,-0x14(%rbp)
   0x00000000004005e3 <+11>:    mov    %rsi,-0x20(%rbp)

20              int mn_top = 0xAAEEEEAA;
   0x00000000004005e7 <+15>:    movl   $0xaaeeeeaa,-0x8(%rbp)

21              greet_visitor();
   0x00000000004005ee <+22>:    mov    $0x0,%eax
=> 0x00000000004005f3 <+27>:    callq  0x400554 <greet_visitor>

22              int mn_btm = 0xAAFFFFAA;
   0x00000000004005f8 <+32>:    movl   $0xaaffffaa,-0x4(%rbp)

23              return 0;
   0x00000000004005ff <+39>:    mov    $0x0,%eax

24      }
   0x0000000000400604 <+44>:    leaveq
   0x0000000000400605 <+45>:    retq
```

prologue of main(…)

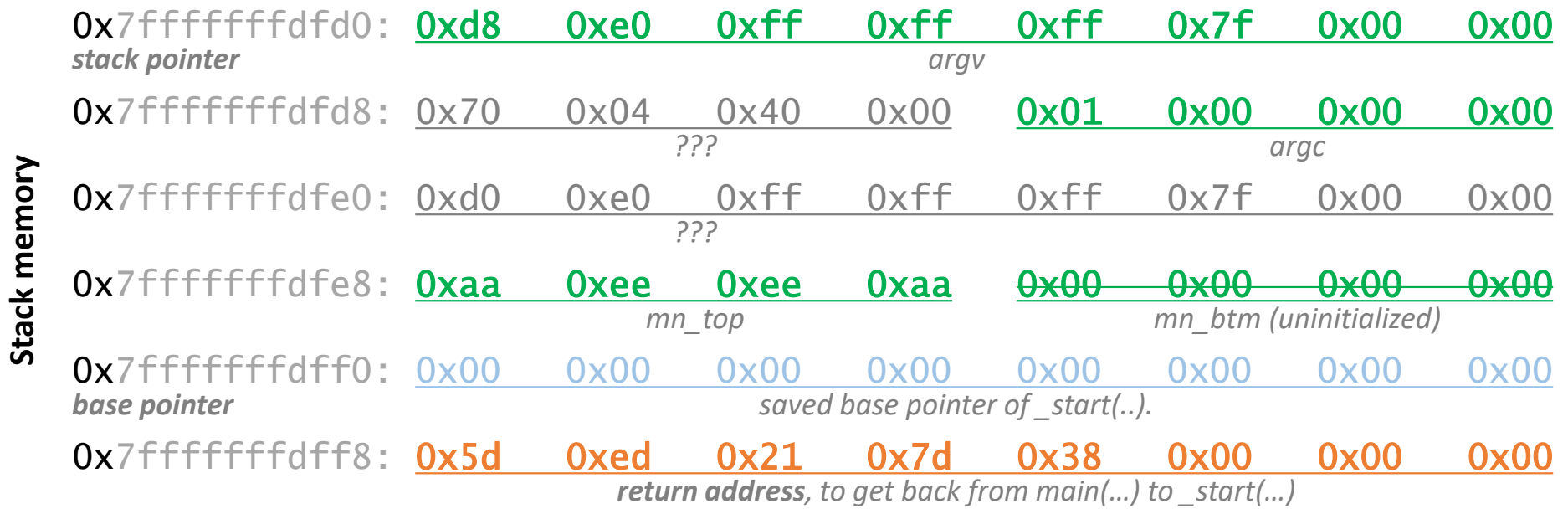
epilogue of main(…)

main(…) » Before calling greet_visitor(…)

Registers

Base pointer: `$rbp = 0x7fffffffdff0`

Stack pointer: `$rsp = 0x7fffffffdfd0`

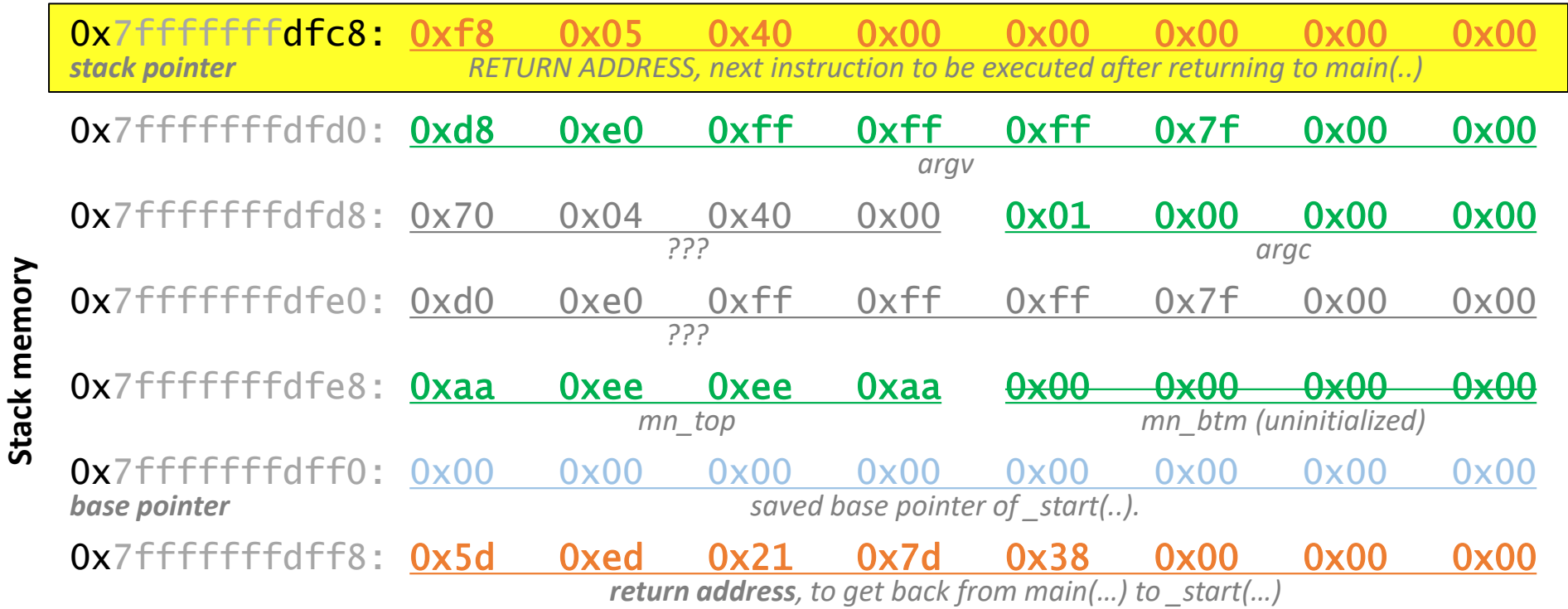


greet_visitor(…) » Before prologue

Registers

Base pointer: `$rbp = 0x7fffffffdff0`

Stack pointer: `$rsp = 0x7fffffffdfc8`



greet_visitor(…) » Disassembly

