
mohan43u's space

Release 0.0.1

Mohan R

May 07, 2024

CONTENTS

1	Creating ta-remington.mim for m17n-db	1
1.1	Problem	1
1.2	Learning m17n-db format	1
1.3	Writing ta-remington.mim	1
1.4	Testing the new Layout	2
1.5	Creating Icon	2
1.6	Pushing to m17n-db	2
2	Reconfiguring Keyboard Keys with XKB	3
2.1	Scan Codes	3
2.2	Kernel Keycodes	3
2.3	Console Keymaps	4
2.4	XKB	5
2.5	Modifications for my broken Keyboard	6
2.6	Drawback of XKB	8
3	Modernising IRC	9
3.1	WeeChat	9
3.2	WeeChat Relay	10
3.3	WeeChat Android	10
3.4	IrssiNotifier	10
3.5	WeeNotifier	10
3.6	Steps to Modernize IRC	11
3.7	Why I wrote WeeNotifier	11
4	Migrated my WordPress blog to GitLab	13
4.1	Inspiration	14
4.2	Website Design	15
4.3	Fixed Vssue GitLab Integration	18
4.4	ModifiedDate Extension for Sphinx	20
4.5	Exporting Blog posts from WordPress to Sphinx	22
4.6	PDF Integration	25
4.7	Publishing the Static Website Automatically	27
5	Emacs LSP for C	29
5.1	LSP (Language Server Protocol)	29
5.2	Installation	30
5.3	Configuration	31
5.4	Running emacs with lsp-mode	32
6	Partial Linking In C	33

7	How I wrote MeeTTY	37
7.1	How we came up with the idea about MeeTTY	38
7.2	How I convinced Tmux developer	42
7.3	How I found one Kiwiirc Bug	44
7.4	How I Isolated Practice Terminal	46
7.5	The Disaster and the Lessons Learnt	48
8	Playing with reStructuredText	51
8.1	this is one internal section	54
9	Howto remap keyboard keys in Linux	57
10	howto change font in linux using fontconfig	59
11	yield in vala and gio concurrency	61
11.1	Gio Async Framework	61
11.2	Vala Async Framework	61
12	Setup Gnome Development Environment in Archlinux	63
12.1	Setup Base	63
12.2	Jhbuild	63
12.3	Compiling	65
12.4	Installing missing Dependencies	65
13	gistnotes - a gnome shell extension	67
14	Issue with Sabayon	69
15	GnomeContinuous using systemd-nspawn	71
16	SystemCtl Restart MyBlog	77
17	Fetchmail	79
18	Dovecot (with mbox)	81
19	Postfix (with smtp.gmail.com as relayhost)	85
20	Virtual Home Network (with Dnsmasq , Bridge, TUN/TAP, Qemu)	91
21	Dnsmasq for Home User	95
22	GNU Gettext - Yet Another Tutorial	99
23	pcapstreamer - A packet dumper	105
24	Systemd for Simple Backup	109
25	Creating Fedora DVD Repo	113
26	Howto bake Persistent Live USB - Debian Way (live-build)	115
27	Milana (2007) - From a Tamil Fan	119
28	Reliance Netconnect Broadband+ on Debian	121
29	Capture screen and mic input using FFmpeg and ALSA	127

30 Aircel GPRS with Linux and Bluetooth	129
31 Wireless Ad-Hoc Network - Linux	135
32 Support for nVidia GeForce graphics card - Debian - Testing	139
33 Support for Broadcom Wireless Devices - Debian - Testing	141
34 Sakleshpur Trekking - A thrilling story	143
35 Howto switch from Ubuntu to Debian - My Experience - Part II	157
36 Howto switch from Ubuntu to Debian - My Experience - Part I	159
37 Starting Another X Server with Different Locale - Ubuntu	161
38 Qemu Networking (Tun/Tap)	163
39 Share your Qemu Guest with your Friends	165
40 Upstart, update-identica and my blog	167
41 ldd2chroot - A script to extract packages	169
42 urlencoder	173
43 Functions from header files	175
44 ‘localtime()’ loves ‘05:30:00’	177
45 Dharmastala - Where you see yourself	179
46 UTF-8 codes in Terminal	181
46.1 unicode2utf8.bash	181
46.2 unicodes.bash	182
47 Howto add license to sourcefiles automatically(vi)	185
48 Mail with Attachments (Unix shellsript)	187
49 FossConf2009(Updated)	191
50 Kesavan’s Birthday with ffmpeg and mencoder	193
51 Thunderbird pet package	195
52 Fossconf2009	197
53 Indian Tamil Movie All BGM	199
54 createpet script	201
55 Mounam Pesiyadhe BGM	203
56 divx_411 on Puppylinux Full install	205
57 About Me	207
Bibliography	209

CREATING TA-REMINGTON.MIM FOR M17N-DB

Updated: May 07, 2024

1.1 Problem

One fine day, I was deep in my office work, Suddenly I got a ping from [shrini](#)¹ in [#ilugc](#)² asking me to take a look at this [issue](#)³. At first, I thought well, it is a simple change, why we need to write a whole new layout for this simple change. But then [shrini](#) insisted that even if it is a simple change, People who were not able to do that simple change are looking for a solution for long time. So, I said to myself, if that is the case, then lets create one.

1.2 Learning m17n-db format

I was not interested on just copying already available mim files and simply modify for our purpose. I wanted to learn what exactly the format is, so I learnt how the parser parses mim through this [link](#)⁴. I studied and understood what each and every word in that file represents.

1.3 Writing ta-remington.mim

Creating this file was fairly easy, I just took rules from [Ekalappai](#)⁵ and a simple M-x query-replace-regexp in my [emacs](#)⁶ did 80% of the job, after that, I just added headers and tail.

```
M-x query-replace-regexp
Query-replace-regexp: ^\([^[:space:]]+\)[[:space:]]+\([^[:space:]]+\)^J
with: ("1" "2")
```

¹ <https://goinggnu.wordpress.com/>

² <https://meet.ilugc.in>

³ <https://github.com/KaniyamFoundation/ProjectIdeas/issues/160>

⁴ <https://www.nongnu.org/m17n/manual-en/m17nDBFormat.html>

⁵ <https://github.com/thamizha/ekalappai/blob/master/installer/app/keyboards/Tamil-typewriter.txt.in>

⁶ <https://www.gnu.org/software/emacs/>

1.4 Testing the new Layout

I gave my changes to `shrini` to see if he can use it, I think he did the basic test and confirmed me that it is working as expected. Then he forwarded my changes to the original requester who wanted this layout. Thankfully, he also was satisfied with my changes.

1.5 Creating Icon

In order to upstream `ta-remington.mim`, we also have to create one icon file for this new layout. With my limited knowledge in gimp, I was able to create one png file for `ta-remington.mim`.

1.6 Pushing to m17n-db

With my changes, I sent a [mail](#)⁷ asking them to include `ta-remington.mim` in [m17n-db mailing list](#)⁸. After few days I got a response that m17n-db [included my changes](#)⁹.

⁷ <https://lists.nongnu.org/archive/html/m17n-list/2021-06/msg00000.html>

⁸ <https://lists.nongnu.org/mailman/listinfo/m17n-list>

⁹ <https://lists.nongnu.org/archive/html/m17n-list/2021-07/msg00003.html>

RECONFIGURING KEYBOARD KEYS WITH XKB

Updated: May 07, 2024

Recently two keys in my laptop keyboard went off, none of my effort successfully bought back those two keys. So, I was thinking what should I do now, here are the list of things I came up

- Replace the keyboard.
- Replace the scan-code of two unused keys through `hwdb`¹.
- Add the missing keys to one already working key as `Level3` key.

The first one is what we normally do, eventually, I will also going to do if I have chance to buy replacement for my laptop keyboard. In the mean time, I need to have a temporary workaround. The second one will make my keyboard loose two other keys instead of the already lost keys. So, I opted for third one.

2.1 Scan Codes

When you press and release a key in your keyboard, it generates `scan codes`². Usually these are two bytes, the first byte gets generated when you press and the second byte get generated when you release flipping the most significant bit. For example, when you press and release `ESC` key, it generates `0x01 0x81` bytes, the first byte is `00000001` which got generated when you pressed the `ESC` key, the second byte is `10000001` which got generated when you released `ESC` key. You can get the scan codes of each key through `showkey -s`³ command.

2.2 Kernel Keycodes

When `kernel`⁴ receives these `scan codes` from keyboard event, the input driver converts these scan codes to `kernel keycodes`⁵. These keycodes are defined under `/usr/include/linux/input-event-codes.h` headers. Kernel maintains one scancode-to-keycode table, you can map any scan-code to any keycode using `setkeycodes`⁶ command. To view the current scancode-to-keycode table, you can use `getkeycodes`⁷ command.

¹ <https://www.freedesktop.org/software/systemd/man/hwdb.html>

² <https://www.win.tue.nl/~aeb/linux/kbd/scancodes-1.html>

³ <https://man.archlinux.org/man/showkey.1#OPTIONS>

⁴ <https://kernel.org>

⁵ <https://github.com/torvalds/linux/blob/master/include/uapi/linux/input-event-codes.h>

⁶ <https://man.archlinux.org/man/setkeycodes.8>

⁷ <https://man.archlinux.org/man/getkeycodes.8>

2.3 Console Keymaps

The kernel's scancode-to-keycode table is not enough for the `tty` driver, so it defines another table called `keymaps`⁸. Apart from the normal keys, there are control keys in our keyboard. This `keymaps` table defines what should happen when a control key and normal key pressed. we can use `dumpkeys`⁹ command to see the current `keymaps` table and we can use `loadkeys`¹⁰ command to load a new `keymaps` table (these `keymaps` table definitions are available under `/usr/share/keymaps` directory). Here is one simple example to to load custom keymap

```
$ # this commands are run from tty3 (you need to press ctrl+alt+f3 to switch to tty3)
$ gzip /usr/share/kbd/keymaps/i386/qwerty/us.map.gz | sed '/ 51 =/s/comma/less/g;/ 51\
→=/s/less/comma/g' | sudo tee /usr/share/kbd/keymaps/i386/qwerty/test.map
$ echo "compose 'h' 'a' to U+0B95" >> /usr/share/keymaps/i386/qwerty/test.map
$ loadkeys test
$ # if you press , key now, it will come as < key. if you press shift + , keys (<
→key) it will come as , key.
$ # if you press left-alt + right-alt, compose will be activated.
$ # With compose activated, if you press 'h' and 'a', then tty driver will generate
$ # (unicode = 0B95, utf-8 = 0xe0 0xae 0x95) in the console.
$ loadkeys us
$ # switched back to default us keymap
```

Most of `Keymaps` table contains “<keycode> = <level0 keysym> <level1 keysym> ... <level256 keysym>” lines. Here <keycode> represents the kernel keycodes from scancode-to-keycodes table, <level0 keysym> means what character to use when that particular keycode key got pressed, <level1 keysym> means what character to use when Shift modifier is active and keycode key got pressed. Shift modifier will get activated when you press Shift key. There are 9 modifier keys available and each modifier carries different weight,

1. Shift (weight = 1)
2. AltGr (weight = 2)
3. Control (weight = 4)
4. Alt (weight = 8)
5. ShiftL (weight = 16)
6. ShiftR (weight = 32)
7. CtrlL (weight = 64)
8. CtrlR (weight = 128)
9. CapsShift (weight = 256)

With the help of these modifiers, we can choose the `level`, the sum of weight of all the activated modifiers decides the level. For example, lets assume the current activated keymap have a line like this,

```
keycode 30 = a b c d e f g h i j k l m n o p q r s t u v w x y z
```

When we press Control + Shift + a, then the level will be 5 (weight of Control is 4, weight of Shift is 1, so the sum of weight of currently activated modifiers is 5, so the current level is 5), thus, f (which is in the level5 position) is the final keysym, so f will be shown in the console.

When we simply press a, then the level will be 0 (there is no modifiers currently active, so the current level is 0), thus a (which is in the level0 position) is the final keysym, so a will be shown in the console.

⁸ <https://man.archlinux.org/man/keymaps.5>

⁹ <https://man.archlinux.org/man/dumpkeys.1>

¹⁰ <https://man.archlinux.org/man/loadkeys.1>

Note: the command `dumpkeys -f` will not show all the levels of a particular keycode, the first line of the output may say `keymaps 0-2,4-6,8-9,12` which means, the output contains only the levels `level0, level1, level2, level4, level5, level6, level8, level9, level12`. So make sure you interpret the `dumpkeys -f` output correctly.

2.3.1 Compose Key

Apart from keycode assignment, you can add compose key sequences in `keymaps`, For example, lets assume the current activated keymap have a line like this,

```
compose 'h' 'a' to U+0B95
```

When we press `leftalt + rightalt`, the compose modifier gets activated (`Alt + AltGr` which is the default key combo to activate compose). After the compose modifier activated, if you press 'h' key and 'a' key, then you will get 'ஃ' in the console.

2.4 XKB

XKB¹¹ is another table similar to `keymaps` but for **Xorg**¹² and also used by **Wayland**¹³. XKB does not use kernel keycodes directly like `keymaps` table, but converts the kernel keycodes to `xkeycodes` (`<kernel-keycode> + 8 = <xkeycode>`). Again, these `xkeycodes` are mapped into `xkeys`. These `xkeycodes` to `xkeys` mapping is defined under `/usr/share/X11/xkb/keycodes` directory. These `xkeys` are then used in symbol tables. These symbol tables are available under `/usr/share/X11/xkb/symbols` directory. Each symbol table contains lines similar to this,

```
// example symbol table saved as /usr/share/X11/xkb/symbols/demo
xkb_symbols "demo" {
    key <AC01> {
        type[Group1] = "TWO_LEVEL",
        type[Group2] = "FOUR_LEVEL",
        symbols[Group1] = [ 'a', 'A' ],
        symbols[Group2] = [ 'a', 'A', 'b', 'B' ]
    };
    include "level3(ralt_switch)"
    include "group(win_space_toggle)"
    include "compose(menu_altgr)"
}
```

XKB contains 8 groups and 256 levels for each group, so you can load 2048 keysyms on a single xkey. The above symbol table only defines Two Groups, The first Group `Group1` type is `TWO_LEVEL`, which means, it has only `Level1` and `Level2`.

