

想知道指定键的下一个键，无须从映射的开头开始遍历。

`bpf_map_get_next_key` 可以发挥的作用还不止如此。你需要注意的另一个行为是许多编程语言会在迭代映射元素前复制映射的值。这样当程序中的其他代码对映射进行修改时，可以阻止未知错误，尤其是从映射中删除元素，这是特别危险的。BPF 使用 `bpf_map_get_next_key` 在遍历映射前不复制映射的值。如果程序正在遍历映射元素，程序的其他代码删除了映射中的元素，当遍历程序尝试查找的下一个值是已删除元素的键时，`bpf_map_get_next_key` 将重新开始查找，下面是示例代码：

```
int next_key, lookup_key;
lookup_key = -1;

while(bpf_map_get_next_key(map_data[0].fd, &lookup_key, &next_key) == 0) {
    printf("The next key in the map is: '%d'\n", next_key);
    if (next_key == 2) {
        printf("Deleting key '2'\n");
        bpf_map_delete_element(map_data[0].fd &next_key);
    }
    lookup_key = next_key;
}
```

下面是程序打印的输出：

```
The next key in the map is: '1'
The next key in the map is: '2'
Deleteing key '2'
The next key in the map is: '1'
The next key in the map is: '3'
The next key in the map is: '4'
The next key in the map is: '5'
```

这个行为不是很直观，所以在使用 `bpf_map_get_next_key` 时请牢记。

在本章中我们介绍的大多数映射类型的行为类似于数组，所以当你想要访问保存在 BPF 映射中的信息时，遍历操作是一种关键操作。然而，这里还有一些附加功能可以用来访问 BPF 映射，我们将在下面介绍。

3.2.5 查找和删除元素

内核为 BPF 映射提供的另一个功能是 `bpf_map_lookup_and_delete_elem`。此功能是在映射中查找指定的键并删除元素。同时，程序将该元素的值赋予