```
3       void greet_visitor() {
=> 0x0000000000400554 <+0>:     push   %rbp
   0x0000000000400555 <+1>:     mov    %rsp,%rbp
   0x0000000000400558 <+4>:     sub    $0x20,%rsp

4               int gt_top = 0xAAAAAAAA;
   0x000000000040055c <+8>:     movl   $0xaaaaaaaa,-0x8(%rbp)

5               char name[10]; // BAD!!!
6               printf("Hello.  What is your name?\n");
   0x0000000000400563 <+15>:    mov    $0x400708,%edi
   0x0000000000400568 <+20>:    callq  0x400438 <puts@plt>

7               gets(name);    // VERY, VERY, VERY BAD!!!
   0x000000000040056d <+25>:    lea    -0x20(%rbp),%rax
   0x0000000000400571 <+29>:    mov    %rax,%rdi
   0x0000000000400574 <+32>:    callq  0x400458 <gets@plt>

8               printf("Hello, %s.", name);
   0x0000000000400579 <+37>:    mov    $0x400723,%eax
   0x000000000040057e <+42>:    lea    -0x20(%rbp),%rdx
   0x0000000000400582 <+46>:    mov    %rdx,%rsi
   0x0000000000400585 <+49>:    mov    %rax,%rdi
   0x0000000000400588 <+52>:    mov    $0x0,%eax
   0x000000000040058d <+57>:    callq  0x400428 <printf@plt>

9               int gt_btm = 0xAABBBBAA;
   0x0000000000400592 <+62>:    movl   $0xaabbbbaa,-0x4(%rbp)

10      }
   0x0000000000400599 <+69>:    leaveq
   0x000000000040059a <+70>:    retq
```

prologue of greet_visitor(…)

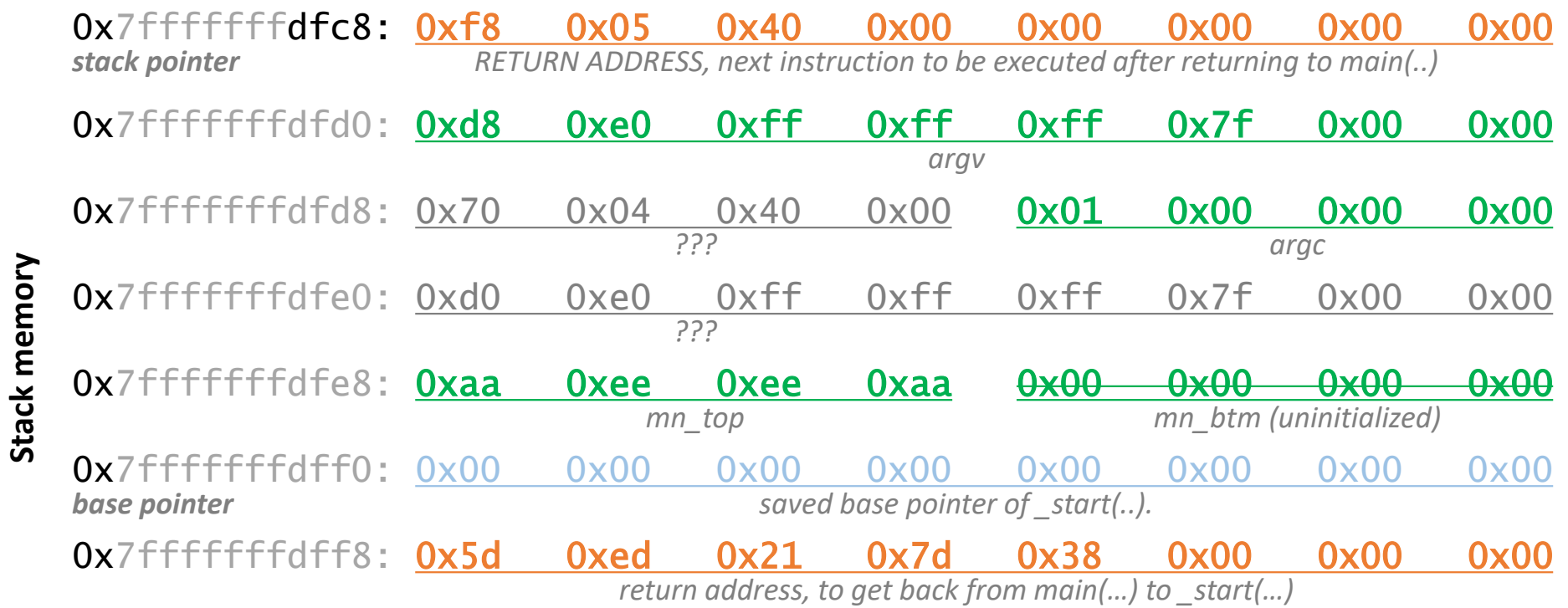
epilogue of greet_visitor(…)

greet_visitor(…)

Registers

Base pointer: `$rbp = 0x7fffffffdff0`

Stack pointer: `$rsp = 0x7fffffffdfc8`



greet_visitor(…) » Prologue

