A Deeper Look at `Struct`

“How structures are structured”

Prerequisite: Structure

Find more contents at
https://sites.google.com/view/cse105june18/home

Md. Saidul Hoque Anik
onix.hoque.mist@gmail.com

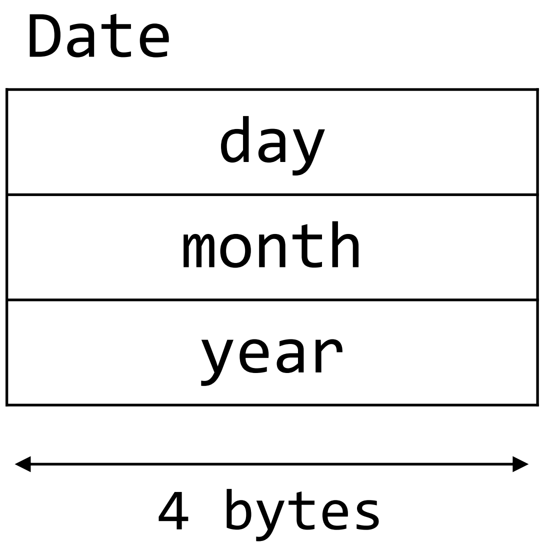
Difference between struct and array?

What is the sizeof a struct?

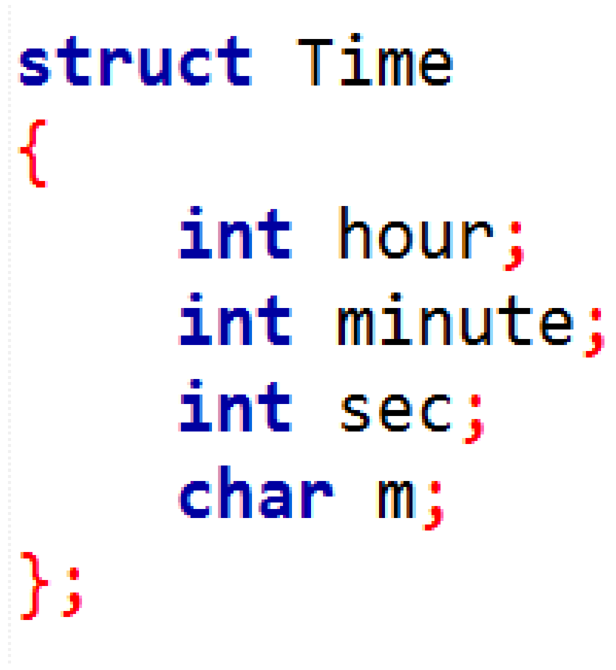
```
struct Date
{
    int day;
    int month;
    int year;
};
```

What is the sizeof a struct?

```
struct Date
{
      int day;
      int month;
      int year;
};
```

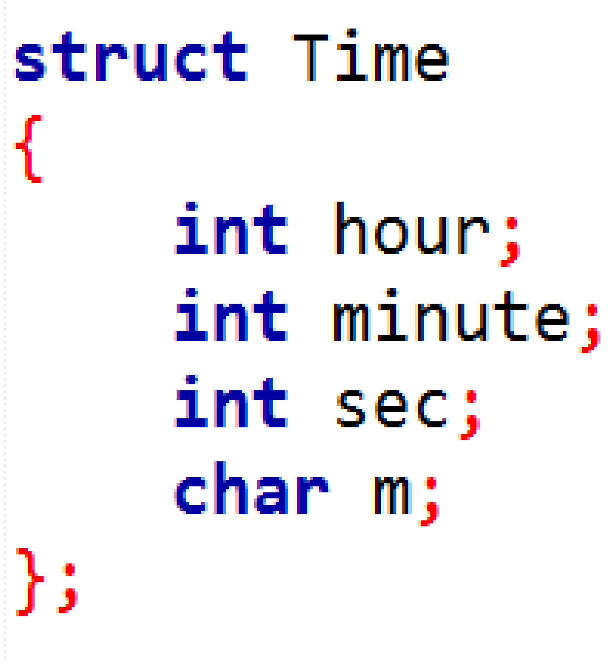


What is the sizeof this struct?

