Module.5

5. Recursive Function Debugging

Description: Reverse debugging helps analyze incorrect recursion logic.

Debugging Tasks:

1. Compile:

2. Use GDB:

Start with record.

Step forward until the stack overflow occurs.

Use reverse-step to identify why the base case is never met.

3. Fix the recursion logic.

To find the Recursive Error in GDB by reverse debugging:----

error_recursive_logic.c

```
#include <stdio.h>

void faultyRecursion(int n) {
    printf("Recursion level: %d\n", n);
    faultyRecursion(n + 1);  // Incorrect base case, no termination condition
}

int main() {
    faultyRecursion(1);  // Start recursion
    return 0;
}
```

Steps to Debug the Recursive Error in GDB

gcc -g -o error_recursive_logic error_recursive_logic.c

gdb ./error_recursive_logic

record

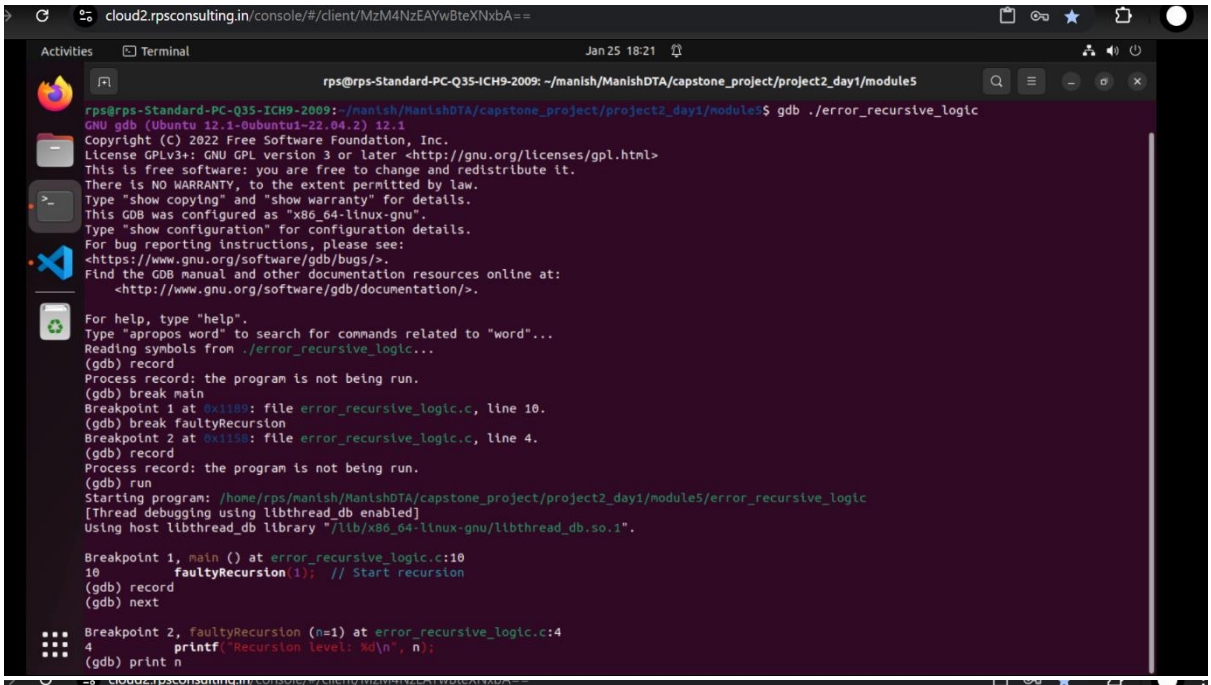
run

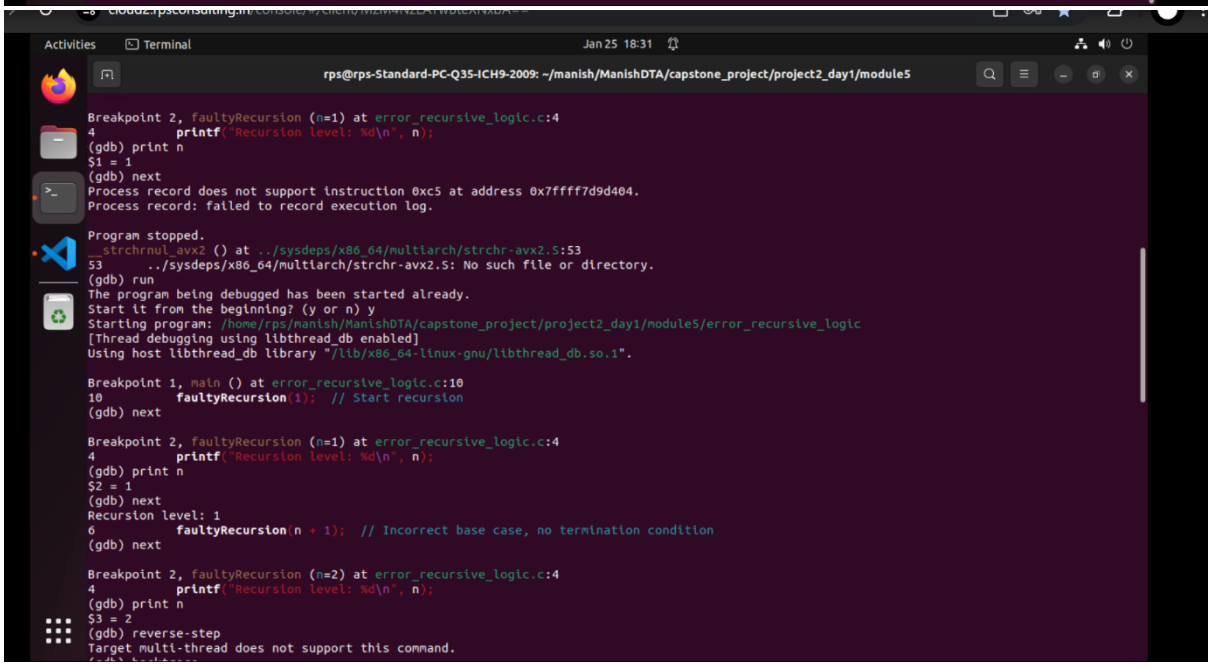
next

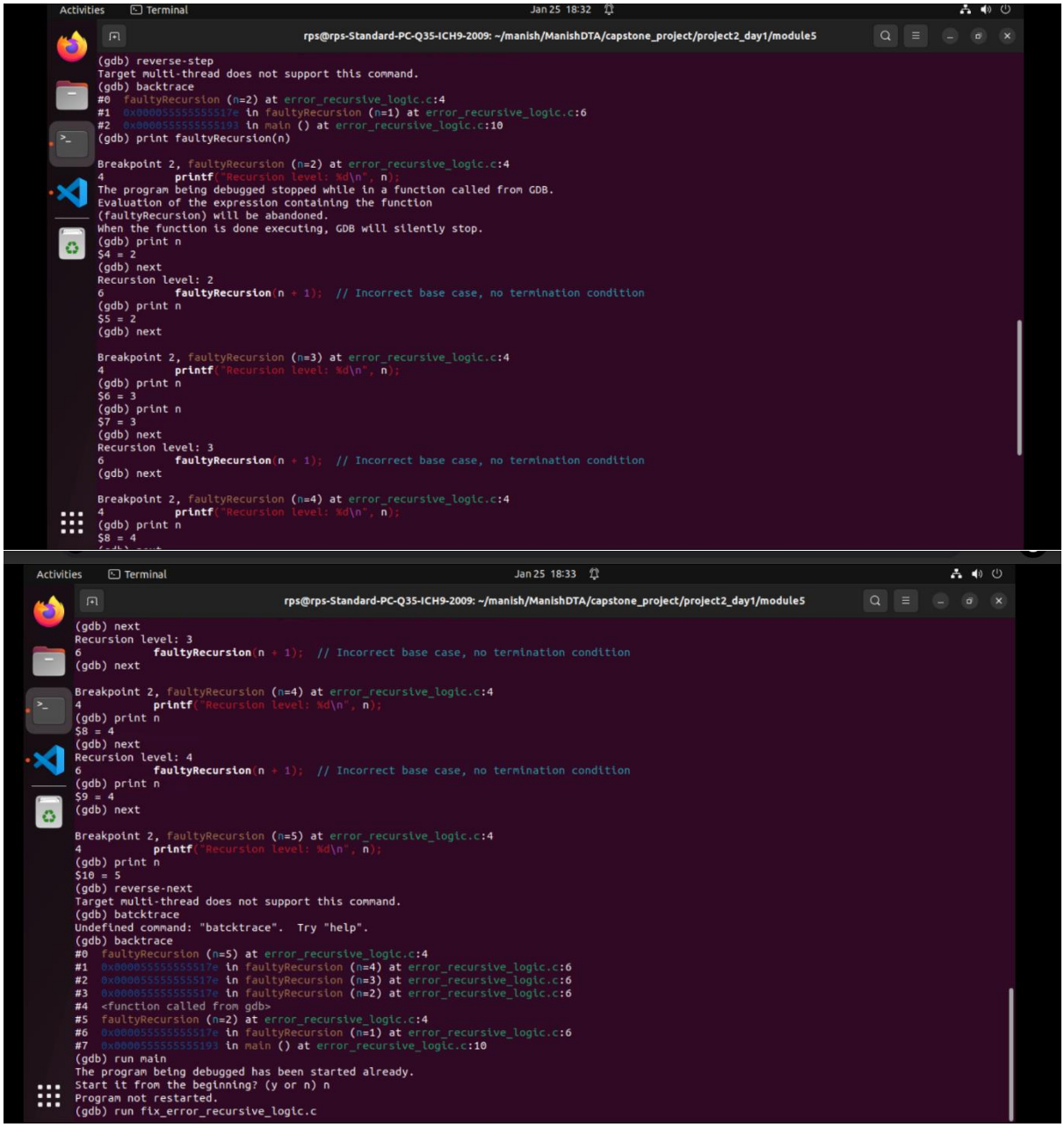
print n

reverse-step

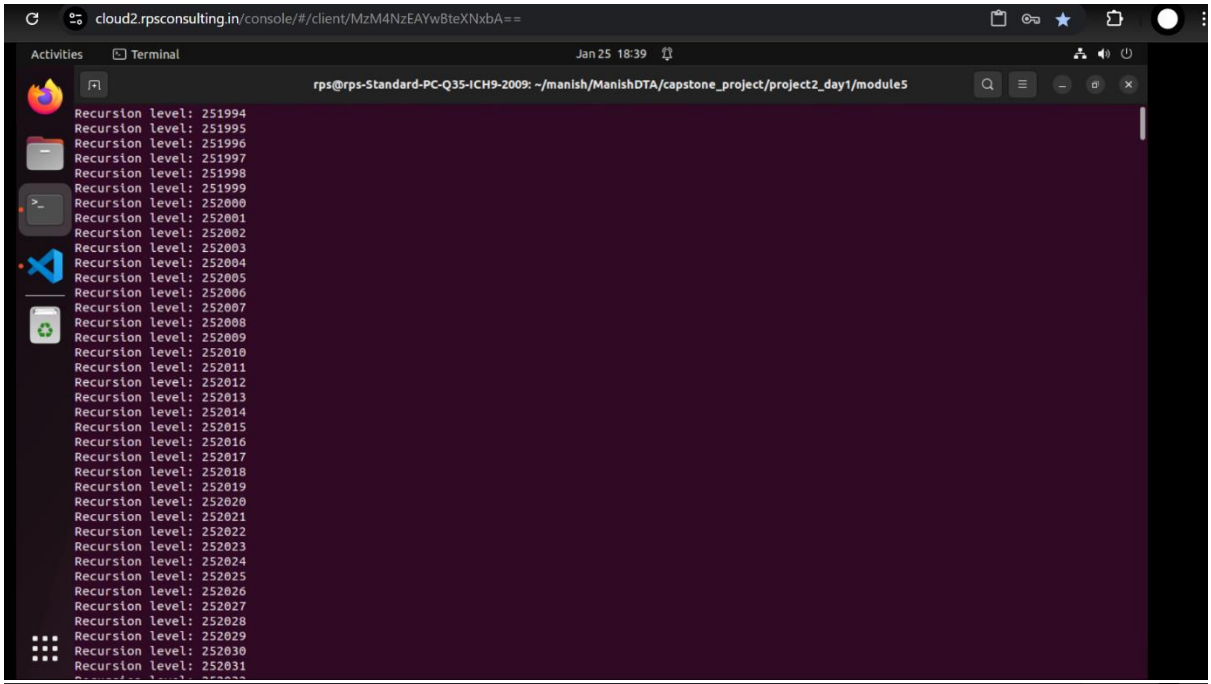
backtrace

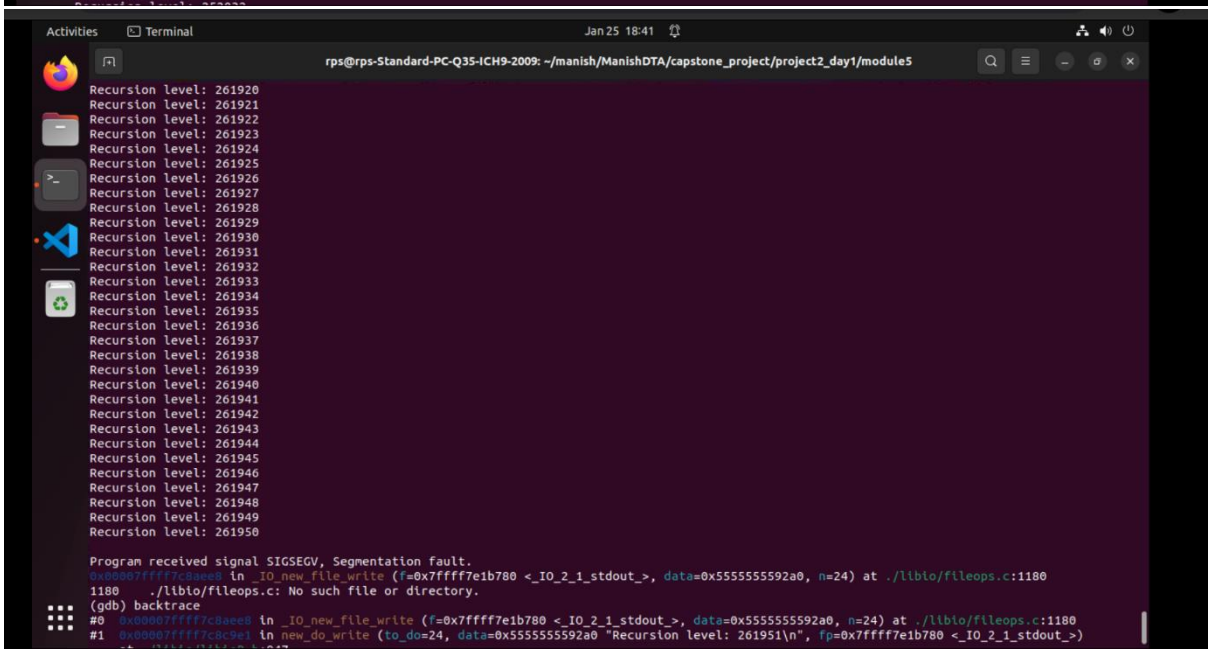


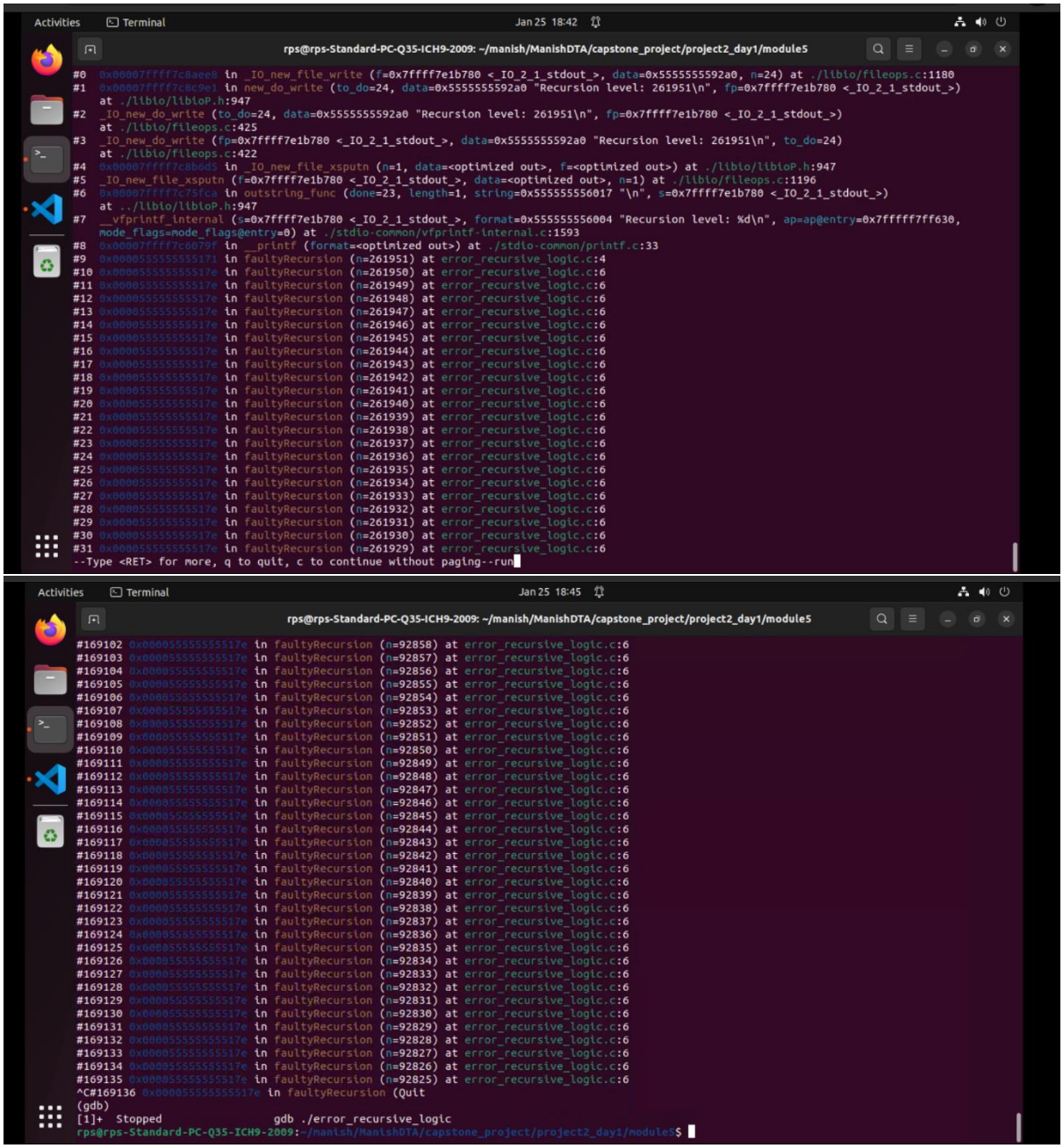



Activities
Terminal
Jan 25 18:32
rps@rps-Standard-PC-Q35-ICH9-2009: ~/manish/ManishDTA/capstone_project/project2_day1/module5
(gdb) reverse-step
Target multi-thread does not support this command.
(gdb) backtrace
#0 faultyRecursion (n=2) at error_recursive_logic.c:4
#1 0x000055555555517e in faultyRecursion (n=1) at error_recursive_logic.c:6
#2 0x0000555555555193 in main () at error_recursive_logic.c:10
(gdb) print faultyRecursion(n)
Breakpoint 2, faultyRecursion (n=2) at error_recursive_logic.c:4
4 printf("Recursion level: %d\n", n);
The program being debugged stopped while in a function called from GDB.
Evaluation of the expression containing the function
(faultyRecursion) will be abandoned.
When the function is done executing, GDB will silently stop.
(gdb) print n
$4 = 2
(gdb) next
Recursion level: 2
6 faultyRecursion(n + 1); // Incorrect base case, no termination condition
(gdb) print n
$5 = 2
(gdb) next
Breakpoint 2, faultyRecursion (n=3) at error_recursive_logic.c:4
4 printf("Recursion level: %d\n", n);
(gdb) print n
$6 = 3
(gdb) print n
$7 = 3
(gdb) next
Recursion level: 3
6 faultyRecursion(n + 1); // Incorrect base case, no termination condition
(gdb) next
Breakpoint 2, faultyRecursion (n=4) at error_recursive_logic.c:4
4 printf("Recursion level: %d\n", n);
(gdb) print n
$8 = 4