Note: XKB levels starts from `level1` instead of `level0` like in `keymaps` table. When there is no modifiers currently active, XKB chooses `level1` keysym but `keymaps` table chooses `level0` keysym.

The second Group `Group2` have `FOUR_LEVEL`, means, `Level1, Level2, Level3, Level4`. So, the above table totally loads 6 keysyms to `<AC01>` xkey ('A' key in keyboard).

¹¹ https://en.wikipedia.org/wiki/X_keyboard_extension

¹² <https://www.x.org/wiki/>

¹³ <https://wayland.freedesktop.org/>

By default, Group1 will be active, so, when we press 'a' key in keyboard, Level1 will be selected and the keysym a will be used, because there is no modifier currently active.

When we press Shift + a key in keyboard, Level2 will be selected and the keysym A will be used, this is because we used `type[Group1] = "TWO_LEVEL"`, this TWO_LEVEL type defines which modifier enables which Level. These type definitions are inside `/usr/share/X11/xkb/types` directory. If you look at `/usr/share/X11/xkb/types/basic` file, for TWO_LEVEL, you can see that `map[Shift] = Level2` line, which means, Shift modifier will enable Level2.

To use the keysyms in Group2, we have to switch to Group2 by pressing Win + Space keys (The above symbol table includes `group(win_space_toggle)`, the definition of `win_space_toggle` is in `/usr/share/X11/xkb/symbols/group` file). Assume that you successfully switched to Group2.

Now, when you press RightAlt + a key in keyboard, Level3 will be selected and keysym b will be used. because we used `type[Group2] = "FOUR_LEVEL"`, this FOUR_LEVEL type definition is inside `/usr/share/X11/xkb/types/extra` file. If you look at this file, for FOUR_LEVEL, you can see `map[LevelThree] = Level3`, which means, LevelThree modifier will enable Level3. Also, you can see `map[Shift+LevelThree] = level4`, which means, Pressing both Shift modifier and LevelThree modifier will enable Level4. The above symbol table includes `level3(ralt_switch)`, the definition of `ralt_switch` is in `/usr/share/X11/xkb/symbols/level3` file. If you look at that definition, it will say that `<RALT>` will generate ISO_Level3_Shift keysym. This ISO_Level3_Shift keysym interpretation is defined in `/usr/share/X11/xkb/compat/iso9995` file, In this file, you can see that ISO_Level3_Shift sets `Modifier=LevelThree`.

When you press Shift + RightAlt + a key in keyboard, Level4 will be selected and keysym B will be used. We already know from the previous paragraph that FOUR_LEVEL type defines Shift + RightAlt selects Level4 level.

2.4.1 Multi Key

Just like keymaps table's Compose Key, XKB also have facility to Compose multiple keys to generate a single keysym. We have to first activate Compose modifier by pressing Menu + AltGr keys, because in the above table, we used `compose(menu_altgr)`, the definition of `menu_altgr` is inside `/usr/share/X11/xkb/symbols/compose` file. If you look at the definition, MENU Key with AltGr sets Multi_key modifier. This modifier enables Compose facility. The key combinations for Compose are under `/usr/share/X11/locale/<locale>/Compose` file, here `<locale>` is the LANG code. If you look at Compose file for en_US.UTF-8 (`/usr/share/X11/locale/en_US.UTF-8/Compose` file), you can see that pressing o and c will produce © character. So, Pressing MENU + Altgr o c will produce © unicode character.

2.5 Modifications for my broken Keyboard

After learning all these things, I finally have this override saved in `/usr/share/X11/xkb/symbols/local` file.

```
partial alphanumeric_keys modifier_keys
xkb_symbols "override" {
    key <I151> {
        type[Group1] = "TWO_LEVEL",
        symbols[Group1] = [ ISO_Level3_Latch ]
    };
    key <AB03> {
        type[Group1] = "THREE_LEVEL",
        symbols[Group1] = [ c, C, Multi_key ]
    };
    key <AC06> {
        type[Group1] = "THREE_LEVEL",
```

(continues on next page)

(continued from previous page)

```

    symbols[Group1] = [ h, H, Left ]
};
key <AC07> {
    type[Group1] = "THREE_LEVEL",
    symbols[Group1] = [ j, J, Down ]
};
key <AC08> {
    type[Group1] = "THREE_LEVEL",
    symbols[Group1] = [ k, K, Up ]
};
key <AC09> {
    type[Group1] = "THREE_LEVEL",
    symbols[Group1] = [ l, L, Right ]
};
key <AC10> {
    type[Group1] = "FOUR_LEVEL",
    symbols[Group1] = [ semicolon, colon, apostrophe, quotedbl ]
};
};
};

```

The meaning of these overrides are as follows

- Pressing Fn activates ISO_Level3_Latch, Latch means, you don't have to keep on pressing the key, one press is enough. So, once I press and release Fn key, LevelThree modifier will be active.
- Pressing Fn c activates Multi_key.
- Fn h will generate Left Arrow keysym.
- Fn j will generate Down Arrow keysym.
- Fn k will generate Up Arrow keysym.
- Fn l will generate Right Arrow keysym.
- Fn ; will generate apostrophe keysym (single quote).
- Fn Shift + ; will generate quotedbl keysym (double quote).

I have to enable my override with the following steps,

- add local:override to /usr/share/X11/xkb/rules/evdev under !option
- enable 'local:override' to 'xkb-options' under gsettings

```
$ gsettings set org.gnome.desktop.input-sources xkb-options "['local:override']"
```

There are few tools which helps to write XKB configuration files, Here are the few ones which I know

setxkbmap¹⁴

Used to compile xkb configuration files and generate data to load into Xorg Server process

xkbcomp¹⁵

Used to get set xkb configuration data from/to Xorg Server process

xev¹⁶

Used to show xkey codes and xkeysyms when we press any in keyboard

xkbcli¹⁷

Used to show details about each key press in keyboard (like xev)

evtest¹⁸

Used to attach with input device and show events from that device

libinput¹⁹

Used to attach with input device and show events from that input device

2.6 Drawback of XKB

All the custom modification works as long as we are working locally, if we try to use any remote application, or a VM, our XKB customization will not work, because those remote apps directly sends scan-codes to the remote location rather than the generated XKB keysyms.

MODERNISING IRC

Updated: May 07, 2024

IRC Protocol

IRC¹ is one of the oldest protocol from the [Dinosaur Age](https://en.wikipedia.org/wiki/Dinosaur_Age)² which is still alive. It is very simple, there is one IRC Server running somewhere in the Internet. You connect to that server using an IRC Client, join one Channel (in other words a Room), and start chatting with people in that channel.

Limitations

Since IRC is very simple text based protocol, it doesn't have a mechanism to store the history, means, if you disconnect from the server, you no longer receive messages from other people in the Channel. In Order to receive all the messages from a particular Channel, your IRC Client need to stay connected with the IRC Server.

Also, we cannot watch what is happening in the Channel all the day. So, the IRC Client should have the ability to provide notification so that when you see the notification, you can decide whether you want to switch look at the conversation happening in the Channel or not. Most of the IRC Clients developed for Desktop Operating Systems do this notification without any issue. Also, they can stay connected with the IRC Server as long as you keep them running.

IRC requires the client to be connected to the server through TCP Socket, but modern mobile operating systems don't like apps which keep on running and stay connected through TCP Socket, these modern mobile operating systems want the apps to use [Push Notification](https://firebase.google.com/docs/cloud-messaging)³ to save battery, these mobile operating systems will kill the IRC Client process if they think that the process is not doing anything. Thus, IRC Clients developed for Mobile Operating Systems cannot effectively stay connected forever with IRC Server through TCP Socket.

3.1 WeeChat

WeeChat⁴ is a terminal based IRC Client. Just like any other IRC Client, WeeChat provides all the functionality of an IRC Client and also provides functionalities more then the normal ordinary IRC Client. We can extend WeeChat through plugins. Plugins can be written using C, Python, Perl etc. Infact, most of the core functionalities are implemented as plugins in WeeChat.

¹ https://en.wikipedia.org/wiki/Internet_Relay_Chat

² <https://en.wikipedia.org/wiki/IRCd#History>

³ <https://firebase.google.com/docs/cloud-messaging>

⁴ <https://weechat.org/>

3.2 WeeChat Relay

One of the plugin in WeeChat is called Relay. This plugin can act as a proxy so that you can use another IRC Client to connect to WeeChat and WeeChat will connect to the actual IRC Server. This plugin also provide another type of communication using [WeeChat Protocol](#)⁵, we can use this protocol and write a WeeChat Client which act like WeeChat but without using IRC Protocol.

3.3 WeeChat Android

[WeeChat Android](#)⁶ is a WeeChat Client for [Android](#)⁷ which communicates with WeeChat running in a remote machine through WeeChat Protocol. WeeChat Android acts like WeeChat so we get a IRC Client for Android. WeeChat Android provides all the functionality of WeeChat but unfortunately it also suffers from the battery optimization in the Android operating system. It cannot stay connected to the remote WeeChat all the time. Also, WeeChat Android not able to use Push Notification because of the [Proprietary Infrastructure](#)^{Page 9, 3} those Push Notifications depends on.

3.4 IrssiNotifier

[IrssiNotifier](#)⁸ is one Android App which uses Push Notification to receive messages from a remote server. The IrssiNotifier Servers which send those Push Notification through the Proprietary Infrastructure are open for any valid clients to send those messages from a remote server.

We have to first create one client through IrssiNotifier website. Using the Client Id, we can develop one WeeChat plugin. This plugin will catch all the notification from WeeChat and send those notification messages to IrssiNotifier Servers. Those IrssiNotifier Servers then send those notification messages to IrssiNotifier Android App in our Android mobile. In this way, we can receive all the notification from WeeChat in our Mobile.

3.5 WeeNotifier

[WeeNotifier](#)⁹ is a plugin for WeeChat which uses IrssiNotifier Infrastructure to broadcast notifications from WeeChat. With the combined use of WeeChat, WeeChat Android and WeeNotifier, we can emulate a Modern Messaging System on top of IRC Protocol

⁵ https://weechat.org/files/doc/stable/weechat_relay_protocol.en.html

⁶ <https://github.com/ubergeek42/weechat-android>

⁷ https://www.android.com/intl/en_in/

⁸ <https://irssinotifier.appspot.com/>

⁹ <https://gitlab.com/mohan43u/weenotifier>

3.6 Steps to Modernize IRC

These are the methods I used to improve my IRC experience on par with any Modern Messaging System especially in Android.

- Lease one very small cloud VM to run WeeChat continuously. You can also use your Desktop/Laptop to run WeeChat, but we need some way for WeeChat Android running in the mobile to connect with WeeChat running in your Desktop/Laptop. You can ask your Internet Service Provider to provide a Static IP or you can use Dynamic DNS method to make your Desktop/Laptop available through Internet. In the end, all we need is to somehow make WeeChat Android connect to WeeChat.
- Keep WeeChat running in that small cloud VM continuously.
- Configure WeeChat to generate notification for all the messages of a particular Channel.
- [Configure WeeChat Relay¹⁰](#) in WeeChat so that WeeChat Android can connect securely.
- Install WeeChat Android App in your Mobile and configure it to connect to your WeeChat running in the small cloud VM.
- Create an account in IrssiNotifier Infrastructure through [their website¹¹](#). You need to generate Client Id to use it in WeeNotifier plugin as well as in IrssiNotifier Android App.
- [Install WeeNotifier¹²](#) plugin in WeeChat to send notification to IrssiNotifier Servers. Configure it so that it is able to send data to the servers.
- Install IrssiNotifier App in your Mobile and configure it to connect to the IrssiNotifier Servers. If you want, you can change the encryption key to a different one from the default. In this case, you have to also configure WeeNotifier plugin in WeeChat with the same encryption key.
- That's all. When WeeChat generate notifications, it will be captured by WeeNotifier and sent to IrssiNotifier Servers, then IrssiNotifier Servers will push those notifications to IrssiNotifier App in your mobile, Once you see the notification, you can then open WeeChat Android App in your mobile to connect back to WeeChat to see whats going in the IRC Channel. Since, WeeChat is running continuously connected to actual IRC Server, the history is preserved.

3.7 Why I wrote WeeNotifier

When I was looking for some way to do push notifications from WeeChat, I came to know about IrssiNotifier and its Infrastructure. Immediately, I found that there is already one plugin available in WeeChat called IrssiNotifier which does exactly what I wanted. But when I tried, It didn't work, I looked it to fix it, then I saw that the plugin invokes openssh for every message to do encryption. I'm not agree to do this way. Invoking a subprocess for each and every message is not a good thing.

So, I decided to write my own plugin. I know that WeeChat will not accept competing plugins for same functionality, but I don't care whether WeeChat accepts or not. It is my own Itch which I scratched. I'm now happy with the result. I [submitted¹³](#) my plugin to WeeChat anyway.

