**Module.2**

Use Case 2: Debugging Null Pointer Dereference with Reverse Debugging

Overview

A null pointer dereference occurs when a program tries to access memory using a null pointer, causing a segmentation fault. Reverse debugging helps identify where the pointer was assigned a null value.  
  
**To find null pointer dereference which causing segmentation fault.**  
segmentation\_fault\_rd.c  
  
#include <stdio.h>

#include <stdlib.h>

int main() {

    int \*array = NULL;

    int n = 5;

    // Logical error: Conditional  that skips memory allocation

    if(n < 0){

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

    }

    // Attempting to dereference a NULL pointer

    array[0] = 10; // Causes segmentation fault

    printf("Value at index 0: %d\n", array[0]);

    free(array); //Free allocated memory

    return 0;

}

**Steps to debug with gdb to find segmentation fault:-**

gcc -g -o segmentation\_fault\_rd segmentation\_fault\_rd.c

gdb ./ segmentation\_fault\_rd

break main

target record-full

run

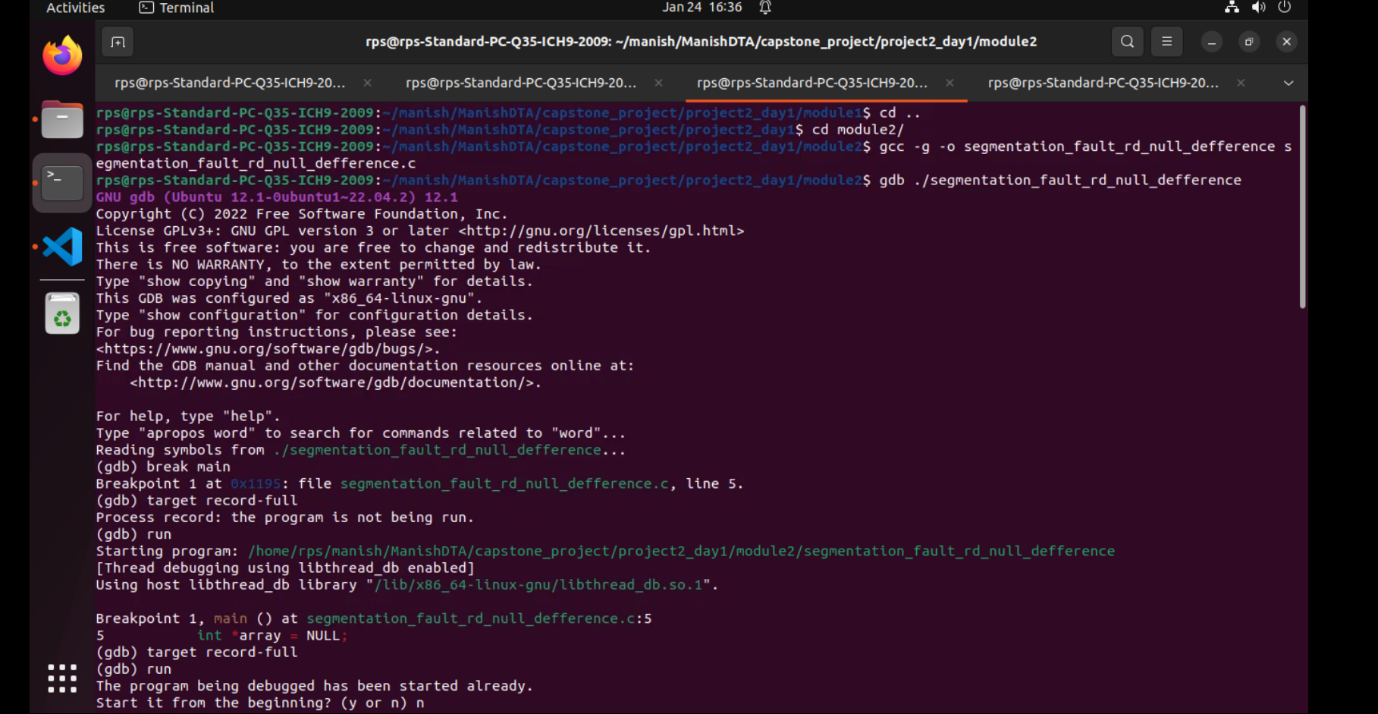
next

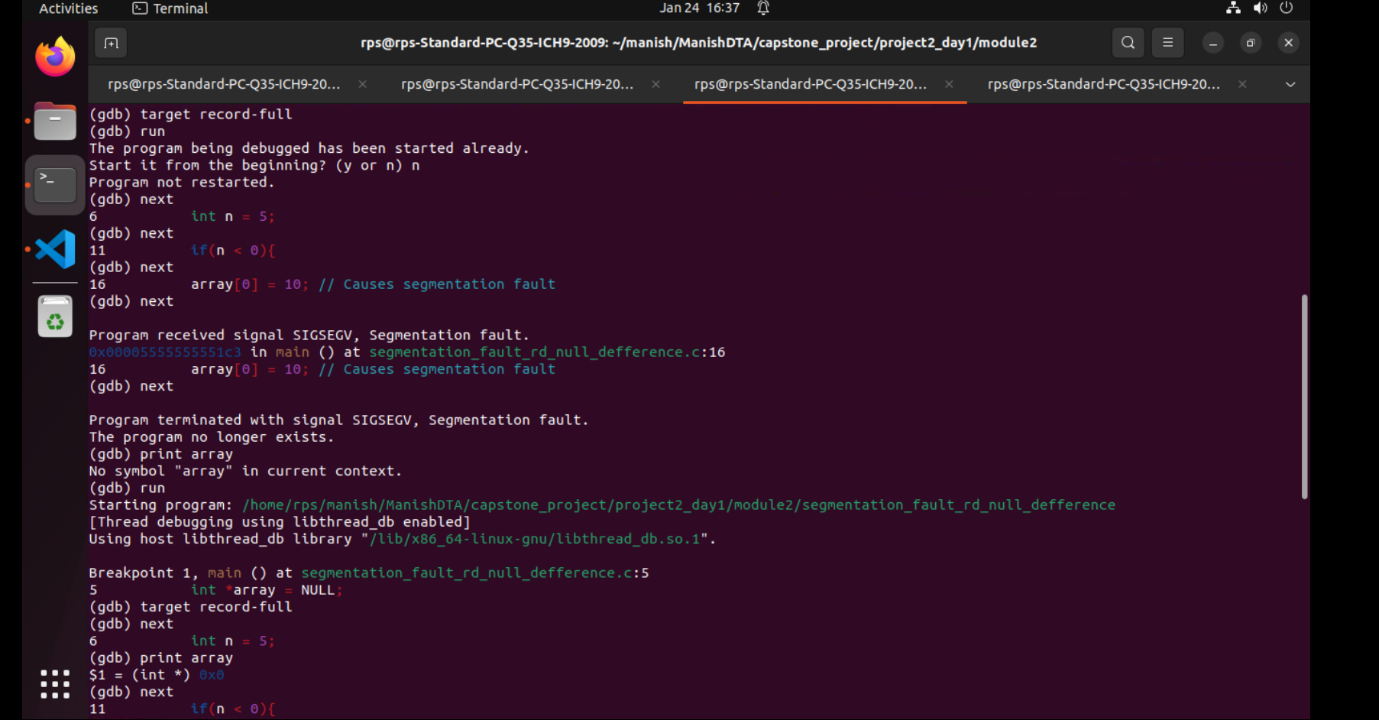
print array

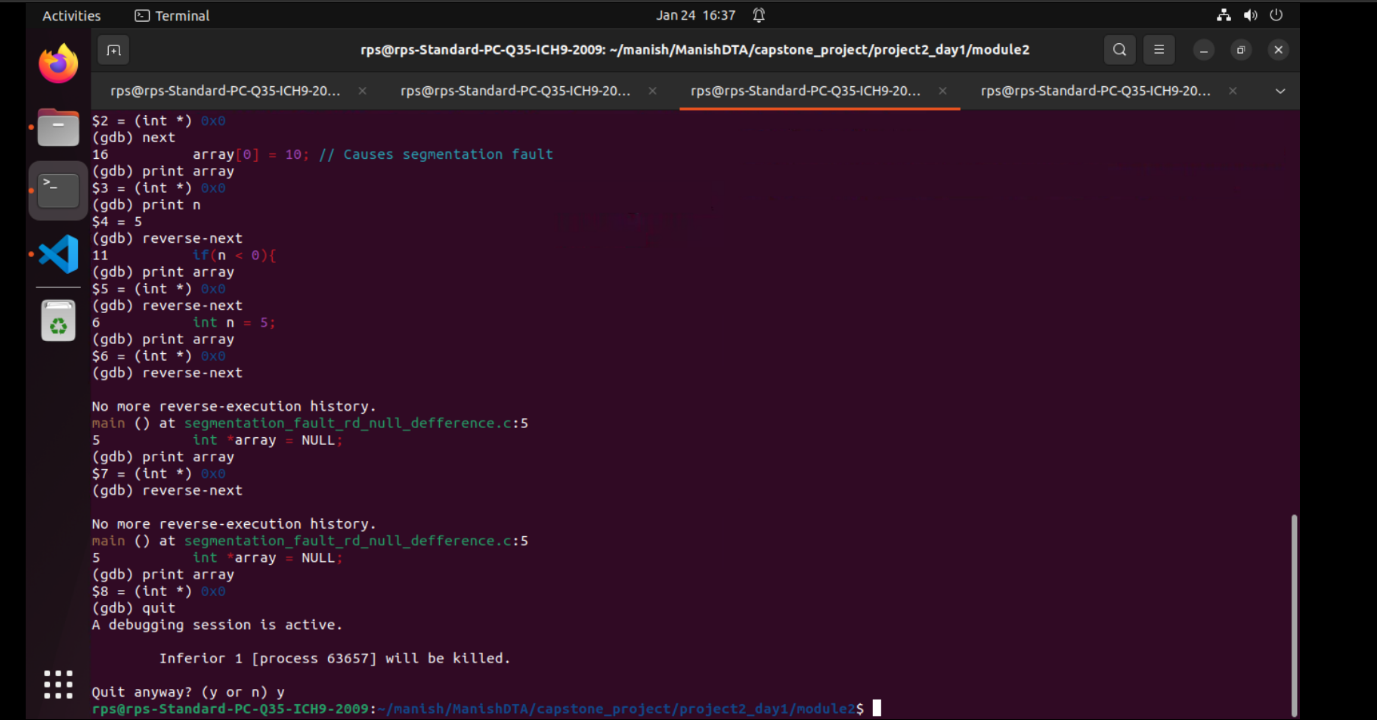
print n

reverse-next

print array

****

****

****

**Fixed segmentation fault which causing segmentation fault.**fixed\_segmentation\_fault\_rd.c

#include <stdio.h>

#include <stdlib.h>

int main() {

    int \*array = NULL;

    int n = 5;

    // Fixed Conditional to ensure proper memory allocation

    if(n > 0){

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

    }

    else{

        printf("Invalid size.\n");

        return -1;

    }

    // Now safe to dereference the pointer

    array[0] = 10; // Causes segmentation fault

    printf("Value at index 0: %d\n", array[0]);

    free(array); //Free allocated memory

    return 0;

}

**Steps to debug with gdb to find segmentation fault fixed or not:-**

gcc -g -o fixed\_segmentation\_fault\_rd fixed\_segmentation\_fault\_rd.c

gdb ./ fixed\_segmentation\_fault\_rd.c

break main

target record-full

run

next

print array

print n

next

print array[0]

reverse-next

print array

continue

gcc -g -o fixed\_segmentation\_fault\_rd.c

./ fixed\_segmentation\_fault\_rd  
  
