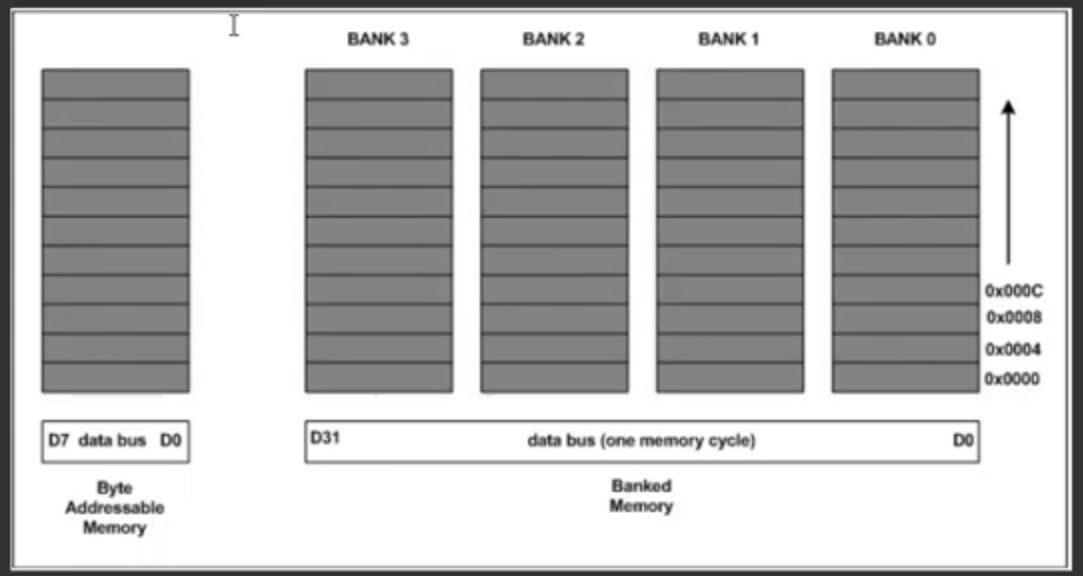
Assumptions

1. System or machine: (32 bit machine)
2. char (1 byte)
3. short int (2 bytes)
4. int (4 bytes)
5. double (8 bytes)

GEEKSFORGEEKS

Structure member alignment and padding

Memory architecture:



GEEKSFORGEEKS

Some Rules

1. Any data type will always store its value at the address which is multiple of the size of that data type.

Example: int will always start from 4,8,12,.....

2. any structure will take the size equal to the multiple of maximum bytes taken by a variable in that structure.

GEEKSFORGEEKS

Sequential memory structure

```
typedef struct structa_tag
{
   char        c;
   short int   s;
} structa_t;
```

Bytes	0	1	2	3	4	5	6	7	8
Allocation	char		Short int	Short int					

Total size: 4bytes

Rule1 -Any data type will always store its value at the address, which is multiple of the size of that data type

Rule2 - **structure will take the size equal to the multiple of maximum bytes taken by a variable in that structure.**

GEEKSFORGEEKS

Sequential memory structure

```
typedef struct structa_tag
{
   char        c;
   short int   s;
} structa_t;
```

Bytes	0	1	2	3	4	5	6	7	8
Allocation	char		Short int	Short int					

Total size: 4bytes

Rule1 -Any data type will always store its value at the address, which is multiple of the size of that data type

Rule2 - **structure will take the size equal to the multiple of maximum bytes taken by a variable in that structure.**

GEEKSFORGEEKS

3

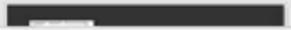
4

5

6

7

8



```
typedef struct structb_tag
{
   short int   s;
   char        c;
   int         i;
} structb_t;
```

Bytes	0	1	2	3	4	5	6	7
Allocation	Short int	Short int	char		int	int	int	int

Total size: 8bytes

Rule1 -Any data type will always store its value at the address, which is multiple of the size of that data type

Rule2 - **structure will take the size equal to the multiple of maximum bytes taken by a variable in that structure.**

GEEKSFORGEEKS

```
typedef struct structb_tag
{
   short int   s;
   char        c;
   int         i;
} structb_t;
```

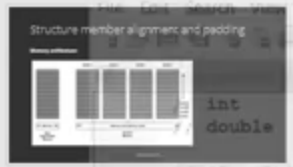
Bytes	0	1	2	3	4	5	6	7
Allocation	Short int	Short int	char		int	int	int	int