¹⁰ https://weechat.org/files/doc/stable/weechat_user.en.html#relay_plugin

¹¹ <https://irssinotifier.appspot.com/#profile>

¹² <https://gitlab.com/mohan43u/weenotifier#setup>

¹³ <https://github.com/weechat/scripts/pull/466>

MIGRATED MY WORDPRESS BLOG TO GITLAB

Updated: May 07, 2024

Recently I moved my [blog](https://mohan43u.wordpress.com)¹ from [WordPress](https://wordpress.com)² to [GitLab](https://gitlab.com)³, you might be asking ‘wait, what? what do you mean by moved to GitLab? when GitLab started providing blogging?’, GitLab don’t provide blogs, but we can use GitLab to host a static website like [GitHub Pages](https://about.gitlab.com/stages-devops-lifecycle/pages/)⁴. So, this is how I converted my WordPress blog to a blog like static website and hosted in GitLab.

¹ <https://mohan43u.wordpress.com>

² <https://wordpress.com>

³ <https://gitlab.com>

⁴ <https://about.gitlab.com/stages-devops-lifecycle/pages/>

4.1 Inspiration

Updated: May 07, 2024

I started working on improving [ILUGC](https://ilugc.in/about/)¹'s online presence after [COVID](https://en.wikipedia.org/wiki/COVID-19)² started. We renewed ILUGC's [irc](https://en.wikipedia.org/wiki/Internet_Relay_Chat)³ channel in [Freenode](https://freenode.net/)⁴ after a long time (but again moved to [Libera.chat](https://libera.chat)⁵, that is another story), it was created around 2009 but nobody cared. During channel registration, freenode stuff asked me does ILUGC have any online presence. I provided ILUGC's [website](https://ilugc.in)⁶, they asked me to add one [TXT](https://en.wikipedia.org/wiki/TXT_record)⁷ record to that website's domain.

At that time, I dint know who owned ILUGC's website, but know that someone from ILUGC community retrived and restored that expired domain few years back. I enquired about that with [shrini](https://goinggnu.wordpress.com/)⁸ and came to know that [cnu](https://cnu.name/)⁹ is the person who retrived and restored that website. So I sent a mail to [cnu](https://cnu.name/) asking to add [TXT](https://en.wikipedia.org/wiki/TXT_record) record which freenode sent to me. He added it immediately.

Joined ILUGC website development

Then [cnu](https://cnu.name/) added me as one of the member of [ILUGC project](https://github.com/ilugc)¹⁰ in [GitHub](https://github.com). I came to know that ILUGC's website is actually a [Hugo](https://gohugo.io/)¹¹ powered static website. I tried to write some article for that website and understood how that website is hosted with the help of [Netlify](https://www.netlify.com/)¹². I wrote one [article](https://ilugc.in/get-involved/)¹³ explaining how one can contribute to ILUGC's website from the knowledge I gained.

ILUGC's website and the way it was hosted without running a webserver inspired me to move my wordpress hosted blog to a static website.

To be continued..

¹ <https://ilugc.in/about/>

² <https://en.wikipedia.org/wiki/COVID-19>

³ https://en.wikipedia.org/wiki/Internet_Relay_Chat

⁴ <https://freenode.net/>

⁵ <https://libera.chat>

⁶ <https://ilugc.in>

⁷ https://en.wikipedia.org/wiki/TXT_record

⁸ <https://goinggnu.wordpress.com/>

⁹ <https://cnu.name/>

¹⁰ <https://github.com/ilugc>

¹¹ <https://gohugo.io/>

¹² <https://www.netlify.com/>

¹³ <https://ilugc.in/get-involved/>

4.2 Website Design

Updated: May 07, 2024

4.2.1 Choosing Markup Language

From my experience contributing to [ILUGC website](https://ilugc.in)¹, I come know that [Hugo](https://gohugo.io/)² uses `Markdown` which is the most popular modern markup language in internet right now. I don't know why but I was hesitant to go with `Markdown`, I searched what are the options we have. So, here are some ways to generate static html pages,

- [Markdown](https://commonmark.org/)³
- [AsciiDoc](https://asciidoc.org/)⁴
- [reStructuredText](https://docutils.sourceforge.io/rst.html)⁵

I'm not going to compare them, because I don't know the benefits/drawbacks of each and everyone. So, I'm just giving what I felt when I tried few.

`Markdown`

Popular format used in everywhere. Very simple to write, literally every static site generator support `Markdown`. But cross referencing is pain, syntax highlighting is lacking compared to other systems, adding caption to an image is not available and there are few more drawbacks people say in [internet](#)⁶.

`reStructuredText`

Not as popular as `Markdown` and not as easy as `Markdown`. but we can pickup easily. One of the amazing thing is to add codes and do syntax highlighting. I use lot of syntax highlighting in my blog articles. So, when I tried `reStructuredText` I immediately fell in love for this one amazing functionality. No wonder why `reStructuredText` is used by [python](https://docs.python.org)⁷, [kernel](https://www.kernel.org/doc/html/latest/)⁸ organizations for documentation.

Once I became comfortable with `reStructuredText` , I decided to go with [Sphinx](https://www.sphinx-doc.org/en/master/)⁹ with its default [Alabaster](https://alabaster.readthedocs.io/en/latest/)¹⁰ theme

4.2.2 RSS Feed

For a blog, having a [RSS](#)¹¹ feed is very important so that people who are interested in our blog will be able to get updates. Thankfully, `Sphinx` already have extension to generate an RSS feed file called `yasfb`¹²,

so I simply used `yasfb` and integrated the output into by sidebar.

¹ <https://ilugc.in>

² <https://gohugo.io/>

³ <https://commonmark.org/>

⁴ <https://asciidoc.org/>

⁵ <https://docutils.sourceforge.io/rst.html>

⁶ <http://ericholscher.com/blog/2016/mar/15/dont-use-markdown-for-technical-docs/>

⁷ <https://docs.python.org>

⁸ <https://www.kernel.org/doc/html/latest/>

⁹ <https://www.sphinx-doc.org/en/master/>

¹⁰ <https://alabaster.readthedocs.io/en/latest/>

¹¹ <https://en.wikipedia.org/wiki/RSS>

¹² <https://github.com/sphinx-contrib/yasfb>

4.2.3 Comment System

Comments and replies are important part of a Blog. But, to write a comment in a blog post, we need to have a server to accept that comment and tag that comment with the blog post. This is impossible in a statically generated website. So, people who use static blogs choose few methods I will explain now

No comment system

Static blog owners will go with this method. They will add their blog post to either [Hacker News](#)¹³ or [Reddit](#)¹⁴ and link back those thread links into their blog post.

Comment System as Service

There are providers like [Disqus](#)¹⁵, [Facebook Comments](#)¹⁶ who basically will provide an embedded widget once you create account in their system. We have to embed that html widget into every article we write. This method means you have to trust the service provider.

Self Hosted Comment System

Instead of using some service like [Disqus](#), we can also host our own Comment System like [Commento](#)¹⁷ or [Isso](#)¹⁸. This also requires embedding one widget in every article, but instead of communicating with some third-party provider, we can use our server with our own hosted comment system.

GitHub/GitLab issues as Comment System

With the help of [GitHub](#)¹⁹ or [GitLab](#)²⁰ api, we can use GitHub/GitLab issues as comment system. Here also, we have to use one embedded HTML widget. But, instead of communicating to a third-party provider or some server which contains self hosted comment system, we use GitHub/GitLab api and use the issue system in GitHub/GitLab as comments. These type of widgets uses [React](#)²¹ or [Vue](#)²² like Front End frameworks. Few such systems are [Gittalk](#)²³ or [Gitment](#)²⁴.

Vssue

[Gittalk](#) or [Gitment](#) only works with GitHub, but I found [Vssue](#)²⁵ which works with following websites

- [GitHub](#)^{Page 16, 19}
- [GitLab](#)²⁰
- [GitTea](#)²⁶
- [BitBucket](#)²⁷

As I'm planning to use GitLab to host my static website. I selected Vssue as my comment system

To be continued..

¹³ <https://news.ycombinator.com/>

¹⁴ <https://www.reddit.com/>

¹⁵ <https://disqus.com>

¹⁶ <https://developers.facebook.com/docs/plugins/comments/>

¹⁷ <https://commento.io>

¹⁸ <https://positiv.org/isso/>

¹⁹ <https://github.com>

²⁰ <https://gitlab.com>

²¹ <https://reactjs.org>

²² <https://vuejs.org>

²³ <https://github.com/gitalk/gitalk>

²⁴ <https://github.com/imsun/gitment>

²⁵ <https://vssue.js.org>

²⁶ <https://gitea.io/en-us/>

²⁷ <https://bitbucket.org/product>

4.3 Fixed Vssue GitLab Integration

Updated: May 07, 2024

Vssue¹ is a comment system which uses GitLab² issues as comments. After I decided to use it. I faced one issue which very much become a blocker issue.

4.3.1 GitLab's Redirect URL Limitation

Basically, GitLab's OAuth2 `redirect_url` will not allow any wildcard characters. So, you cannot use a `redirect_url` like <https://mohan43u.space/>, you have to provide full url like <https://mohan43u.space/path/to/article.html>. For static websites, this means, we have to add each and every article's path into GitLab which is impossible. The author of Vssue raised this [issue](#)³ with GitLab but due to security concerns, they are not going to fix this. Without fixing this issue, I cannot use Vssue with my static site.

I was frustrated. I thought I should just forget about the comment system entirely, but somehow decided to see what actually happening inside Vssue when OAuth2 takes place. Soon, I found that there is a `parameter`⁴ in the Authorize call which we can utilize to wrap the current url. When the redirect takes place and response comes from GitLab, we check for this `parameter` and then unwrap to see the original url which requested OAuth2. Thus, we can again do local redirect to that original url.

4.3.2 Pull Request for Vssue

Vssue is written in [Vue.js](#)⁵. Without properly understanding how to create PR for Vssue, I just added my fix and created a pull request. But my pull request failed in CI, then I came to know that I have to write proper test case for my change.

4.3.3 Writing Test case in Vssue

Vssue uses [jest](#)⁶ as its test framework. Since my pull request failed in CI, I have to write proper test case for my change. The actual fix is basically very simple, but writing test case become difficult for me. Every time I thought few test cases is enough, the code coverage in CI dont accept because not all code path was tested. So, I have to again and again write test cases to finally make CI succeed for my pull request.

My [pull request](#)⁷ is still waiting for approval in Vssue but I'm already using it in my static website. People also tried my changes in their static websites and it seems it is working

To be continued..

¹ <https://vssue.js.org>

² <https://gitlab.com>

³ <https://gitlab.com/gitlab-org/gitlab/-/issues/23054>

⁴ <https://docs.gitlab.com/ce/api/oauth2.html#authorization-code-flow>

⁵ <https://vuejs.org/>

⁶ <https://jestjs.io/>

⁷ <https://github.com/meteorlxy/vssue/pull/101>

4.4 ModifiedDate Extension for Sphinx

Updated: May 07, 2024

4.4.1 Idea Behind the Extension

When writing article in [Hugo](https://gohugo.io/)¹ you have to provide some date in `date` field so that Hugo can show it in the statically generated website. Every time you update one article, you have to manually update this field.

I was looking for some way to generate this date automatically in [Sphinx](https://www.sphinx-doc.org/en/master/usage/restructuredtext/roles.html#substitutions)². There is a way to add current date using `|today|`³ substitution. I was looking for similar kind of substitution which is capable of automatically fetching the current file's modified date. But unfortunately, I'm not able to find any extension for Sphinx which satisfy my need. So, I decided why not write one?

4.4.2 ModifiedDate Extension

So, I looked into the source code of Sphinx which handles `|today|` [substitution](https://www.sphinx-doc.org/en/master/usage/restructuredtext/roles.html#substitutions)⁴, after understanding that code, I started implementing my own extension to Sphinx and came up with my own substitution called `|modifieddate|`. Here is my extension,

```
"""modifieddate.

This extension provides `|modifieddate|` substitution

.. moduleauthor:: Mohan R <mohan43u@gmail.com>
"""

import os
import datetime
from docutils import nodes
from docutils.transforms import Transform
from sphinx.util.i18n import format_date
from sphinx.locale import _

class ModifiedDateTransform(Transform):
    """ModifiedDateTransform class."""

    default_priority = 210

    def apply(self, **kwargs):
        """Transform modifieddate substitution."""
        for ref in self.document.traverse(nodes.substitution_reference):
            name = ref['refname']
            if name == 'modifieddate':
                source = self.document['source']
                modifieddate = None
                try:
                    modifieddate = os.stat(source).st_mtime
                except Exception as exception:
```

(continues on next page)

¹ <https://gohugo.io/>

² <https://www.sphinx-doc.org/>

³ <https://www.sphinx-doc.org/en/master/usage/restructuredtext/roles.html#substitutions>

⁴ https://github.com/sphinx-doc/sphinx/blob/60203e34a4ea6b5a1e2c9279bfbdeab697305d97e/sphinx/transforms/__init__.py#L114

(continued from previous page)

```

        raise Exception('failed to get modifieddate for %s' % (source))
    ↪from exception
        if modifieddate is not None:
            config = self.document.settings.env.config
            modifieddate_fmt = config.today_fmt or _('%b %d, %Y')
            timezone = datetime.timezone.utc
            modifieddate = datetime.datetime.fromtimestamp(modifieddate,
                                                            timezone)
            text = format_date(modifieddate_fmt, date=modifieddate,
                              language=config.language)
            ref.replace_self(nodes.Text(text))

def setup(app):
    """extension."""
    app.add_transform(ModifiedDateTransform)
    return {
        'version': '0.0.1',
        'parallel_read_safe': True,
        'parallel_write_safe': True
    }

```

4.4.3 Automating Updated Date

Adding `|modifieddate|` substitution is not enough to automate. Every time I update one article, the current modified date in the disk will change, but I need to preserve the timestamp when I create that article. So I wrote one small script which will look at the prefix of each article filename, If the filename contains a timestamp, then it compares the current modified date for that file in the disk with that timestamp from filename, if they are not equal, then the script will update the modified date to the timestamp from the filename. In this way, I make sure the modified date is always preserved to the timestamp when the article was written. Since the modified date in the disk and the timestamp from the filename got synced, `|modifieddate|` substitution always return the timestamp from filename as intended.