```
3       void greet_visitor() {
=> 0x0000000000400554 <+0>:     push   %rbp
   0x0000000000400555 <+1>:     mov    %rsp,%rbp
   0x0000000000400558 <+4>:     sub    $0x20,%rsp

4               int gt_top = 0xAAAAAAAA;
   0x000000000040055c <+8>:     movl   $0xaaaaaaaa,-0x8(%rbp)

5               char name[10]; // BAD!!!
6               printf("Hello.  What is your name?\n");
   0x0000000000400563 <+15>:    mov    $0x400708,%edi
   0x0000000000400568 <+20>:    callq  0x400438 <puts@plt>

7               gets(name);    // VERY, VERY, VERY BAD!!!
   0x000000000040056d <+25>:    lea    -0x20(%rbp),%rax
   0x0000000000400571 <+29>:    mov    %rax,%rdi
   0x0000000000400574 <+32>:    callq  0x400458 <gets@plt>

8               printf("Hello, %s.", name);
   0x0000000000400579 <+37>:    mov    $0x400723,%eax
   0x000000000040057e <+42>:    lea    -0x20(%rbp),%rdx
   0x0000000000400582 <+46>:    mov    %rdx,%rsi
   0x0000000000400585 <+49>:    mov    %rax,%rdi
   0x0000000000400588 <+52>:    mov    $0x0,%eax
   0x000000000040058d <+57>:    callq  0x400428 <printf@plt>

9               int gt_btm = 0xAABBBBAA;
   0x0000000000400592 <+62>:    movl   $0xaabbbbaa,-0x4(%rbp)

10      }
   0x0000000000400599 <+69>:    leaveq
   0x000000000040059a <+70>:    retq
```

prologue of greet_visitor(…)

epilogue of greet_visitor(…)

greet_visitor(…) » push %rbp

```
3       void greet_visitor() {
=> 0x0000000000400554 <+0>:     push   %rbp
   0x0000000000400555 <+1>:     mov    %rsp,%rbp
   0x0000000000400558 <+4>:     sub    $0x20,%rsp

4               int gt_top = 0xAAAAAAAA;
   0x000000000040055c <+8>:     movl   $0xaaaaaaaa,-0x8(%rbp)

5               char name[10]; // BAD!!!
6               printf("Hello.  What is your name?\n");
   0x0000000000400563 <+15>:    mov    $0x400708,%edi
   0x0000000000400568 <+20>:    callq  0x400438 <puts@plt>

7               gets(name);    // VERY, VERY, VERY BAD!!!
   0x000000000040056d <+25>:    lea    -0x20(%rbp),%rax
   0x0000000000400571 <+29>:    mov    %rax,%rdi
   0x0000000000400574 <+32>:    callq  0x400458 <gets@plt>

8               printf("Hello, %s.", name);
   0x0000000000400579 <+37>:    mov    $0x400723,%eax
   0x000000000040057e <+42>:    lea    -0x20(%rbp),%rdx
   0x0000000000400582 <+46>:    mov    %rdx,%rsi
   0x0000000000400585 <+49>:    mov    %rax,%rdi
   0x0000000000400588 <+52>:    mov    $0x0,%eax
   0x000000000040058d <+57>:    callq  0x400428 <printf@plt>

9               int gt_btm = 0xAABBBBAA;
   0x0000000000400592 <+62>:    movl   $0xaabbbbaa,-0x4(%rbp)

