Лекція 8. Структури

- З метою підвищити зручність використання мови асемблер в нього також був введений такий тип даних, як структура.

- По визначенню ***структура*** — це тип даних, який складається з фіксованого числа елементів різного типу.

Структури

- Для використання структур у програмі необхідно виконати три дії:
- Задати *шаблон структури*.

  За змістом це означає визначення нового типу даних, що згодом можна використовувати для визначення перемінних цього типу.
- Визничити шаблон структури.

  Цей етап має на увазі ініціалізацію конкретної змінної заздалегідь визначеною (за допомогою шаблона) структурою.
- Організувати *звернення до елементів структури*.

Структури

- ***ОПИСАТИ*** **СТРУКТУРУ в програмі означає лише вказати її схему чи шаблон - пам'ять при цьому не виділяється.**
- **Цей шаблон можна розглядати лише як інформацію для транслятора про розташування полів і їхні значення за замовчуванням.**

- ***ВИЗНАЧИТИ*** **СТРУКТУРУ — передбачає надавання вказівки транслятору виділити пам'ять і привласнити цій області пам'яті символічне ім'я.**
- **Описати структуру в програмі можна тільки один раз, а визначити — будь-яку кількість разів.**

Описання шаблона структури

- **Описання шаблона структури має наступний синтаксис:**

ім'я_структури STRUC

<описання полів>

ім'я_структури ENDS

- ***<описання полів>*** являє собою послідовність директив опису даних **db, dw, dd, dq** і **dt**.
- Їх операнди визначають розмір полів і, при необхідності, початкові значення. Цими значеннями можуть ініціалізуватися відповідні поля при визначенні структури

Опис шаблона структури

- *Місце розташування* шаблона в програмі може бути будь-яким, але за логікою роботи однопрохідного транслятора, воно повинно розташовуватись до того місця, де визначається змінна з типом даної структури. Тобто при описанні в сегменті даних змінної з типом деякої структури її шаблон необхідно помістити на початку сегменту даних або перед визначенням змінної даної структури.

Структури

- **Розглянемо роботу зі структурами на прикладі моделювання бази даних про співробітників деякого відділу.**
- **Для простоти, щоб вирішити проблему перетворення інформації при введенні, домовимося, що всі поля символьні.**
- **Визначимо структуру запису цієї бази даних наступним шаблоном:**

Структури

```
worker  struc
;інформація про співробітника
nam db30dup  (' ')    ;прізвище, ім'я, по батькові
sex db  ‘ч‘           ;стать, за замовчуванням ‘ч'
position    db30 dup (' ')    ;посада
age  db2 dup(‘ ’)         ;вік
standing   db2 dup(‘ ’)   ;стаж
salary   db4 dup(‘ ’)     ;оклад у гривнях
birthdate   db8 dup(‘ ’)  ;дата народження
worker  ends
```

Визначення даних з типом структури

- Для використання описаної за допомогою шаблона структури в програмі необхідно визначити змінну з типом даної структури. Для цього використовується наступна синтаксична конструкція:

- [ім'я змінної] ім'я_структури <[список значень]>

Визначення даних з типом структури

- *ім'я змінної* — ідентифікатор перемінної даного структурного типу.
- Завдання імені *змінної* необов'язково. Якщо його не вказати, просто буде виділена область пам'яті розміром у суму довжин всіх елементів структури.
- *список значень* — взятий у кутові дужки список початкових значень елементів структури, розділених комами. (Його визначення також необов'язкове. Якщо список зазначений не цілком, то всі поля структури для даної змінної ініціалізуються значеннями із шаблона, якщо такі задані)

Визначення даних з типом структури

- Допускається ініціалізація окремих полів, але в цьому випадку пропущені поля повинні відокремлюватися комами. Пропущені поля будуть ініціалізовані значеннями із шаблона структури. Якщо при визначенні нової змінної з типом даної структури ми згодні з усіма значеннями полів у її шаблоні (тобто заданими за замовчуванням), то потрібно просто написати кутові дужки.
- Наприклад: **victor worker <>**.