If there is no timestamp in the filename, then my script will add the current time to the filename, this will make sure the filename of the article contains the original timestamp when it was created.

the date you see at the top of this article after “Updated:” field comes from `|modifieddate|` substitution

To be continued..

4.5 Exporting Blog posts from WordPress to Sphinx

Updated: May 07, 2024

4.5.1 Taking Backup from WordPress

WordPress¹ provides way to `export`² the entire blog as one single XML file, it also provides way to backup multimedia data in the same page where you take the blog backup. I took both the backups and kept it in my laptop. My entire blog is now in a xml file. I need to parse that file and generate rst files for each blog post. I tried to see if there is any, but I'm not able to find. So, I decided to write one.

4.5.2 xml2rst.py

This is the simplest parser I wrote which only captures the `title`, `pubdate` and `blogpost` and creates rst file. This will not capture comments or tags.

```
#!/usr/bin/env python3

import os
import re
import sys
import time
import calendar
import xml.parsers.expat

item = 0
capturetitle = 0
capturepubdate = 0
capturedata = 0
ctitle = None
cpubdate = None
cdata = None

def start_element(name, attrs):
    global item
    global capturedata
    global capturetitle
    global capturepubdate
    global ctitle
    global cpubdate
    global cdata

    if name == "item":
        item = 1
    if item == 1 and name == "title":
        capturetitle = 1
        ctitle = ""
    if item == 1 and name == "pubDate":
        capturepubdate = 1
        cpubdate = ""
    if item == 1 and name == "content:encoded":
```

(continues on next page)

¹ <https://wordpress.com>

² <https://wordpress.com/support/export/>

(continued from previous page)

```

        capturedata = 1
        cdata = ""

def char_data(data):
    global capturedata
    global capturetitle
    global capturepubdate
    global ctitle
    global cpubdate
    global cdata

    if capturetitle == 1:
        ctitle += data
    if capturepubdate == 1:
        cpubdate += data
    if capturedata == 1:
        cdata += data

def end_element(name):
    global item
    global capturedata
    global capturetitle
    global capturepubdate
    global ctitle
    global cpubdate
    global cdata

    if item == 1 and name == "title":
        capturetitle = 0
    if item == 1 and name == "pubDate":
        capturepubdate = 0
    if item == 1 and name == "content:encoded":
        capturedata = 0

    if name == "item":
        item = 0
        if len(cdata) > 0:
            ptime = time.strptime(cpubdate, '%a, %d %b %Y %H:%M:%S %z')
            ptime_epoch = calendar.timegm(ptime)
            if not os.path.exists(str(ptime.tm_year)):
                os.makedirs(str(ptime.tm_year), 0o755)
            articlefile = ctitle.lower().replace(' ', '_')
            articlefile = re.sub('[^a-zA-Z0-9_]', '', articlefile)
            articlefile = str(ptime.tm_year) + '/' + str(ptime_epoch) + '_' +
→articlefile + '.rst'
            with open(articlefile, 'w') as article:
                article.write('.. title:: ' + ctitle)
                article.write('\n\n%s\n' % (''.join(['*' for n in
→range(len(ctitle))]))
                article.write(ctitle)
                article.write('\n\n%s\n' % (''.join(['*' for n in range(len(ctitle))]))
                article.write('\n| Updated: |modifieddate|\n\n')
                article.write(cdata)
                os.utime(articlefile, (ptime_epoch, ptime_epoch))

def main():

```

(continues on next page)

(continued from previous page)

```
parser = xml.parsers.expat.ParserCreate()
parser.StartElementHandler = start_element
parser.EndElementHandler = end_element
parser.CharacterDataHandler = char_data
parser.Parse(open(sys.argv[1]).read(), 1)

if __name__ == "__main__":
    main()
```

4.5.3 Manually fixing syntax Highlighting

Even though the parser helped generating rst files. I still have to manually fix the source code highlighting. This was boring and repeated work. but I thought instead of spending time to improve the parser, I just go through the generated rst files and fix it by hand, because I don't have a lot of blog posts to fix.

To be continued..

4.6 PDF Integration

Updated: May 07, 2024

Sphinx¹ not only capable of generating HTML, but it can also generate PDF. Instead of doing `make html`, we can run `make latexpdf` to generate pdf document. To generate PDF, Sphinx first converts reStructuredText to tex document then calls any one of tex engine (PdfTeX², XeTeX³ or LuaTeX⁴). You can configure any one of this engine through `latex_engine` config variable in Sphinx's `conf.py`. If you don't specify this variable, `pdftex` will be used by default.

4.6.1 Issue with PdfTeX

PdfTeX, the default TeX engine in Sphinx works great if you are sticking with `ascii` encoding. But, it fails to render unicode fonts properly. There are some workarounds to improve PdfTeX, but I find out that XeTeX by default supports unicode. So I decided to use XeTeX.

4.6.2 Dependencies for XeTeX

We have to make sure the following packages installed in order to use XeTeX in Ubuntu⁵

```
$ sudo apt install texlive-xetex fonts-freefont-otf latexmk xindy
```

4.6.3 Switch Sphinx to use XeTeX

We have to instruct Sphinx to use XeTeX as the default latex engine in `conf.py`

```
latex_engine = 'xelatex'
```

4.6.4 Font Integration in XeTeX

In Linux, we use `fontconfig`⁶ to utilize fonts of different type. But in XeTeX, we have to use `fontspec`⁷ and explicitly mention which font to use. It can be done in Sphinx using `fontpkg` key in `latex_elements` dictionary in `conf.py`

```
latex_elements = {
    'fontpkg': r'''
\setmainfont{FreeSerif}[
    Path = /usr/share/fonts/opentype/freefont/,
    BoldFont = *Bold,
    ItalicFont = *Italic
]
\setsansfont{FreeSans}[
    Path = /usr/share/fonts/opentype/freefont/,
    BoldFont = *Bold,
```

(continues on next page)

¹ <https://http://sphinx-doc.org/>

² <https://www.tug.org/applications/pdftex/>

³ <https://www.tug.org/xetex/>

⁴ <http://www.luaatex.org/>

⁵ <https://ubuntu.com/>

⁶ <https://www.freedesktop.org/wiki/Software/fontconfig/>

⁷ <https://github.com/wspr/fontspect/>

(continued from previous page)

```
    ItalicFont = *Oblique
]
\setmonofont{FreeMono}[
    Path = /usr/share/fonts/opentype/freefont/,
    BoldFont = *Bold,
    ItalicFont = *Oblique
]
```

The generated PDF document is available in the sidebar

To be continued..

4.7 Publishing the Static Website Automatically

Updated: May 07, 2024

4.7.1 GitLab Pages

Just like [GitHub Pages](https://pages.github.com/)¹, [GitLab Pages](https://about.gitlab.com/stages-devops-lifecycle/pages/)² are also available. GitLab Pages are basically a git repository with special `.gitlab-ci.yml` which will push the static html content you have in that repository under `your-name.gitlab.io/projectname`. It is easy to setup³.

4.7.2 Using Gitlab CI to build and publish

There are two choices for me to publish my website

- Build html and pdf locally in my laptop and push the resulted content to gitlab. By using their default GitLab Pages specific `.gitlab-ci.yml`, I can just publish easily
- Write my own `.gitlab-ci.yml` and build both html and pdf content using [Gitlab's CI](https://docs.gitlab.com/ee/ci/)⁴ and publish.

I selected the second way and here is my custom `.gitlab-ci.yml`

```
default:
  image: ubuntu

build0:
  stage: build
  only:
    - master
  script:
    - useradd -d "${CI_PROJECT_DIR}" -m -U -s /bin/bash user0
    - chown -R user0:user0 "${CI_PROJECT_DIR}"
    - export DEBIAN_FRONTEND=noninteractive
    - apt update -y -q && apt full-upgrade -y -q && apt autopurge -y -q && apt clean -
    ↪y
    - apt install -y -q python-is-python3 python3-venv make git texlive-xetex texlive-
    ↪latex-recommended fonts-freefont-otf latexmk xindy
    - su user0 -c 'cd; ./scripts/venvrn.sh make'
  artifacts:
    name: "${CI_PROJECT_NAME}"
    paths:
      - "${CI_PROJECT_NAME}/${CI_PROJECT_NAME}.html.tar.gz"
      - "${CI_PROJECT_NAME}/${CI_PROJECT_NAME}.pdf"

pages:
  stage: deploy
  only:
    - master
  script:
    - cd "${CI_PROJECT_DIR}"
    - mkdir public && tar xvzf "${CI_PROJECT_NAME}/${CI_PROJECT_NAME}.html.tar.gz" -C
    ↪public/
```

(continues on next page)

¹ <https://pages.github.com/>

² <https://about.gitlab.com/stages-devops-lifecycle/pages/>

³ <https://docs.gitlab.com/ee/user/project/pages/>

⁴ <https://docs.gitlab.com/ee/ci/>

(continued from previous page)

```
- cp "${CI_PROJECT_NAME}/${CI_PROJECT_NAME}.pdf" public/_static/misc/
artifacts:
  paths:
    - public
```

As you see, there are three jobs I defined,

- html
- pdf
- pages

html job basically build html pages, pdf job builds pdf document and pages job is job specifically for publishing the contents to GitLab Pages.

4.7.3 Custom Domain

GitLab Pages also supports [Custom Domain](#)⁵ with SSL/TLS automatically generated through [LetsEncrypt](#)⁶. But, There are bit of work to do in our DNS Records. Once we do the required changes in the DNS Records, GitLab will do the rest automatically.

Thus, everything is automated, all I need to do is write blog posts consistently, I hope I'll try to do after this migration, because I feel I really like writing in reStructuredText.

End.

⁵ https://docs.gitlab.com/ee/user/project/pages/custom_domains_ssl_tls_certification/

⁶ <https://letsencrypt.org/>

EMACS LSP FOR C

Updated: May 07, 2024

5.1 LSP (Language Server Protocol)

Language Server Protocol¹ is a standard way of assisting programmers by providing set of features (like on the fly compilation, symbol resolution, references resolution, providing help or documentation etc). This is a client/server model, implementation of this protocol consist of a server process (eg: `clangd`²) parsing the code which is currently written (or already written) and listening for any client to connect with it and ask for details. The client (eg: `lsp-mode`³ in `emacs`⁴) will communicate with the server on the fly and ask for details based on the current context of the code currently written.

`clangd`

This LSP server is provided by `llvm`⁵ project for C/C++. Also, there are other LSP servers for C/C++ available like `ccls`⁶. This `ccls` depend on `libclang`⁷ which is also used by `clangd`, basically `libclang` is the one which parses the source code. Both `ccls` and `clangd` utilize the `libclang` library for code parsing.

`lsp-mode`

This is a emacs package. This acts as a LSP client communicating with different LSP servers based on the current source file. There are additional packages like `lsp-ui`⁸ to provide visual response.

¹ <https://microsoft.github.io/language-server-protocol/>

² <https://clangd.llvm.org/>

³ <https://emacs-lsp.github.io/lsp-mode/>

⁴ <https://www.gnu.org/software/emacs/>

⁵ <https://llvm.org/>

⁶ <https://github.com/MaskRay/ccls>

⁷ <https://clang.llvm.org/docs/Tooling.html>

⁸ <https://emacs-lsp.github.io/lsp-ui/>

```

File Edit Options Buffers Tools C Help
build
resources
umoci
kiwiirc
libfuse
+ .github/ISSUE_TEMPLATE
+ doc
+ example
+ include
- lib
+ modules
buffer.c
fuse_lowlevel.c
fuse.c
fuse.i.h
fuse_log.c
fuse_loop.c
fuse_loop_mt.c
fuse_lowlevel.c
fuse_misc.h
fuse_opt.c
fuse_signals.c
fuse_versionscript

void fuse_unmount(struct fuse *f) {
    fuse_session_unmount(fuse_get_session(f))
}

int fuse_version(void)
{
    return FUSE_VERSION;
}

const char *fuse_pkgversion(void)
{
    return PACKAGE_VERSION;
}

static void helloworld(void) {
    printf
    printf(const char *restrict __format, ...) int (Function)
    obstack_printf(struct obstack *restrict __obstack, const char *restrict __format, ...) int (Function)
}

### function printf
---
→ int
Parameters:
- const char *restrict __format
Write formatted output to stdout.
This function is a possible cancellation point and therefore not marked with __THROW.
---
extern int printf(const char *restrict __format, ...)

Treemacs -UU-:***-F1 fuse.c Bot L5123 Git-master (C/*l company-capf LSP[clangd:5516] EIDoc Abbrev) 21/4
libfuse 21/4 /home/mohan/Devel/notmine/meetty-deps/fuse-overlays
fuse.c 21/4 lib/
+ [clang] Too many errors emitted, stopping now (0:0)
+ [clang] 'config.h' file not found (15:9)
+ [clang] Field has incomplete type 'struct fuse_operations' (60:24)
+ [clang] Unknown type name 'fuse_ino_t' (74:1)
+ [clang] Unknown type name 'fuse_ino_t' (78:1)
+ [clang] Unknown type name 'fuse_ino_t' (118:1)
LSP Errors View
[mohan-moh0:journalctl 1:ssh* 2:top] "mohan@container0:~/De" 81:56 86-Jan-21

```

5.2 Installation

LSP consist of two parts, the LSP server and the LSP client. For different languages, we need to install different LSP servers based on our requirement. On the client side, as we are going to use emacs, we only need to install lsp-mode (additionally lsp-ui) once, this same client can be used with multiple LSP servers for different languages.

