التاريخ: / / الموضوع: _____

Lec 14

The c language supports the user-defined data types like:
- structure
- Union
- Enum

structure:-
- A structre is used to and store a (Package of data), related to each other, that describe a thing
- You need to specify, how this Package of data looks like?
- How many elements (structure members) in this structure?
- How this elements ordered in this structure?
- What is the type of each element in this structre?

- in c Programming, a struct (or structure) is collection of variables [can be different types] under a single name.

التاريخ: / /

الموضوع: _____

syntax of structure in c

- The struct key keyword
- struct name
- struct element

```
struct  struct_name{
        Data type member1_name;
        Data type member2_name;

};
```

for EX:-

```
struct student{
  char name[30];
  float degree;
  unsigned short Id;
};
```

* لحينا كدا حجزنا مساحة في الميموري

* بحجز مساحة في الميموري لما اعمل object من الـ structure

* الـ structure بعملها في الـ header file ولما نعوز استخدمه اعمل include للفايل h. في الـ main.c مثلا

syntax of creating object from structure in c

Local → inside main or { }
Global → out side Main → in space.

```
Struct student student1;
```

↳ student1 is object or variable from type struct student

```
Struct student ahmed, mohamed; list of creating object
```

```
int main()
{
 struct student_type{      <- Local structure type
   char Name [20];
   int degree;
   int id;
 }; // you can add object here befor; and still local

 struct student_type S1;   <- Local structure object

 return 0;
}
```

لازم الـobject يكون معاه
في نفس الـ scope ← {}

* بعمل حاجه زي كدا مثلاً لو أنا شغال و مش عايز حد يشوف الـ struct دي غير المكان اللي عرفتها فيه ← {}

```
struct student_type{        <- Global structure Type
    char Name [20];
    int degree;
    int id;
 } amr, ramy;               <- Global structure object

struct student_type sayed;

int main(){          Global structure object <- sayed, amr, ramy

}
```

التاريخ: / /   الموضوع: ______

* it's optional to provide your structure a name, we suggest you to give it aname.

```
struct {
    char student_name [30];
    float grade;
    int id;
} samy;    ← Global structure object
```

* ممكن مكتبش اسم الـ struct عند كلمة struct ولكن لازم اكتبها بعد { وقبل ; سواء كان local أو global

```
int main(){

    struct {
        char student_name[30];
        float grade;
        int id;
    } samy;    Local structure object

}
```

الموضوع: التاريخ: / /

Special characteristics of structure in c

- A struct doesn't contain a member with incomplete or function type (except the flexible arrays).

- That is the reason at the time of structure declaration it cannot contains the instance of it self but contains a pointer to itself.

[1]

```
struct student_type{
 char name [30];
 float degree;
 struct student_type std; -> error -> incomplete type
};
```

مينفعش أكريت obJ وأنا بعمل def لل struct بداخله

[2]

```
struct student_type{
 char name [30];
 float degree
struct student_type *std_ptr -> true
};
```

ينفع اعمل create لبوينتر
في نفس ال struct من نفس النوع

[3]

```
struct student_type{
 char name [30];
 float degree;
 char student_address[ ];
};        Valid in c99 and c11
```

ينفع اعمل array و محددش
ال size بتاعها بس لازم
تيجي في آخر ال struct

التاريخ: / / الموضوع: _____

3] struct student_type{
char student address[ ]; → error
};

* مينفعش اعمل array وحددش الsize بتاعها وخصوصا لو الاراى دى هى العنصر الوحيد اللى فال struct

-A flexible array member is the official cool name for what used to (usually) be called → the "struct hack"

4]

```
struct student{
  char name[30];
  float degree;
  int id;
  char std_address[.];
};
```

Valid on cygwin, c11

```
struct student{
  char name[30];
  char std_address[ ];
  float degree
};
```

error flexible array member not at end of struct

Notes

Gcc permits a c structure to have no members.

* Also you can create objects from this type

```
struct student{
};
struct student std_empty;   //This object doesn't consumed any size in the memory
```

Note

- in c++, the struct keyword is optional before in declaration of a variable.

- in C, it is mandatory.

لازم وأنا شغال في ال C اكتب كلمة struct وأنا بستخدم ال كلمة لما ال c++ لا مش لازم الموضوع بيبقى إختياري.

Initialization of structure in C:

- We cannot initialize the member of the structure at the time of the structure declaration because there is no memory is allocated to the members at the time of declartion.

```
Struct student_type{
    char name[30];
    float degree=2.5;
    int id = 34;
};
```

→ ILLEGAL

- A structure type declaration is only template:
- There is no memory reserved for the structure until a variable is declared.

