Phone Book Data Structure

All the phone numbers in the world in 2.5 GigaBytes

Pratik Deoghare

March 28, 2015

Contents

1 Problem

A phone company wants to keep track of subscribers to their service. They want to be able to perform following operations,

1. Assign new customer a phone number that is not already in use.
2. Remove customer from the database and make his number available.
3. Find an unused number to assign to new customer.

Range of phone numbers is 000-000-0000 to 999-999-9999.

2 Solution

3 Usage

```
#include "PhoneBook.h"

int main(){
    my::PhoneBook<9999999999> pb;
    ...
    pb.is_used(number); // Check if number is used.
    pb.set_used_status(number, true); // Add user
    pb.set_used_status(number, false); // Remove user
    pb.get_unused(); // Get an unused number

    ...
}
```

Find code in `./code/`.

Basic test code in `./code/test.cc`.

Valgrind output:

```
 ~/code/phonebook/code/ master* valgrind ./bigtest
==22343== Memcheck, a memory error detector
==22343== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==22343== Using Valgrind-3.9.0 and LibVEX; rerun with -h for copyright info
==22343== Command: ./bigtest
==22343==
==22343== Warning: set address range perms: large range [0x39fe1040, 0x847f8cc0) (undefined)
==22343== Warning: set address range perms: large range [0x847f9040, 0xcf010cc0) (undefined)
==22343== Warning: set address range perms: large range [0x847f9028, 0xcf010cd8) (noaccess)
==22343== Warning: set address range perms: large range [0x39fe1028, 0x847f8cd8) (noaccess)
==22343==
==22343== HEAP SUMMARY:
==22343==     in use at exit: 0 bytes in 0 blocks
==22343==   total heap usage: 2 allocs, 2 frees, 2,500,000,000 bytes allocated
==22343==
==22343== All heap blocks were freed -- no leaks are possible
==22343==
==22343== For counts of detected and suppressed errors, rerun with: -v
==22343== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 2 from 2)
 ~/code/phonebook/code/ master*
```