5.2.1 Installing LSP Server for C/C++ (clangd)

For the C/C++ Language, we need to install clangd LSP server. This clangd is available on all major distros. You have to install based on your distro. Following command will install clangd in Arch⁹.

```
$ sudo pacman -S clang
```

Installing bear^{Page 30, 10}

Merely installing clangd will not going to provide all the LSP server functionality. You need to tell clangd about how-to parse each source file in your project so that clangd will be able to parse the source code same like the compiler will parse. To know how-to compile a source file, clangd uses [compilation database](https://clangd.lvm.org/config.html)¹¹. This compilation database is nothing but a file called compile_commands.json which contains compilation command lines to compile each source code in the project. The [cmake](https://cmake.org/)¹² can generate this compile_commands.json automatically when -DCMAKE_EXPORT_COMPILE_COMMANDS=ON added to cmake. The [meson](https://mesonbuild.com/)¹⁶ can generate compile_commands.json automatically. But for non cmake or meson projects, bear tool is needed to produce

⁹ <https://archlinux.org/>

¹⁰ <https://github.com/rizotto/Bear>

¹¹ <https://clangd.lvm.org/config.html>

¹² <https://cmake.org/>

¹⁶ <https://mesonbuild.com/>

`compile_commands.json`. This bear tool is available on all major distros. You have to install based on your distro. For Arch, bear is available in [AUR](#)¹³.

5.2.2 Installing LSP Client for emacs (lsp-mode and lsp-ui)

Following emacs packages needed and can be installed through [ELPA](#)²⁰/[MELPA](#)²¹ (`M-x list-packages`).

- `lsp-mode` (core LSP client)
- `lsp-ui` (adds UI to `lsp-mode`)
- `company`¹⁴ (text completion Framework for emacs)

5.3 Configuration

5.3.1 Configuring clangd

For clangd, we need to generate `compile_commands.json` file in project root directory. If you are using `autotools`¹⁵ based projects, instead of running `make`¹⁷, you have to run `make` with `bear` like this,

```
$ bear -- make
```

This will run `make` along with `bear` and records all the compilation commands from `make`, finally `bear` saves all the recorded compilation commands in `compile_commands.json`.

5.3.2 Configuring emacs LSP packages

We need to add the following lines to `.emacs`¹⁸ to configure the LSP packages,

```
;; company
(add-hook 'after-init-hook 'global-company-mode)

;; lsp (language server)
(require 'lsp-mode)
(add-hook 'prog-mode-hook #'lsp-deferred)
```

¹³ <https://aur.archlinux.org/packages/bear/>

²⁰ <https://elpa.gnu.org/>

²¹ <https://melpa.org/>

¹⁴ <https://company-mode.github.io/>

¹⁵ https://www.gnu.org/software/automake/manual/html_node/Autotools-Introduction.html#Autotools-Introduction

¹⁷ <https://www.gnu.org/software/make/>

¹⁸ https://www.gnu.org/software/emacs/manual/html_node/emacs/Init-File.html

5.4 Running emacs with lsp-mode

Just open a C/C++ source file or header file, lsp-mode will be automatically become active, emacs will also start clangd process. Following keybindings will be helpful once lsp-mode is active.

- M-. (to go to definition of the symbol at point)
- M-, (to traverse back in call stack)
- M-? (to list all references of the symbol at point)
- s-l G r (peek all references of the symbol at point)

The screenshot shows the Emacs editor with a C source file named `fuse.c`. The code defines several functions: `int fuse_mount`, `void fuse_unmount`, `int fuse_version`, `const char *fuse_pkgversion`, and `void fuse_pollhandle_destroy`. The `fuse_pkgversion` function is highlighted, and a sidebar on the right displays a list of 9 references to this function across various files in the project, including `example/hello_ll.c`, `example/invalidate_path.c`, `example/notify_inval_entry.c`, `example/notify_inval_inode.c`, `example/notify_store_retrieve.c`, `example/passthrough_ll.c`, `example/printcap.c`, `/include/fuse_common.h`, and `/lib/fuse.c`. The status bar at the bottom indicates the current file is `fuse.c`, the commit is `99% L5117 Git-master`, and the LSP server is `clangd:4138`.

More [lsp keybindings](https://emacs-lsp.github.io/lsp-mode/page/keybindings/)¹⁹ are available for lsp-mode.

¹⁹ <https://emacs-lsp.github.io/lsp-mode/page/keybindings/>

PARTIAL LINKING IN C

Updated: May 07, 2024

Its better to give an example first then explain. Here is the source code of a library called `libplic_math.a`, it is a very simple library with `addition(plic_add)`, `subtraction(plic_sub)` and `multiplication(plic_mul)` functions.

plic_math.h

```
1  #ifndef __PLIC_MATH__
2  #define __PLIC_MATH__
3
4  #ifdef __cplusplus
5  extern "C" {
6  #endif
7
8      int plic_add(int a, int b);
9      int plic_sub(int a, int b);
10     int plic_mul(int a, int b);
11
12     #ifdef __cplusplus
13     }
14     #endif
15
16     #endif /* __PLIC_MATH__ */
```

plic_add.c

```
1  #include <stdio.h>
2  #include "plic_math.h"
3
4  int plic_add(int a, int b) {
5      return a + b;
6  }
```

plic_sub.c

```
1  #include <stdio.h>
2  #include "math.h"
3
4  int plic_sub(int a, int b) {
5      return a - b;
6  }
```

plic_mul.c

```
1 #include <stdio.h>
2 #include "plic_math.h"
3
4 int plic_mul(int a, int b) {
5     return a * b;
6 }
```

Makefile

```
1 CC ?= gcc
2
3 all: libplic_math.a
4
5 plic_add.o: plic_add.c plic_math.h
6     $(CROSS_COMPILE)$(CC) $(CFLAGS) -fPIC -c -o $@ $<
7
8 plic_sub.o: plic_sub.c plic_math.h
9     $(CROSS_COMPILE)$(CC) $(CFLAGS) -fPIC -c -o $@ $<
10
11 plic_mul.o: plic_mul.c plic_math.h
12     $(CROSS_COMPILE)$(CC) $(CFLAGS) -fPIC -c -o $@ $<
13
14 libplic_math.a: plic_add.o plic_sub.o plic_mul.o
15     ar rcUs $@ $^
16
17 clean:
18     rm -fr *.o *.a
19
20 .PHONY: clean
```

Here is the source code of another library called `libplic_sum.a`, this library uses addition(`plic_add`) function from `libplic_math.a` library and provides another function to sum the results(`plic_sum`).

`plic_sum.h`

```
1 #ifndef __PLIC_SUM__
2 #define __PLIC_SUM__
3
4 #ifdef __cplusplus
5 extern "C" {
6 #endif
7
8     int plic_sum(int i, int j);
9
10 #ifdef __cplusplus
11 }
12 #endif
13
14 #endif /* __PLIC_MATH__ */
```

`plic_sum.c`

```
1 #include <stdio.h>
2 #include <plic_math.h>
3 #include "plic_sum.h"
4
5 int plic_sum(int i, int j) {
6     int n = 0;
```

(continues on next page)

(continued from previous page)

```

7  int sum = 0;
8
9  for(n = i; n < j; n++) {
10     sum = plic_add(sum, n);
11 }
12
13 return sum;
14 }

```

Here is the source code of one program `plic_sum_elf` which uses `sum(plic_sum)` function from `libplic_sum.a`.

plic_sum_elf.c

```

1  #include <stdio.h>
2  #include "plic_sum.h"
3
4  int main(int argc, char *argv[]) {
5      printf("sum: %d\n", plic_sum(10, 20));
6      return 0;
7  }

```

In `plic_sum_elf.c`, The function `plic_sum` from `libplic_sum.a` depends on `plic_add` function from `libplic_math.a`, so in order to generate `plic_sum_elf` executable, we have to link both `libplic_sum.a` and `libplic_math.a` like this,

```

$ cc -static -o plic_sum_elf plic_sum_elf.c -L/path/to/libplic_sum_dir -lplic_sum -L/
↪path/to/libplic_math_dir -lplic_math

```

If we omit `-lplic_math`, then compiler will throw error saying undefined reference to `plic_add` symbol. Imagine that you are writing one library like `libplic_sum.a` which itself depends on various libraries like `libplic_math.a`. The user of your library also need to be aware of all the libraries which your library depends.

It is unnecessary for the downstream user to know all the details about your library in order to link with your library

Partial Linking with '-r' flag

To simplify and make linking easier for your downstream user, you can provide your library as partially linked library, means, when you generate your library, you can [instruct the compiler through -r flag¹](#) to resolve all the symbols required for your library and generate one big object file in order to generate a partially linked library.

```

$ cc -r -o plic_sum_nodeps.o plic_sum.c -L/path/to/libplic_math_dir -lplic_math
$ ar rcUs libplic_sum_nodeps.a plic_sum_nodeps.o

```

Now, it becomes easy to generate `plic_sum_elf` executable, no need to link with `libplic_math.a`, downstream user only need to link with `libplic_sum_nodeps.a`

```

$ cc -static -o plic_sum_elf plic_sum_elf.c -L/path/to/libplic_sum_nodeps_dir -lplic_
↪sum_nodeps

```

Here is the Makefile which generates `libplic_sum.a`, `libplic_sum_nodeps.a` as well as `plic_sum_elf` executable.

Makefile

¹ <https://gcc.gnu.org/onlinedocs/gcc/Link-Options.html>

```
1 CC := gcc
2
3 all: libplic_sum.a libplic_sum_nodeps.a plic_sum_elf
4
5 plic_sum.o: plic_sum.c plic_sum.h
6     $(CROSS_COMPILE)$(CC) $(CFLAGS) -fPIC -c -o $@ $< -I../plic-math
7
8 libplic_sum.a: plic_sum.o
9     ar rcUs $@ $^
10
11 plic_sum_nodeps.o: plic_sum.o
12     $(CROSS_COMPILE)$(CC) $(LDFLAGS) -r -o $@ $< -L../plic-math -lplic_
13 ↪math
14
15 libplic_sum_nodeps.a: plic_sum_nodeps.o
16     ar rcUs $@ $^
17
18 plic_sum_elf: plic_sum_elf.c libplic_sum_nodeps.a
19     $(CROSS_COMPILE)$(CC) $(CFLAGS) $(LDFLAGS) -static -o $@ $< -L. -
20 ↪lplic_sum_nodeps
21
22 clean:
23     rm -fr *.o *.a plic_sum_elf
24
25 .PHONY: clean
```

HOW I WROTE MEETTY

Updated: May 07, 2024

MeeTTY is a webapp to do terminal based meetings. It consists of three parts

- IRC chat Window
- Presenter's Terminal
- Practice Terminal

Proper documentation about MeeTTY is available in [README.md](https://gitlab.com/mohan43u/meetty/-/blob/master/README.md)¹. This article is not a MeeTTY documentation, but rather a recollection of how I wrote it.

¹ <https://gitlab.com/mohan43u/meetty/-/blob/master/README.md>

7.1 How we came up with the idea about MeeTTY

Updated: May 07, 2024

7.1.1 Pandemic

Apr 2020, Its the start of [COVID-19 pandemic](https://en.wikipedia.org/wiki/Coronavirus_disease_2019)⁴, governments all around the world started lock-down, no one should roam around outside their houses. I'm one of the coordinator of ilugc.in⁵ who conducts monthly meetings every 2nd Saturday. Last month meet (Mar, 2020) was conducted physically at [IIT](https://www.iitm.ac.in)⁶. It is not possible to conduct this month meet physically. We have to go online.

7.1.2 Frustration

This month's meet is due in next week, obviously we are going online this month like every other special interest groups around the world. But, we are yet to decide on how we are going to conduct the online meet. As our group only promotes [Free and Open Source software](https://en.wikipedia.org/wiki/Free_and_open-source_software)⁷, we cannot go with online meeting platforms like [Zoom](https://zoom.us)⁸ or [Google Meet](https://meet.google.com)⁹ or some other commercial platforms. We have to stick with FOSS based solutions. These are the choices we considered at that time,

[Jitsi](https://jitsi.org)¹⁰

Just like most FOSS group around us, we would have went with Jitsi, but somehow we didn't want to go with jitsi. It may be because we thought of installing Jitsi on our own server to improve video playback but ended up not choosing it because of the thought of buying/creating new instance in aws and doing the actual installation. At this time, we were looking for a solution which don't involve anything to setup in a server.

[Jami](https://jami.net)¹¹

Jami looked very interesting, its peer to peer and no server involved¹. But when we tried it in different OSES, the application crashed in OSX, Ubuntu LTS, it only somehow stable in Arch. Also functionalities are not consistent, and some functionalities didn't work. So we decided not to use at this time.

[Big Blue Button](https://bigbluebutton.org/)¹²

At least for Jitsi, we can use it without doing any server setup through <https://meet.jit.si/>, but for BBB, we have to setup server. Also, it doesn't support [latest distros](https://github.com/bigbluebutton/bbb-install/issues/37)¹³. So, BBB is out of our choice.

[Matrix.org](https://matrix.org)¹⁴

Matrix can do video conferencing using jitsi bridge. Again, this also involved setting up a server with jitsi and matrix and connecting each other. So, we didn't choose Matrix.

All the platforms we considered are proper audio/video based conference solutions. They need proper servers with enough bandwidth to operate in a useful way. We lacked a server and not ready to invest.

⁴ https://en.wikipedia.org/wiki/Coronavirus_disease_2019

⁵ <https://ilugc.in>

⁶ <https://www.iitm.ac.in>

⁷ https://en.wikipedia.org/wiki/Free_and_open-source_software

⁸ <https://zoom.us>

⁹ <https://meet.google.com>

¹⁰ <https://jitsi.org>

¹¹ <https://jami.net>

¹ actually TURN is needed, but its another story

¹² <https://bigbluebutton.org/>

¹³ <https://github.com/bigbluebutton/bbb-install/issues/37>

¹⁴ <https://matrix.org>

7.1.3 The Trigger

We were all discussing and evaluating the platforms at [#ilugc](#)¹⁵² channel in [Freenode](#)¹⁶. At that time, [shrini](#)¹⁷ asked this question,

“Why do we need full audio/video platforms, why can’t we just use IRC and terminals?”

---shrini

We started reasoning this question, like what we loose if we go with IRC based communication instead of live video conference. We already know that IRC works for communication, its text based and doesn’t suck bandwidth. The meetings we conduct are mostly like someone will talk about a technical topic which involves showing terminal commands and explain about what those commands do to achieve specific goals (like setting up [nginx](#)¹⁹ etc.,) so IRC is enough to convey the message to audience.

Now, as I previously said, most of our ilugc talks involves showing commands and explaining what they do. For the explanation part, we already arrived at the conclusion that IRC is enough. Now, we need to find a way for presenter to share terminal to the outside world.

The Idea

This is where things got interested, I already know that there are ways to show terminal output in a webpage³. I also know that python can do [pty](#)²¹, means, it can create a terminal, run a program on that created terminal and handle input/output of that created terminal which in turn will handle input/output to that program.

So I thought why not just combine python’s [pty](#) with whatever [solution](#)²⁰ exists to show terminal output in web?

I come up with this concept,

- run a server to serve a webpage
- the webpage should contain two parts which uses [xterm.js](#)²⁰ to show terminal output
- the first part should show the terminal of presenter, it should be read-only, whatever the presenter do in the server should be displayed in this presenter’s terminal.
- the second part should show another terminal to practice what the presenter is explaining. This terminal should also be capable of getting input from user

I quickly started prototyping my idea, I used the below tools

[flask](#)²², [flask-socketio](#)²³ and [socket.io](#)²⁴

Flask is used to serve webpage, flask-socketio is used to handle inputs from [xterm.js](#)²⁰, whenever a new page get served, socket.io will immediately initiate two websocket back to server, flask-socketio will handle those connection, one websocket is for presenter's terminal called `tmux` and another websocket is for practice terminal called `nspawn`.

¹⁵ <https://meet.ilugc.in>

² we reactivated and acquired admin rights to this channel after the start of the pandemic when we noticed that [#ilugc](#) channel was abandoned long before

¹⁶ <https://freenode.net>

¹⁷ <https://goinggnu.wordpress.com>

¹⁹ <https://nginx.org>

³ I don’t know exactly how to do it^{Page 39, 20}, at this time, but learnt later.

²⁰ <https://xtermjs.org>

²¹ <https://docs.python.org/3/library/pty.html>

²² <https://flask.palletsprojects.com/en/1.1.x/>

²³ <https://flask-socketio.readthedocs.io/en/latest/>

²⁴ <https://socket.io/>

Whenever `tmux` websocket connection comes in, flask-socketio will create a new `pty` and will start `tmux-cmdline` defined in the environment. The input and output of that commandline will be sent back and forth through `tmux` websocket so that it will appear in the `presenter's` terminal part of the webpage

Whenever `nspawn` websocket connection comes in, flask-socketio will create a new `pty` and will start `nspawncmdline` defined in the environment. The input and output will be handled through `nspawn` websocket so that it will appear in `practice` terminal part of the webpage

`tmux`²⁵

`tmux` is a fantastic screen multiplexer. It uses client-server model, for our purpose we basically start `tmux` client through `tmuxcmdline` to wait and connect to a particular `tmux` session called `broadcast` in read-only mode. So all the `presenter's` terminal created through MeeTTY's webpage will be read-only clients to a `broadcast` `tmux` session.

If we want to show something to the audience, all we need is to connect to the same `broadcast` `tmux` session as read-write client and start typing, everything we type will be replicated automatically to all the read-only clients which are connected to the same `broadcast` `tmux` session. Thus, we achieve the goal of broadcasting presenter's terminal to multiple audience.

`linux` `usernamespace`²⁶

Opening a terminal in a webpage and expose it to the whole world is the stupidest craziest idea. But, the thought of providing a place for audience to practice what the presenter is teaching is very interesting. So, instead of starting a direct shell, we utilize `linux` `usernamespace` container to start a shell through `nspawncmdline` and attach it to `practice` terminal through `nspawn` websocket. Thus, `practice` terminal will have a restricted shell which audience can execute commands.

`kiwiirc`²⁷

I completed my prototype and showed it to people in `#ilugc`, People provided great feedback, shrini come up with the idea to include a web IRC client called `kiwiirc` so that people can use a single web page to login into `#ilugc` channel, see what the presenter is teaching through commands and practice those commands. `mbuf`¹⁸ suggested to have single page and show all three parts of the webpage (`IRC` window, `Presenter's` Terminal and `Practice` Terminal) in a single page instead of having three tabs to switch between them. He also suggested to put `Powered By` label somewhere in the webpage. I didn't realize the significance of putting that label, but after sometime I understood how important that label is, The word `Attribution` in `Open Source` licenses refers to this label.

So, this is how we establish two terminals and an IRC client in MeeTTY's webpage.

In the mean time, April month meet took place through `#ilugc` channel. But we didn't introduce MeeTTY to ILUGC monthly meet participants. At this point, you may think that we almost reached our goals, but this is just a start.

To be continued..

²⁵ <https://github.com/tmux/tmux/wiki>

²⁶ https://man7.org/linux/man-pages/man7/user_namespaces.7.html

²⁷ <https://kiwiirc.com/>

¹⁸ <http://shakthimaan.com/>

tmuxcmdline

This environment variable contains the commandline which gets executed whenever there is new `tmux` websocket establishes. This will connect to the `broadcast` `tmux` session as read-only client.

nspawncmdline

This environment variable contains the commandline which gets executed whenever there is new `nspawn` websocket establishes. This will start new shell under linux usernamespace.

7.2 How I convinced Tmux developer

Updated: May 07, 2024

7.2.1 Tmux Problem

As I *already mentioned*, MeeTTY uses `tmux`² as a terminal multiplexer to show Presenter's Terminal and Practice Terminal. While developing, I noticed that tmux shown in MeeTTY web page does not show all the contents of the `broadcast session`, I first thought it may be because of `xterm.js`³ not sending current size of its `div` element properly to `pty`⁴, but, even after syncing `xterm.js` and `pty` to correctly set the `terminal size`⁵, still tmux not showing full contents of `broadcast session` in Presenter's Terminal. All the web terminals which MeeTTY creates through `xterm.js` and `pty` will run one tmux client with read-only permission.

A little back story, When tmux server gets new connection from a tmux client, it sets the window size of all the attached clients to the size of the client which has the smallest terminal size, but *this commit*⁶ changed this logic to not consider read-only clients if there is one read-write client available at the time of attaching the new client to the tmux session. This affected MeeTTY, because MeeTTY web page only creates read-only tmux clients, but the tmux client created through `tmux-share.sh`⁷ which is responsible to provide `broadcast session` have write permission.

7.2.2 Power of FOSS^{Page 42, 13}

With this problem, I approached people in `#tmux`⁹ IRC channel, at first few guys told me that read-only clients should not affect the window size, but I argued that read-only clients affected the size until recently¹ and something got changed this behaviour recently. Then one person with nick `nicm`¹² replied me to set the read-only clients size through `resize-window` command in tmux, I then argued that running `resize-window` after gathering terminal size is bit complicated when terminal size is going to get changed dynamically through `pty` and told that it will be useful to bring back the old window size logic with a flag, then he said he was hesitant to add another commandline flag. At this point I realized that I was speaking to the Author of tmux `Nicholas Marriot`¹². Anyway, I explained my situation with bit more context by explaining what I'm trying with MeeTTY.

After few seconds, `nicm` decided to bring back the old window size logic, within ten minutes, he pushed *one commit*⁸ and asked me to test his changes, he separated two internal flags `read-only` and `ignore-size` from `-r` option on `attach-session` command in tmux and introduced a new option `-f` to explicitly set different internal flags. I was very excited to work with a `OpenBSD`¹⁰ developer and immediately cloned `tmux repository`¹¹ and started building tmux. All in all, It only took just 30 minutes from starting my conversation in `#tmux` IRC channel to get tmux executable with the desired functionality, I realized that this is the power of FOSS, no second opinions, no waiting for approvals, simple uncomplicated development. I love FOSS.

So this is how I was able to convince Tmux developer to bring back the old default window size logic

² <https://github.com/tmux/tmux/wiki>

³ <https://xtermjs.org>

⁴ <https://docs.python.org/3/library/pty.html>

⁵ <https://man.openbsd.org/tty.4#TIOCSWINSZ>

⁶ <https://github.com/tmux/tmux/commit/b905c5d455b354f7210dc118f748a658f0358cd7>

⁷ <https://gitlab.com/mohan43u/meetty/-/blob/master/scripts/tmux-share.sh>

¹³ https://en.wikipedia.org/wiki/Free_and_open-source_software

⁹ <irc://irc.freenode.net/tmux>

¹ I don't know this commit changed the window size logic at that time

¹² <https://www.undeadly.org/cgi?action=article&sid=20090712190402>

⁸ <https://github.com/tmux/tmux/commit/cf9badd6f844e8a26f1e7c59ba1c9eb3358571f>

¹⁰ <https://www.openbsd.org/>

¹¹ <https://github.com/tmux/tmux>

To be continued..

7.3 How I found one Kiwiirc Bug

Updated: May 07, 2024

7.3.1 First ILUGC meet with MeeTTY

After Initial development, I introduced MeeTTY to participants on [May month ILUGC meet](#)³ through my [tmux](#)⁴ talk. Participants were able to see what I do with tmux through `Presenter's Terminal`. I provided explanation of each steps I do in [#ilugc](#)⁵ IRC channel. Participants were also able to see what I type in [#ilugc](#) IRC channel through MeeTTY. So, I was able to conduct a pure text based online meeting through MeeTTY.

7.3.2 The Bug

After my talk about tmux, I asked participants what they feel about having text only meeting through MeeTTY, participants gave positive feedback. One participant [stof999](#)⁶ pointed me that he used his Tab to open MeeTTY, due to Tab having small screen size, one has to scroll down to see the `Presenter's Terminal`, but whenever I type something in [#ilugc](#) IRC channel, [Kiwiirc](#)⁷ will scroll up the web page to keep it visible, so `Presenter's Terminal` will be scrolled down invisible. He felt that this is inconvenience, every time kiwiirc scrolls up, he has to scroll down to `Presenter's Terminal` to see what I was doing.

So I looked into the [sources of Kiwiirc](#)⁸, and found out that [this commit](#)⁹ introduced this behaviour, this behaviour takes place everywhere except [iOS](#)¹⁴. So I decided to add another exception similar to the ios exception to disable this behaviour if kiwiirc is [inside Iframe](#)¹⁰. kiwiirc maintainers accepted my change.

7.3.3 The Real Problem

All these code change and commit to upstream went fine, but here comes the real issue, I was using kiwiirc through one [embedded widget](#)¹¹ which creates one Iframe in MeeTTY web page, kiwiirc does not talk directly to IRC servers, it requires a [gateway](#)¹² to talk to IRC servers. This embedded widget downloads kiwiirc related JavaScript files from a CDN network, as my fix is not yet released, those JavaScript files from CDN networks does not have my fix¹. I have to workaround this real issue.

³ <https://www.freelists.org/post/ilugc/Meeting-Minutes-ILUGC-May-2020-Meet-conducted-on-09052020>

⁴ <https://github.com/tmux/tmux/wiki>

⁵ <https://meet.ilugc.in>

⁶ <https://stof999.ch/o.php>

⁷ <https://kiwiirc.com/>

⁸ <https://github.com/kiwiirc/kiwiirc>

⁹ <https://github.com/kiwiirc/kiwiirc/commit/a7bb8e1f3b0ad40b0117c7432e6635f71e3d4765>

¹⁴ <https://en.wikipedia.org/wiki/IOS>

¹⁰ <https://github.com/kiwiirc/kiwiirc/pull/1316>

¹¹ <https://kiwiirc.com/embedding>

¹² <https://github.com/kiwiirc/webircgateway>

¹ my changes are still not released to CDN networks as of May 07, 2024

7.3.4 Self Hosting Kiwiirc

kiwiirc project also open sourced their gateway. So, I decided to self host this gateway in the same server where I run MeeTTY. Also decided to generate kiwiirc JavaScript files from master branch which contains my fix and serve those JavaScript files from the same server where I run MeeTTY. This will make MeeTTY fully independent of kiwiirc infrastructure².

So, whenever MeeTTY web page loads, it will download the kiwiirc JavaScript files directly from MeeTTY server instead of downloading from CDN network, when user login to IRC through kiwiirc in MeeTTY web page, instead of connecting to kiwiirc hosted gateway, it connects to the gateway which is running in the same server which runs MeeTTY. This self hosted gateway will establish connection to [Freenode](#)¹³ and open #ilugc IRC channel for communication.

So this is how I fixed Kiwiirc related problems (or did I?)

To be continued..

