**PROCESS MEMORY MAP**

**1. What are the various segments of memory created for a process?**

**A:** Text Segment

Data Segment

Heap

Stack

Memory-mapped segment

**2. Refer the code and identify which variable goes in which segment? (Analyse line by line)**

**#include <stdio.h>**

**int glob\_array[2000];**

**void func()**

**{**

**static int count = 0;**

**int func\_local=10;**

**char \*ptr; /\* Assume pointer size is 4 or 8 bytes. Which segment do these belong to? \*/**

**char array[1000];**

**ptr = malloc(100); /\* From which segment, do these 100 bytes come? \*/**

**count++;**

**printf("func called %d times\n", count);**

**}**

**int main() /\* Which segment do the compiled machine instructions of source code go into? \*/**

**{**

**int main\_local;**

**func();**

**return 0;**

**}**

**A:**

**glob\_array[2000]**: Goes into the **Data Segment** (global variable, initialized to zero).

**static int count = 0;**: Goes into the **Data Segment** (static variable, retains value across function calls).

**int func\_local = 10;**: Goes into the **Stack** (local variable in the function).

**char \*ptr;**: Goes into the **Stack** (local pointer variable).

**char array[1000];**: Goes into the **Stack** (local array).

**ptr = malloc(100);**: Allocates memory in the **Heap** (dynamic memory).

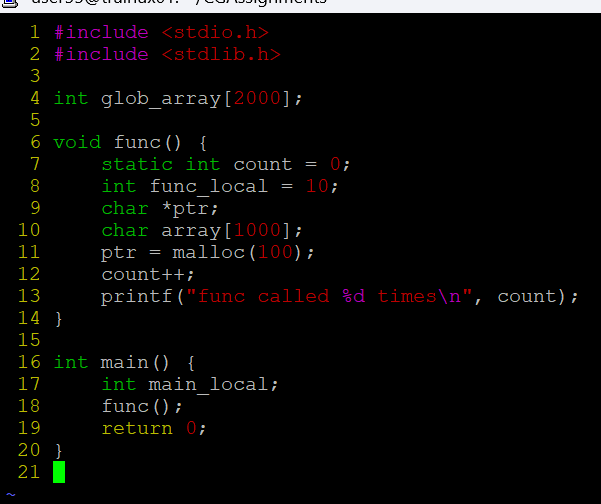
**count++;**: Refers to a **Data Segment** variable (static, retains value).

**int main\_local;**: Goes into the **Stack** (local variable in main()).

**3. How many functions would be in stack when func() is being executed**

**A:** When func() is being executed, there would be two functions on the stack: one for func() and one for main().

**4. Create an executable of above code, run size command and view the size**

****

**OUTPUT:**

****