- We can only initialize the structure member using the curly braces {}

- There are Many Methods to initialize a structure object "Variable"

EX:

```
struct student_type {
        char student_name [30];
        float student_degree;
        int   student_id;
};
```

①

```
struct student_type ahmed {"Ahmed", 3.4, 8};
```

- الطريقة دى بعمل initi لعناصر الـ struct بالترتيب على حسب كل نوع من العناصر يعنى الأول array of char يبقا لازم الأول يبقا string بين " " والتانى float والتالت عدد صحيح عشان لو غلطت عكس كدا هيحصل مشاكل فى الكود ممكن اخطو بعد آخر عنصر ومكنش لسه آخر عنصر فقط

②

تانى طريقة وهى بان access عناصر الـ struct باستخدام (.) dot operator ⟶ objectname . element ;

```
struct student_type ahmed;
strcpy ( ahmed.student_name, "Ahmed");
ahmed . student_degree = 4.5;
ahmed . student_id = 8;
```

③

```
struct student_type ramy = {
    . student_name = "Ramy Elsayed",
    . student_id = 22,
    . student_degree = 5.6,
};
```

Valid on C99 and C11 {designated initializer}

Note

* لو عايز اعرف ال size بتاع ال obj struct :-

```
Printf("%d", sizeof(obj_name));
```

* ال size بتاع ال obj بيبقا أكبر من أو يساوى مجموع حجم كل العناصر مع بعض

Use of typedef with a structure

- When we use typedef with structure then it creates the alias of the structure. (شبيه)
- There is no need to write struct keyword every time with a variable declaration that means typedef save extra keystroke and make the code cleaner and readable.

```
typedef struct {
    char name[30];
    float degree;
    int id;
} Student_t;
```

Alias

ممكن اكتب هنا إسم واستعمل الطريقة السابقة
ex:
struct student_type ahmed; ①

على إفتراض إن دى مكتوبة بعد struct

②
student_t ahmed;
→ object ليس
وإنما alias فقط

← فى اللحظة دى ال إتنين دول متساويين ①②

* وبالتالى يفضل إن استعمل typedef دايماً بالشكل السابق دا واكتب بعدها alias.

Accessing structure member using Pointers

* عشان اقدر استعمل البوينتر مع الـ struct لازم البوينتر يكون من نفس نوع الـ struct اللي هيشاور عليه

* يقدر الوصول العناصر بتاع الـ struct من خلال البوينتر باستخدام الـ arrow operator ->

From last ex:-

```
Student_t ahmed = {"ahmed", 3.4, 10};
Student_t *Ptr = NULL;
```

* كدا عملت Pointer من نوع student_t وبشاور على NULL علشان اخليه يشاور على ahmed

```
Ptr = &ahmed;

Printf("%s\n", Ptr->name); // ahmed
Printf("%0.2f", Ptr->degree); // 3.4
Printf("%d", Ptr->id); // 10
```

Dynamic memory allocation for a struct object

* لو عايز اعمل أوبجكت من نوع struct واخزنه وال malloc هستعمل heap

```
student_t *Ptr = NULL
Ptr = (student_t *) malloc (size of (student_t));
```

Casting لان الـ malloc بترجع void Pointer فلازم اعملها

* لازم اعمل Validation على الـ Pointer دا عشان لو معرفش يحجز مساحة في الـ heap هيرجع NULL

* ممكن اعمل function اباصيلها من نوع struct Pointer
ومن خلال البوينتر دا اقدر اطبع الداتا الـ obj

* ممكن أيضا اعمل فانكشن اباصي ليها struct object وتطبع من خلال obj
دا الداتا الخاصة بيه
← هباصي by refrence ← & قبل اسم obj ← مهم

* الأفضل إن استخدم طريقة Pointer عشان بيوفر مساحة بعني طريقة
الـ Pointer بعمل Pointer فقط على عكس طريقة الفانكشن بتعمل copy لمحتوى
الفانكشن عشان كدا بستخدمها by refrence عشان ميعملش copy
← وبالتالي طريقة الـ Pointer بتجزي حجم الـ Pointer وفقط
إنما طريقة الـ struct بتجزي حجم الـ function كامل وممكن يحتوي
على recursive ← فانكشن

