Basic C concepts and C memory layout

- Static (local/global) (initialized/uninitialized)
- Global variable(initialized/uninitialized)
- Const (local/global) (initialized/uninitialized)
- Static const (local/global) (initialized/uninitialized)

Command used in this exercise

- $ objdump -S a.out
- $ size --format=SysV a.out | (grep -e .text -e .data -e .bss)
- $size -o a.out
- $readelf -h a.out

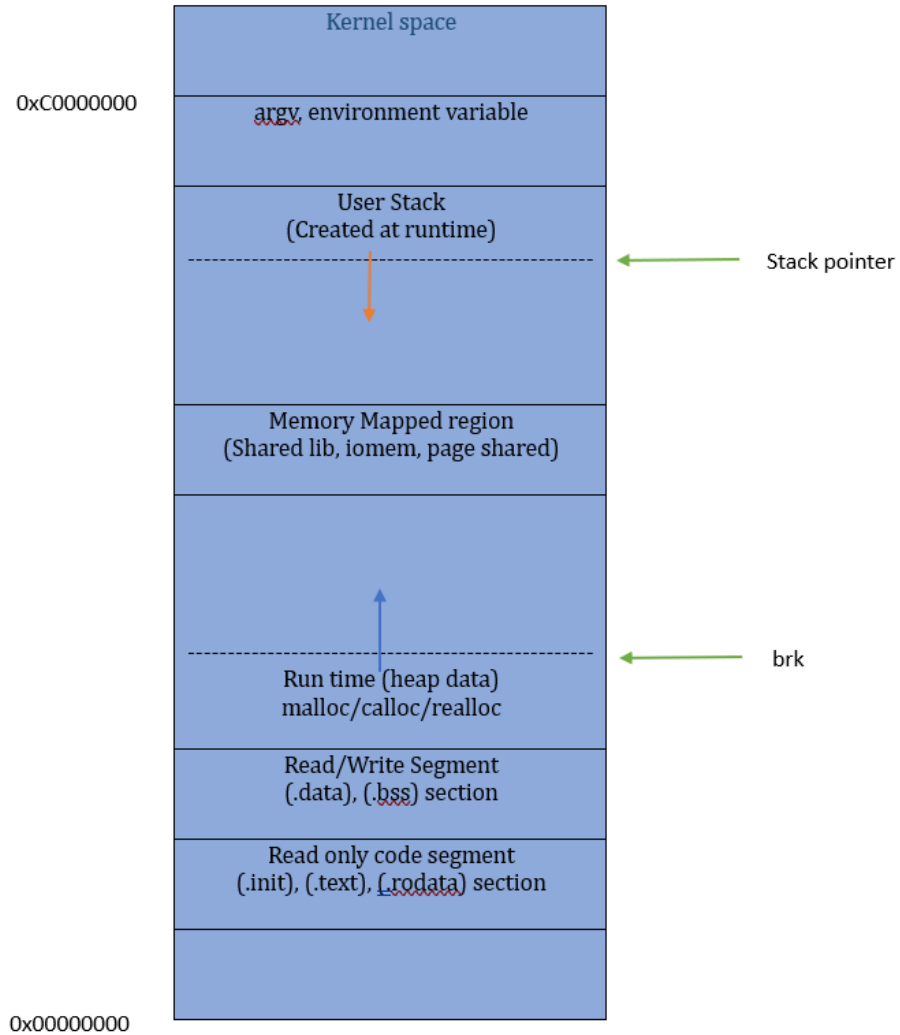
GitHub link for Code

https://github.com/krishneshpathak/c-basic-concept/tree/main

Code:

```c
#include <stdio.h>

 // int g_var;                                  /* .bss section */

 // int g_var = 0;                              /* .bss section */

 // int g_var = 7;                              /* .data section */

 //static  int g_var;                           /* .bss section */

 // static int g_var = 0;                       /* .bss section */

 //static int g_var = 7;                        /* .data section */

 //const int  g_var;                            /* .bss section */
 //const int  g_var = 0;                        /* cannot modify (.rodata section) */
 //const int  g_var = 7;                        /* cannot modify (.rodata section) */

 //static const int  g_var;                     /* cannot modify (.rodata section) */
 static const int  g_var = 0;                   /* cannot modify (.rodata section) */
 //static const int  g_var = 7;                 /* cannot modify (.rodata section) */

 int main(void)
 {
     //static const int const l_var;            /* cannot modify */
     //static const int const l_var = 0;        /* cannot modify */
     //static const int const l_var = 10;       /* cannot modify */

     const int  l_var=10;                       /* modify by pointer (stack section) */
     printf("variable = (%d)\n", g_var);
     printf("variable = (%d)\n", l_var);

     return 0;
 }
```

C memory Layout



Virtual Memory Layout