System Programming CSE25101 Fall 2017 Midterm Exam

Professor Sam H. Noh Oct. 19, 2016

1. (6 points) Select the answers for the following multiple choice questions. If there are multiple correct answers, you MUST select ALL the correct answers.
   a. What is the minimum (most negative) value of a 16-bit two's complement integer?
      (a) -2^{15} (b) $-2^{15}+1$ (c) -2^{16} (d) $-2^{16}+1$
   b. On an x86-64 Linux system, which of these takes up the most bytes in memory?
      (a) `char *a` (b) `short b[3]` (c) `char c[7]` (d) `double d`
   c. Select the two's complement negation of the following binary value:: 1111010011
      (a) 0000101100 (b) 0000011100 (c) 1000101100 (d) 0000101101
   d. How many lines does a direct-mapped cache have in a set?
      (a) 0 (b) 1 (c) 2 (d) 3 (e) 4
   e. Suppose register `%rax` holds value `x` and `%rcx` holds value `y`. What is the formula indicating the value that will be stored in register `%rdx` for the following assembly code instruction: `leaq 0xA(,%rcx,4), %rdx`
      (a) x + 4y (b) x + y (c) 10 + 4y (d) 7 + 9x (e) 4 + x
   f. Consider an `int *a` and an `int n`. If the value of `%rcx` is `a` and the value of `%rdx` is `n`, which of the following assembly snippets best corresponds to the C statement `return a[n]`?

      (a)
      ```
      mov (%rcx,%rdx,4),%rbx
      ret
      ```
      (b)
      ```
      leal (%rcx,%rdx,4),%rax
      ret
      ```
      (c)
      ```
      mov (%rcx,%rdx,4),%rax
      ret
      ```
      (d)
      ```
      mov (%rcx,%rdx,1),%rax
      ret
      ```

2. (14 points) Given

   `int x = foo( ); int y = bar( ); unsigned ux = x; unsigned uy = y;`

   State whether the following is TRUE or FALSE. If your answer is FALSE, describe the reason it can be FALSE.
   a. ux >> 3 = = ux/8

   b. x & (x-1) != 0

   c. x & 7 = = 7 ==> (x<<30) < 0

   d. x > y ==> -x < -y

   e. x >= 0 ==> -x <= 0

   f. ux > -1

   g. (x|-x)>>31 = = -1

3. (6 points; 2 points each) Fill in the relation (<, >, ==, <=, >=) between the two constants on the left columns. Give a short description of the reason on the rightmost column. Assume the word size is 16 bits such that UMax = 65,535, TMax = 32,767 and TMin = -32,768.

Constant1	Constant2	Relation	Reason
32767	(int)32768U		
32767U	-32767-1		
-1	(unsigned)0		

Your Name: Student ID:

4. (8 points) There is something wrong with the following code. (Assume all other declarations are appropriately made.)

```
unsigned i;
for (i = cnt - 2; i >= 0; i - -)
a[i] += a[i+1];
```

a. (3 points) Explain the problem.

b. (5 points) Fix the problem and explain why your solution will correct the problem.

5. (15 points; 3 points each) The C code on the right box is compiled into assembly code on the left box. Answer the following questions.

a. The first instruction of this function is push %rbp. What is the purpose of this instruction?

b. Show how the C code n*10 (in code n = n*10 + p[i] - '0') is being translated. Provide explanations for the translated code.

c. Specify the memory addresses for variables n and i.
d. Specify the address for [(a)] in the assembly code.
e. Specify the address for [(b)] in the assembly code.

```
0x00000000004004d8:     push   %rbp
0x00000000004004d9:     mov    %rsp,%rbp
0x00000000004004dc:     mov    %rdi,-0x18(%rbp)
0x00000000004004e0:     movl   $0x0,-0x4(%rbp)
0x00000000004004e7:     movl   $0x0,-0x8(%rbp)
0x00000000004004ee:     jmp    0x400543
0x00000000004004f0:     mov    -0x8(%rbp),%eax
0x00000000004004f3:     cltq
0x00000000004004f5:     add    -0x18(%rbp),%rax
0x00000000004004f9:     movzbl (%rax),%eax
0x00000000004004fc:     cmp    $0x2f,%al
0x00000000004004fe:     jle    [(a)]
0x0000000000400500:     mov    -0x8(%rbp),%eax
0x0000000000400503:     cltq
0x0000000000400505:     add    -0x18(%rbp),%rax
0x0000000000400509:     movzbl (%tax),%eax
0x000000000040050c:     cmp    $0x39,%al
0x000000000040050e:     jle    [(b)]
0x0000000000400510:     movl   $0xffffffff,-0x1c(%rbp)
0x0000000000400517:     jmp    0x400559
0x0000000000400519:     mov    -0x4(%rbp),%edx
0x000000000040051c:     mov    %edx,%eax
0x000000000040051e:     shl    $0x2,%eax
0x0000000000400521:     add    %edx,%eax
0x0000000000400523:     add    %eax,%eax
0x0000000000400525:     mov    %eax,%edx
0x0000000000400527:     mov    -0x8(%rbp),%eax
0x000000000040052a:     cltq
0x000000000040052c:     add    -0x18(%rbp),%rax
0x0000000000400530:     movzbl (%rax),%eax
0x0000000000400533:     movsbl %al,%eax
0x0000000000400536:     lea    (%rdx,%rax,1),%eax
0x0000000000400539:     sub    $0x30,%eax
0x000000000040053c:     mov    %eax,-0x4(%rbp)
0x000000000040053f:     addl   $0x1,-0x8(%rbp)
0x0000000000400543:     mov    -0x8(%rbp),%eax
0x0000000000400546:     cltq
0x0000000000400548:     add    -0x18(%rbp),%rax
0x000000000040054c:     movzbl (%rax),%eax
0x000000000040054f:     test   %al,%al
0x0000000000400551:     jne    [(c)]
0x0000000000400553:     mov    -0x4(%rbp),%eax
0x0000000000400556:     mov    %eax,-0x1c(%rbp)
0x0000000000400559:     mov    -0x1c(%rbp),%eax
0x000000000040055c:     leaveq
0x000000000040055d:     retq
```

```
int test1(char *p)
{
    int i, n;

    n = 0;

    for (i = 0; p[i] != '\0'; i++)
    {
        if (p[i] < '0' || p[i] > '9')
        {
            return -1;
        }
        n = n*10 + p[i] - '0';
    }
    return n;
}
```