---

* وأنا بستعمل الـ scanf مع الـ struct Pointer لازم احطهم بين ()
واحط & قبل () بس مع الـ string بستخدم دالة الـ gets وليكن هنا

```
void get_student_data(student_t *student){
    gets(student -> name); // take one parameter -> array of character
    scanf("%f", &(student -> degree));
    scanf("%d", &(student -> id));
}
```

scanf بسبب الـ string عشان الـ space

* بنفع اعمل return لفانكشن من نوع struct بس احدد لها
اسم الـ struct اللي بتعامل بيه يعني الـ return بتاع الـ فانكشن
لازم يكون نوع student_t بس

* وبالتالي كدا بنفع اعمل return لأكثر من عنصر زي الـ Pointer في نفس الوقت

التاريخ: / /

الموضوع: _____

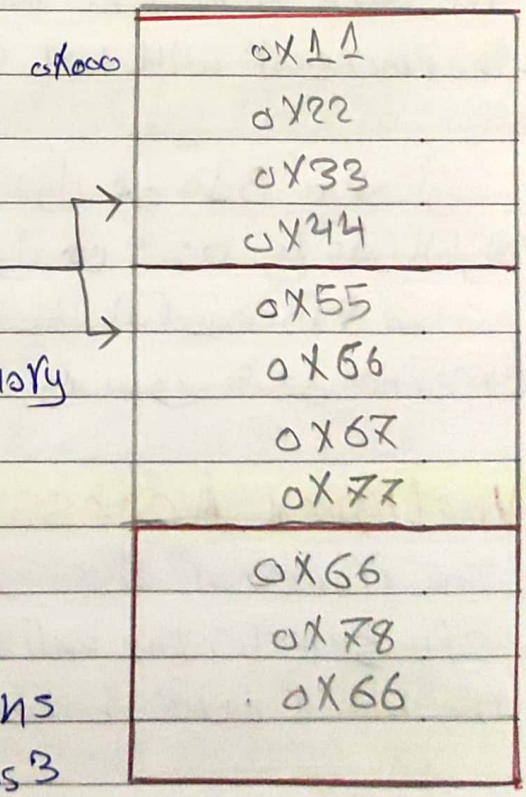
Understanding of structure Padding in C with alignment

Memory Access Boundry: -

- in real-world, Processor, does not read or write the memory byte by byte but accually, for Performance reason it accesses in the formates like 2, 4, 8, 16 and 32 bytes of chunk at a time.

- the hardware designers often restrict the Processor such that it can access memory only at certain boundaries.

Ex. Processor be able to acces the memory only at four byte boundaries



- The Processor has to access the memory location with address 0 and then reads the four consecutive bytes (from address 0 to 3).

- Next it has to use shift operations to separate the content of address 3 from other theree bytes (from 0 to 2).

- similarly, the Processor can access address 4 and reads another 4 bytes chunk from 4 to 7

- finally, shift operations can be used to separate the desired byte (at address 4 from other three bytes.

- Memory access boundary limitation exists because making certain assumptions about the address can simplify the hardware design.

Memory Acces Boundary And Alignment :
(Aligned Access Restriction)

- When memory is aligned then Processor easily fetches the data from the memory
- the Processor will not generate any exception or fault

* الطريقة دى لو أنا عايز 4byte والـ 4bytes دول من 3 لـ 10 مثلا أو أنا مخزن داتا من نوع int فحجز الـ 4byte دول من 3 : 0 عكس الطريقة اللى فاتت دى لو أنا مخزن من 1 : 4 وبالتالى الـ Processor هياخد exception وقت ما يروح يعملها access وفى النوع دا البروسيسور هيعمل

Unaligned Access:

- the Processor take some ticks to access the unligned memory
- CPU selects the unligned memory which represents through the black dark border.

الموضوع: _____ التاريخ: / /

الفرق بين Aligned & unAligned ؟

* لو أنا مثلا عندي مجموعة من ال var مثلا int char short غالبا Aligned لما يعمل تخزين الداتا بيقا فيه فراغات وبالتالي دي أماكن فاضية في الميموري ← waste in memory
على عكس ال unaligned بيخزن كله ورا بعضه ومش بيقا عندي اماكن فاضية أو بعمل آخر فراغات ← waste

- Generally, the compiler handles the scenario of alignment and it aligned at the variable in their boundary.

Note

Alignment of datatypes mandated by the Processor architecture, not by language

Structure Padding :-

- When you create the structure of union then compiler inserts some extra bytes between the member of structure or union for the alignment

- These extra unused bytes are called Padding bytes and this technique is called structure Padding inc

- Padding increasses the Performance of the Processor at the Penalty of memory.