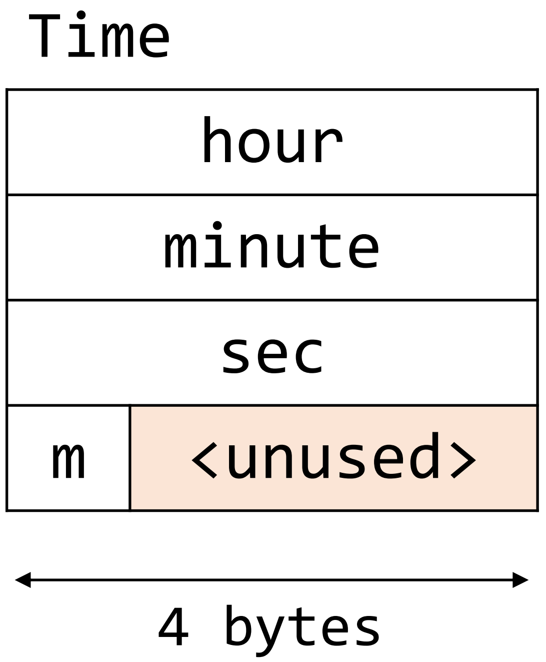


```c
struct Time
{
	int hour;
	int minute;
	int sec;
	char m;
};
```

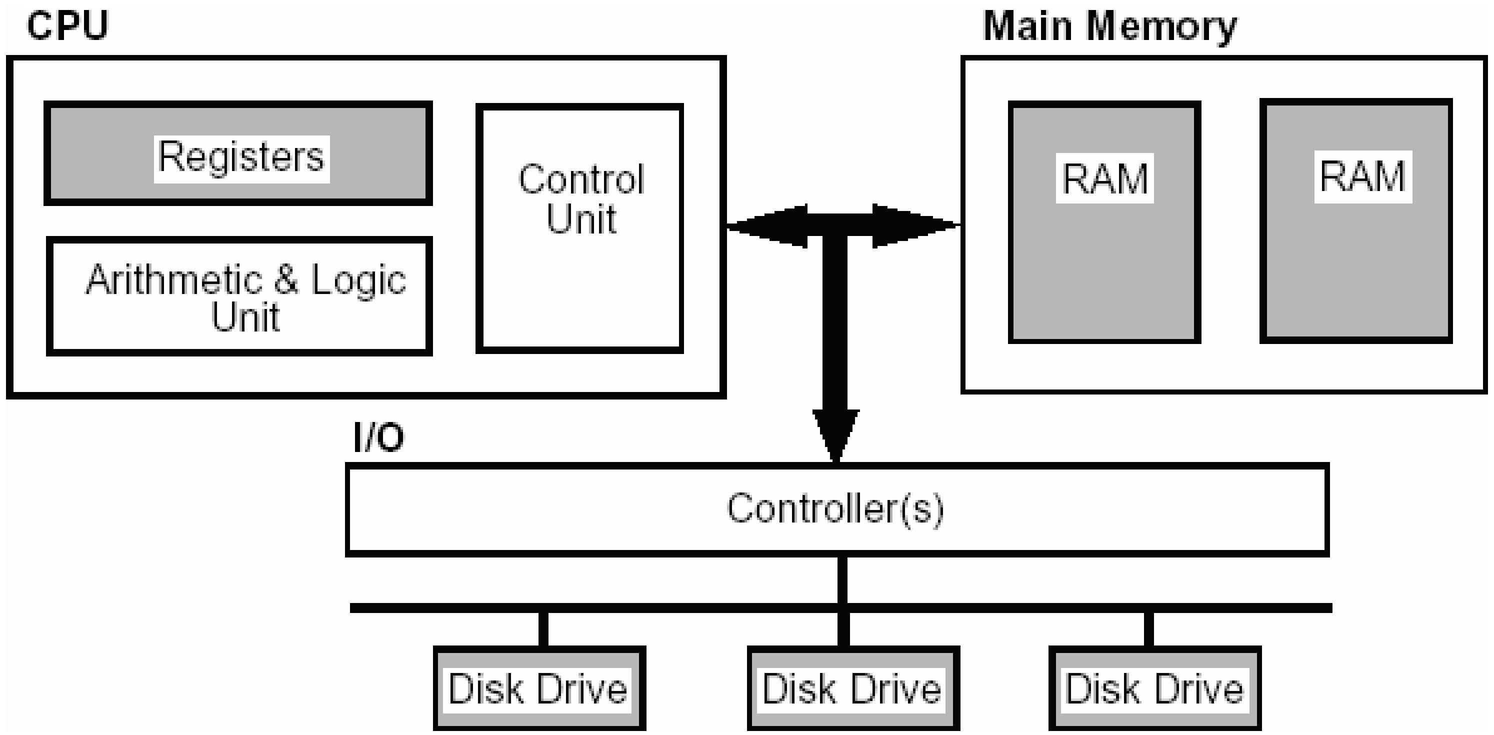
What is the sizeof this struct?



```
struct Time
{
    int hour;
    int minute;
    int sec;
    char m;
};
```