Як приклад визначимо декілька змінних з типом описаної вище структури.

```
Data   segment
Sotr1  worker    <’Гурко Андрій Вячеславович’, ,’ художник’,’33’,‘15’,‘1800’,’26.01.64’>
Sotr2  worker    <’Михайлова Наталя Генадієвна’,’ ж',’програміст’,’30’,’10’,’1680’,’27.10.58’>
Sotr3  worker    <’Степанов Юрій Лонгинович’,, ’ художник’,’38’,’20’,’1750’,’01.01.58’>
Sotr4  worker    <’Юрова Олена Олександрівна’, ’ж', ’связист’,’32’,’2’,,’09.01.66’>
Sotr5  worker    <>
;тут усі значення за замовчуванням
Data ends
```

Методи роботи зі структурою

- Ідея введення структурного типу в будь-яку мову програмування складається з об'єднання різнотипних змінних в один об'єкт.
- В мові повинні бути засоби доступу до цих змінних усередині конкретного екземпляра структури. Для того щоб послатися в команді на поле деякої структури, використовується спеціальний оператор — *символ "." (крапка)*. Він використовується в наступній синтаксичній конструкції:

Адресний_вираз.ім'я_поля_структури

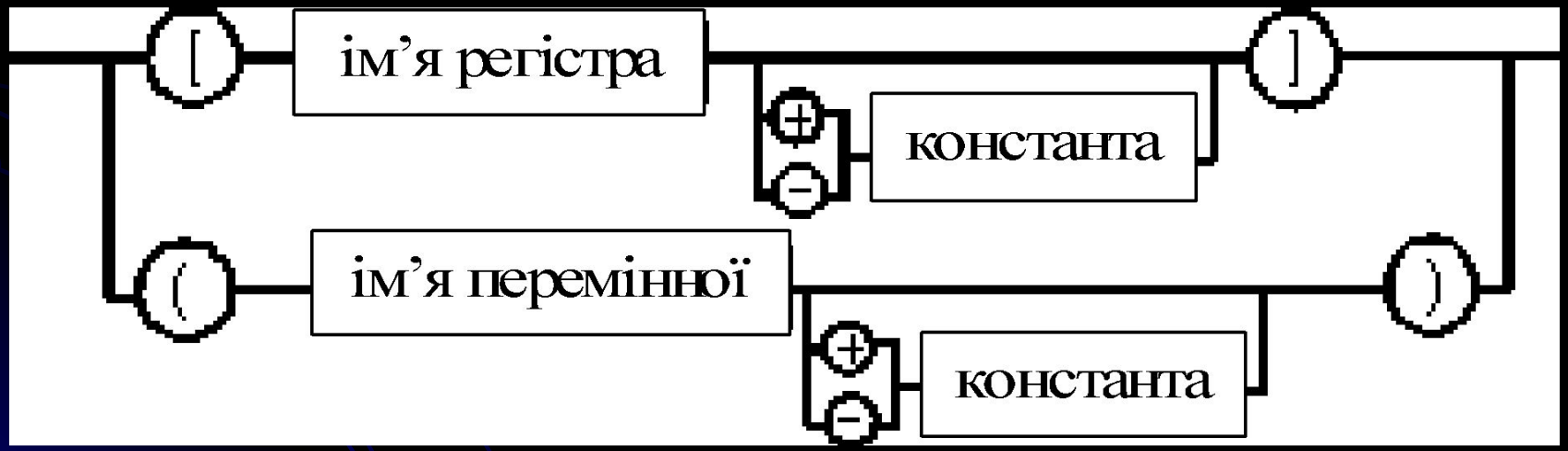
Методи роботи зі структурою

- *адресний_вираз* — ідентифікатор змінної деякого структурного типу чи вираз в дужках відповідно до зазначеного нижче синтаксичних правил (мал. 1);
- *ім'я_поля_структури* — ім'я поля із шаблона структури (це та ж адреса, а точніше, зсув поля від початку структури)

Методи роботи зі структурою

- У такий спосіб оператор "." (крапка) обчислює вираження

(адресний_вираз)+(ім'я_поля_структури)



Методи роботи зі структурою

- Продемонструємо на прикладі визначеної нами структури ***worker*** деякі прийоми роботи зі структурами.
- Наприклад, витягти в **ax** значення поля з віком. Тому що навряд чи вік працездатної людини буде більше величини 99 років, то після переміщення вмісту цього символьного поля в регістр **ax** його буде зручно перетворити в двійкове представлення командою **aad**.

