Michael DesRoches

CPE301 Dr. Egbert

06FEB19

Homework 2

Description of purpose:

This homework's objective was to get me familiarized with basic code operations like pointers and functions. Also, to get me familiar with reading and understanding those basic principles. I'll also be familiarizing myself with bit math operations and determining binary and hexadecimal results. It appears that I'll also be messing with the BIOS.

1.

a) int *y_addr where y_addr is a variable that points to an integer

b) unsigned char *ch_addr where ch_addr is a variable that points to an unsigned character

c) int *z

d) int *date_pt

e) unsigned char *pt_chr

2.

A,H,N,S are all true

Reason:
Pointer: A pointer is a variable which holds the address of another variable
y_addr = &a;

here *x_pt is a pointer to an integer, which can refer the address of the memory location of int type variable a.
dt_addr = &b;

here *dt_addr is a pointer to a long, which can refer to the address of the memory location of long type varible b
pt_addr = &b;

here *pt_addr is a pointer to a long, which can refer to the address of the memory location of the long type variable b
y_addr = x_pt;

3.

The data type must be declared

4. it means the sddress of the variable

5. The sizeof operator returns size of a variable or datatype in bytes.

program to determine the number of bytes used by your PC to store the address of an integer, character, and double precision number by using the sizeof ( ) operator :-

```
#include<stdio.h>

main()

{

int i;

char c;

double d;

printf(" size of an integer : %d\n ", sizeof( i ) ) ;

printf(" size of a character : %d\n", sizeof( c ) ) ;

printf(" size of a double : %d\n", sizeof( d ) ) ;
}
```

output :-

```
size of an integer : 2
size of a character : 1
size of a double : 8
```

Would you expect the size of each address to be the same? Why or why not?

I didn't expect the size of each address of an integer,character, and double precision number to be the same.

Because..

The sizeof operator returns size according to given or specific a variable or datatype in bytes.

HW 01

(1.5) given $\{0111\ 0110\ 1101\ 0001\ 1001\ 1011\ 1111\ 0000\}_2$.
determine equivalent # in each base
a). Decimal
b). hexidecimal

* recall $\overset{8}{1}\ \overset{4}{0}\ \overset{2}{1}\ \overset{1}{0}$ $(2^3\ 2^2\ 2^1\ 2^0)$ → 8020 = 8+0+2+0 = 10 = A

Bin → Hex

0111 | 0110 | 1101 | 0001 | 1001 | 1011 | 1111 | 0000
7 6 D 1 9 B F 0

$(76D19BF0)_{16} = (0111\ 0110\ 1101\ 0001\ 1001\ 1011\ 1111\ 0000)_2$

* recall ex. write binary # ; list powers of 2 Right to left

0 1 1 1 0 1 1 0 1 1 0 1 0 0 0 1 1 0 0 1 1 0 1 1 1 1 1 1 0 0 0 0
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bin → Dec

$1\times2^{30} + 1\times2^{29} + 1\times2^{28} + 1\times2^{26} + 1\times2^{25} + 1\times2^{23} + 1\times2^{22} + 1\times2^{20}$ .... calculator

$= (1993448432)_{10}$

Hex → Dec

(1.6) (c) $(24687531)_{16}$ (7 6 5 4 3 2 1 0)

$2\times16^7 + 4\times16^6 + 6\times16^5 + 8\times16^4 + 7\times16^3 + 5\times16^2 + 3\times16^1 + 1\times16^0$

$(24687531)_{16} = (616825521)_{10}$

Hex → Binary

(5)

2	4	6	8	7	5	3	1
0010	0100	0110	1000	0111	0101	0011	0001

$(24687531)_{10} = (0010\ 0100\ 0110\ 1000\ 0111\ 0101\ 0011\ 0001)_2$