² I didn't realize how self hosting gateway is going to affect me later, more details in the next post.

¹³ <https://freenode.net/>

7.4 How I Isolated Practice Terminal

Updated: May 07, 2024

7.4.1 Practice Terminal

The primary reason to provide Practice Terminal in MeeTTY is to make newcomers to Linux practice the commands shown by the Presenter during meeting without worrying about having Linux installed in their system. But as we all know, giving a shell in web is the stupidest way to ruin your server. So, the Shell we provide in the Practice Terminal should be isolated as much as possible. In Linux, we have [user namespace](#)² to isolate process without using root permission, [cgroup](#)³ to control few aspects of the isolated process and [fuse-overlayfs](#)⁴ to create overlayed root filesystem without root permission.

7.4.2 User Namespace

There are various tools which uses [namespaces](#)⁵ to run containers like [runc](#)⁶ which is used by [docker](#)⁷ and [podman](#)⁸, [lxc](#)⁹ which is used by [lxd](#)¹⁰, [systemd-nspawn](#)¹¹ which is used by [machinectl](#)¹² etc. These container tools focus heavily to provide better way to run containers¹, but I needed a simple tool to work like the traditional [chroot](#)¹³ without root permission to run Practice Terminal shell. The nearest tool I found was [unshare](#)¹⁴. There are others like [bwrap](#)¹⁵ used by [flatpak](#)¹⁶, [firejail](#)¹⁷ which provides simple way to run process in isolated environment using namespaces.

Before we do chroot, we need to manually mount pseudo filesystems like [proc](#)¹⁸, [devtmpfs](#)¹⁹, [sysfs](#)²⁰ etc. For unshare too we have to do this manual mount step before running unshare. I don't want to do this manual mount, it should be automatically done, so I looked for a tool which does this automatically, the nearest tool I found is [arch-chroot](#)²¹ which does mount step automatically, both bwrap and firejail can do the mount step but we have to specify which pseudo filesystems to mount each time.

I would have used arch-chroot, but it requires root privilege, I'm not willing to run anything with root privilege. Also, I need the sandboxed root filesystem to be used as a base to create identical instances of root filesystem for the Practice Terminal shell, this way, the changes made through Practice Terminal will not affect the actual sandboxed root filesystem, this involves [overlayfs](#)²² filesystem, but overlayfs needs root privilege, thankfully fuse-overlayfs is there to rescue me.

² https://man7.org/linux/man-pages/man7/user_namespaces.7.html

³ <https://man7.org/linux/man-pages/man7/cgroups.7.html>

⁴ <https://github.com/containers/fuse-overlayfs>

⁵ <https://man7.org/linux/man-pages/man7/namespaces.7.html>

⁶ <https://github.com/opencontainers/runc>

⁷ <https://www.docker.com/>

⁸ <https://podman.io/>

⁹ <https://linuxcontainers.org/#LXC>

¹⁰ <https://linuxcontainers.org/#LXD>

¹¹ <https://www.freedesktop.org/software/systemd/man/systemd-nspawn.html>

¹² <https://www.freedesktop.org/software/systemd/man/machinectl.html>

¹ there are other tools like [umoci](#)^{Page 46, 31} [buildah](#)³² to prepare container images.

³¹ <https://github.com/opencontainers/umoci>

³² <https://buildah.io/>

¹³ <https://man7.org/linux/man-pages/man1/chroot.1.html>

¹⁴ <https://man7.org/linux/man-pages/man1/unshare.1.html>

¹⁵ <https://github.com/containers/bubblewrap>

¹⁶ <https://flatpak.org/>

¹⁷ <https://github.com/netblue30/firejail>

¹⁸ <https://man7.org/linux/man-pages/man5/proc.5.html>

¹⁹ <https://lwn.net/Articles/330985/>

²⁰ <https://man7.org/linux/man-pages/man5/sysfs.5.html>

²¹ https://wiki.archlinux.org/index.php/chroot#Using_arch-chroot

²² <https://www.kernel.org/doc/Documentation/filesystems/overlayfs.txt>

So, I ended up writing my own `guest`²³ commandline tool which uses fuse-overlaysfs to implement my own isolation mechanism. I also wrote `limit`²⁴ commandline tool to manage resources for Practice Terminal through cgroup.

7.4.3 The guest commandline tool

This guest tool was heavily inspired by unshare because of its simplicity. It uses fuse-overlaysfs to create a root filesystem then automatically mounts proc, devtmpfs and sysfs pseudo filesystems, then isolates itself and creates the shell which appears in Practice Terminal. It also communicates with limit process through `/run/meetty.sock` `unix socket`²⁵ to initialize limits for the Practice Terminal shell.

7.4.4 The limit commandline tool

This limit tool runs as root and starts listening on `/run/meetty.sock` unix socket. When guest commandline tool starts, it will provide its `pid` to the limit process. Once limit gets the `pid`, it will create a new `network namespace`²⁶, add this `pid` to this namespace, creates `veth`²⁷ pair and attach one end of the veth to this new network namespace, then attach other end with a `bridge`²⁸ in the default network namespace. It also limits maximum number of process in the Practice Terminal to 50, It also limits maximum memory Practice Terminal can use to 100 MB, It also limits maximum number of open file descriptors to 1024. All these limits are achieved through cgroup and `rlimit`²⁹.

This limit command line also disables packet routing in Practice Terminal if there is no `tmux-share.sh`³⁰ process running in the host.

writing guest and limit tools taught me lot of things about Linux Namespaces and Cgroups

To be continued..

²³ <https://gitlab.com/mohan43u/meetty/-/blob/master/guest/guest.c>

²⁴ <https://gitlab.com/mohan43u/meetty/-/blob/master/limit/limit.c>

²⁵ <https://man7.org/linux/man-pages/man7/unix.7.html>

²⁶ https://man7.org/linux/man-pages/man7/network_namespaces.7.html

²⁷ <https://man7.org/linux/man-pages/man4/veth.4.html>

²⁸ <https://www.kernel.org/doc/Documentation/networking/bridge.txt>

²⁹ <https://man7.org/linux/man-pages/man2/setrlimit.2.html>

³⁰ <https://gitlab.com/mohan43u/meetty/-/blob/master/scripts/tmux-share.sh>

7.5 The Disaster and the Lessons Learnt

Updated: May 07, 2024

7.5.1 The Disaster

After *conducting* my first talk in ILUGC Meet through MeeTTY, I assumed that MeeTTY is good enough to conduct online training. So, I sent invite mail in the *ILUGC mailing list*¹ to conduct one week session in *Posix Shell*² through MeeTTY. At that time I really don't know the amount of people who will be joining for the session.

On that fateful first day of the week long session, one hour before starting the session, I logged into the MeeTTY server to prepare for my talk, not even considering to check MeeTTY is working fine or not. Suddenly, I decided to see how many connections I'm getting, at first, it looked good, but then I realized that there are too many disconnects happening. Then I tried to open MeeTTY URL, The URL opened, when I tried to login to the *#ilugc*³ channel, *kiwiirc*⁴ received error from *Freenode*⁵ saying more than 10 connection is not possible. Also the practice terminal is not showing anything.

In another 10 minutes, I have to start the talk, I was sitting without knowing what to do, I still not able to figure out how to fix these two issues in the MeeTTY website, in between these, *#ilugc* started going crazy, I learnt how difficult it is to control people in IRC. There were too many people joining *#ilugc* and everyone start sending messages irrelevant to the session. I have no idea how to control people as I'm very new to op role in IRC. I previously studied some flags to do op role, but I was not able to concentrate either on the MeeTTY website issue or the *#ilugc* channel chaos.

Then *shrini*⁶ and *mbuf*⁷ came to rescue. *shrini* started sending messages to calm down the people in *#ilugc* channel. I also promoted both of them to ops and asked them to change the channel mode to *+m* which will prevent normal users from sending messages and only allows messages from voiced users and ops. *mbuf*^{Page 48, 7} took the initiative and enabled *+m* mode. This allowed me to somehow relax.

It was 30 minutes past the start time and I'm still not able to come up with a solution to fix the connectivity issue between *kiwiirc* and *freenode*. I decide to give up, I disabled *kiwiirc* widget and disabled Practice Terminal in the MeeTTY web page, asked participants to join *#ilugc* from their own IRC client or through *Freenode's own kiwiirc*⁸. Also, asked them to practice the commands in some other way.

After 40 minutes delay, I was somehow convinced myself that the issues are managed, then with the remaining energy, started my talk. Even though the start was a disaster, I was able to convey and conduct the session as I intended. People were able to see what I teach through Presenter's Terminal in MeeTTY web page and learnt something on that day.

That day I was exhausted like anything. I was not even ready to do the postmortem which I usually do when shit hits the roof in my work place. Somehow after some time, I recovered with enough energy to look into what went wrong with MeeTTY.

¹ <https://www.freelists.org/archive/ilugc/>

² <https://www.freebsd.org/cgi/man.cgi?query=sh&sektion=&n=1>

³ <https://meet.ilugc.in>

⁴ <https://kiwiirc.com>

⁵ <https://freenode.net/>

⁶ <https://goinggnu.wordpress.com/>

⁷ <http://shakthimaan.com/>

⁸ <https://webchat.freenode.net/>

7.5.2 The Lessons

Problem with self hosting kiwiirc's gateway^{Page 49, 9}

As I already *mentioned*, I was running kiwiirc's gateway locally in my MeeTTY server. The kiwiirc widget in the MeeTTY web page will use the gateway running on the same MeeTTY server. This gateway is responsible for communicating with freenode and establish IRC session. This gateway acts like a IRC client on behalf of kiwiirc.

But, freenode have limitation on how many connections one IP address can establish. If there are more than 10 IRC sessions started from the same IP address, freenode will not allow further connection from the same IP address. But, kiwiirc project run their own gateway in their infrastructure to communicate with freenode, somehow their infrastructure didn't have this issue. Anyway, running my own gateway proved to be a disastrous move. I looked into how *kiwiirc's own website*^{Page 48, 4} connect with their gateway, It turned out to be setting the kiwiirc's gateway URL in a simple configuration variable called `kiwiServer` on kiwiirc's config file. I copied the same URL to my kiwiirc's config file. It worked fine. Now, the kiwiirc widget in MeeTTY web page connects directly with kiwiirc's own gateway running in their infrastructure to communicate with freenode.

Fork is expensive in Python

The other issue I faced on that day in MeeTTY web page was that the Practice Terminal never showed up. Also, during that chaos, I noticed that the server's memory was gone. Initially I thought the *guest* tool (written in C) going ape shit and leaking memory, but it was not the case. The issue was with the way I used Python's `pty`¹⁰ module.

The `pty` module offers a simple way to create PTY terminals with `fork()`¹¹, which will automatically create a child process and attach PTY terminal's one end to the child and another end to the parent. It looked easy for me to use `fork()` as a developer because all I need to do is to do the `fork()` and define the child. I was wrong, stupidly very wrong. When we do fork in Python, the entire Python interpreter get forked into a new process, so we are doubling the memory. This is why, the MeeTTY server quickly run out of memory even with just 20 session.

The Re-Design

To get out of this fork mess, I have to redesign the whole MeeTTY core logic to use `openpty()`¹² and `threads`¹³ instead of `fork()`. I also came across this issue of hitting maximum open file descriptors when using `openpty()`, because, when a PTY pair file descriptors allocated through `openpty()`, we have to close it manually because this call is a python wrapper on top of C's `openpty()`¹⁴, so we have to close it properly, otherwise we will hit the maximum open file descriptors limit.

⁹ <https://github.com/kiwiirc/webircgateway>

¹⁰ <https://docs.python.org/3/library/pty.html>

¹¹ <https://docs.python.org/3/library/pty.html#pty.fork>

¹² <https://docs.python.org/3/library/pty.html#pty.openpty>

¹³ <https://docs.python.org/3/library/threading.html>

¹⁴ <https://man7.org/linux/man-pages/man3/openpty.3.html>

7.5.3 The Moral of the Story

After I re-wrote MeeTTY, I stress tested with help from shrini. We were able to reach more than 150 connections compared to going out of memory by mere 20 connections. This is on a 2 core 2GB VM. Also, I was able to setup scaling for the Practice Terminal in [caddy](https://caddyserver.com/)¹⁵ so that If I add another instance of MeeTTY in a different VM, it will automatically use that VM only to run Practice Terminal.

I learnt a lot about how the bottlenecks raises their head when we start scaling, we may have to re-write the entire project to achieve performance, this whole experience taught me that the code should be replaceable at anytime by anyone to grow. To me, this is the moral of the story.

So, this is how I wrote MeeTTY

End

¹⁵ <https://caddyserver.com/>

PLAYING WITH RESTRUCTUREDTEXT

Updated: May 07, 2024

This is a paragraph. This is an example to *italics*, this is an example to **bold**, this is an example to *passthrough*, this is an example to `literal text`, this is an example to single word hyper link [google](#)³, this is an example to [same google but with space between](#)⁴, this is also an example example to google [link](#)^{Page 51, 3}, this is also an example to [google link with space between](#)⁴

this is an example to footnote [CIT01] reference, footnote reference can be¹ auto² numbered, this is an example to <https://reddit.com> url. Something need to be changed.

