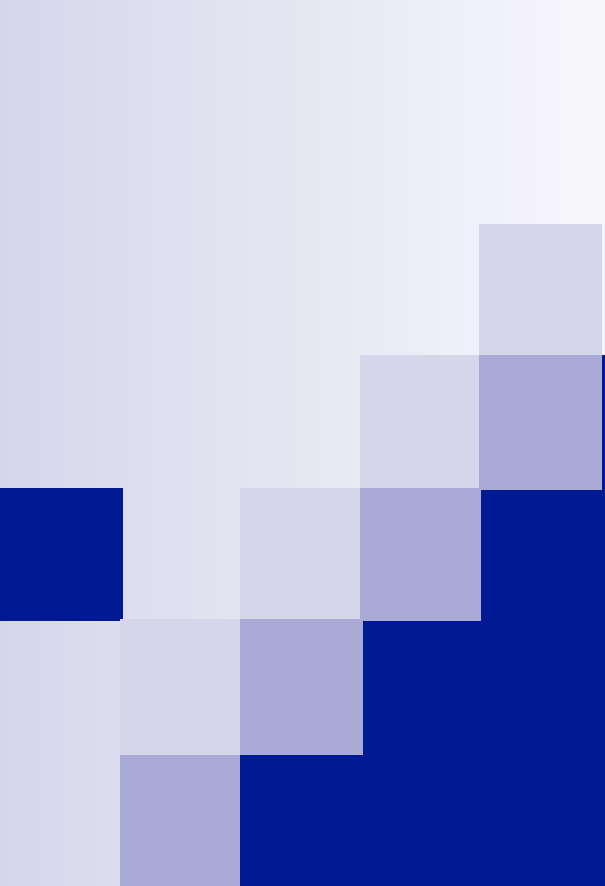




NASM Basics II

Data declarations

(Practice)

ICS312

Machine-Level and

Systems Programming

Henri Casanova (henric@hawaii.edu)

(q1) Data Declaration Sizes

- For each C declaration below give the corresponding numbers of bytes in RAM:
 - `short int x;`
 - `char *s = "hello\n";`
 - `unsigned int y;`
 - `unsigned char z = 'a';`
 - `int z = 034; // OCTAL`

(q1) Solutions

- For each C declaration below give the corresponding numbers of bytes in RAM:

- `short int x;` // 4 bytes
- `char *s = "hello\n";` // 7 bytes
- `unsigned int y;` // 4 bytes
- `unsigned char z = 'a';` // 1 byte
- `int z = 034;` // 4 bytes

(q2) Little vs. Big Endian

- For each declaration below say whether the defined RAM content would be different or the same on a Little Endian and a Big Endian machine:
 - `char *str="hello";`
 - `int a[3]={-1, -2, -3};`
 - `int b[2]={-1, 0};`
 - `unsigned char x[2]={12,1};`
 - `short foo=1;`

(q2) Solutions

- For each declaration below say whether the defined RAM content would be different or the same on a Little Endian and a Big Endian machine:

- `char *str="hello";` // SAME
- `int a[3]={-1, -2, -3};` // DIFF
- `int b[2]={-1, 0};` // SAME!
- `unsigned char x[2]={12,1};` // SAME
- `short foo=1;` // DIFF

