Description
Some time ago, I submitted a patch to allow submitting number ranges not only as decimal, but also as hexadecimal or octal: https://gerrit.osmocom.org/c/libosmocore/+/19422
However, it was never merged.

Today a similar topic was raised again due to SMSCB Message IDs being printed in the vty "show messages" command as hexadecimal (and without 0x prefix) while the "show message ID" only accepted decimal numbers, which makes the user experience really confusing (I even spent some time debugging why "show message ID" was saying no message existing and I was seeing it in the list).

It was also raised that cisco console seems to also support hex numbers, see for instance "lac" in https://www.cisco.com/c/en/us/td/docs/wireless/asr_5000/21-1/CLI/books/R-Z/21-1_R-Z_CLI_Reference/21-1_R-Z_CLI_Reference_chapter_0100001.html
Supporting ranges in hexadecimal would also us to set stuff like lac as hexadecimal, and even store the config that way, because the vty would be able to read it back.

Similary, supporting octal would allow us to easily specify file permissions bitmasks in the VTY.

I see 2 main options to support those:
1- Specify range parameters as decimal ranges (ex: <0-200>) but allowing passing any format of integer (dec, hex, oct) by specifying the format prefix. That's what the libosmocore patch above adds support for.
For instance:

- cmd definition: "lac <0-6534324>"
- cmd write: "lac 0x30"

2- Specify accepted range formats explicitly. This means each param needs to be more detailed but also allows accepting specific formats on specific parameters. For instance, one would allow only decimal and hex in lac cmd: "lac (<0-6534324>|<0x0-0x4352>), or only accept hex: "lac <0x0-0x4352>"

I'm happy with any of the two above. Let's see what others want to say about it.

History
#1 - 07/25/2022 04:43 PM - pespin
- Description updated

#2 - 07/25/2022 04:53 PM - laforge
pespin wrote:
Similary, supporting octal would allow us to easily specify file permissions bitmasks in the VTY.

do we actually do that anywhere? I would guess most people who're not unix sysadmins or C programmers wouldn't know that 01234 is != 1234 and hence I would really avoid that notation. And 0/O is just so close in terms of "hamming distance of the character".... so I fear more danger than benefit.

I see 2 main options to support those:
1- Specify range parameters as decimal ranges (ex: <0-200>) but allowing passing any format of integer (dec, hex, oct) by specifying the format prefix. That's what the libosmocore patch above adds support for.
For instance:
For the input, I think that is the best

2- Specify accepted range formats explicitly. This means each param needs to be more detailed but also allows accepting specific formats on specific parameters. For instance, one would allow only decimal and hex in lac cmd: "lac <0-6534324>|<0x0-0x4352>, or only accept hex: "lac <0x0-0x4352>"

it could also be both:

- if a decimal range is specified, allow decimal or hex input (I'd avoid octal)
- if a hexadecimal range is specified in the command, allow only hexadecimal input

#3 - 07/25/2022 05:22 PM - pespin
laforge wrote in #note-2:

pespin wrote:

Similary, supporting octal would allow us to easily specify file permissions bitmaks in the VTY.

do we actually do that anywhere? I would guess most people who're not unix sysadmins or C programmers wouldn't know that 01234 is != 1234 and hence I would really avoid that notation. And 0/O is just so close in terms of "hamming distance of the character".... so I fear more danger than benefit.

https://gitea.osmocom.org/osmocom/osmo-pcap/src/branch/master/src/osmo_server_vty.c#L138
https://gitea.osmocom.org/osmocom/osmo-pcap/src/branch/master/src/osmo_server_vty.c#L91

it could also be both:

- if a decimal range is specified, allow decimal or hex input
- if a hexadecimal range is specified in the command, allow only hexadecimal input

That's a bit confusing at first but I see your point. I'm fine with either this solution (let's call it "3") or any of the 2 I presented before.

(I'd avoid octal)

Bear in mind allowing octal makes it actually easier implementation, because (see presented libosmocore patch before) one can simply pass 0 to the base to stroul and let it parse properly whatever base it is thrown at it.

#4 - 07/26/2022 08:00 AM - laforge
On Mon, Jul 25, 2022 at 05:22:31PM +0000, pespin wrote:

Bear in mind allowing octal makes it actually easier implementation, because (see presented libosmocore patch before) one can simply pass 0 to the base to stroul and let it parse properly whatever base it is thrown at it.

I'd rather invest a few more dozens of minutes of R&D time rather than sending our users into a trap.

#5 - 08/30/2022 05:51 PM - laforge
- Status changed from Feedback to Stalled
- Assignee changed from laforge to pespin

#6 - 09/14/2022 05:11 PM - pespin
I have been working a bit more on this.
I already implemented the check of the numeric ranges so that:

- If a decimal range is specified in the vty command (eg <0-300>), then input arg can be either decimal or hex, and both are accepted.
- If a hexadecimal range is specified in the vty command (always prefixed by "0x"|"0X" to avoid confusions, eg <0x0-0x500>), then input arg must also be hexadecimal.

However, this may cause lots of unexpected errors. Because even if the checking is fine, then now a lot of existing VTY commands may receive a hexadecimal input string, while this was not possible before. Lots of them are probably not using strotul() with base=0, or even worse, using atoi(), which only accepts decimal base and has no way to return an error.
So that would end up in lots of unexpected behaviors if hexadecimal values are typed for existing commands.

Hence, I think the best is to only accept decimal arg values if decimal ranges are specified. If one wishes to support both types of ranges in a specific command, then the arg can be specified as "(<0-255>|<0x00-0xff>)" and then use strtol(base=0).

#7 - 09/14/2022 05:50 PM - pespin
We may want to support the same in CTRL interface:

CTRL_CMD_DEFINE_RANGE(bts_lac, "location-area-code", struct gsm_bts, location_area_code, 0, 65535);

#8 - 09/14/2022 06:09 PM - pespin
- Status changed from Stalled to Feedback
- % Done changed from 0 to 80

Here's the new patch for libosmocore allowing to specify hexadecimal numeric ranges:
https://gerrit.osmocom.org/c/libosmocore/+/19422 vty: Allow hex representations in cmd numeric ranges

And here's a patch for osmo-bsc using this feature for location_area_update:
https://gerrit.osmocom.org/c/osmo-bsc/+/29358 vty: Allow setting LAC as hexadecimal value