10      }
   0x0000000000400599 <+69>:    leaveq
   0x000000000040059a <+70>:    retq
```

prologue of greet_visitor(…)

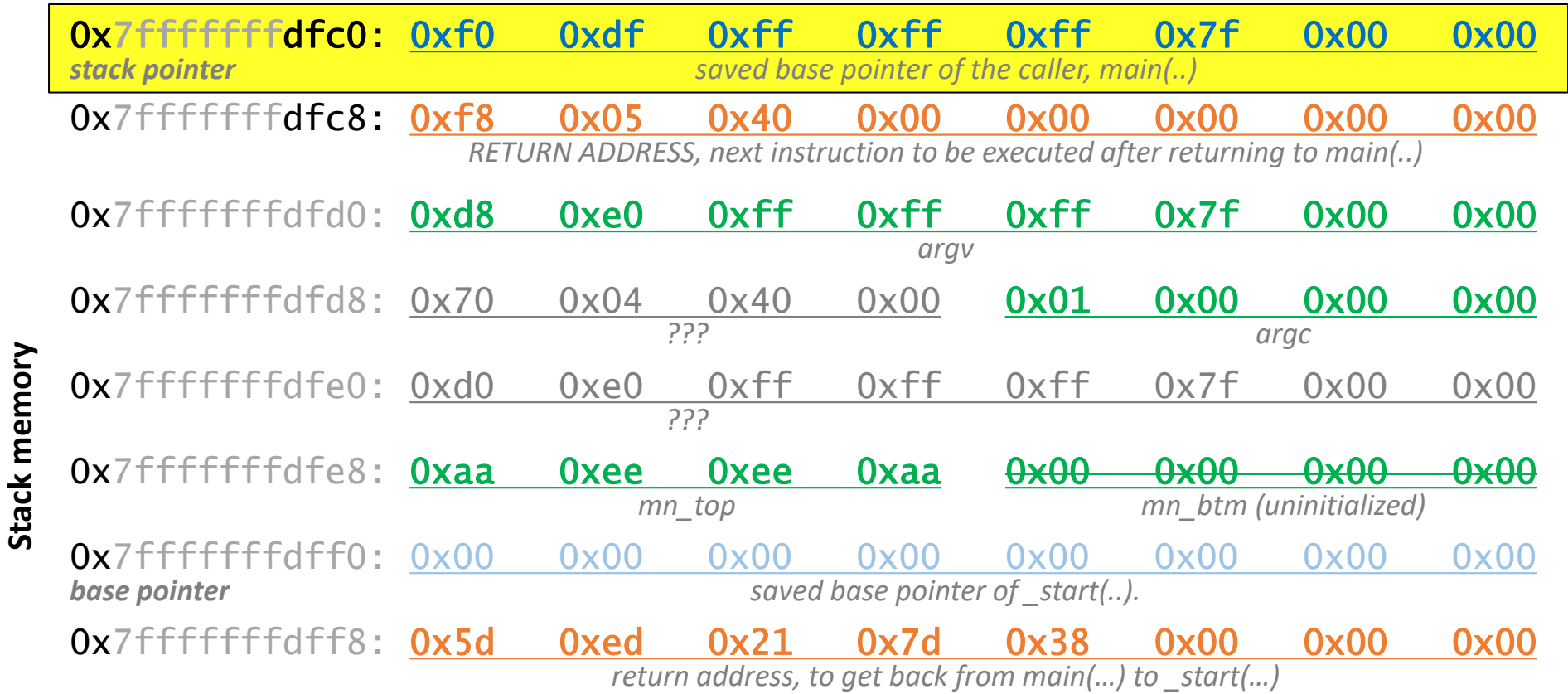
epilogue of greet_visitor(…)

greet_visitor(…) » push %rbp

Registers

Base pointer: `$rbp = 0x7fffffffdff0`

Stack pointer: `$rsp = 0x7fffffffdfc0`

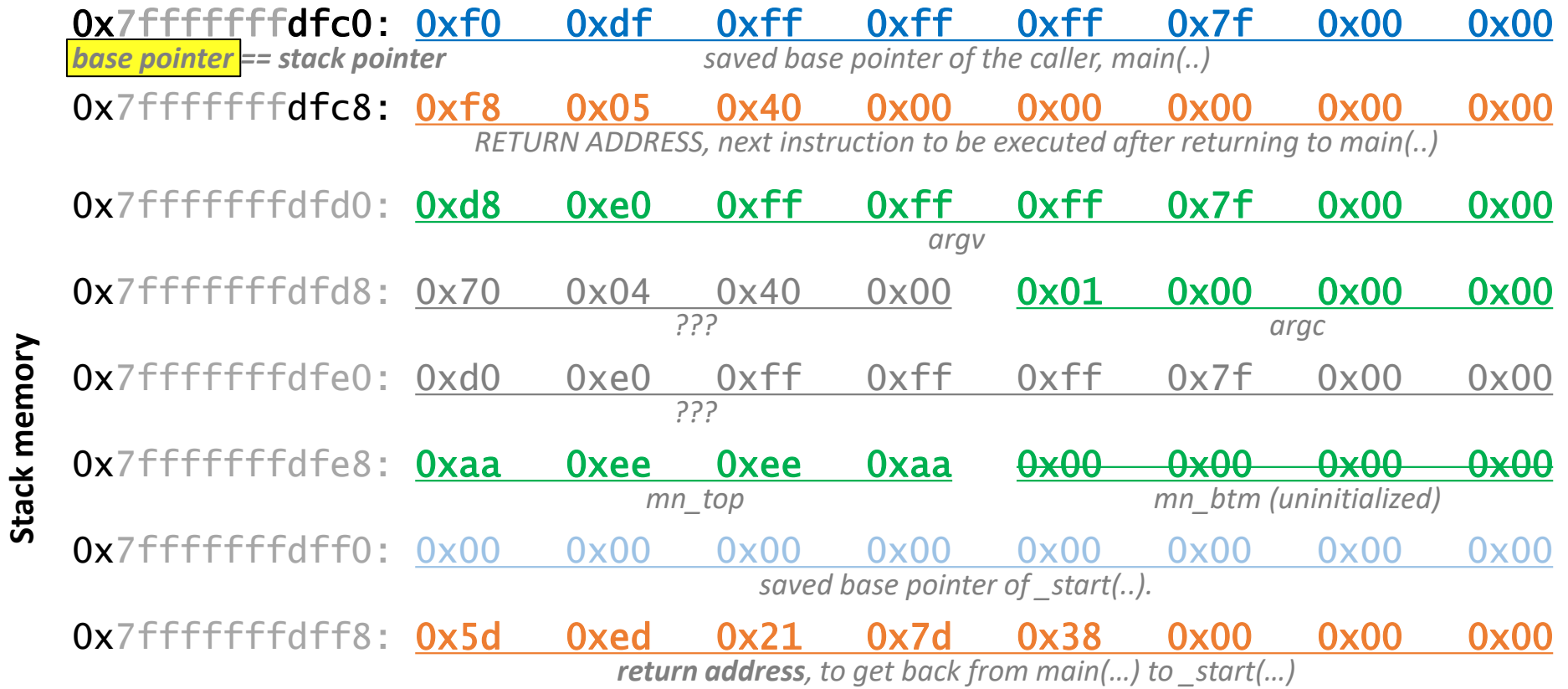


greet_visitor(…) » mov %rsp %rbp

Registers

Base pointer: `$rbp = 0x7fffffffdfc0`

Stack pointer: `$rsp = 0x7fffffffdfc0`



greet_visitor(…) » sub $0x20, %rsp

Registers Base pointer: `$rbp = 0x7fffffffdfc0`
Stack pointer: `$rsp = 0x7fffffffdfa0`

greet_visitor(…) » movl $0xaaaaaaaa,−0x8(%rbp)

