**Digital assignment-2**

**Name:tej sai reddy**

**Regno:17mis1021**

**Gdb debugging:**

Code:

#include <stdio.h>

//function for factorial

long int factorial(int n)

{

if(n==1) return 1;

return n\*factorial(n-1);

}

int main()

{

int num;

long int fact=0;

printf("Enter an integer number: ");

scanf("%d",&num);

fact=factorial(num);

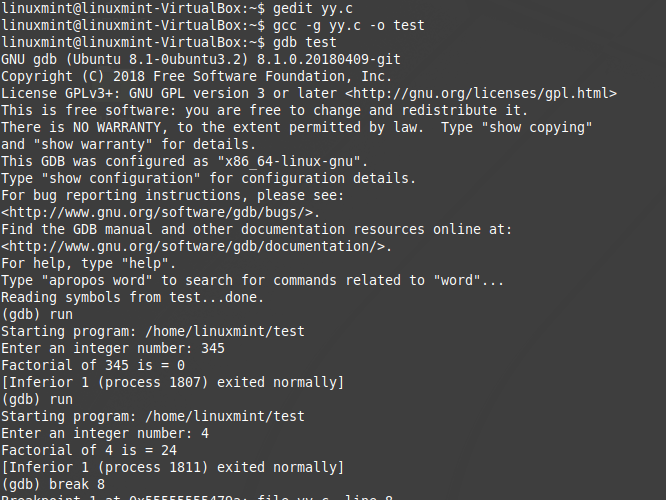
printf("Factorial of %d is = %ld",num,fact);

printf("\n");

return 0;

}

Output:

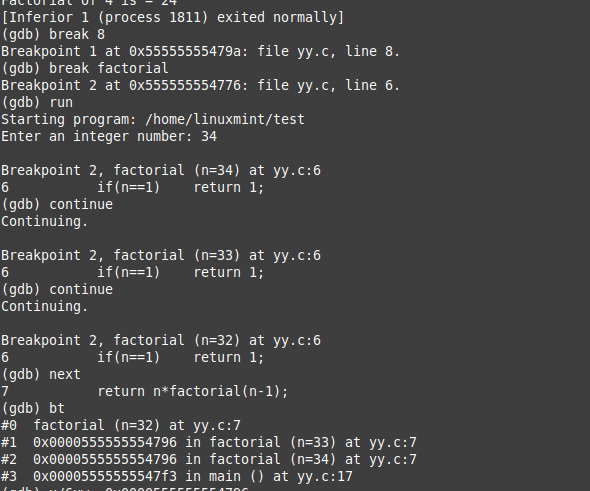


The break command is used to create a pause in the program

The continue command is used to resume the program code .

The next command is used for the execution of the next instruction

The backtrace command is used for frames created created in the program.



**Memory debugging:**

Syntax: x/nfu

Where n is number of lines

F is format

U is byte/word

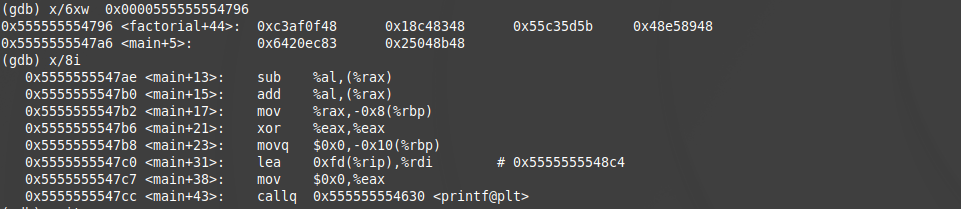
the values for u can be in unsigned,hexadecimal,gaintwords,bytes.

The values for format are binary,octal,unsigned,string,float,unsigned etc.

Example:

X/6xw is used to create 6 hexadecimal words.

And x/8i is to create 8 machine instructions.