Методи роботи зі структурою

- Будьте уважні, тому що через принцип збереження даних *“молодший байт за молодшою адресою”* старша цифра віку буде поміщена в **al**, а молодша — у **ah**.
- Для коректування досить використовувати команду **xchg al,ah**:

```
mov ax,word ptr sotr1.age
;у al вік sotr1
xchg ah,al
```

Методи роботи зі структурою

- а можна і так:

```
lea   bx,sotr1
mov   ax,word ptr [bx].age
xchg  ah,al
```

Методи роботи зі структурою

- Якщо співробітників не четверо, а набагато більше, і до того ж їхнє число й інформація про них постійно змінюється. У цьому випадку губиться зміст явного визначення змінних з типом ***worker*** для конкретних особистостей.
- Мова асемблера дозволяє визначати не тільки окрему змінну з типом структури, але і масив структур.
- Наприклад, визначимо масив з 10 структур типу ***worker***:

```
mas_sotr worker 10 dup (<>)
```

Методи роботи зі структурою

- Подальша робота з масивом структур виконуеться так само, як і з одномірним масивом. *Як бути з розміром і як організувати індексацію елементів масиву?*
- Аналогічно іншим ідентифікаторам, визначеним у програмі, транслятор призначає імені типу структури й імені змінної з типом структури атрибут типу. Значенням цього атрибута є розмір у байтах. Витягти це значення можна за допомогою оператора **type**.
- Після того як став відомий розмір екземпляра структури, організувати індексацію в масиві структур не представляє особливої складності.

Методи роботи зі структурою

```
worker      struc
...
worker      ends
...
mas_sotr worker      10 dup (<>)
...
mov     bx,type worker          ;bx=77
lea di,mas_sotr
;витягти і вивести на екран стать усіх співробітників:
mov     cx,10
cycl:
mov     al,[di].sex
...
;виведення на екран умісту поля sex структури worker
add     di,bx   ;до наступного структурі в масиві mas_sort
loop    cycl
```

