Patrick Austin

CPE 301 - 1104, Fall 2016

Homework 3

9/27/2016

1. a. volatile int\* y\_addr;

b. volatile unsigned char\* ch\_addr;

c. volatile int\* z;

d. volatile int\* date\_pt;

e. volatile unsigned char\* pt\_chr;

2. a. Valid b. Invalid c. Invalid

d. Invalid e. Invalid f. Invalid

g. Invalid h. Valid i. Invalid

j. Invalid k. Invalid l. Invalid

m. Invalid n. Valid o. Invalid

p. Invalid q. Invalid r. Invalid

s. Valid t. Invalid u. Invalid

3. A variable contained as a pointer stores the address in memory of another variable.

4. If var2 is a variable, &var2 will provide the address in memory at which var2 is stored.

5. According to sizeof(), my PC stores the addresses of integers, characters, and doubles on 8 bytes. I would expect this to be the case, since all of these types of pointers are doing the same thing, referring to locations in memory, regardless of the type.

6. a. 1000 00002

b. 1110 11112

c. 0110 11112

7. a. 8016

b. EF16

c. 6F16

8. a. 0x015716 = 0000 0001 0101 01112

After a left shift by one bit: 0000 0010 1010 1110 2 = 0x02AE16

b. 0x070116 = 0000 0111 0000 00012

After a left shift by two bits: 0001 1100 0000 01002 = 0x1C0416

c. 0x067316 = 0000 0110 0111 00112

After a right shift by two bits: 0000 0001 1001 11002 = 0x019C16

d. 0x005716 = 0000 0000 0101 01112

After a right shift by three bits: 0000 0000 0000 10102 = 0x000A16

9. a. The mask will be 0001 10002 = 0x1816

b. The mask will be 0101 11112 = 0x5F16

c. The mask will be 0000 10102 = 0A16 using bitwise XOR.

10. void capsLockOn()

{

//get the address of the register, pointing at 8 bits

unsigned char\* keyboard\_flags = 0x417 ;

//at the address, make bit 6 equal 1: bitwise OR with binary 0100 0000, ie hex 40

\*keyboard\_flags = \*keyboard\_flags | 0x40 ;

}