

Chương trình bị lỗi. Vì vậy hãy xem loại thông tin nào có thể thu thập. Kiểm tra sự cố, ta có thể thấy chương trình ở dòng 28 của main.cc, điểm này chỉ đến 0 và chúng ta có thể thấy dòng mã đã được thực thi. Nhưng ta cũng muốn biết cái gì đã gọi phương thức này và muốn có thể kiểm tra các giá trị trong các phương thức gọi. Vì vậy, tại dấu nhắc gdb, nhập backtrace cho kết quả sau:

(gdb) backtrace

```
ubuntu@gdb-cpp: ~
File Edit View Search Terminal Help
Creating Node, 4 are in existence right now
The fully created list is:
4
3
2
1

Now removing elements:
Creating Node, 5 are in existence right now
Destroying Node, 4 are in existence right now
4
3
2
1

Program received signal SIGSEGV, Segmentation fault.
0x000055555555586c in Node<int>::next (this=0x0) at main.cc:28
28      Node<T>* next () const { return next_; }
(gdb) backtrace
#0  0x000055555555586c in Node<int>::next (this=0x0) at main.cc:28
#1  0x0000555555555763 in LinkedList<int>::remove (this=0x55555556aeb0,
      item_to_remove=@0x7fffffffe43c: 1) at main.cc:77
#2  0x00005555555553b1 in main (argc=1, argv=0x7fffffffe558) at main.cc:120
(gdb)
```

Vì vậy, ngoài những gì đã biết về phương thức hiện tại và các biến cục bộ, bây giờ ta cũng có thể xem những phương thức nào được gọi và tham số của chúng là gì. Ví dụ, có thể thấy rằng chương trình đã được gọi bởi LinkedList <int> :: remove () trong đó mục tham số cần loại bỏ ở địa chỉ 0x7fffffffe43c. Nó có thể giúp hiểu lỗi nếu biết giá trị của mục cần xóa, vì vậy cần xem tiếp giá trị tại địa chỉ của mục cần xóa. Điều này có thể được thực hiện bằng lệnh x sử dụng địa chỉ làm tham số. ("X" có thể được coi là viết tắt của "exam".) Đây là kết quả của việc chạy lệnh:

(gdb) x 0x7fffffffe43c

--> Vì vậy, chương trình gặp sự cố trong khi cố gắng chạy LinkedList <int> :: remove với tham số là 1. Bây giờ ta đã thu hẹp vấn đề xuống một hàm cụ thể và một giá trị cụ thể cho tham số.