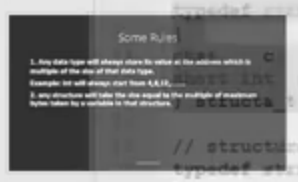
Total size: 8bytes

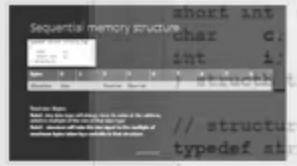
Rule1 -Any data type will always store its value at the address, which is multiple of the size of that data type

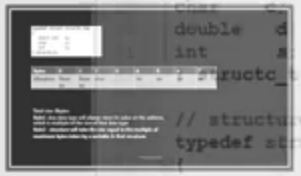
Rule2 - **structure will take the size equal to the multiple of maximum bytes taken by a variable in that structure.**

GEEKSFORGEEKS









```
typedef struct structb_tag
{
   short int   s;
   char        c;
   int         i;
} structb_t;
```

Bytes	0	1	2	3	4	5	6	7
Allocation	Short int	Short int	char		int	int	int	int

Total size: 8bytes

Rule1 -Any data type will always store its value at the address, which is multiple of the size of that data type

Rule2 - **structure will take the size equal to the multiple of maximum bytes taken by a variable in that structure.**

GEEKSFORGEEKS

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

programming questions.txt | ramdump.py | gpu_info_script.py | new 1 | basic.cpp

```
// int      4 bytes
// double   8 bytes

// structure A
typedef struct structa_tag
{
char    c;
short int  s;
} structa_t;

// structure B
typedef struct structb_tag
{
short int  s;
char    c;
int     i;
} structb_t;

// structure C
typedef struct structc_tag
{
char    c;
double  d;
int     s;
} structc_t;

// structure D
typedef struct structd_tag
{
double  d;
int     s;
char    c;
} structd_t;

int main()
{
printf("sizeof(structa_t) = %d\n", sizeof(structa_t));
//printf("sizeof(structb_t) = %d\n", sizeof(structb_t));
//printf("sizeof(structc_t) = %d\n", sizeof(structc_t));
//printf("sizeof(structd_t) = %d\n", sizeof(structd_t));

return 0;
}
```

```
typedef struct structb_tag
{
   short int   s;
   char        c;
   int         i;
} structb_t;
```

Bytes	0	1	2	3	4	5	6	7
Allocation	Short int	Short int	char		int	int	int	int

Total size: 8bytes

Rule1 -Any data type will always store its value at the address, which is multiple of the size of that data type

Rule2 - **structure will take the size equal to the multiple of maximum bytes taken by a variable in that structure.**

GEEKSFORGEEKS

C++ source file length : 858 lines : 51 Ln : 12 Col : 1 Sel : 74 | 5 Windows (CR LF) UTF-8 INS

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

programming questions.txt | ramdump.py | gpu_info_script.py | new 1 | basic.cpp

```
// int     4 bytes
// double    8 bytes

// structure A
typedef struct structa_tag
{
char    c;
short int  s;
} structa_t;

// structure B
typedef struct structb_tag
{
short int  s;
char    c;
int     i;
} structb_t;

// structure C
typedef struct structc_tag
{
char    c;
double   d;
int      s;
} structc_t;

// structure D
typedef struct structd_tag
{
double   d;
int      s;
char    c;
} structd_t;

int main()
{
printf("sizeof(structa_t) = %d\n", sizeof(structa_t));
//printf("sizeof(structb_t) = %d\n", sizeof(structb_t));
//printf("sizeof(structc_t) = %d\n", sizeof(structc_t));
//printf("sizeof(structd_t) = %d\n", sizeof(structd_t));

return 0;
}
```

C++ source file | length : 858 lines : 51 | Ln : 12 Col : 1 Sel : 74 | 5 | Windows (CR LF) | UTF-8 | INS

6:25 PM
5/23/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

programming questions.txt | ramdump.py | gpu_info_script.py | new 1 | basic.cpp

```
// int     4 bytes
// double    8 bytes

// structure A
typedef struct structa_tag
{
char    c;
short   s;
} structa_t;

// structure B
typedef struct structb_tag
{
short int s;
char    c;
int     i;
} structb_t;

// structure C
typedef struct structc_tag
{
char    c;
double  d;
int     s;
} structc_t;

// structure D
typedef struct structd_tag
{
double  d;
int     s;
char    c;
} structd_t;

int main()
{
printf("sizeof(structa_t) = %d\n", sizeof(structa_t));
printf("sizeof(structb_t) = %d\n", sizeof(structb_t));
//printf("sizeof(structc_t) = %d\n", sizeof(structc_t));
//printf("sizeof(structd_t) = %d\n", sizeof(structd_t));

return 0;
}
```