Registers Base pointer: `$rbp = 0x7fffffffdfc0`
Stack pointer: `$rsp = 0x7fffffffdfa0`

greet_visitor(...) » After calling gets(name)

Registers

Base pointer: `$rbp = 0x7fffffffdfc0`

Stack pointer: `$rsp = 0x7fffffffdfa0`

greet_visitor(...) » movl $0xaabbbbaa,−0x4(%rbp)

Registers Base pointer: `$rbp = 0x7fffffffdfc0`
Stack pointer: `$rsp = 0x7fffffffdfa0`

Stack memory

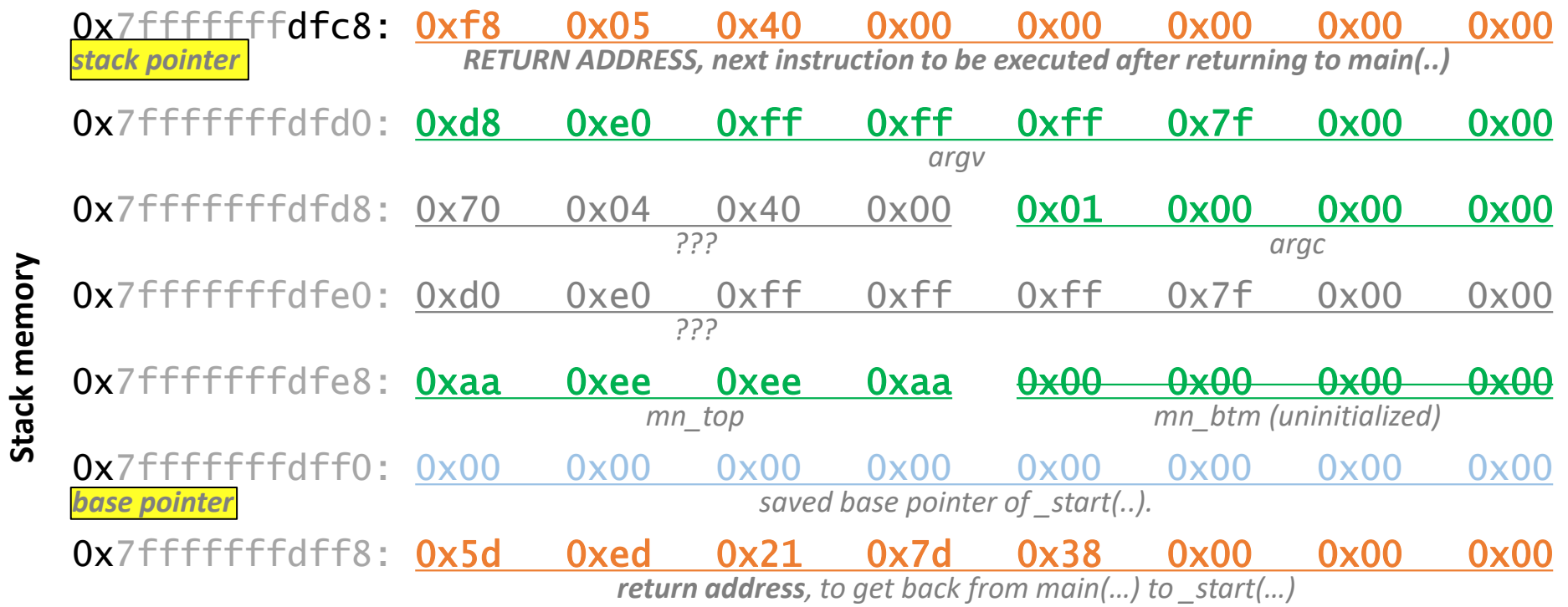
0x7fffffffdfa0:	0x41	0x6c	0x65	0x78	0x61	0x6e	0x64	0x65
stack pointer	*name[0]*	*name[1]*	*name[2]*	*name[3]*	*name[4]*	*name[5]*	*name[6]*	*name[7]*
0x7fffffffdfa8:	0x72	0x00	0x40	0x00	0x00	0x00	0x00	0x00
	name[8]	*name[9]*	*garbage / padding*					