Memory Retrieval

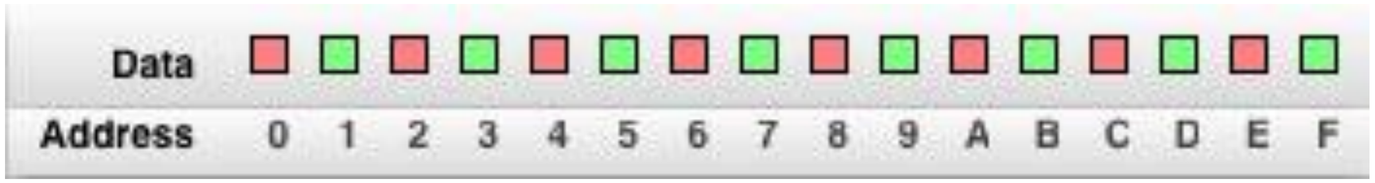


https://www.doc.ic.ac.uk/~eedwards/compsys/memory/index.html
https://www.ibm.com/developerworks/library/pa-dalign/index.html

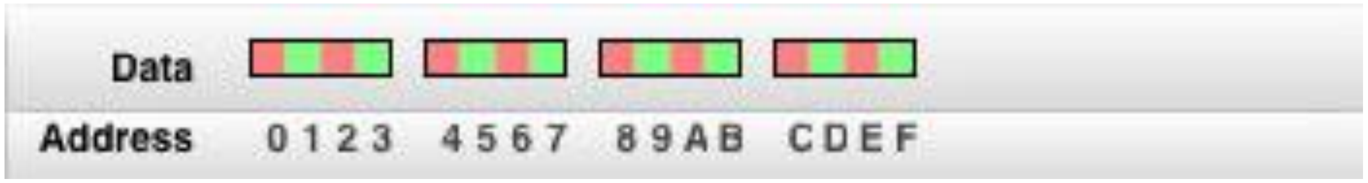
Memory Granularity

How much information will be read at a time?

How programmers see memory

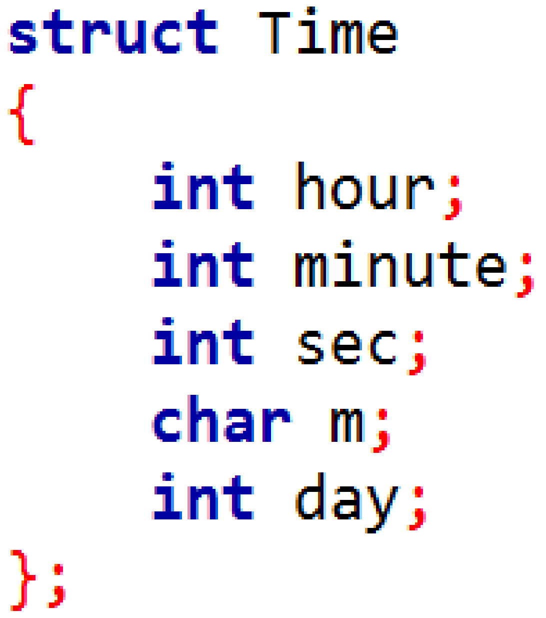


How processors see memory



https://www.doc.ic.ac.uk/~eedwards/compsys/memory/index.html
https://www.ibm.com/developerworks/library/pa-dalign/index.html

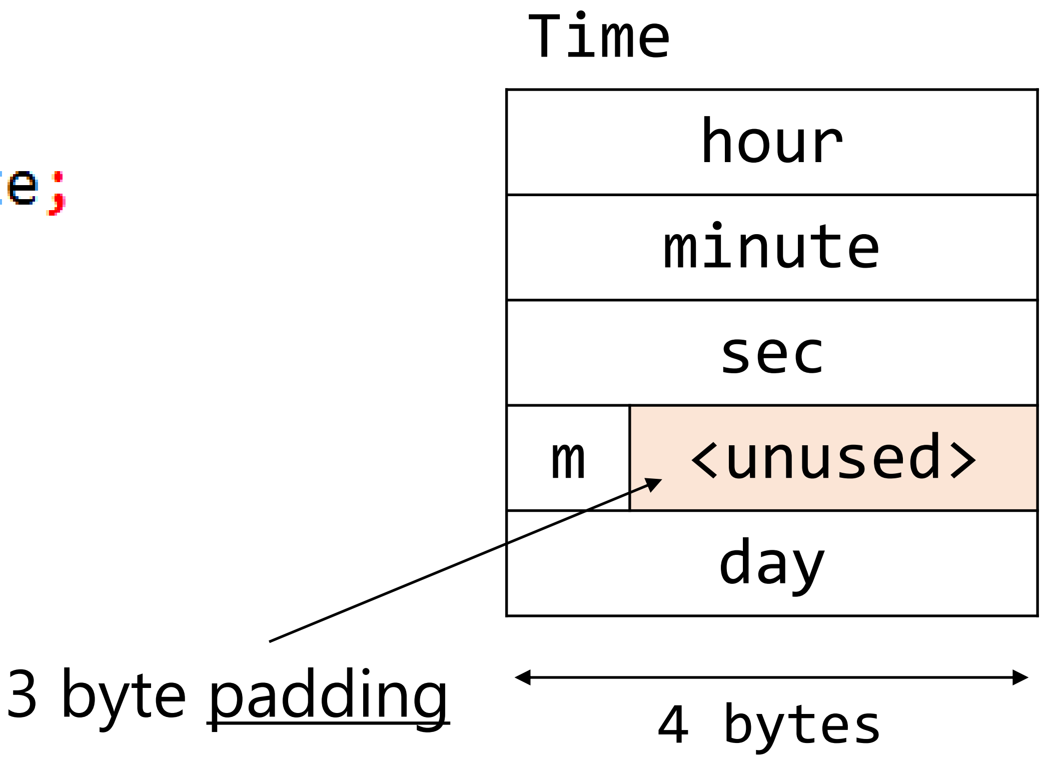
What is the sizeof this struct?



```
struct Time
{
        int hour;
        int minute;
        int sec;
        char m;
        int day;
};
```

What is the sizeof this struct?

```c
struct Time
{
    int hour;
    int minute;
    int sec;
    char m;
    int day;
};
```



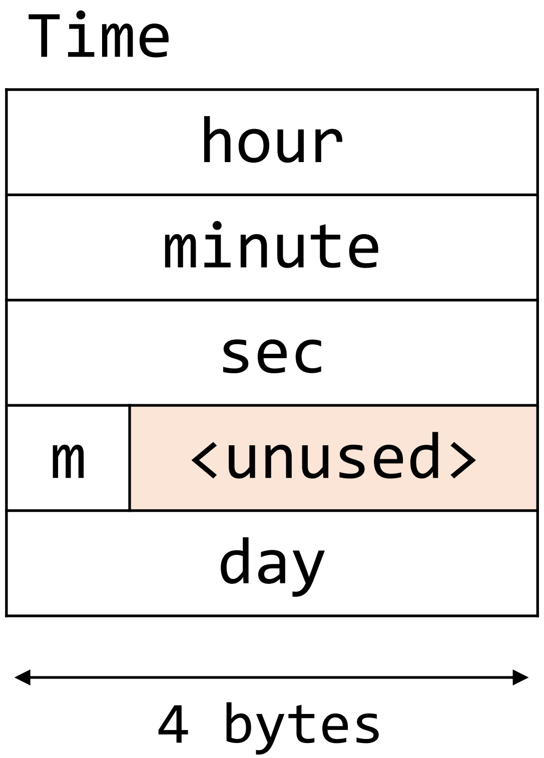
Memory Alignment rules for struct

- Size of a struct will be divisible by the size of largest member

- Starting address of each member will be divisible by it’s size

- char and char[] are special, they can be placed anywhere

- Padding is order-dependent



Task

Find out the size of the following `structs`

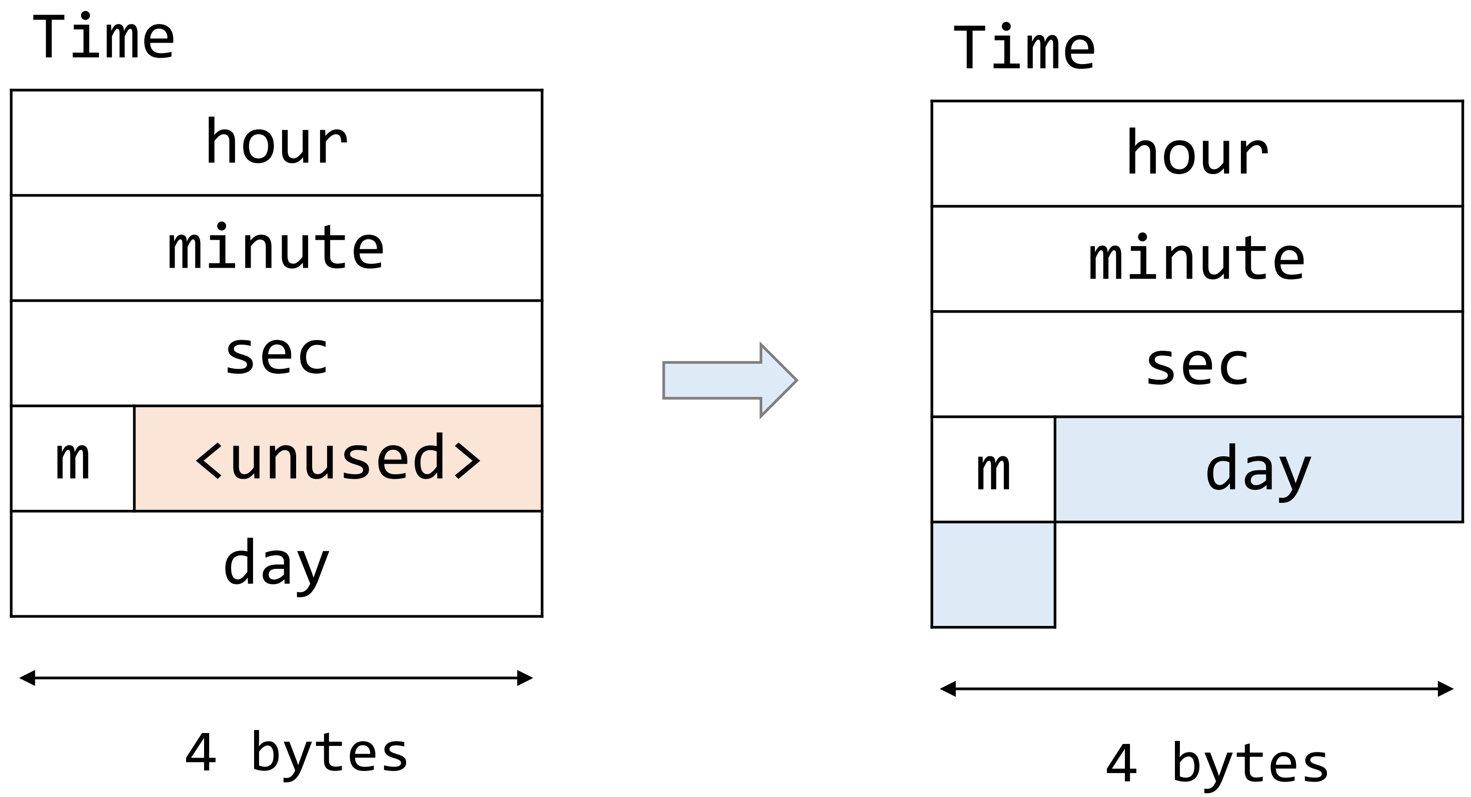
```
struct A
{
    char c;             //1 byte
    short int i;        //2 byte
};


struct B
{
    short int a;        //2 byte
    char c;             //1 byte
    int i;              //4 byte
};
```