C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.16299.1146]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\vgulia>cd Documents

C:\Users\vgulia\Documents>cd personal_documents\C_Pointers

C:\Users\vgulia\Documents\personal_documents\C_Pointers>ls
 'New Text Document.txt'   basic.c     basic.exe    basic1.exe   basic2.cpp   basic3.cpp   basic4.exe   theory.docx
  a.exe                    basic.cpp   basic1.cpp   basic1.o     basic2.exe   basic3.exe   test.p

C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic.cpp

C:\Users\vgulia\Documents\personal_documents\C_Pointers>ls
 'New Text Document.txt'   basic.c     basic.exe    basic1.exe   basic2.cpp   basic3.cpp   basic4.exe   theory.docx
  a.exe                    basic.cpp   basic1.cpp   basic1.o     basic2.exe   basic3.exe   test.p

C:\Users\vgulia\Documents\personal_documents\C_Pointers>a.exe
sizeof(structa_t) = 4
sizeof(structb_t) = 8

C:\Users\vgulia\Documents\personal_documents\C_Pointers>
```

C++ source file | length : 856 | lines : 51 | Ln : 45 | Col : 1 | Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

6:25 PM 5/23/2019

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.16299.1146]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\vgulia>cd Documents

C:\Users\vgulia\Documents>cd personal_documents\C_Pointers

C:\Users\vgulia\Documents\personal_documents\C_Pointers>ls
'New Text Document.txt'   basic.c     basic.exe    basic1.exe   basic2.cpp   basic3.cpp   basic4.exe   theory.docx
 a.exe                    basic.cpp   basic1.cpp   basic1.o     basic2.exe   basic3.exe   test.p

C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic.cpp

C:\Users\vgulia\Documents\personal_documents\C_Pointers>ls
'New Text Document.txt'   basic.c     basic.exe    basic1.exe   basic2.cpp   basic3.cpp   basic4.exe   theory.docx
 a.exe                    basic.cpp   basic1.cpp   basic1.o     basic2.exe   basic3.exe   test.p

C:\Users\vgulia\Documents\personal_documents\C_Pointers>a.exe
sizeof(structa_t) = 4

C:\Users\vgulia\Documents\personal_documents\C_Pointers>
```

C:\Users\vgulia\Documents\personal_documents\C_Pointers\basic.cpp - Notepad++

File

C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.16299.1146]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\vgulia>cd Documents

C:\Users\vgulia\Documents>cd personal_documents\C_Pointers

C:\Users\vgulia\Documents\personal_documents\C_Pointers>ls
'New Text Document.txt'   basic.c     basic.exe    basic1.exe   basic2.cpp   basic3.cpp   basic4.exe   theory.docx
 a.exe                    basic.cpp   basic1.cpp   basic1.o     basic2.exe   basic3.exe   test.p

C:\Users\vgulia\Documents\personal_documents\C_Pointers>g++ basic.cpp

C:\Users\vgulia\Documents\personal_documents\C_Pointers>ls
'New Text Document.txt'   basic.c     basic.exe    basic1.exe   basic2.cpp   basic3.cpp   basic4.exe   theory.docx
 a.exe                    basic.cpp   basic1.cpp   basic1.o     basic2.exe   basic3.exe   test.p

C:\Users\vgulia\Documents\personal_documents\C_Pointers>a.exe
sizeof(structa_t) = 4

C:\Users\vgulia\Documents\personal_documents\C_Pointers>
```

C++ s

6:25 PM
5/23/2019


```
typedef struct structb_tag
{
   short int   s;
   char        c;
   int         i;
} structb_t;
```

Bytes	0	1	2	3	4	5	6	7
Allocation	Short int	Short int	char		int	int	int	int

Total size: 8bytes

Rule1 -Any data type will always store its value at the address, which is multiple of the size of that data type

Rule2 - **structure will take the size equal to the multiple of maximum bytes taken by a variable in that structure.**

GEEKSFORGEEKS