Activities
Terminal
Jan 25 18:33
rps@rps-Standard-PC-Q35-ICH9-2009: ~/manish/ManishDTA/capstone_project/project2_day1/module5
(gdb) next
Recursion level: 3
6 faultyRecursion(n + 1); // Incorrect base case, no termination condition
(gdb) next
Breakpoint 2, faultyRecursion (n=4) at error_recursive_logic.c:4
4 printf("Recursion level: %d\n", n);
(gdb) print n
$8 = 4
(gdb) next
Recursion level: 4
6 faultyRecursion(n + 1); // Incorrect base case, no termination condition
(gdb) print n
$9 = 4
(gdb) next
Breakpoint 2, faultyRecursion (n=5) at error_recursive_logic.c:4
4 printf("Recursion level: %d\n", n);
(gdb) print n
$10 = 5
(gdb) reverse-next
Target multi-thread does not support this command.
(gdb) batcktrace
Undefined command: "batcktrace". Try "help".
(gdb) backtrace
#0 faultyRecursion (n=5) at error_recursive_logic.c:4
#1 0x000055555555517e in faultyRecursion (n=4) at error_recursive_logic.c:6
#2 0x000055555555517e in faultyRecursion (n=3) at error_recursive_logic.c:6
#3 0x000055555555517e in faultyRecursion (n=2) at error_recursive_logic.c:6
#4 <function called from gdb>
#5 faultyRecursion (n=2) at error_recursive_logic.c:4
#6 0x000055555555517e in faultyRecursion (n=1) at error_recursive_logic.c:6
#7 0x0000555555555193 in main () at error_recursive_logic.c:10
(gdb) run main
The program being debugged has been started already.
Start it from the beginning? (y or n) n
Program not restarted.
(gdb) run fix_error_recursive_logic.c


cloud2.rpsconsulting.in/console/#/client/MzM4NzEAYwBteXNxbA==
Activities
Terminal
Jan 25 18:39
rps@rps-Standard-PC-Q35-ICH9-2009: ~/manish/ManishDTA/capstone_project/project2_day1/module5
Recursion level: 251994
Recursion level: 251995
Recursion level: 251996
Recursion level: 251997
Recursion level: 251998
Recursion level: 251999
Recursion level: 252000
Recursion level: 252001
Recursion level: 252002
Recursion level: 252003
Recursion level: 252004
Recursion level: 252005
Recursion level: 252006
Recursion level: 252007
Recursion level: 252008
Recursion level: 252009
Recursion level: 252010
Recursion level: 252011
Recursion level: 252012
Recursion level: 252013
Recursion level: 252014
Recursion level: 252015
Recursion level: 252016
Recursion level: 252017
Recursion level: 252018
Recursion level: 252019
Recursion level: 252020
Recursion level: 252021
Recursion level: 252022
Recursion level: 252023
Recursion level: 252024
Recursion level: 252025
Recursion level: 252026
Recursion level: 252027
Recursion level: 252028
Recursion level: 252029
Recursion level: 252030
Recursion level: 252031


Activities
Terminal
Jan 25 18:41
rps@rps-Standard-PC-Q35-ICH9-2009: ~/manish/ManishDTA/capstone_project/project2_day1/module5
Recursion level: 261920
Recursion level: 261921
Recursion level: 261922
Recursion level: 261923
Recursion level: 261924
Recursion level: 261925
Recursion level: 261926
Recursion level: 261927
Recursion level: 261928
Recursion level: 261929
Recursion level: 261930
Recursion level: 261931
Recursion level: 261932
Recursion level: 261933
Recursion level: 261934
Recursion level: 261935
Recursion level: 261936
Recursion level: 261937
Recursion level: 261938
Recursion level: 261939
Recursion level: 261940
Recursion level: 261941
Recursion level: 261942
Recursion level: 261943
Recursion level: 261944
Recursion level: 261945
Recursion level: 261946
Recursion level: 261947
Recursion level: 261948
Recursion level: 261949
Recursion level: 261950
Program received signal SIGSEGV, Segmentation fault.
0x00007ffff7c8aee8 in _IO_new_file_write (f=0x7ffff7e1b780 <_IO_2_1_stdout_>, data=0x5555555592a0, n=24) at ./libio/fileops.c:1180
1180 ./libio/fileops.c: No such file or directory.
(gdb) backtrace
#0 0x00007ffff7c8aee8 in _IO_new_file_write (f=0x7ffff7e1b780 <_IO_2_1_stdout_>, data=0x5555555592a0, n=24) at ./libio/fileops.c:1180
#1 0x00007ffff7c8c9e1 in new_do_write (to_do=24, data=0x5555555592a0 "Recursion level: 261951\n", fp=0x7ffff7e1b780 <_IO_2_1_stdout_>)



Fixed the Recursive logic Error done reverse debugging to verify:----

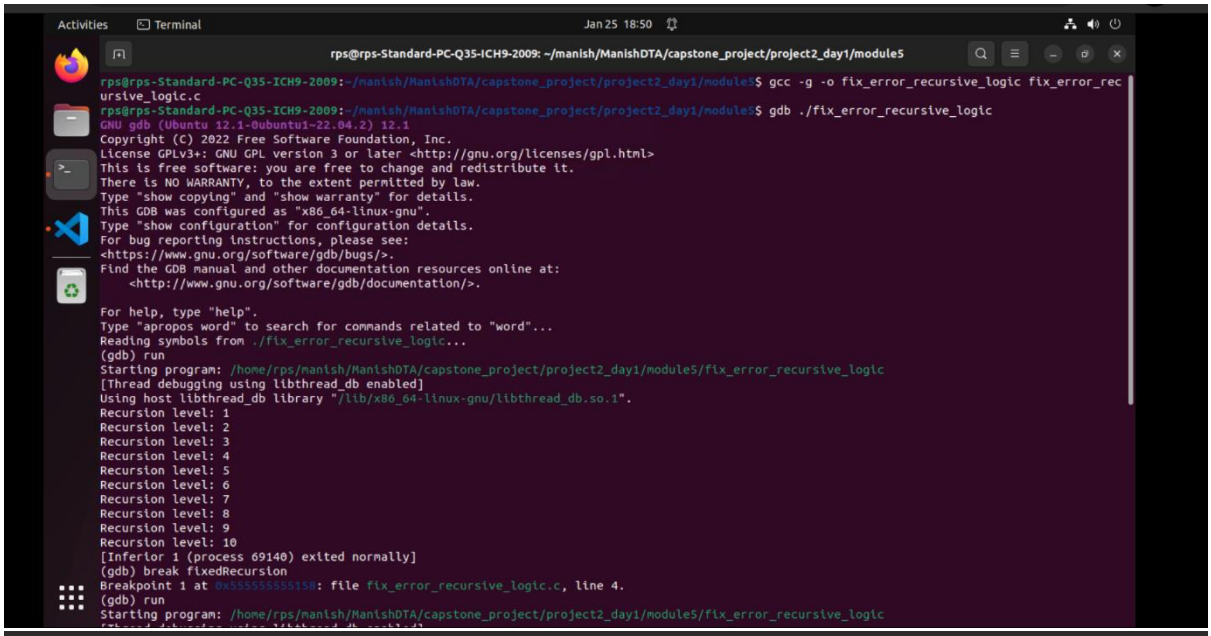
fix_error_recursive_logic.c

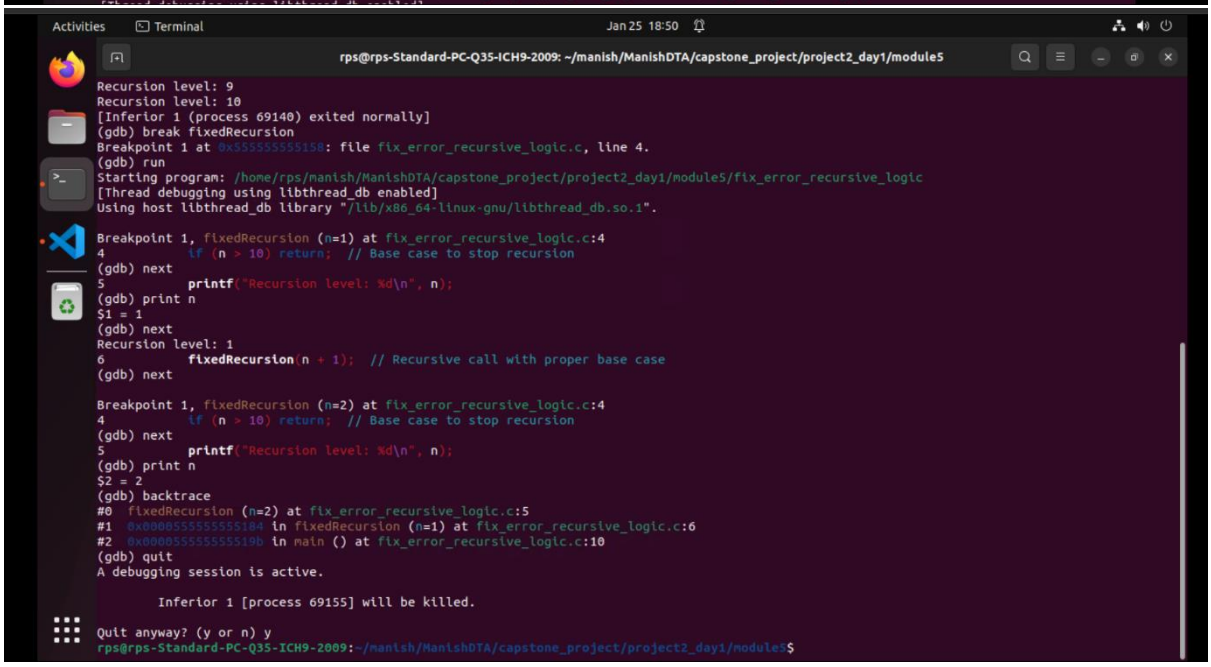
```
#include <stdio.h>


void fixedRecursion(int n) {
    if (n > 10) return;  // Base case to stop recursion
    printf("Recursion level: %d\n", n);
```

```
    fixedRecursion(n + 1);  // Recursive call with proper base case
}

int main() {
    fixedRecursion(1);  // Start recursion
    return 0;
}
```

Steps to Verify if Recursion is Fixed in GDB:--

```
gcc -g -o fix_error_recursive_logic fix_error_recursive_logic.c
gdb ./fix_error_recursive_logic
break fixedRecursion
run
next
print n
Backtrace
quit
```


Activities
Terminal
Jan 25 18:50
rps@rps-Standard-PC-Q35-ICH9-2009: ~/manish/ManishDTA/capstone_project/project2_day1/module5
rps@rps-Standard-PC-Q35-ICH9-2009:~/manish/ManishDTA/capstone_project/project2_day1/module5$ gcc -g -o fix_error_recursive_logic fix_error_rec
ursive_logic.c
rps@rps-Standard-PC-Q35-ICH9-2009:~/manish/ManishDTA/capstone_project/project2_day1/module5$ gdb ./fix_error_recursive_logic
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04.2) 12.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./fix_error_recursive_logic...
(gdb) run
Starting program: /home/rps/manish/ManishDTA/capstone_project/project2_day1/module5/fix_error_recursive_logic
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Recursion level: 1
Recursion level: 2
Recursion level: 3
Recursion level: 4
Recursion level: 5
Recursion level: 6
Recursion level: 7
Recursion level: 8
Recursion level: 9
Recursion level: 10
[Inferior 1 (process 69140) exited normally]
(gdb) break fixedRecursion
Breakpoint 1 at 0x555555555158: file fix_error_recursive_logic.c, line 4.
(gdb) run
Starting program: /home/rps/manish/ManishDTA/capstone_project/project2_day1/module5/fix_error_recursive_logic


Activities
Terminal
Jan 25 18:50
rps@rps-Standard-PC-Q35-ICH9-2009: ~/manish/ManishDTA/capstone_project/project2_day1/module5
Recursion level: 9
Recursion level: 10
[Inferior 1 (process 69140) exited normally]
(gdb) break fixedRecursion
Breakpoint 1 at 0x555555555158: file fix_error_recursive_logic.c, line 4.
(gdb) run
Starting program: /home/rps/manish/ManishDTA/capstone_project/project2_day1/module5/fix_error_recursive_logic
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Breakpoint 1, fixedRecursion (n=1) at fix_error_recursive_logic.c:4
4 if (n > 10) return; // Base case to stop recursion
(gdb) next
5 printf("Recursion level: %d\n", n);
(gdb) print n
$1 = 1
(gdb) next
Recursion level: 1
6 fixedRecursion(n + 1); // Recursive call with proper base case
(gdb) next
Breakpoint 1, fixedRecursion (n=2) at fix_error_recursive_logic.c:4
4 if (n > 10) return; // Base case to stop recursion
(gdb) next
5 printf("Recursion level: %d\n", n);
(gdb) print n
$2 = 2
(gdb) backtrace
#0 fixedRecursion (n=2) at fix_error_recursive_logic.c:5
#1 0x0000555555555184 in fixedRecursion (n=1) at fix_error_recursive_logic.c:6
#2 0x000055555555519b in main () at fix_error_recursive_logic.c:10
(gdb) quit
A debugging session is active.
Inferior 1 [process 69155] will be killed.
Quit anyway? (y or n) y
rps@rps-Standard-PC-Q35-ICH9-2009:~/manish/ManishDTA/capstone_project/project2_day1/module5$