OpenVox Communication Co.Ltd

How to Filter Number Over

OpenVox GSM Cards

V1.0

OpenVox Communication Co.Ltd

OpenVox-Best Cost Effective Asterisk Cards

Address: F/3, Block No.127, Jindi Industrial Zone,
Shazui Road, Futian District, Shenzhen, Guangdong 518048, China
Tel:+86-755-82535461, 82535095, 82535362, Fax:+86-755-83823074
Business Contact: sales@openvox.com.cn
Technical Support: support@openvox.com.cn

Business Hours: 09:00-18:00(GMT+8) from Monday to Friday
URL: www.openvox.cn

Thank You for Choosing OpenVox Products!

Copyright

Confidentiality

Information contained herein is of a highly sensitive nature and is confidential and proprietary to OpenVox Inc. No part may be distributed, reproduced or disclosed orally or in written form to any party other than the direct recipients without the express written consent of OpenVox Inc.

Disclaimer

OpenVox Inc. reserves the right to modify the design, characteristics, and products at any time without notification or obligation and shall not be held liable for any error or damage of any kind resulting from the use of this document.
OpenVox has made every effort to ensure that the information contained in this document is accurate and complete; however, the contents of this document are subject to revision without notice. Contact OpenVox to ensure you have the latest revision of this document.

Trademarks

All other trademarks mentioned in this document are the property of their respective owners.

Contents

1. Test Environments

Software Platform：

1) Install the driver chan_extra-2.0.2 or higher versions

2) Get the latest software packet from the following URL:

   http://www.chan-extra.org/index.php/Main_Page

3) Centos 5.6 or higher versions

Hardware Platform：

1) PC with PCI or PCI-E slot(s)

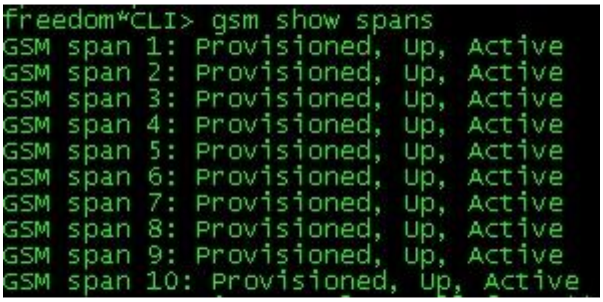
2) G400P/E telephony card(s)

2. Operation Instructions

Filtering phone number is achieved by running the command “gsm check phone stat” on CLI, and it will return the cellphone status. The complete format of command “**gsm check phone stat**” is like：

gsm check phone stat <span> <number> <hangup> [timeout]

a. The parameter **span** stands for the gsm channel and starts from 1. You will be able to check status of all spans by “**gsm show spans**”. For example, the following figure shows all 10 spans in the system.



```
freedom*CLI> gsm show spans
GSM span 1: Provisioned, Up, Active
GSM span 2: Provisioned, Up, Active
GSM span 3: Provisioned, Up, Active
GSM span 4: Provisioned, Up, Active
GSM span 5: Provisioned, Up, Active
GSM span 6: Provisioned, Up, Active
GSM span 7: Provisioned, Up, Active
GSM span 8: Provisioned, Up, Active
GSM span 9: Provisioned, Up, Active
GSM span 10: Provisioned, Up, Active
```

b. The parameter **`number`** stands for the number to be detected.

c. The parameter **`hangup`** stands for whether to be hang up if has detected the number is existed. The value "0" stands for not hang up automatically and "1" is on hook. When has selected "0" and detected the number, it will be continued to make the dial-up. If need to be hang up, you should manual do it or wait for the other party hangs up firstly.

d. The parameter **`timeout`** stands for detecting timeout. When the number detected is empty, busy or power off, with operators' voice prompt, will just return the status. As a result, it needs to set timeout. The longer waiting time, the higher accuracy ratio, but the slower the filtering speeds. The value of timeout should be set according to the operator because different operators different voice durations.

Result format returned：

Type：XXXXXX status

The following table instructs two types:

Name	**Introduction**
PHONE	Corresponding to the current result cellphone status.
SPAN	Corresponding to the channel status. Just when more programs call one and the same channel to filter number, it will return such type result. It only allowed that a channel filters a number in the same moment.