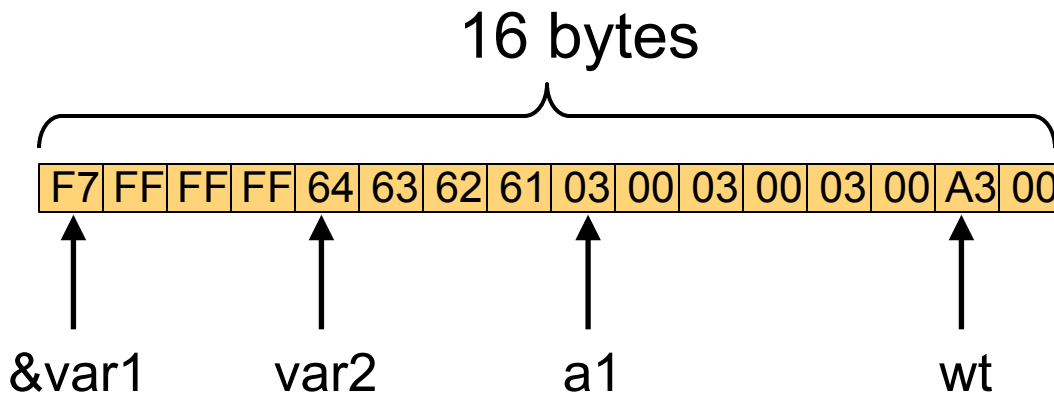
(q3) Data Layout

```
int  var1 = -9;
char *var2 = "dcba";
unsigned short a1[3] = {0b11, 0b11, 0b11};
char wt[2] = {0xA3, 0};
```

- What is the layout and the content of the data memory segment on a **LITTLE ENDIAN** machine?
 - Byte per byte, in hex

(q3) Solutions

```
int  var1 = -9;  
char *var2 = "dcba";  
unsigned short a1[3] = {0b11, 0b11, 0b11};  
char wt[2] = {0xA3, 0};
```



(q4) Data Layout

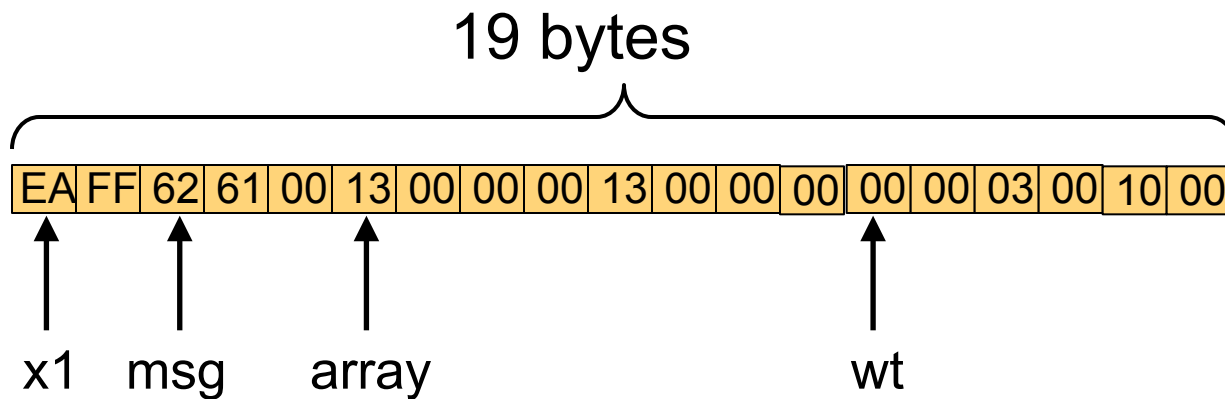
```
short x1 = -22;  
char *msg = "ba";  
signed int array[2] = {023, 023};  
short wt[3] = {0, 0b011, 020};
```

- What is the layout and the content of the data memory segment on a **LITTLE ENDIAN** machine?
 - Byte per byte, in hex

(q4) Solutions

octal!

```
short x1 = -22;  
char *msg = "ba";  
signed int array[2] = {023, 023};  
short wt[3] = {0, 0b011, 020};
```



(q5) Data Layout

```
char A[6]={ 'd' ,0,' a' ,' f' ,027,' \0' };  
short B = 0x11FA;  
short C[2] = {-15, -15};  
char D[2] = {0x43, 0xAA};  
short E = 0x09;  
short F[2] = {-034, 0};
```

- What is the layout and the content of the data memory segment on a **LITTLE ENDIAN** machine?
 - Byte per byte, in hex

```
char A[6]={ 'd' ,0,'a' , 'f' ,027,' \0' };
short B = 0x11FA;
short C[2] = {-15, -15};
char D[2] = {0x43, 0xAA};
short E = 0x09;
short F[2] = {-034, 0};
```