Technique to reduce wastage

Technique to reduce wastage

1. Tell the compiler not to pad



Technique to reduce wastage

2. Declare variables in ascending/descending order of size

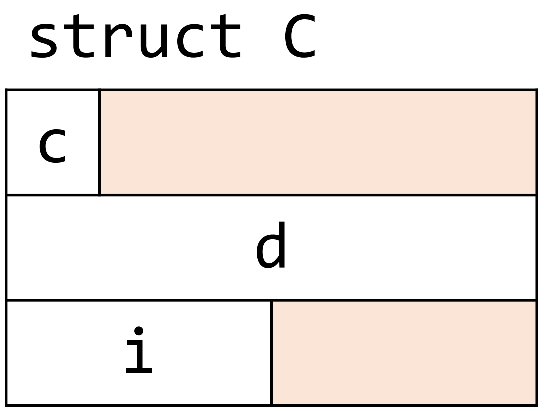
```c
struct C
{
    char c;           //1 byte
    double d;         //8 byte
    int i;            //4 byte
};
```

```c
struct D
{
    char c;           //1 byte
    int i;            //4 byte
    double d;         //8 byte
};
```

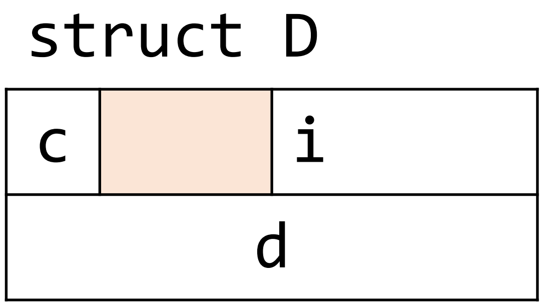
Technique to reduce wastage

2. Reorder the variable declaration sequence

```
struct C
{
    char c;          //1 byte
    double d;        //8 byte
    int i;           //4 byte
};
```

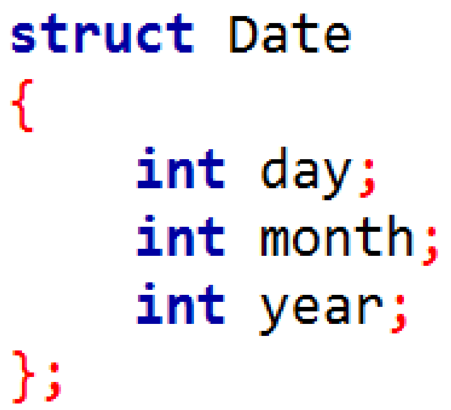


```
struct D
{
    char c;          //1 byte
    int i;           //4 byte
    double d;        //8 byte
};
```



Technique to reduce wastage

3. Use bit fields



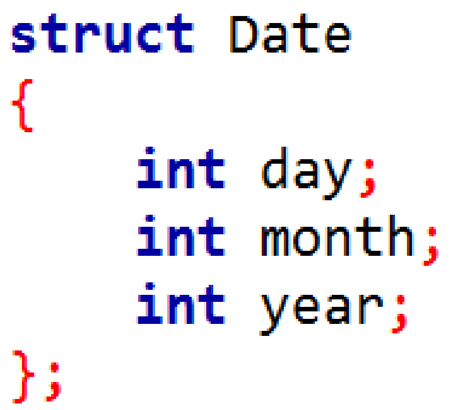
```
struct Date
{
	int day;
	int month;
	int year;
};
```

Each int (if unsigned) can hold = $2^{32} - 1$ = 4,29,49,67,295

How many bits should a day require?

Technique to reduce wastage

3. Use bit fields



```
struct Date
{
	int day;
	int month;
	int year;
};
```

Each int (if unsigned) can hold = $2^{32} - 1$ = 4,29,49,67,295

How many bits should a day require?

Technique to reduce wastage

3. Use bit fields

```
struct Day
{
    unsigned int day : 5;
    unsigned int month : 4;
    unsigned int year : 11;
};
```

Number of bits day should occupy

What will be the highest value of year?

What will be the overall size of struct Day?

Restrictions of Bit Fields

- We cannot have pointers to bit field members as they may not start at a byte boundary.

- It is implementation defined to assign an out-of-range value to a bit field member.

- Bit fields cannot be static in C.

- Array of bit fields is not allowed.