XXXXXX： Stands for the tested number or channel

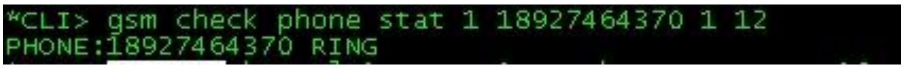
status： Including 5 different statuses

Name	**Introduction**
USING	The current channel has been using.
CONNECT	The other party is connected or busy.
RING	The other party is ringing.
BUSY	The other party is calling with else.
TIMEOUT	The other party is busy, power off or not existed. According to your operator to set timeout value limit.
NOTEXIST	The number is not existed

3. Typical Examples

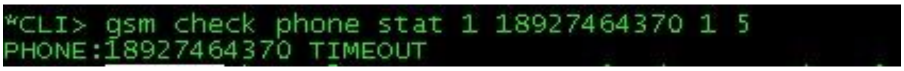
a. Testing a number by commands

Ringing status:



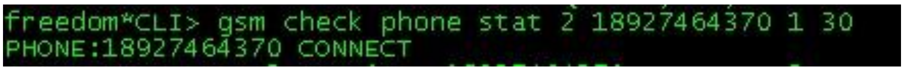
```
*CLI> gsm check phone stat 1 18927464370 1 12
PHONE:18927464370 RING
```

Timeout status：



```
*CLI> gsm check phone stat 1 18927464370 1 5
PHONE:18927464370 TIMEOUT
```

Connected or busy status：



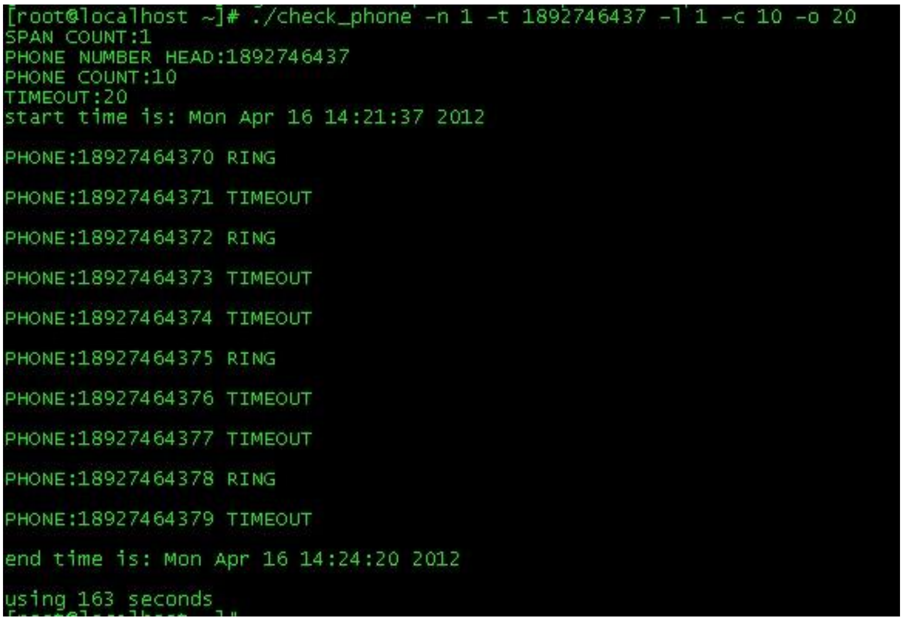
```
freedom*CLI> gsm check phone stat 2 18927464370 1 30
PHONE:18927464370 CONNECT
```

b. Multi-channel filter numbers

Multi-channel filter numbers simultaneously by Multithreaded

Program calls the filtering instructs, and then returns results.

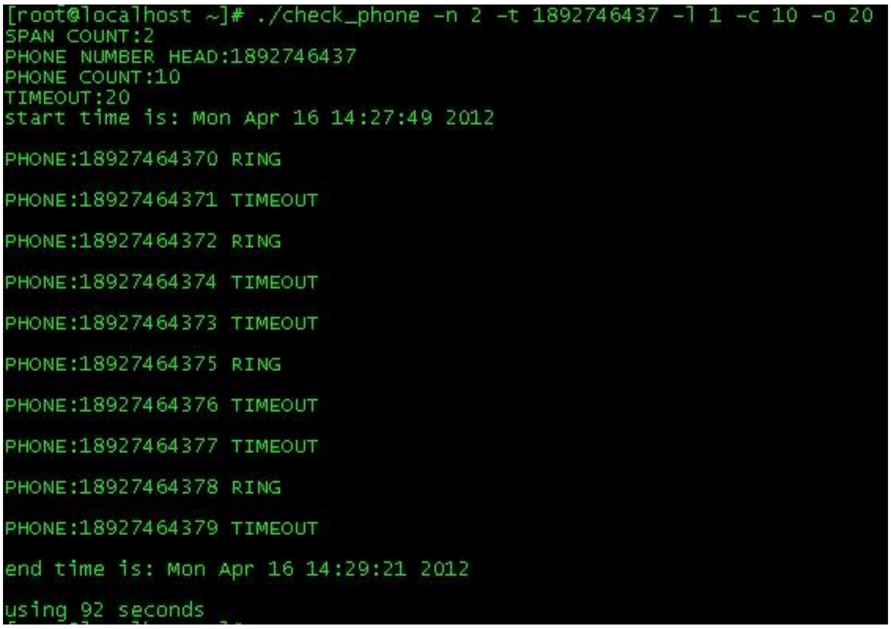
1）Single channel filters 10 numbers



```
[root@localhost ~]# ./check_phone -n 1 -t 1892746437 -l 1 -c 10 -o 20
SPAN COUNT:1
PHONE NUMBER HEAD:1892746437
PHONE COUNT:10
TIMEOUT:20
start time is: Mon Apr 16 14:21:37 2012

PHONE:18927464370 RING

PHONE:18927464371 TIMEOUT

PHONE:18927464372 RING

PHONE:18927464373 TIMEOUT

PHONE:18927464374 TIMEOUT

PHONE:18927464375 RING

PHONE:18927464376 TIMEOUT

PHONE:18927464377 TIMEOUT

PHONE:18927464378 RING

PHONE:18927464379 TIMEOUT

end time is: Mon Apr 16 14:24:20 2012

using 163 seconds
```

It takes 163 seconds in total for a single channel to filter 10 numbers, and 16 seconds in average for a number. Due to some empty numbers waste much time, so it takes long seconds to be checked for a single number in average.

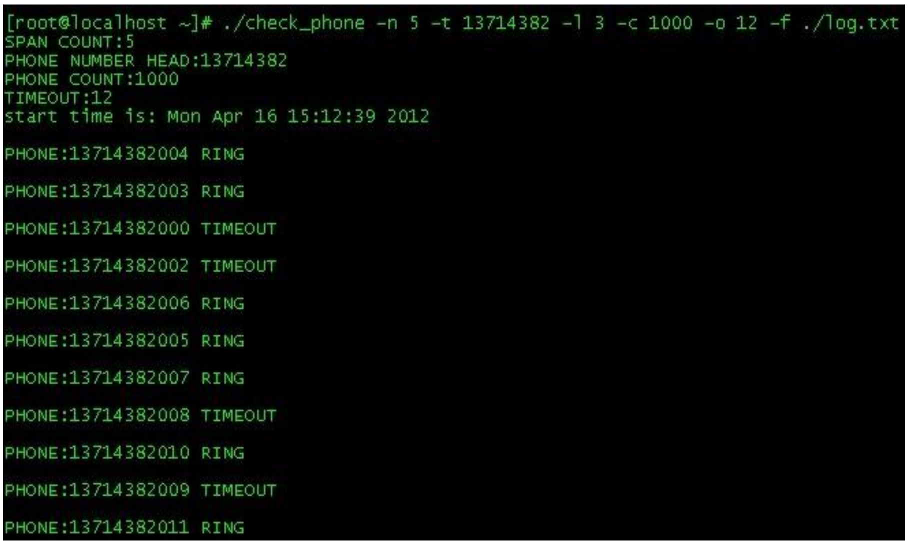
2）Dual-channel filters 10 numbers



```
[root@localhost ~]# ./check_phone -n 2 -t 1892746437 -l 1 -c 10 -o 20
SPAN COUNT:2
PHONE NUMBER HEAD:1892746437
PHONE COUNT:10
TIMEOUT:20
start time is: Mon Apr 16 14:27:49 2012

PHONE:18927464370 RING

PHONE:18927464371 TIMEOUT

PHONE:18927464372 RING

PHONE:18927464374 TIMEOUT

PHONE:18927464373 TIMEOUT

PHONE:18927464375 RING

PHONE:18927464376 TIMEOUT

PHONE:18927464377 TIMEOUT

PHONE:18927464378 RING

PHONE:18927464379 TIMEOUT

end time is: Mon Apr 16 14:29:21 2012

using 92 seconds
```

It takes 92 seconds in total for two channels to filter 10 numbers, and 9.2 seconds in average for a number. It is faster than single channel filters 10 numbers. Due to some empty numbers waste much time, the time filtered by dual channels will not be half of that by single channel.

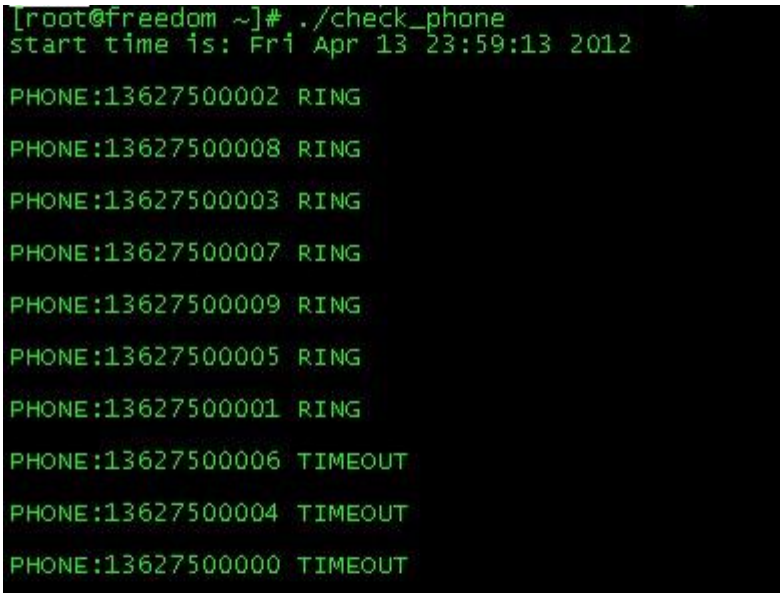
3）Five channels filter 1000 numbers



```
[root@localhost ~]# ./check_phone -n 5 -t 13714382 -l 3 -c 1000 -o 12 -f ./log.txt
SPAN COUNT:5
PHONE NUMBER HEAD:13714382
PHONE COUNT:1000
TIMEOUT:12
start time is: Mon Apr 16 15:12:39 2012

PHONE:13714382004 RING

PHONE:13714382003 RING

PHONE:13714382000 TIMEOUT

PHONE:13714382002 TIMEOUT

PHONE:13714382006 RING

PHONE:13714382005 RING

PHONE:13714382007 RING

PHONE:13714382008 TIMEOUT

PHONE:13714382010 RING

PHONE:13714382009 TIMEOUT

PHONE:13714382011 RING
```

It takes 58 minutes at a rough estimate for 5 channels to filter 1000 numbers and 3.5 seconds filter a number in average.

4）10 channels filter 20000 numbers



```
[root@freedom ~]# ./check_phone
start time is: Fri Apr 13 23:59:13 2012

PHONE:13627500002 RING

PHONE:13627500008 RING

PHONE:13627500003 RING

PHONE:13627500007 RING

PHONE:13627500009 RING

PHONE:13627500005 RING

PHONE:13627500001 RING

PHONE:13627500006 TIMEOUT

PHONE:13627500004 TIMEOUT

PHONE:13627500000 TIMEOUT
```

It takes about 5 hours for 10 channels to filter 20000 numbers and 1.5 seconds filter a number in average.

From the above four testing, did you find the filtering time law ?

4. Sample Program

There is a simple sample program source code for convenience calling filter function, and this is the URL.

check_phone.c

The following is the compiler method：

Gcc –o check_phone check_phone.c –lpthread

The test program parameter is like below:

./check_phone -n <spans_number> -t <phone_title> -l <phone_buttom_length> -c <phone_numbers> -o <timeouts>

Parameter definitions：

Spans_number： Channel number (1-X) used in filtering

phone_title： The front part of the phone number, for instance, 13530667 (whole number is 13506678880)

phone_buttom_length： Tail number， which corresponds to phone_title. For example, when phone_title is 13530667 and its tail number is 3 (last 3 digits)

phone_numbers: Filtering phone number count in total

Timeouts: Single filtering number timeout duration