

C, Memory

CS 2130: Computer Systems and Organization 1

Xinyao Yi Ph.D.
Assistant Professor

Announcements

- Homework 8 due tonight on Gradescope
- Homework 9 available soon
- Lab 10 tomorrow: Memory Errors

Common Memory Bugs (reading)

More on man pages

```
char *strsep(char **s, const char *d);
```

Processes

Process - approximately what we think of as a “running program”

- Operating System effectively has a giant array of processes started since computer turned on
- Try `ps -A`
- Has access to all memory (but only its own!)
- Operating System maintains data structure about each process
 - What program is running, who ran it, when it started, ...
 - Array of “file like objects”

Processes

1. [18 points] Consider the following C code. Write the function `examtwo()` in x86-64 assembly. Your function should store x in register `rax` and y in register `rbx`. When using 32-bit values, use the correct 32-bit instruction and register.

```
1 long mathfun(long x, int y);
2
3 long examtwo() {
4     long x = 0x42;           // store in rax
5     int y = 1;              // store in rbx
6     while (y <= 5) {
7         x = mathfun(x, y);
8         y++;
9     }
10    return x;
11 }
```

For the next two questions, assume the first eight registers and the given segment of memory have the following values before the next few instructions.

Register	Value (hex)
rax	0x2130
rcx	0xc501e1f0
rdx	0x100000088
rbx	0x4
rsp	0xc501e20b
rbp	0xc501e20f
rsi	0x67
rdi	0xf000000ff

Mem Addr.	Value (hex)
0xc501e201	0x01
0xc501e202	0xe2
0xc501e203	0x01
0xc501e204	0x50
0xc501e205	0xff
0xc501e206	0xa0
0xc501e207	0x04
0xc501e208	0x34
0xc501e209	0x12

Mem Addr.	Value (hex)
0xc501e20a	0xb2
0xc501e20b	0x59
0xc501e20c	0x01
0xc501e20d	0x46
0xc501e20e	0xcf
0xc501e20f	0xea
0xc501e210	0x65
0xc501e211	0x7d
0xc501e212	0x4e

2. [16 points] Which program registers are modified, and to what values, by the following instructions? Leave spaces blank if fewer registers change than there are lines. If no registers are changed, write "none" in the first register box with no new value. *Each instruction below is independent; do not use the result of one as input for the next.* (4 points each)

`popl %eax`

Register	New Value

`andl %edi, %edx`

Register	New Value

`leaq 0x6(%rcx, %rbx, 4), %rdi`

Register	New Value

`movq 0x6(%rcx, %rbx, 4), %rdi`

Register	New Value

For the next two questions, assume the first eight registers and the given segment of memory have the following values before the next few instructions.

Register	Value (hex)
rax	0x2130
rcx	0xc501e1f0
rdx	0x100000088
rbx	0x4
rsp	0xc501e20b
rbp	0xc501e20f
rsi	0x67
rdi	0xf000000ff

Mem Addr.	Value (hex)
0xc501e201	0x01
0xc501e202	0xe2
0xc501e203	0x01
0xc501e204	0x50
0xc501e205	0xff
0xc501e206	0xa0
0xc501e207	0x04
0xc501e208	0x34
0xc501e209	0x12

Mem Addr.	Value (hex)
0xc501e20a	0xb2
0xc501e20b	0x59
0xc501e20c	0x01
0xc501e20d	0x46
0xc501e20e	0xcf
0xc501e20f	0xea
0xc501e210	0x65
0xc501e211	0x7d
0xc501e212	0x4e

3. [6 points] Assume we stored an array of 16-bit unsigned numbers that started at memory address 0xc501e20a (i.e., the far right column). Write the hexadecimal values of the first two numbers in the array, if the array was stored in memory using:

A. little-endian storage.

--	--

B. big-endian storage.

--	--

4. [12 points] True/False questions on our discussion of a backdoor in our Toy ISA processor. For each of the statements below, fill in the T circle *completely* if you think the statement is True. If you think it's False, fill in the F circle *completely*.

T F A backdoor can allow others to take control of your computer without your knowledge.

T F Our exploit only provided a way to execute code that was programmed into the hardware.

T F Our backdoor's malicious code is executed when the user's program read the bytes of the passcode in order.

T F Our backdoor required additional gates for calculation, which would noticeably slow the normal operation of the processor.

5. [6 points] We discussed copyrights and patents in class. Assume that Jane has written a new program—using a novel approach—to solve matrix multiplication faster than anyone else. Which of the following would allow Jane to protect her intellectual property? *Fill in the circle complete for all that apply.*
- The implementation (source code) is immediately copyrighted, preventing others from using it.
 - Jane patents her algorithm, requiring anyone who wants to use it to pay royalties.
 - Jane does not release the source code, but requires users to accept a license agreement restricting re-distribution.
 - Jane publishes her code on her blog, including a copyright notice and text describing limits on re-using the code.

6. [4 points] Suppose you have a file named `hello.c`, which includes a `main()` function. What command on the CS portal would you use to compile this C code to an executable, with a moderate level of optimization (i.e., 2 levels or level 2), so that it can be run later as `./a.out`?

Answer

7. [6 points] True/False questions. For each of the statements below, fill in the **T** circle *completely* if you think the statement is True. If you think it's False, fill in the **F** circle *completely*.

- An error will occur in the **type-checking** stage of compilation when compiling our code if we call the function `csprintf()` which does not exist.
- If we accidentally wrote Java instead of C when initializing an array, an error would occur in the **parsing** stage of compilation: `int[100] arr = new arr[100];`

For the next two questions, consider the following C code snippet.

```
1 void addressfun() {  
2     int a[10];  
3     int *p = a + 4;  
4     _____ = 42;  
5     size_t asize = sizeof(a);  
6     return;  
7 }
```

8. [6 points] Which of the expressions below could be used in the blank on line 4 to set the 5th element in `a` (i.e., `a[4]`) to 42? *Fill in the circle completely for all that apply.*

- `*(a + 4)`
- `p[0]`
- `*p`
- `*(a + 16)`
- `*a + 16`
- `p[3]`

9. [4 points] What value is stored in `asize` on line 5?

Answer

For the next question, consider the following struct and function header.

```
1 typedef struct {
2     char *name;
3     unsigned long age;
4     double latitude;
5     double longitude;
6 } person;
7
8 int locate(person waldo);
```

10. [4 points] When the code is compiled to x86-64 assembly, where is the first parameter to the function `locate` placed when calling the function?

- In register `%rdi`
- In register `%rax`
- On the stack
- This is not valid C

11. [18 points] Write a C function named `missingNumber` that takes two parameters (in order): an array of integers named `nums` given as a pointer, and an integer `n`; it should return an integer.

`nums` points to a valid array of length `n`, which contains `n` distinct integers selected from the range 0 through `n`, inclusive, in random order. Exactly one integer in that range is missing. Your function should find and return the missing integer. If `n = 0`, return `-1`.

Do not allocate another array or modify the contents of `nums`.

For example: given `nums = {3, 0, 1}` and `n = 3`, return 2; given `nums = {1, 4, 3, 2}` and `n = 4`, return 0; given `nums = {}` and `n = 0`, return `-1`.