Structures

Lecture 5

Structures (a record)

A structure is a *user defined heterogeneous* data structure; e.g.

```
struct personnel /* Structure template for employee */
{
    int number;
    char surname[26];
    char initial;
    char  dob [8];
    int dept;
    char date_joined [8];
}; typedef struct personnel Personnel
// declare a variable for a structure
Personnel person;
```

Nested structures and pointers

```
struct date                    /* Structure template for a date */
{
    int day;
    int month;
    int year;
};

struct personnel     /* Structure template for employee */
{
    int number;
    char surname[26];
    char initial;
    struct date dob;
    int dept;
    struct date joined;
}; typedef struct personnel as Personnel
Personnel  person;  // declaring a variable to a structure
Personnel *ptr;   //declaring a pointer to a structure
```

Using structure

- The following are examples of accessing fields
- Using the (dot) . operator
  - person.surname // it's a string (a primary field)
  - person.dob.day // a field of a nested structure
  - person.dept

- Using pointer notations (->)
  - ptr = &person // assign struct to a pointer
  - ptr -> surname;
  - ptr ->dob.day;

Passing a structure to a function

- Like other data types a structure can be passed by value or by reference.
- By value “ pass the structure” and by reference “a pointer to the structure”.

- Do not forget to use the appropriate notation in each function (if struct use the dot operator; if pointer to struct use the ->

Passing structures to functions

denis.manley@apollo: ~/OS2/week3

```
/* A program to illustrate passing structure values to a function.
        The program reads data for a student and displays it  */

#include <stdio.h>

struct student_rec       /* global structure template   */
{
  int number ;
  char name[26] ;
  int age ;
  int scores[5] ;
};
  // prototype
void display_student_data(struct student_rec) ;
void get_student_data(struct student_rec *) ;

void main()
{

        struct student_rec student ;
        struct student_rec *student_ptr ;

        student_ptr = &student ;

        /* use a pointer to a structure variable as an argument */
        get_student_data( student_ptr ) ;

        /* use a structure variable as an argument */
        display_student_data( student ) ;
}
```

1,1 Top

Display and Get functions

denis.manley@apollo: ~/OS2/week3

```
/* Function: display_student_data() - displays student data
         Arguments: the student structure                    */
void display_student_data (struct student_rec stud)
{
        int i;
        printf( "\n\nThe data in the student structure is:" ) ;
        printf( "\nnumber is %d" , stud.number ) ;
        printf( "\nname is %s" , stud.name ) ;
        printf( "\nage is %d" , stud.age ) ;
        printf( "\nscores are: " ) ;
        for ( i= 0 ; i < 5 ; i++ )
                printf( " %d " , stud.scores[i] ) ;
        printf( "\n" ) ;
}

/* Function: get_student_data() - This function reads student data
        Arguments: a pointer to the student structure      */
void get_student_data( struct student_rec *ptr )
{
        int i ;
        printf( "Student Number: " ) ;
        scanf( "%d" , &(ptr->number) ) ;
        printf( "Student Name: " ) ;
        scanf( "%25s" , ptr->name ) ;
        printf( "Age: " ) ;
        scanf( "%d" , &(ptr->age) ) ;
        printf( "5 Test Scores: " ) ;
        for ( i= 0 ; i < 5 ; i++ )
                scanf( "%d" , &(ptr->scores[i]) ) ;
}
```

63,1 96%

Sample output

```
denis.manley@soc-apollo:~/OS2/week3$ ./Structure_functions
Student Number: 21
Student Name: denis
Age: 25
5 Test Scores: 2
45
68
12
34


The data in the student structure is:
number is 21
name is denis
age is 25
scores are:  2  45  68  12  34
denis.manley@soc-apollo:~/OS2/week3$
```

Array of structure

- An array can also hold structures
  - Typedef
    - Typedef struct personnel EMPLOYEE
  - Declare an array of structures
    - EMPLOYEE   staff[20];
  - The **empdb.c** is a program that declares an array of structures and then allows the user to
    - Add a record to array of structures
    - Delete a record
    - Display a record
    - Edit a record
  - *Implementation: Empty position in array denoted by number field contain 0*

Emp.c Main functions

```
void main()
{
        EMPLOYEE persons[MAX_PERSONS];
        int menu_choice;



        init_database(persons);


        do
        {
                menu_choice = menu();

                switch ( menu_choice )
                {
                        case 1 :
                                                add_an_employee( persons );
                                                break;

                        case 2 :
                                                delete_an_employee( persons );
                                                break;

                        case 3 :
                                                display_an_employee( persons );
                                                break;

                        case 4:                 edit_an_employee(persons);
                                                break;

                }
        }
        while ( menu_choice != 0) ;
}
```

Menu and search functions

```
int menu(void)
{
        int choice;

        /* Display the menu. */
        printf("\n\n 1. Add     an Employee\n\n");
        printf(" 2. Delete  an Employee\n\n");
        printf(" 3. Display an Employee\n\n");
        printf(" 4  Edit an Employee\n\n");
        printf(" 0. Quit\n\n");
        printf( "Please enter your choice (0 to 4) " );

        /* Get the option. */
        do
                scanf( "%d", &choice );
        while ( choice <0 || choice > 4 );

        return (choice);
}


int search_database( EMPLOYEE person_array[], int emp_number )
{
        int i = 0;


        while ( i < MAX_PERSONS && person_array[i].number != emp_number )
                i++;

         return (i);
}
```

Delete Function (assign 0 to number field)

denis.manley@apollo: ~/OS2/week3

```
void delete_an_employee( EMPLOYEE person_array[] )

{

        int employee_number;
        int pos;


        printf("Employee Number to Delete (1 to 3 digits, except 0) :");
        do
          scanf( "%3d",&employee_number );
        while ( employee_number <= 0 ) ;


        pos = search_database( person_array, employee_number );


        if ( pos == MAX_PERSONS )
          printf( "This employee is not in the database\n" );
        else
          {
                printf("Employee %3d deleted", employee_number);
                person_array[pos].number = 0;
          }
}
```

137,0-1 46%

The add function

denis.manley@apollo: ~/OS2/week3

```
void add_an_employee( EMPLOYEE person_array[] )
{
        int i=0;

              while ( person_array[i].number != 0 && i < MAX_PERSONS )
          i++;

        if ( i == MAX_PERSONS )
          printf("\nSorry, the database is full\n");
        else {
          printf( "\n\nEmployee Number (1 to 3 digits, except 0) : " );
          do
                 scanf( "%3d",&person_array[i].number );
          while ( person_array[i].number <= 0 );

          printf( "\nEmployee  Surname (Maximum 25 characters) : " );
          scanf( "%25s",person_array[i].surname );
          printf( "\n          First Name (Maximum 10 characters) : " );
          scanf( "%10s",person_array[i].first_name );
          printf( "\nDate of Birth\n" );
          printf( "      Day (1 or 2 digits) : " );
          scanf( "%2d",&person_array[i].dob.day );
          printf( "    Month (1 or 2 digits) : " );
          scanf( "%2d",&person_array[i].dob.month );
          printf( "     Year (1 or 2 digits) : " );
          scanf( "%2d",&person_array[i].dob.year );
          printf( "\nDepartment Code (1 to 4 digits): " );
          scanf( "%4d",&person_array[i].dept );
          printf( "\nDate Joined\n" );
          printf( "      Day (1 or 2 digits) : " );
          scanf( "%2d",&person_array[i].joined.day );
          printf( "    Month (1 or 2 digits) : " );
          scanf( "%2d",&person_array[i].joined.month );
          printf( "     Year (1 or 2 digits) : " );
          scanf( "%2d",&person_array[i].joined.year );
        }
}
                                                          109,1         31%
```

Why is there no & for scanf("s",....)?

Home work

- Modify the empdb.c source code:
  - To include a function to *edit* a record the array of structures.
    - ask the user for a record number;
    - If the position exists: modify the contents
    - you can simply prompt the user to re-enter all the data again
  - Test the *modified source code* to confirm it does as expected: Use specific examples