الموضوع: ______ التاريخ: / /

- In structure or union data member aligned as per the size of the higest bytes member to prevent the penalty of performance.

size ≥ {+}

* لو جبت ال size لـ obj من نوع struct ولقيته أكبر من مجموع العناصر كدا في Padding لو بيساوى كدا مفيش Padding

* ال size بتاع ال obj لـ struct بيتحجز عن طريق إن بحجز لكل عنصر من عناصر ال struct بأكبر size من العناصر.

```
struct test{
  char var1;
  unsigned int var2;
  char var3;
}obj;
```

* هنا هيحجز 12 byte ←
4 ← int ← أكبر حاجة
1 ← char ← var1 ← +3 ← Padding
1 ← char ← var3 ← +3 ← Padding

```
struct test {
    char var1;
    int var3; char var2;
    char *ptr;
}obj;
```

* هنا هيحجز على حجم ال Pointer لأن حجم ال Pointer أكبر وبالتالى هيحجز 24 byte

1 → char → +3 Padding
4 → int
1 → char → +7 Padding
8 → Pointer

* ومن ظهر مشاكل ان عندى بحجز أماكن وبيبقا عندى Padding كبير ممكن أقلل حاجة زى كدا عن طريق إن رتب العناصر حسب الحجم من الكبير للصغير

التاريخ: / / الموضوع: _____

* عشان احجز حجم العناصر وميتقاش فيه Padding بستعمل structure Packing ← عكس الـ Padding

*Pragma Pack(1) ← بكتب قبل وبعد الـ struct

struct test{
char var1;
double var2;
char var3;
}obj;

* هنا هيحجز في الميموري 10 bytes فقط

*Pragma Pack(1)

* لو غيرت الـ 1 لـ 2 كده هيعمل bdd 2bytes & Padding وبالتالي الحجم على المثال الاخير هيبقى 12

Conclusion:-

- Memory alignment increase the Performance of the processor and we have to care the alignment of the memory for the better Performance of the Program.

- CPU Performs better with aligned data as compared to unaligned data because some Processor takes an extra cycle to access the unaligned data.

- When we create the structure, union the we have to rearrange the member (largest to smallest) in a careful way for the better performance of the program.

- the size of structure must be divided by the size of largest member in the structure, to force the alignment when we create array of this structure.

- we can avoid structure padding by structure packing using the preprocessor directive provided by the compiler #pragma pack (this is compiler dependent).

Understanding of structure and Bit field:-

* في الإمبيديد سيستم في الميكروكنترولر تحديدًا الأطراف أو الرجول الـ Pins بتبقا متقسمة إلى Ports كل Port بيبقا مثلا 8 Pins مع بعض فأنا بعوز مثلا أكتب في 8 Pins مع بعض أو أكتب على Pin معين مثلا لو عايز أخرج على Pin معين 1 فأنا لما أخرج 1 على Pin دا فكدا بيكافئ 5 volt وأيضا 0 بيكافئ zero volt فأنا بعمل حاجة زي دي مثلا إزاي ؟!

```
typedef struct {
    unsigned char Pin0   :1
    unsigned char Pin1   :1
    unsigned char Pin2   :1
    unsigned char Pin3   :1
    unsigned char Pin4   :1
    unsigned char Pin5   :1
    unsigned char Pin6   :1
    unsigned char Pin7   :1
} Port_reg_t ;
```

(:1 →) لو حددت له الـ Pin دا هياخد 1 bit فقط

التاريخ: / /　　　　　　　　　　　　　　　　　　　　الموضوع:

* فأنا لما أحتاج اعمل access للـ pins دا Write / Read

```
Port_reg_t reg1;
reg1.Pin0 = 1;   /* 0000 0001 */
reg1.Pin1 = 1;   /* 0000 0011 */
reg1.Pin2 = 1;   /* 0000 0111 */
```

* طب لو أنا محتاج اكتب في العناصر كلها مع بعضها
استعمل حاجة إسمها union

```
typedef union {
    struct {
unsigned char Pin0 :1
unsigned char Pin1 :1
unsigned char Pin2 :1
unsigned char Pin3 :1
unsigned char Pin4 :1
unsigned char Pin5 :1
unsigned char Pin6 :1
unsigned char Pin7 :1
    };
    unsigned char AllPort;
} Port_t;
```

* الـ Union معناه إن أي حاجة بتبقا مشتركة
يعني الـ هكتبه في AllPort هيكتب في عناصر الـ struct
ويبقوا متحدين مع بعض

* علشان اجمع تلت Pins مع بعض وتحددهم مثلا :3 ← :1

* الـ size بتاع الـ Port_t هيساوي ← 1 byte

```
Port_t Port;
Port.AllPort = 0x55; // /* 0101 0101 */
```

