Program 11-Memory Layout of a C Program

Submitted by,

Ambily Francis

1947204

CODE:

/\*Lab11 : write a c program to print out the memory locations of BSS, Stack, Variable on stack, Heap and etc

Name:Ambily Francis

Reg No:1947204

Date:1/9/2020

\*/

//header files

#include <stdio.h>

#include <stdlib.h>

//global variables //All global variable gets stored in the data segment

const float pi=3.14; //It will be stored in read only block of initialized data segment

int f\_no=3; //Initialized data segment

int data1; // Uninitialized global variable stored in BSS

int factorial\_fun(int num) //Arguments passed to the function used to store in stack segment

{

static int fact=1; //Initialized local static variable will be stored in the initialized data segment

int i; //local variable of function will be stored in stack segment

static int data2; // Uninitialized static variable stored in BSS

if (num < 0)

{

printf("OOPS..!Factorial is only existing "); //Code segment

}

else

{

for (i = 1; i <= num; ++i)

{

fact \*= i;

}

}

return fact; //Return addresses are always stored in the stack segment

}

int main()

{

char \*ptr = malloc(sizeof(char)\*4); //heap will manage memory management functions like malloc, calloc and free

int num = 5; //local variable of function will be stored in stack segment

int result; //local variable of function will be stored in stack segment

result=factorial\_fun(num); //calling function

printf("\nNAME:AMBILY FRANCIS \nREGNO:1947204\n\n");

printf("\n---Address of variables in Stack--- \n");

printf("\nnum : %p", &num);

printf("\nresult : %p\n", &result);

printf("\n---Address of variables in heap--- ");

printf("\nptr : %p\n", &ptr);

printf("\n---Address of Uninitialized global variable stored in BSS--- ");

printf("\ndata1 :%p\n",&data1);

printf("\n---Address of variables in initialized data segments---");

printf("\npi :%p",&pi);

printf("\nf\_no :%p\n",&f\_no);

return 0;

}

OUTPUT:

