Module.1

Use Case 1: Debugging Logical Errors in Loops with Reverse Debugging

Overview

Logical errors in loops can produce incorrect results, such as exceeding bounds or miscalculating outputs. Reverse debugging in GDB allows you to step backward to trace how the loop state evolved.

Logical Error:- logical_error.c

```
#include <stdio.h>


int main(){
   int n = 5;  //calculate factorial of 5
   int factorial = 1;


   //logical error: Incorrect initialization of the loop variable
   for(int i = 0; i <= n; i++){
      factorial *= i;   //causes multiplication by 0 in the first iteration


   }
   printf("Factorial of %d is %d\n", n, factorial);
   return 0;


}
```

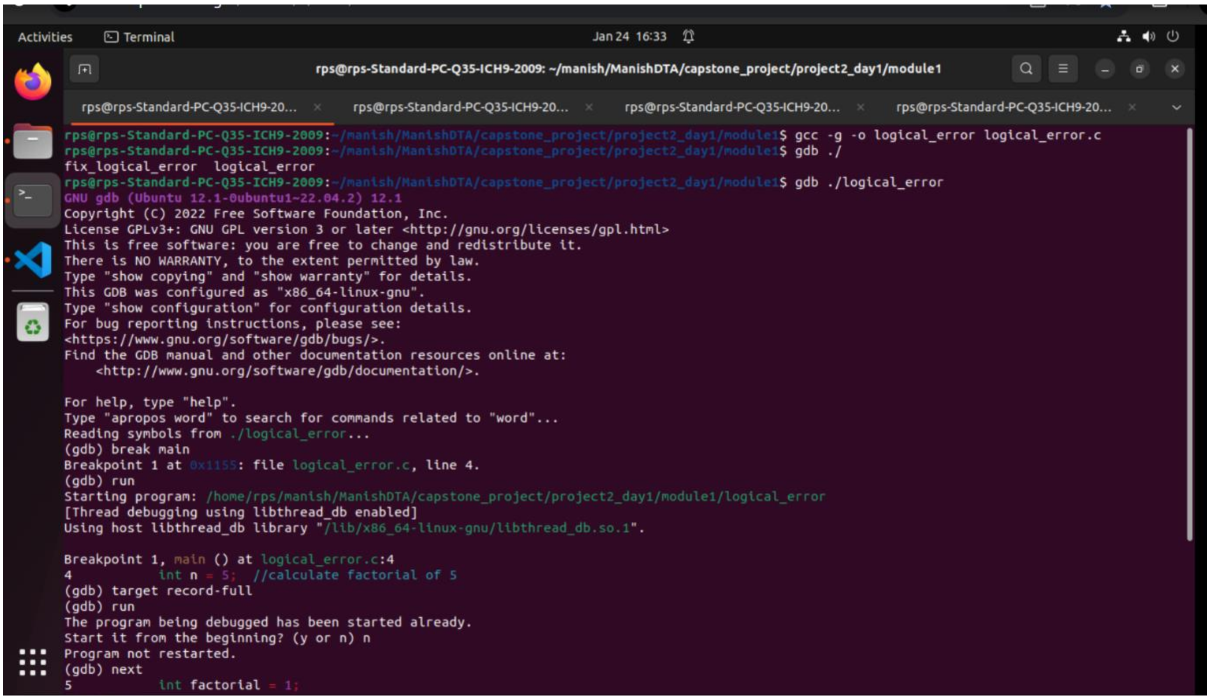
Steps to debug with gdb to find logical error:-

```
gcc -g -o logical_error logical_error.c
gdb ./logical_error
break main
target record-full
run
next
print i
print factorial
```

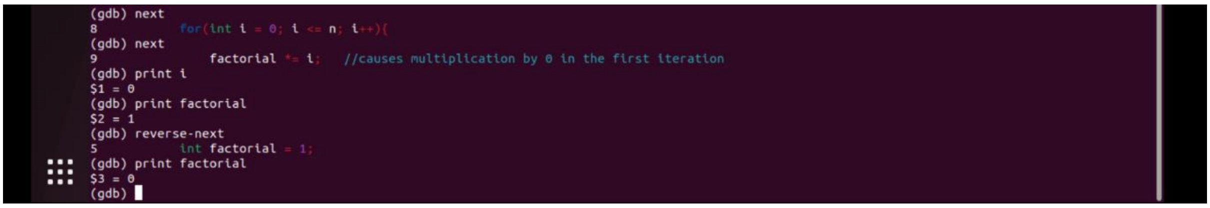
reverse-next

print i

print factorial


Activities
Terminal
Jan 24 16:33
rps@rps-Standard-PC-Q35-ICH9-2009: ~/manish/ManishDTA/capstone_project/project2_day1/module1
rps@rps-Standard-PC-Q35-ICH9-20...
rps@rps-Standard-PC-Q35-ICH9-20...
rps@rps-Standard-PC-Q35-ICH9-20...
rps@rps-Standard-PC-Q35-ICH9-20...

```
rps@rps-Standard-PC-Q35-ICH9-2009:~/manish/ManishDTA/capstone_project/project2_day1/module1$ gcc -g -o logical_error logical_error.c
rps@rps-Standard-PC-Q35-ICH9-2009:~/manish/ManishDTA/capstone_project/project2_day1/module1$ gdb ./
fix_logical_error  logical_error
rps@rps-Standard-PC-Q35-ICH9-2009:~/manish/ManishDTA/capstone_project/project2_day1/module1$ gdb ./logical_error
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 ./logical_error...
(gdb) break main
Breakpoint 1 at 0x1155: file logical_error.c, line 4.
(gdb) run
Starting program: /home/rps/manish/ManishDTA/capstone_project/project2_day1/module1/logical_error
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at logical_error.c:4
4           int n = 5;  //calculate factorial of 5
(gdb) target record-full
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) n
Program not restarted.
(gdb) next
5           int factorial = 1;
```



```
(gdb) next
8           for(int i = 0; i <= n; i++){
(gdb) next
9               factorial *= i;   //causes multiplication by 0 in the first iteration
(gdb) print i
$1 = 0
(gdb) print factorial
$2 = 1
(gdb) reverse-next
5           int factorial = 1;
(gdb) print factorial
$3 = 0
(gdb) 
```

Fix Logical Error:- fix_logical_error.c

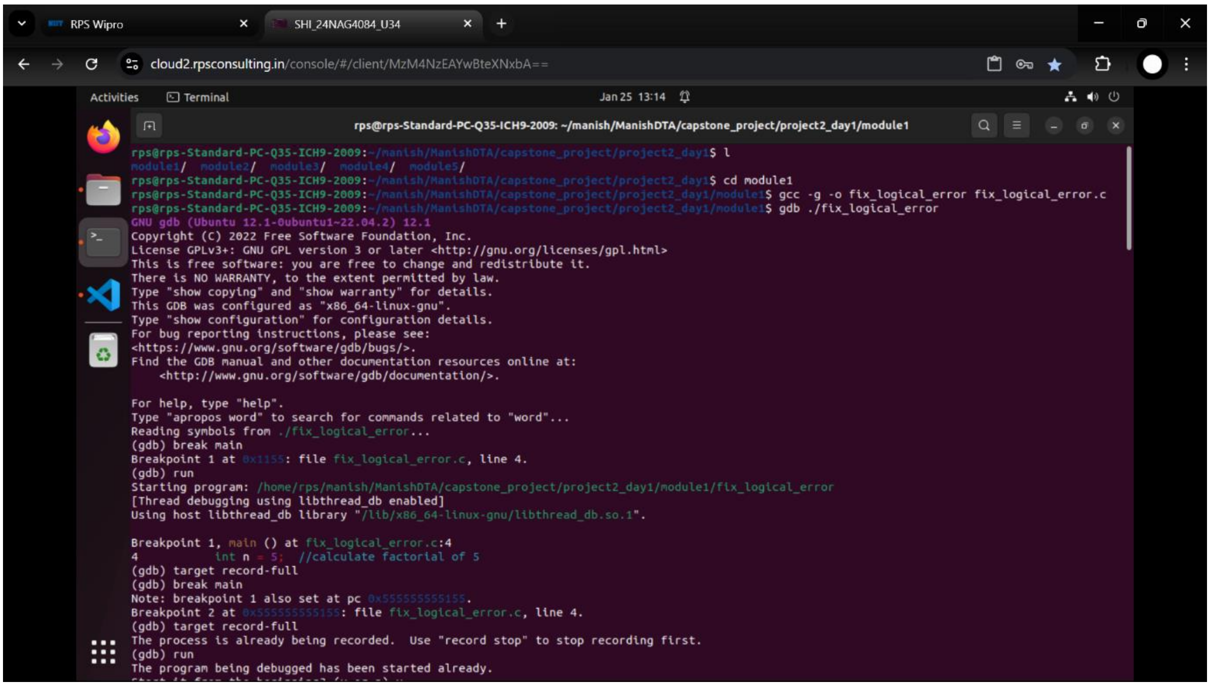
factorial_debug_fixed example.c

```
#include <stdio.h>

int main(){

   int n = 5;  //calculate factorial of 5
```

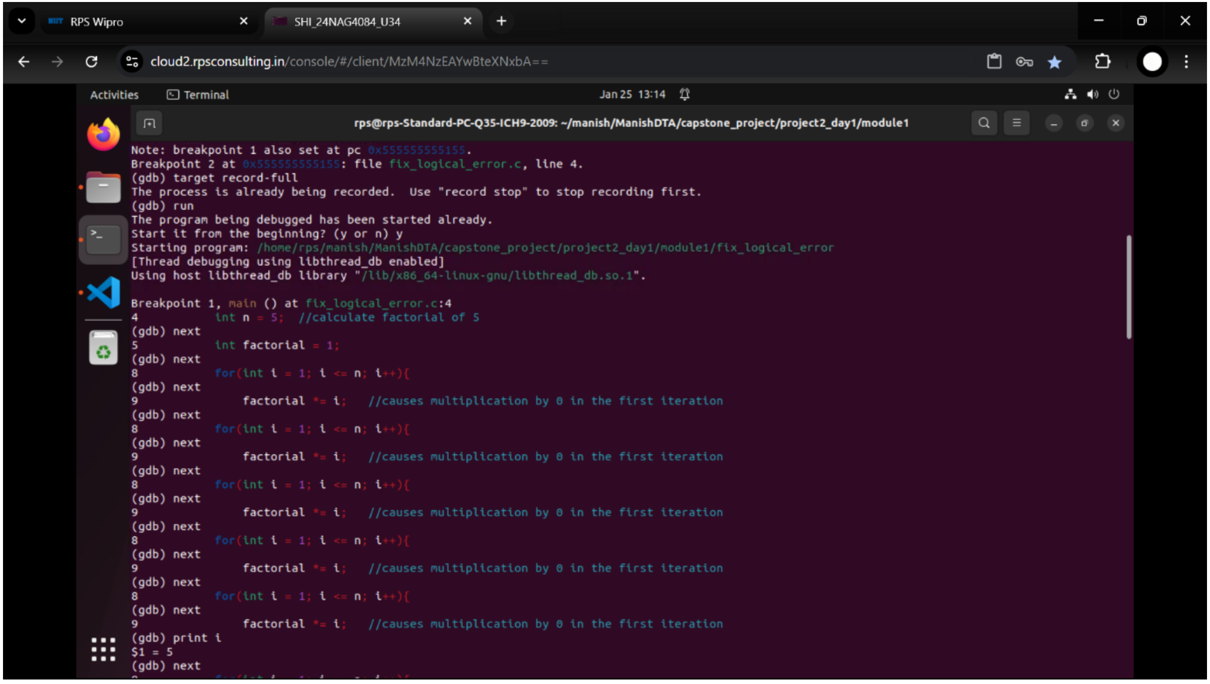
```
    int factorial = 1;


    //logical error: Incorrect initialization of the loop variable
    for(int i = 1; i <= n; i++){
        factorial *= i;   //causes multiplication by 0 in the first iteration


    }
    printf("Factorial of %d is %d\n", n, factorial);
    return 0;


}
```

Steps to debug with gdb to find logical error fixed or not :-

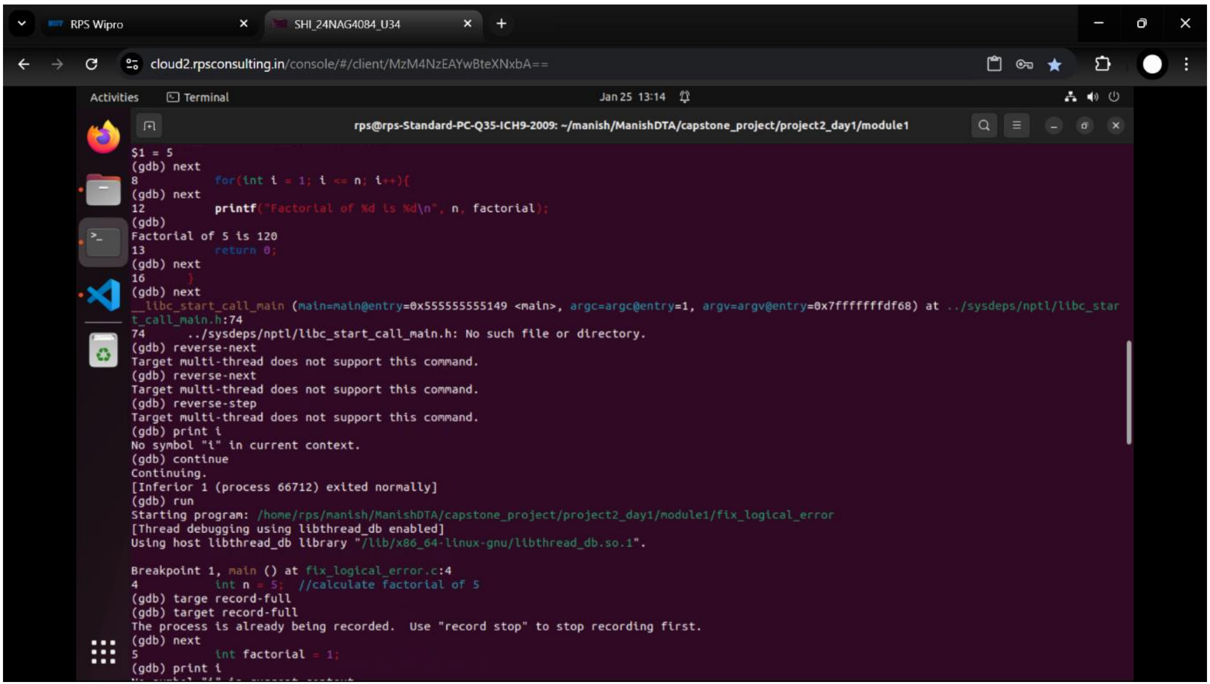
```
gcc -g -o factorial_debug_fixed example.c
gdb ./factorial_debug_fixed
break main
target record-full
run
next
print i
print factorial
reverse-next
print i
print factorial
continue
run
```


RPS Wipro
SHI_24NAG4084_U34
cloud2.rpsconsulting.in/console/#/client/MzM4NzEAYwBteXNxbA==
Activities
Terminal
Jan 25 13:14
rps@rps-Standard-PC-Q35-ICH9-2009: ~/manish/ManishDTA/capstone_project/project2_day1/module1

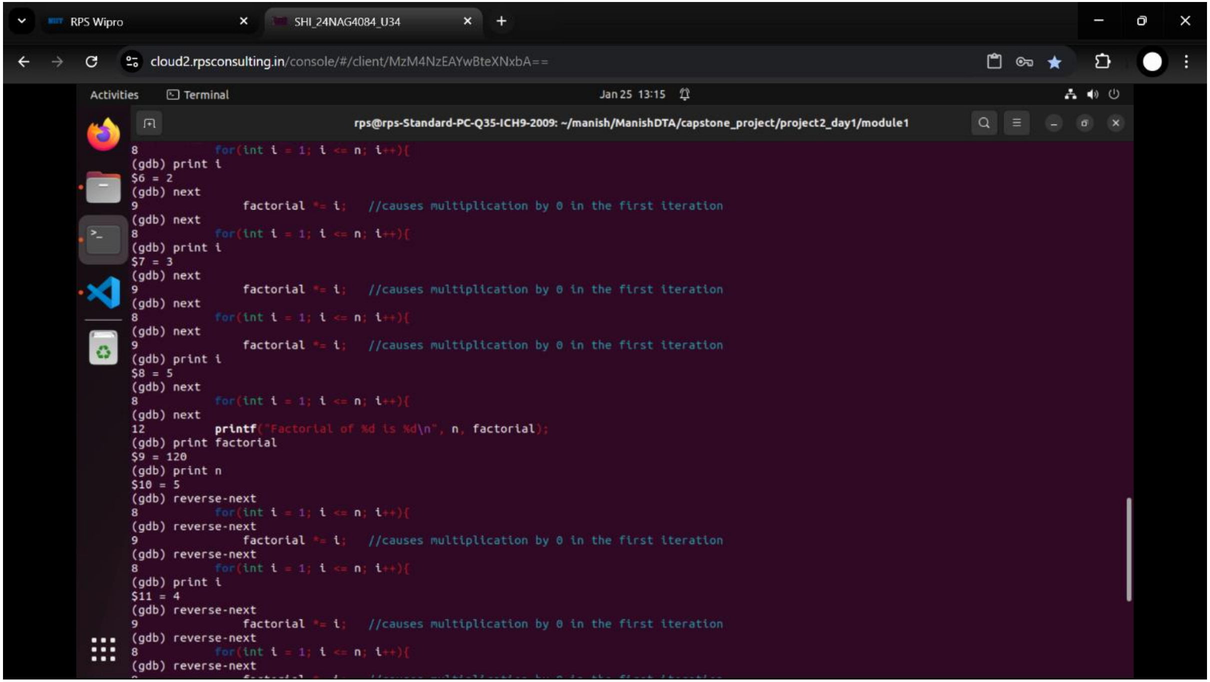
```
rps@rps-Standard-PC-Q35-ICH9-2009:~/manish/ManishDTA/capstone_project/project2_day1$ l
module1/  module2/  module3/  module4/  module5/
rps@rps-Standard-PC-Q35-ICH9-2009:~/manish/ManishDTA/capstone_project/project2_day1$ cd module1
rps@rps-Standard-PC-Q35-ICH9-2009:~/manish/ManishDTA/capstone_project/project2_day1/module1$ gcc -g -o fix_logical_error fix_logical_error.c
rps@rps-Standard-PC-Q35-ICH9-2009:~/manish/ManishDTA/capstone_project/project2_day1/module1$ gdb ./fix_logical_error
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_logical_error...
(gdb) break main
Breakpoint 1 at 0x1155: file fix_logical_error.c, line 4.
(gdb) run
Starting program: /home/rps/manish/ManishDTA/capstone_project/project2_day1/module1/fix_logical_error
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at fix_logical_error.c:4
4           int n = 5;  //calculate factorial of 5
(gdb) target record-full
(gdb) break main
Note: breakpoint 1 also set at pc 0x555555555155.
Breakpoint 2 at 0x555555555155: file fix_logical_error.c, line 4.
(gdb) target record-full
The process is already being recorded.  Use "record stop" to stop recording first.
(gdb) run
The program being debugged has been started already.
```


RPS Wipro
SHI_24NAG4084_U34
cloud2.rpsconsulting.in/console/#/client/MzM4NzEAYwBteXNxbA==
Activities
Terminal
Jan 25 13:14
rps@rps-Standard-PC-Q35-ICH9-2009: ~/manish/ManishDTA/capstone_project/project2_day1/module1

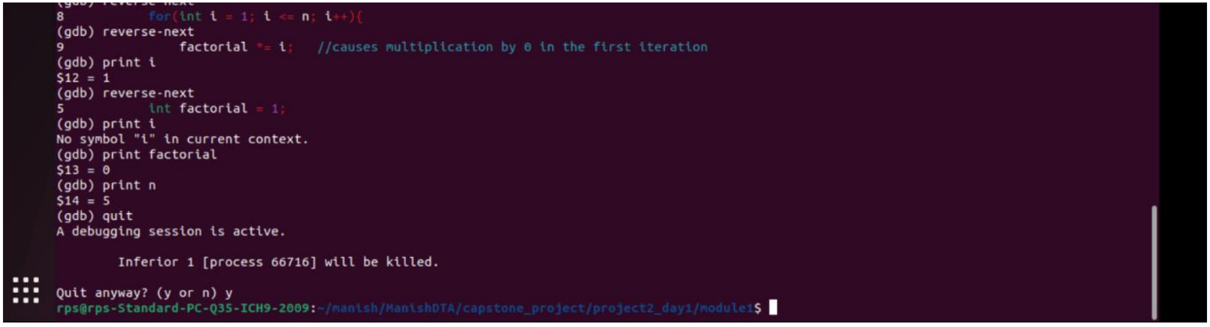
```
Note: breakpoint 1 also set at pc 0x555555555155.
Breakpoint 2 at 0x555555555155: file fix_logical_error.c, line 4.
(gdb) target record-full
The process is already being recorded.  Use "record stop" to stop recording first.
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/rps/manish/ManishDTA/capstone_project/project2_day1/module1/fix_logical_error
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at fix_logical_error.c:4
4           int n = 5;  //calculate factorial of 5
(gdb) next
5           int factorial = 1;
(gdb) next
8           for(int i = 1; i <= n; i++){
(gdb) next
9               factorial *= i;   //causes multiplication by 0 in the first iteration
(gdb) next
8           for(int i = 1; i <= n; i++){
(gdb) next
9               factorial *= i;   //causes multiplication by 0 in the first iteration
(gdb) next
8           for(int i = 1; i <= n; i++){
(gdb) next
9               factorial *= i;   //causes multiplication by 0 in the first iteration
(gdb) next
8           for(int i = 1; i <= n; i++){
(gdb) next
9               factorial *= i;   //causes multiplication by 0 in the first iteration
(gdb) next
8           for(int i = 1; i <= n; i++){
(gdb) next
9               factorial *= i;   //causes multiplication by 0 in the first iteration
(gdb) print i
$1 = 5
(gdb) next
```


RPS Wipro
SHI_24NAG4084_U34
cloud2.rpsconsulting.in/console/#/client/MzM4NzEAYwBteXNxbA==
Activities
Terminal
Jan 25 13:14
rps@rps-Standard-PC-Q35-ICH9-2009: ~/manish/ManishDTA/capstone_project/project2_day1/module1

```
$1 = 5
(gdb) next
8           for(int i = 1; i <= n; i++){
(gdb) next
12          printf("Factorial of %d is %d\n", n, factorial);
(gdb)
Factorial of 5 is 120
13          return 0;
(gdb) next
16      }
(gdb) next
__libc_start_call_main (main=main@entry=0x555555555149 <main>, argc=argc@entry=1, argv=argv@entry=0x7fffffffdf68) at ../sysdeps/nptl/libc_star
t_call_main.h:74
74      ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
(gdb) reverse-next
Target multi-thread does not support this command.
(gdb) reverse-next
Target multi-thread does not support this command.
(gdb) reverse-step
Target multi-thread does not support this command.
(gdb) print i
No symbol "i" in current context.
(gdb) continue
Continuing.
[Inferior 1 (process 66712) exited normally]
(gdb) run
Starting program: /home/rps/manish/ManishDTA/capstone_project/project2_day1/module1/fix_logical_error
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at fix_logical_error.c:4
4           int n = 5;  //calculate factorial of 5
(gdb) targe record-full
(gdb) target record-full
The process is already being recorded.  Use "record stop" to stop recording first.
(gdb) next
5           int factorial = 1;
(gdb) print i
```


RPS Wipro
SHI_24NAG4084_U34
cloud2.rpsconsulting.in/console/#/client/MzM4NzEAYwBteXNxbA==
Activities
Terminal
Jan 25 13:15
rps@rps-Standard-PC-Q35-ICH9-2009: ~/manish/ManishDTA/capstone_project/project2_day1/module1

```
8           for(int i = 1; i <= n; i++){
(gdb) print i
$6 = 2
(gdb) next
9               factorial *= i;   //causes multiplication by 0 in the first iteration
(gdb) next
8           for(int i = 1; i <= n; i++){
(gdb) print i
$7 = 3
(gdb) next
9               factorial *= i;   //causes multiplication by 0 in the first iteration
(gdb) next
8           for(int i = 1; i <= n; i++){
(gdb) next
9               factorial *= i;   //causes multiplication by 0 in the first iteration
(gdb) print i
$8 = 5
(gdb) next
8           for(int i = 1; i <= n; i++){
(gdb) next
12          printf("Factorial of %d is %d\n", n, factorial);
(gdb) print factorial
$9 = 120
(gdb) print n
$10 = 5
(gdb) reverse-next
8           for(int i = 1; i <= n; i++){
(gdb) reverse-next
9               factorial *= i;   //causes multiplication by 0 in the first iteration
(gdb) reverse-next
8           for(int i = 1; i <= n; i++){
(gdb) print i
$11 = 4
(gdb) reverse-next
9               factorial *= i;   //causes multiplication by 0 in the first iteration
(gdb) reverse-next
8           for(int i = 1; i <= n; i++){
(gdb) reverse-next
```



```
8           for(int i = 1; i <= n; i++){
(gdb) reverse-next
9               factorial *= i;   //causes multiplication by 0 in the first iteration
(gdb) print i
$12 = 1
(gdb) reverse-next
5           int factorial = 1;
(gdb) print i
No symbol "i" in current context.
(gdb) print factorial
$13 = 0
(gdb) print n
$14 = 5
(gdb) quit
A debugging session is active.

        Inferior 1 [process 66716] will be killed.

Quit anyway? (y or n) y
rps@rps-Standard-PC-Q35-ICH9-2009:~/manish/ManishDTA/capstone_project/project2_day1/module1$
```