($P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0$)

كدا أنا كتبت في كل عناصر الـ struct ده القيمة دي بالترتيب كدا

Array of structure :-

Ex:-

```
typedef struct {
   char student_name[30]; //Array with in structure.
   float student_degree
   int id;
} student_t;

student_t student[2];
```

* كما أنا عرفت array من نوع struct كل عنصر في الاراى عبارة عن struct.

```
strcpy(
student[0].student_name, Mostafa);
student[0].student_degree = 30.0;
```

} Method one for initilize

```
student_t student[2] = {
{
   "Mostafa", 3.5, 2
},
{
   "Mohamed", 5.6, 1
}
};
```

} Method two for initilize

* ممكن استعمل طريقة designated initializer.

الموضوع: ________ التاريخ: / /

student_t *student_ptr [2] ;

* هنا انا عامل array من نوع struct كل عنصر فيها عبارة عن pointer بيخزن عنوان داتا من نوع struct obj

* لما اجي access الelem الي في الstruct بإستخدام الpointer هعملها access بإستخدام ال ->

* ممكن يقابلنا struct inside struct ولما اجي اعمل access لعنصر منك هستعمل . مرتين عشان اوصل لstruct جوا struct

obj name . outside struct element . inside struct element

---

* مينفعش اعمل obj من نفس نوع ال struct

```
typedef struct {
    student_t ahmed; //error
    char name[20];
    int id ;
} student_t;
```

حتى لو عملت pointer وحط * قبل ahmed هيعملي error

عشان اتجاوز الerror دا هعمل حاجه من اتنين

1- هكتب كلمة struct قبل student_t وفي الحالة دي هتقبلها عادي لكن انا بقوله اعمل create لpointer of struct ; هكتب struct name وهستخدم الname عشان اعرف الpointer

2- همسح كلمة typedef وال alias الي تحت واعملها بطريقة الstruct العادية

← ومن هنا جت فكرة ال linked_list وهي عبارة حاجه بتشاور على حاجه من نفس نوع هنشرحها قدام في DS

Ex:-

```
typedef struct student{
    struct student *std_freind;
    char name[20];
    int id; details d;
}student_t;
```

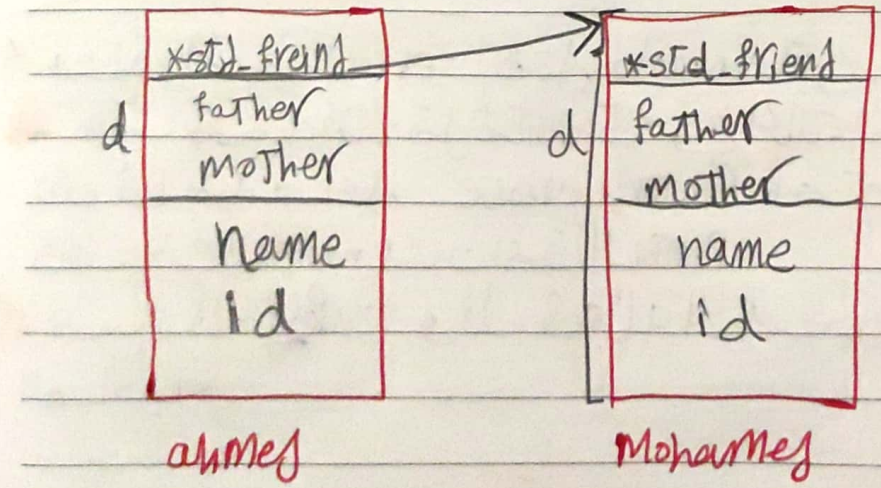
```
typedef struct{
char Father[20];
char Mother[30];
}details
```

```
student_t ahmed, Mohamed;

int main()
{
ahmed.std_freind = &Mohamed;

strcpy(Mohamed.d.Father, "Ali");

printf("%s\n", ahmed.std_freind->d.father);
```

خزنت عنوان Mohamed في ahmed

* طبعت اول اسم أبو محمد من تحت ahmed

IMPORTANT



*implement a pointer to struct type in the same struct type "self Referential structure"

التاريخ: / /

Notes

الموضوع: _____

* ممكن استعمل Pointer to function كـ element داخل struct

* استعمل جاهزي كدا في Softwure Application layer
توضيحها في المحاضرة السابقة

* مينفعش اكتب function داخل الـ structure

* متنساش () بعد الـ Pointer عشان استعمل الـ funtion اللي الـ Pointer بيشاورعليها