**Valgrind**:

checks for the memory leakage problems

**CODE:**

#include <stdio.h>

#include <stdlib.h>

int main()

{

int \*ptr, \*ptr1;

int n, i;

n = 5;

printf("Enter number of elements: %d\n", n);

ptr = (int\*)malloc(n \* sizeof(int));

if (ptr == NULL) {

printf("Memory not allocated.\n");

exit(0);

}

else {

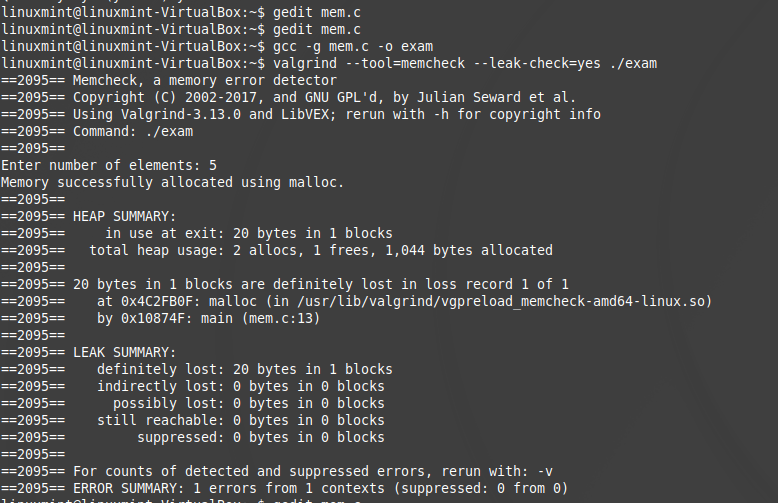
printf("Memory successfully allocated using malloc.\n");

}

return 0;

}

OUTPUT:



There is one error because the initialized pointer is not free after used.

Example:2

Code:

#include <stdio.h>

#include <stdlib.h>

int main(void)

{

char \*p = malloc(1);

\*p = 'a';

char c = \*p;

printf("\n [%c]\n",c);

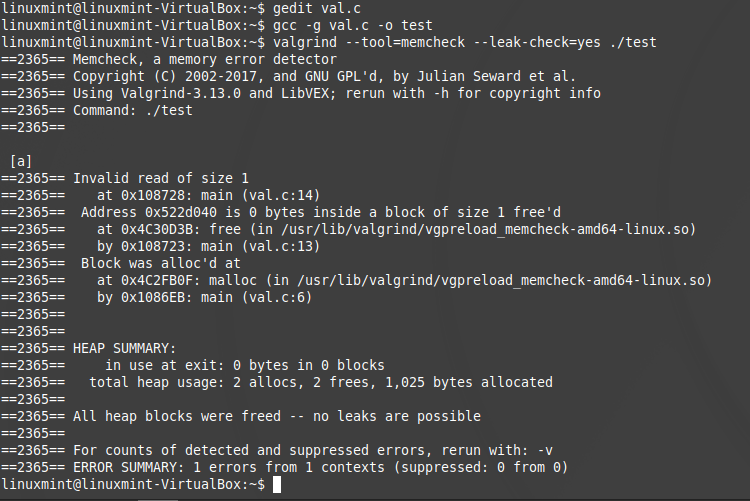
free(p);

c = \*p;

return 0;

}

Output:



The error summary is provided with 1 error because of calling the pointer after free function of that pointer.

**Strace for programs and system calls for tracing:**

**Code:**

#include<stdio.h>

#include<fcntl.h>

#include<errno.h>

#include<stdlib.h>

extern int errno;

int main()

{

int fd = open("tej.txt", O\_RDWR );

printf("fd = %d/n", fd);

if (fd ==-1)

{

printf("Error Number % d\n", errno);

perror("Program");

}

int sz;

sz = write(fd, "Hello world\n", strlen("Hello world\n"));

if (sz == -1 && errno != EINTR) {

perror("Write to output file");

exit(EXIT\_FAILURE);

}

return 0;

}