Your Name: Student ID:

6. (9 points) Answer the following questions. You MUST show how you reached your solution.
   a. (3 points) Calculate the average time to access a sector on HDD that has the following characteristics: rotational rate of 12,000RPM, average seek time of 5 ms, and average number of sectors/track 300.

   b. (6 points) A potential weakness of SSDs is that the underlying flash memory can wear out. Assume that an SSD can guarantee 128 petabytes of writes before the drive wears out. Estimate the lifetime (in years) of this SSD for the following workloads.
      (a) The SSD is written to continuously at a rate of 470 MB/s, the average sequential write throughput for the device.
      (b) The SSD is written to at a rate of 20 GB/day, the typical write rate for a typical user.

7. (7 points) Consider an SRAM-based cache for a DRAM-based main memory. Neglect the possibility of other caches or levels of the memory hierarchy below main memory. Let the access time of the cache and memory be C and M, respectively. Denote the hit rate on the cache as h.
   a. (4 points) Describe the Average Access Time (AAT) for a reference to this system.

   b. (3 points) Assume that memory access time is 100 times higher than cache access time. If the cache hit rate increases from 98% to 99%, how much of a performance improvement can be expected? You must explain the reason behind your conclusion.

8. (10 points) In 2002, it was discovered that code supplied by Sun Microsystems to implement the XDR library, a widely used facility for sharing data structures between programs, had a security vulnerability arising from the fact that multiplication can overflow without any notice being given to the program. Code similar to that containing the vulnerability is shown below:
   a. (4 points) Explain what the code below does.

```
1  /*
2   * Illustration of code vulnerability similar to that found in
3   * Sun's XDR library.
4   */
5  void* copy_elements(void *ele_src[], int ele_cnt, size_t ele_size) {
6      /*
7       * Allocate buffer for ele_cnt objects, each of ele_size bytes
8       * and copy from locations designated by ele_src
9       */
10     void *result = malloc(ele_cnt * ele_size);
11     if (result == NULL)
12         /* malloc failed */
13         return NULL;
14     void *next = result;
15     int i;
16     for (i = 0; i < ele_cnt; i++) {
17         /* Copy object i to destination */
18         memcpy(next, ele_src[i], ele_size);
19         /* Move pointer to next memory region */
20         next += ele_size;
21     }
22     return result;
23 }
```

b. (6 points) Describe, using an example and specific code line numbers given, under what specific conditions a malicious programmer could make use of the vulnerability of the program.

Your Name: Student ID:

9. (25 points) The following is disassembled code of two functions fct1 and fct2. The left most numbers are Line numbers to be used for reference only and are not part of the assembly code.

```
Disassembly of fct2( )
1       0000000000400540 <fct2>:
2       400540:     48 89 f8        mov %rdi, %rax
3       400543:     48 0f af c6     imul %rsi, %rax
4       400547:     c3              retq

Disassembly of fct1( )
5       0000000000400548 <fct1>
6       400548:     48 8d 77 01     lea 0x1(%rdi), %rsi
7       XXXX:       48 83 ef 01     sub $0x1, %rdi
8       400550:     e8 eb ff ff ff  callq 400540 <fct2>
9       400555:     f3 c3           repz retq
.
.
10      400560:     e8 e3 ff ff ff  callq 400548 <fct1>
11      400565:     48 89 c2        mov %rax, %rdx
```

a. (2 points) Explain what the lea instruction does.

b. (15 points) Assuming that execution starts from Line 10 and the states are as below. Fill in the following table to trace instruction execution through to the point where the program returns to Line 11.

Instruction		State values at beginning				
Line #	%rip	%rdi	%rsi	%rax	%rsp	*%rsp
10	0x400560	10	15	20	0x7fffffffe820	-
11						

c. (8 points) Based on all of the above, construct the C language code for fct1( ) and fct2( ).