0x7fffffffdfb0:	0xe8	0xe0	0xff	0xff	0xff	0x7f	0x00	0x00
			garbage					
0x7fffffffdfb8:	0xaa	0xaa	0xaa	0xaa	0xaa	0xbb	0xbb	0xaa
			gt_top		*gt_btm*			
0x7fffffffdfc0:	0xf0	0xdf	0xff	0xff	0xff	0x7f	0x00	0x00
base pointer	*saved base pointer of the caller, main(..)*							
0x7fffffffdfc8:	0xf8	0x05	0x40	0x00	0x00	0x00	0x00	0x00
	RETURN ADDRESS, next instruction to be executed after returning to main(..)							
0x7fffffffdfd0:	0xd8	0xe0	0xff	0xff	0xff	0x7f	0x00	0x00
				argv				
0x7fffffffdfd8:	0x70	0x04	0x40	0x00	0x01	0x00	0x00	0x00
			???		*argc*			
0x7fffffffdfe0:	0xd0	0xe0	0xff	0xff	0xff	0x7f	0x00	0x00
			???					
0x7fffffffdfe8:	0xaa	0xee	0xee	0xaa	0x00	0x00	0x00	0x00
			mn_top		*mn_btm (uninitialized)*			
0x7fffffffdff0:	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
	saved base pointer of _start(..).							
0x7fffffffdff8:	0x5d	0xed	0x21	0x7d	0x38	0x00	0x00	0x00
	***return address**, to get back from main(...) to _start(...)*							

greet_visitor(…) » leaveq

Registers

Base pointer: `$rbp` = `0x7fffffffdff0`

Stack pointer: `$rsp` = `0x7fffffffdfc8`

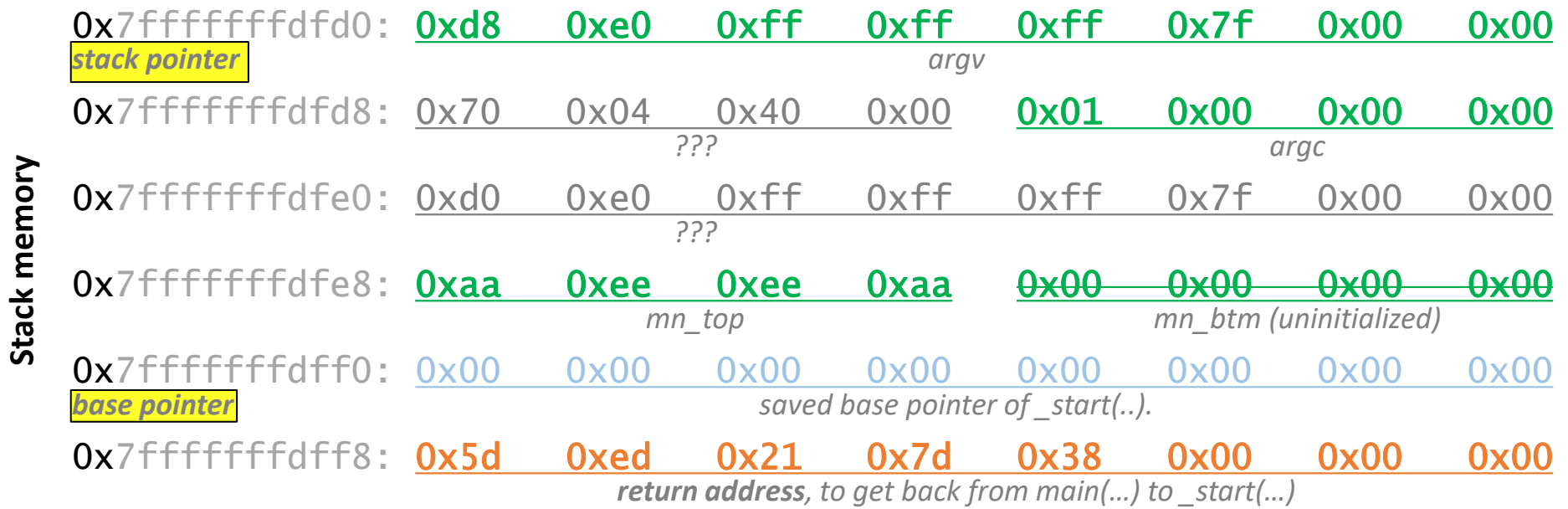


greet_visitor(…) » retq

Registers

Base pointer: `$rbp = 0x7fffffffdff0`

Stack pointer: `$rsp = 0x7fffffffdfd0`



main(…) » Just before returning

Registers

Base pointer: `$rbp = 0x7fffffffdff0`

Stack pointer: `$rsp = 0x7fffffffdfd0`