

Bây giờ ta biết segfault đang xảy ra ở đâu và khi nào, chúng ta muốn xem chương trình đang làm gì trước khi nó bị treo. Một cách để làm điều này là thực hiện lần lượt từng câu lệnh của chương trình cho đến khi đi đến điểm thực thi mà ta muốn xem điều gì đang xảy ra. Cách làm này đúng, nhưng đôi khi ta có thể muốn chỉ chạy đến một phần mã cụ thể và dừng thực thi tại thời điểm đó để có thể kiểm tra dữ liệu tại vị trí đó. Nếu đã từng sử dụng trình gỡ lỗi, bạn có thể quen thuộc với khái niệm điểm ngắt (breakpoint). Về cơ bản, điểm ngắt là một dòng trong mã nguồn nơi trình gỡ lỗi sẽ ngắt thực thi. Trong ví dụ này, để xem mã trong LinkedList <int> :: remove (), ta đặt một điểm ngắt ở dòng 52 của main.cc.

```
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=@0x7fffffffef43c: 1) at main.cc:77
#2  0x00005555555553b1 in main (argc=1, argv=0x7fffffffef558) at main.cc:120
(gdb) x 0x7fffffffef43c
0x7fffffffef43c: 0x00000001
(gdb) break 52
Breakpoint 1 at 0x555555555f9: file main.cc, line 52.
(gdb)
```

Lúc này Breakpoint 1 được đặt tại main.cc, dòng 52. (Lý do điểm ngắt nhận một số là vì ta có thể tham khảo điểm ngắt sau này, chẳng hạn như nếu muốn xóa nó.) Vì vậy, khi chương trình được chạy, nó sẽ trả lại quyền điều khiển cho trình gỡ lỗi mỗi khi nó đến dòng 52. Điều này có thể không mong muốn nếu phương thức được gọi nhiều lần nhưng chỉ có vấn đề với một số giá trị nhất định được truyền. Các điểm ngắt có điều kiện có thể giúp ta lúc này. Ví dụ: ta biết rằng chương trình bị treo khi LinkedList <int> :: remove () được gọi với giá trị là 1. Vì vậy, ta có thể yêu cầu trình gỡ lỗi chỉ ngắt ở dòng 52 nếu mục cần xóa bằng 1. Điều này có thể được thực hiện bằng cách ra lệnh sau:

```
(gdb) condition 1 item_to_remove==1
```

Như vậy ta đã đặt một điểm ngắt có điều kiện và bây giờ muốn xem qua hàm này từng dòng một và xem liệu có thể xác định được nguồn gốc của lỗi hay không. Điều này được thực hiện bằng cách sử dụng lệnh step. gdb có một tính năng hay là khi nhấn enter mà không cần gõ lệnh thì lệnh cuối cùng sẽ tự động được sử dụng. Bằng cách đó, chúng ta có thể thực hiện một cách đơn giản bằng cách nhấn vào phím enter sau khi bước đầu tiên đã được nhập.