memory\_structure.md 26/11/2022

## Memory structure

```
1. Check the following simple C program
  #include <stdio.h>
     int main(void)
  \triangleright
          return 0;
  C
  [narendra@CentOS]$ gcc memory-layout.c -o memory-layout
  [narendra@CentOS]$ size memory-layout
  text
              data
                           bss
                                      dec
                                                  hex
                                                          filename
  960
              248
                            8
                                     1216
                                                  4c0
                                                         memory-layout
2. Let us add one global variable in the program, now check the size of bss (highlighted in red color).
 #include <stdio.h>
  int global; /* Uninitialized variable stored in bss*/
     int main(void)
 C
  [narendra@CentOS]$ gcc memory-layout.c -o memory-layout
  [narendra@CentOS]$ size memory-layout
  text
              data
                          bss
                                                  hex
                                                          filename
   960
               248
                            12
                                     1220
                                                  4c4
                                                          memory-layout
  3. Let us add one static variable which is also stored in bss.
    #include <stdio.h>
    int global; /* Uninitialized variable stored in bss*/
    int main(void)
     [narendra@CentOS]$ gcc memory-layout.c -o memory-layout
     [narendra@CentOS]$ size memory-layout
     text
                data
                             bss
                                         dec
                                                             filename
                                                     hex
      960
                  248
                              16
                                        1224
                                                     4c8
                                                             memory-layout
```

memory\_structure.md 26/11/2022

```
4. Let us initialize the static variable which will then be stored in the Data Segment (DS)
 С
 #include <stdio.h>
 \mathcal O int global; /* Uninitialized variable stored in bss*/
 int main(void)
         static int i = 100; /* Initialized static variable stored in DS*/
         return 0;
  [narendra@CentOS]$ gcc memory-layout.c -o memory-layout
  [narendra@CentOS]$ size memory-layout
                                                           filename
  text
              data
                           bss
                                       dec
                                                   hex
  960
               252
                            12
                                      1224
                                                   4c8
                                                           memory-layout
5. Let us initialize the global variable which will then be stored in the Data Segment (DS)
 С
  #include <stdio.h>
  int global = 10; /* initialized global variable stored in DS*/
  int main(void)
  C
          static int i = 100; /* Initialized static variable stored in DS*/
  [narendra@CentOS]$ gcc memory-layout.c -o memory-layout
  [narendra@CentOS]$ size memory-layout
  text
              data
                           bss
                                       dec
                                                    hex
                                                            filename
  960
               256
                              8
                                       1224
                                                    4c8
                                                           memory-layout
```