Методи роботи зі структурою

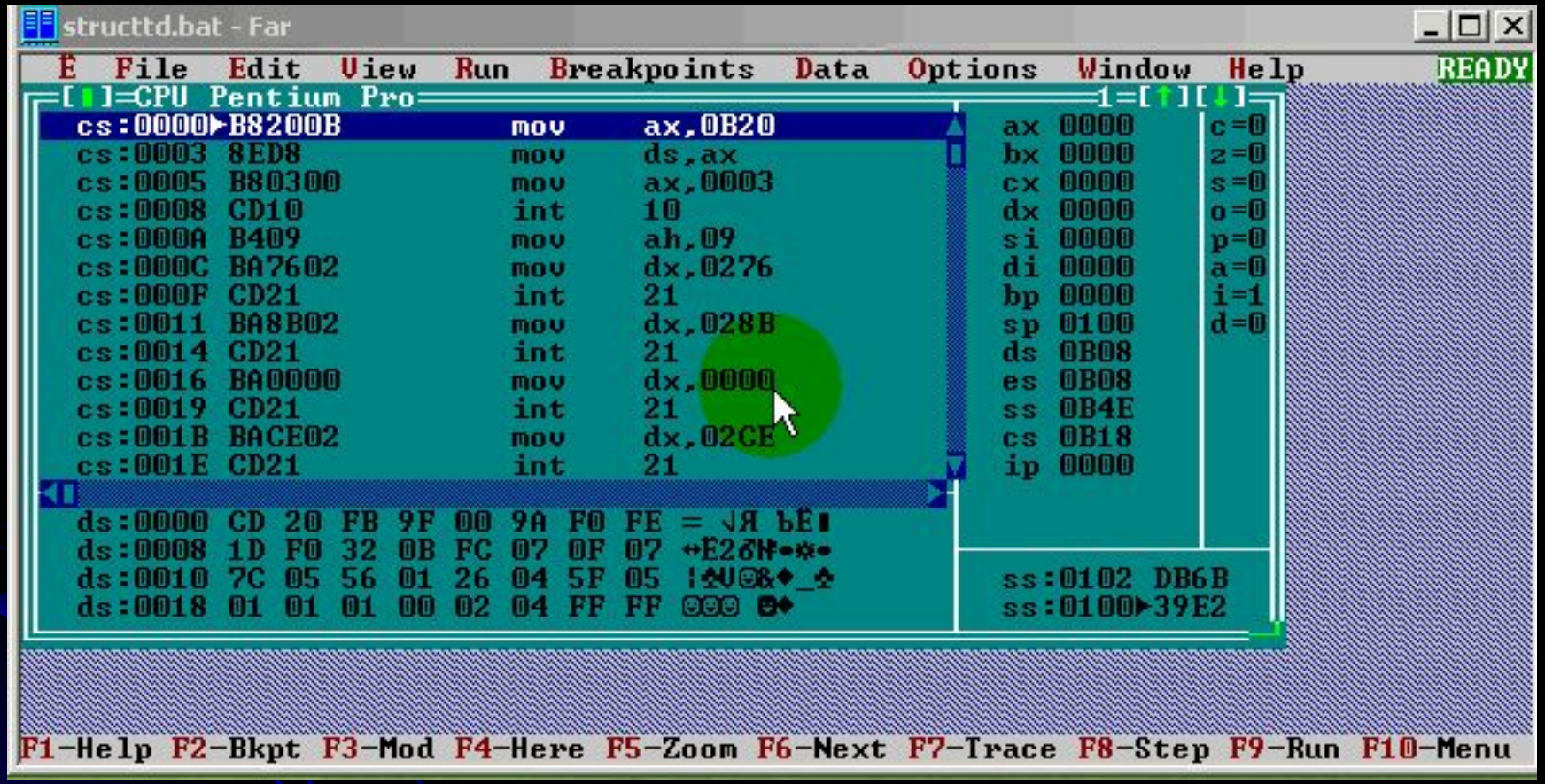
- Як виконати копіювання поля з однієї структури у відповідне поле іншої структури? Чи як виконати копіювання всієї структури? Давайте виконаємо копіювання полючи *nam* третього співробітника в поле *nam* п'ятого співробітника:

Методи роботи зі структурою

```
worker    struc
...
worker    ends
...
mas_sotrworker    10 dup (<>)
...
mov   bx,offset mas_sotr
mov   si,(type worker)*2  ;si=77*2
add   si,bx
mov   di,(type worker)*4  ;si=77*4
add   di,bx
mov   cx,30
rep   movsb
```

Припустимо, що у нас є декілька комп’ютерів (3) різної конфігурації і володіють ними різні власники.

Суть задачі полягає в тому, щоб створити структуру, заповнити її та вивести вкінці її на екран.


structd.bat - Far
File Edit View Run Breakpoints Data Options Window Help
READY
CPU Pentium Pro
cs:0000▶B8200B mov ax,0B20
cs:0003 8ED8 mov ds,ax
cs:0005 B80300 mov ax,0003
cs:0008 CD10 int 10
cs:000A B409 mov ah,09
cs:000C BA7602 mov dx,0276
cs:000F CD21 int 21
cs:0011 BA8B02 mov dx,028B
cs:0014 CD21 int 21
cs:0016 BA0000 mov dx,0000
cs:0019 CD21 int 21
cs:001B BACE02 mov dx,02CE
cs:001E CD21 int 21
ds:0000 CD 20 FB 9F 00 9A F0 FE
ds:0008 1D F0 32 0B FC 07 0F 07
ds:0010 7C 05 56 01 26 04 5F 05
ds:0018 01 01 01 00 02 04 FF FF
ax 0000
bx 0000
cx 0000
dx 0000
si 0000
di 0000
bp 0000
sp 0100
ds 0B08
es 0B08
ss 0B4E
cs 0B18
ip 0000
c=0
z=0
s=0
o=0
p=0
a=0
i=1
d=0
ss:0102 DB6B
ss:0100▶39E2
F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu