EDUCBA
(https://www.educba.com/software-development/)

← (https://www.educba.com/hash-else-in-c/)

→ (https://www.educba.com/nested-structure-in-c/)

Introduction to Structure Padding in C

Structure padding mainly talks about memory for variables which are aligned based on of the variable. Let suppose a “char” of 1 byte memory can be assigned anywhere in between like 0x5000 to 0x5001. Same way if we have an “int” of 4 bytes memory can be assigned

How Does Structure Padding Work in C?

- Structure padding is said to be in order to align the data in memory 1 or more un-occupied bytes (empty bytes) are kept between any memory addresses which are actually assigned for other data structure members at the time of memory allocation.
- If we observe the architecture of the computer processor can be read 1 word means bytes in 32 bit processor from memory at a time.
- Utilize this advantage of processor then data is always inserted as 4 bytes package which will becomes insert empty address spaces in between other existing members address.
- After introducing this structure padding concept in we got to know that size of the structure is not always same.

Syntax:

```
Struct member{
Char character;
Int number;
Double salary;
}
```


Explanation: “Char” data type takes only 1 byte after 3 byte padding(Char, Int and Dou the number will starts at 4 bytes boundary and rest “Int” and “Double” will takes 4 and 8 bytes respectively.

(https://www.educba.com/software-development/)

.

Code:

```
//include basic C library files
#include <stdio.h>
//including string data member in C
#include <string.h>
//creating first structure
struct first_structure
{
int rollNo1, rollNo2;
char firstName;
char character;
float salary;
};
//creating second structure
struct second_structure
{
int rollNo1;
char firstName;
int rollNo2;
char character;
float salary;
};
//main method to run the C application
int main()
```

(https://www.educba.com/software-development/)

```
struct second_structure s2;
//displaying first_structure and second_structure output
printf("==================FIRST
STRUCTURE==============================\n");
printf("size of first_structure in bytes : %d\n",sizeof(s1));
printf ( "\n   Address of rollNo1        = %u",&s1.rollNo1 );
printf ( "\n   Address of rollNo2        = %u",&s1.rollNo2 );
printf ( "\n   Address of firstName       = %u",&s1.firstName );
printf ( "\n   Address of character          = %u",&s1.character);
printf ( "\n   Address of salary = %u",&s1.salary);
printf("\n==================SECOND
STRUCTURE==============================\n");
printf("size of second_structure in bytes : %d\n",sizeof(s2));
printf ( "\n   Address of rollNo1        = %u",&s2.rollNo1 );
printf ( "\n   Address of rollNo2        = %u",&s2.rollNo2 );
printf ( "\n   Address of firstName       = %u",&s2.firstName );
printf ( "\n   Address of character          = %u",&s2.character);
printf ( "\n   Address of salary = %u",&s2.salary);
getchar();
return 0;
}
```

Output:

```
==================FIRST STRUCTURE==============================
size of first_structure in bytes : 16

   Address of rollNo1        = 1698979216
   Address of rollNo2        = 1698979220
```

(https://www.educba.com/software-development/)

```
Address of rollNo2        = 1698979240
Address of firstName       = 1698979236
Address of character          = 1698979244
Address of salary = 1698979248
```

Example #2

Code:

```
//include basic C library files
#include<stdio.h>
//including string data member in C
#include <string.h>
//creating first structure
struct employee
{
char first_name[40];
char last_name[30];
};
//main method to run the C application
int main()
{
//taking first structure reference
struct employee e;
printf("Enter your first name:");
scanf("%s", &e.first_name);
printf("Enter your last name:");
scanf("%s",&e.last_name);
```

(https://www.educba.com/software-development/)

```
printf("size of employee in bytes : %d\n",sizeof(e));
printf ( "\n   Address of first_name        = %u",&e.first_name);
printf ( "\n   Address of last_name        = %u",&e.last_name );
return 0;
}
```

Output:

```
Enter your first name: Paramesh
Enter your last name: Nathi
First Name of Employee is :Paramesh
Last Name of Employee is :Nathi
=================1ST STRUCTURE=======================
size of employee in bytes : 70

   Address of first_name        = 597898080
   Address of last_name        = 597898120
```

Example #3

Overcome Structure padding problem in C

Code:

```
//include basic C library files
#include <stdio.h>
//including string data member in C
#include <string.h>
#pragma pack(1)
//creating first structure
```

(https://www.educba.com/software-development/)

```
char character;
float salary;
};
//creating second structure
struct second_structure
{
int rollNo1;
char firstName;
int rollNo2;
char character;
float salary;
};
//main method to run the C application
int main()
{
//taking first structure reference
struct first_structure s1;
//taking second structure reference
struct second_structure s2;
//displaying first_structure and second_structure output
printf("==================FIRST
STRUCTURE==============================\n");
printf("size of first_structure in bytes : %d\n",sizeof(s1));

printf ( "\n   Address of rollNo1        = %u",&s1.rollNo1 );
printf ( "\n   Address of rollNo2        = %u",&s1.rollNo2 );
printf ( "\n   Address of firstName       = %u",&s1.firstName );
```

(https://www.educba.com/software-development/)

```
printf ( "\n   Address of rollNo1        = %u",&s2.rollNo1 );
printf ( "\n   Address of rollNo2        = %u",&s2.rollNo2 );
printf ( "\n   Address of firstName       = %u",&s2.firstName );
printf ( "\n   Address of character          = %u",&s2.character);
printf ( "\n   Address of salary = %u",&s2.salary);
getchar();
return 0;
}
```

(https://www.educba.com/software-development/)

```
==================FIRST STRUCTURE=============================
size of first_structure in bytes : 14

   Address of rollNo1        = 2700968948
   Address of rollNo2        = 2700968952
   Address of firstName       = 2700968956
   Address of character          = 2700968957
   Address of salary = 2700968958
==================SECOND STRUCTURE=============================
size of second_structure in bytes : 14

   Address of rollNo1        = 2700968962
   Address of rollNo2        = 2700968967
   Address of firstName       = 2700968966
   Address of character          = 2700968971
   Address of salary = 2700968972
```

Conclusion

Structure padding is said to be in order to align the data in memory 1 or more un-occupied bytes (empty bytes) are kept between any memory addresses which are actually assigned for other data structure members at the time of memory allocation.

Recommended Articles

This is a guide to Structure Padding in C. Here we discuss the working of Structure Padding in C along with different examples and code implementation. You can also go through our other related articles to learn more –

1. Tokens in C (https://www.educba.com/tokens-in-c/)
2. Void Pointer in C (https://www.educba.com/void-pointer-in-c/)
3. Power Function in C (https://www.educba.com/power-function-in-c/)
4. Double Pointer in C (https://www.educba.com/double-pointer-in-c/)

(https://www.educba.com/software-development/)

source=footer)

Privacy Policy (https://www.educba.com/privacy-policy/?source=footer)

Apps

iPhone & iPad (https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8)

Android (https://play.google.com/store/apps/details?id=com.educba.www)

Resources

Free Courses (https://www.educba.com/software-development/free-courses/?source=footer)

Java Tutorials (https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer)

Python Tutorials (https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer)

All Tutorials (https://www.educba.com/software-development/software-development-tutorials/?source=footer)

Certification Courses

All Courses (https://www.educba.com/software-development/courses/?source=footer)

Software Development Course - All in One Bundle (https://www.educba.com/software-development/courses/software-development-course/?source=footer)

Become a Python Developer (https://www.educba.com/software-development/courses/python-certification-course/?source=footer)

Java Course (https://www.educba.com/software-development/courses/java-course/?source=footer)

(https://www.educba.com/software-development/)

VB.NET Course (https://www.educba.com/software-development/courses/vb-net-course/?source=footer)

PHP Course (https://www.educba.com/software-development/courses/php-course/?source=footer)