Example bullet list,

- one
- two
 - one one
 - one two

1. one
2. two
 1. one one
 2. one two

This is also numbered list,

1. one
2. two
 1. one one
 2. one two

Term 1

Definition 1

Term 2

Definition 2

This is a normal paragraph without broken lines

³ <https://google.com>

⁴ <https://google.co.in>

¹ this is first reference

² this is second reference

this paragraph
contains broken lines

this is normal paragraph introducing field definition

variable0
variable0 definition
variable1
variable1 definition

this is normal paragraph introducing option definition

-a short option
--analog long option

```
this is literal block
anything written here will show
differently
```

this is normal paragraph to show shorthand literal block beginning

```
this is literal block
with broken lines
```

this is example for quoted block

```
> this line is not intended
> and also lines can be broken
> this block begins with ">" character
```

this is example to line blocks

```
| this line is
| a line block
```

this is example to block quote

“this is a quote”

---Mohan R

this is normal paragraph introducing c code highlighting through `highlight::` directive

```
#include <stdio.h>
int main(int argc, char* argv[]) {
    printf("hello world\n");
    return 0;
}
```

this is normal paragraph introducing bash code highlighting through `highlight::` directive

```
$ echo "this is import bash" or is this python?
```

this is normal paragraph introducing python highlighting through `highlight::` directive

```
import os
import sys
def hello():
    print("hello world")
```

this is a normal paragraph introducing c highlighting through code-block:: directive

```
1 #include <stdio.h>
2 int main(int argc, char* argv[]) {
3     printf("hello world\n");
4     return 0;
5 }
```

this is a normal paragraph introducing console highlighting through code-block:: directive

```
$ echo "is this import bash" or is this python?
```

this is a normal paragraph introducing python highlighting through code-block:: directive

```
1 import os
2 import sys
3 def hello():
4     print("hello world")
```

this is a normal paragraph introducing literalinclude:: directive

```
1 #include <stdio.h>
2
3 static void helloworld(void) {
4     printf("helloworld\n");
5 }
6
7 int main(int argc, char *argv[]) {
8     helloworld();
9     return 0;
10 }
```

this is a normal paragraph introducing same literalinclude:: directive with specific lines set as :lines: 3-5, :lines:8 and lines:1-

```
static void helloworld(void) {
    printf("helloworld\n");
}
```

```
helloworld();
```

```
#include <stdio.h>

static void helloworld(void) {
    printf("helloworld\n");
}

int main(int argc, char *argv[]) {
    helloworld();
    return 0;
}
```

this is normal paragraph introducing doctest

```
>>> 1 + 1
2
```

this is a normal paragraph introducing table

head1	head2
col1	col2

this is a paragraph introducing inline [link to python](#).

8.1 this is one internal section

this paragraph contains [this is one internal section](#) link to the section it contains.

this is a paragraph this contains inline roles like *italic*, **bold text**, `literal text`, subscript _{text}, superscript ^{text}

Attention: this is an attention text

Caution: this is caution text

Danger: this is danger text

Error: this is error text

Hint: this is hint text

Important: this is important text

Note: this is note text

Tip: this is tip text

Warning: this is warning text

Generic Admonition

this is admonition text

This is a centered text

- one
- two
- three
- four

this is a normal paragraph introducing image



Topic

this is a topic which is not attached to document flow

Sidebar

Sidebar Subtitle

this is a sidebar with subtitle

some stupid shit called ephigraph

---Mohan R

Table 1: Table with Title

head1	head2
col1	col2

Table 2: CSV Table

head1	head2
col1	col2

Table 3: List Table

head1	head2
col1	col2

HOWTO REMAP KEYBOARD KEYS IN LINUX

Updated: May 07, 2024

Almost two years since I wrote a blog post, but learnt a lot in these past two years. I'll be writing everything I can remember in the coming days, but first, I need to record down something I learnt recently.

I came across one challenge to generate a key press event where the key is not at all in the normal US keyboard. That special key (Hangul/English Toggle key) is only available in Korean Keyboard. So, here is how I did

We need to choose any one key available in the normal US keyboard and remap it to produce Hangul/English Toggle Key's code. For my purpose I'm going to remap F7 key to produce Hangul/English keypress events. First, I need to know what F7 produces when it gets pressed. There is a tool called `evtest` which will show the code generated by physical keyboard

```
Event: time 1496043435.462606, type 4 (EV_MSC), code 4 (MSC_SCAN), value 41
Event: time 1496043435.462606, type 1 (EV_KEY), code 65 (KEY_F7), value 0
```

Here, the `MSC_SCAN` value of 41 (hexadecimal) is the code generated by the physical keyboard. Using this value, we need to create one file (`/usr/lib/udev/hwdb.d/63-keyboard.hwdb`) with the following content

```
keyboard:name:AT Translated Set 2 keyboard:dmi:*
KEYBOARD_KEY_41=hangeul
```

Here, The first line starting with `keyboard:` indicates the name of the keyboard. There are lot of way to match the keyboard (see `/usr/lib/udev/hwdb.d/60-keyboard.hwdb` file for more details). The second line which starting with `KEYBOARD_KEY_` should have one space before `KEYBOARD`. The `hangeul` string at the end of the second line is a suffixed lowercase substring of a macro defined in `/usr/include/linux/input-event-codes.h` file. You can use more than one `KEYBOARD_KEY_` lines to remap the keys in your physical keyboard.

Once you create the `63-keyboard.hwdb` file. Run the following command to update `/etc/udev/hwdb.bin` file which is used by `udev`

```
$ sudo udevadm hwdb --update
```

Once the `hwdb.bin` file updated, Either reboot your machine or trigger `udev` events for the particular keyboard (here, `/dev/input/event1` may vary depend on your keyboard devfile mapping)

```
$ sudo udevadm trigger /dev/input/event1
```

Once you have the remapping, you can test it again with `evtest`.

```
Event: time 1496046275.166879, type 4 (EV_MSC), code 4 (MSC_SCAN), value 41
Event: time 1496046275.166879, type 1 (EV_KEY), code 122 (KEY_HANGUEL), value 0
```

In the above output, MSC_SCAN value 41 generates KEY_HANGUEL instead of KEY_F7.

Update (2018-02-06)

I recently have to remap Left Window Key to Right Window Key. So I created a new file in `/etc/udev/hwdb.d/92-keyboard-override.hwdb` and added following lines

```
# keyboard overrides
# laptop keyboard
libinput:keyboard:*
    KEYBOARD_KEY_db=rightmeta
    KEYBOARD_KEY_dd=rightmeta

# USB keyboard
evdev:*
    KEYBOARD_KEY_700e3=rightmeta
```

Here `KEYBOARD_KEY_dd=rightmeta` changes Menu key to Right Window Key, `KEYBOARD_KEY_db=rightmeta` and `KEYBOARD_KEY_700e3=rightmeta` both changes Left Window Key to Right Window Key but former is for laptop keyboard and later is for USB keyboard.

HOWTO CHANGE FONT IN LINUX USING FONTCONFIG

Updated: May 07, 2024

To those who are not familiar to [fontconfig](#), it is the library which provides fonts to each program in your linux desktop (atleast in gnome3). It maintains the details about all the font files available in your linux box and provides best matched font when your program requests a specific type of font.

If you use only english in day-to-day activities, fontconfig dont come in your way, but its a nightmare if you are a multilingual user. Even if you are a multilingual user, fontconfig tries its best to make your life easy by matching the best font for your language automatically, but the pain starts if you don't like what fontconfig provides and want to change the default font for your language to something which looks good in the editor.

I went through the pain and want to share what I did to fix. For me, fontconfig said TAMu_Kadambari is the best font for Tamil and gedit by default shows tamil fonts using TAMu_Kadambari. But, this font doesn't look good in gedit. After much research, I'm able to change the default font for Tamil language from TAMu_Kadambari to TAMu_Kalyani using this xml file.

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<!--save this file as ~/.config/fontconfig/fonts.conf-->
<fontconfig>
  <match>
    <test name="lang">
      <string>ta</string>
    </test>
    <edit name="family" mode="assign">
      <string>TAMu_Kalyani</string>
    </edit>
  </match>
</fontconfig>
```

Here, the `<match>` tag contains one test which checks if the provided language is Tamil (`lang=ta`). If it is, then edit the original request coming from the application and assign family name as TAMu_Kalyani (`family=TAMu_Kalyani`) so that fontconfig matching algorithm will match only TAMu_Kalyani.

Thats it, my gedit now picks TAMu_Kalyani and it looks good. More details about fontconfig is available [here](#).

fontconfig uses patterns for its matching algorithm, and it goes like this

```
<families>-<point sizes>:<name1>=<values1>:<name2>=<values2>..
```

for example, here is how you can check which is the best matched font available in your system for language Tamil. Output also contains top 10 ranked fonts for tamil. The first preferred font is TAMu_Kalyani, because I have overridden in `~/.config/fontconfig/fonts.conf`

```
[mohan@mohanlaptop0 ~]$ fc-match -s 'monospace-10:lang=ta' | head
TAMu_Kalyani.ttf: "TAMu_Kalyani" "Regular"
TSCu_Comic.ttf: "TSCu_Comic" "Normal"
DejaVuSans.ttf: "DejaVu Sans" "Book"
DejaVuSansMono.ttf: "DejaVu Sans Mono" "Book"
DejaVuSerif.ttf: "DejaVu Serif" "Book"
Pothana2000.ttf: "Pothana2000" "Pothana2000"
MalOtf.ttf: "MalOtf" "Book"
TSCu_Paranar.ttf: "TSCu_Paranar" "Regular"
SyrCOMAdiabene.otf: "East Syriac Adiabene" "Regular"
SyrCOMKharput.otf: "Serto Kharput" "Regular"
[mohan@mohanlaptop0 ~]$
```

If I remove `~/.config/fontconfig/fonts.conf`, it goes back to the original behaviour and provides TAMu_Kadambari as preferred font for Tamil

```
[mohan@mohanlaptop0 ~]$ fc-match -s 'monospace-10:lang=ta' | head
TAMu_Kadampari.ttf: "TAMu_Kadambri" "Regular"
TAMu_Kalyani.ttf: "TAMu_Kalyani" "Regular"
DejaVuSansMono.ttf: "DejaVu Sans Mono" "Book"
DejaVuSansMono-Bold.ttf: "DejaVu Sans Mono" "Bold"
DejaVuSans.ttf: "DejaVu Sans" "Book"
DejaVuSerif.ttf: "DejaVu Serif" "Book"
Pothana2000.ttf: "Pothana2000" "Pothana2000"
MalOtf.ttf: "MalOtf" "Book"
TSCu_Comic.ttf: "TSCu_Comic" "Normal"
TSCu_Paranar.ttf: "TSCu_Paranar" "Regular"
[mohan@mohanlaptop0 ~]$
```

fontconfig configuration files are very versatile and one can do lot of things, but we need to first understand its patterns to do anything useful. Anyway, I hope this article provide good example for fc-match command-line tool and fontconfig patterns.

YIELD IN VALA AND GIO CONCURRENCY

Updated: May 07, 2024

By no means, I'm an expert in [Vala](#) or [Gio](#). Infact, I'm just a beginner, but I have learned something while playing with vala which I want to record (for future reference). Here I'll explain how Gio's async framework works and how vala uses Gio beneath the tree.

11.1 Gio Async Framework

Better to explain with an example,

Here, I have created a new GObject class called `TestGioAsync` with `something_to_say` property. I have implemented a simple async function called `test_gio_async_say`. When called, `test_gio_async_say` will setup `GSimpleAsync` object, attach another function called `test_gio_async_say_idle_func` into `GMainLoop`'s next idle iteration and return.

Once `test_gio_async_say_idle_func` triggered, it takes the value from `something_to_say` property and put it into `g_simple_async_result_set_op_res_pointer` as `op_res` (which will be picked-up later) and completes the async transaction with `g_simple_async_result_complete_in_idle` which will call the `test_gio_async_say_cb` provided in `test_gio_async_say` in `GMainLoop`'s next idle iteration.

Once `test_gio_async_say_cb` triggered, it will call `test_gio_async_say_finish` which will simply return `op_res` which contains the value taken from `something_to_say` property.

11.2 Vala Async Framework

This is the equivalent vala example for the above C example

Vala async functions returns immediately whenever it reaches `yield` keyword, In this example, I have hooked `this.say.callback` into `GMainLoop`'s next idle iteration. In the next idle iteration, `this.say` will be called and will start execution exactly after the `yield` statement which will eventually trigger the anonymous lambda function provided in `this.say.begin` with `sentence` as return value.

Once the lambda function triggered, it will call `this.say.end` which will return the `sentence` value to the lambda function.

In Vala example, `this.say.callback` is equivalent to `test_gio_async_say_idle_func` in C example.

SETUP GNOME DEVELOPMENT ENVIRONMENT IN ARCHLINUX

Updated: May 07, 2024

Many people who want to contribute to Gnome will go with Fedora, because it is the best suitable for gnome developer. But, Archlinux is as good as Fedora in terms of providing latest gnome libraries. Here I'll explain how I setup my development environment in Archlinux using [pacstrap](#), [jhbuild](#) and [arch-chroot](#).

12.1 Setup Base

In your home directory, create a folder for gnome and install archlinux packages

```
$ cd ~/Devel/gnome
$ su
# pacstrap -d runtime base base-devel
# chown -R <username>:<groupname> runtime
# exit
$
```

Here, <username>:<groupname> must be your username and groupname to make sure you are able to modify the files inside ~/Devel/gnome/runtime without any issues. The above steps will install Archlinux base packages. Now, we need to install [jhbuild](#),

12.2 Jhbuild

```
$ cd ~/Devel/gnome/checkout
$ git clone git://git.gnome.org/jhbuild
$ cd jhbuild
$ PYTHON=/usr/bin/python2 ./autogen.sh --prefix="${HOME}/Devel/gnome/runtime/usr"
$ make install
```

done, next step is to configure [jhbuild](#) to setup the paths properly, to do this, we need to copy the sample `jhbuildrc` file from `jhbuild` source directory to ~/ .config directory and edit that file,

```
$ cp ~/Devel/gnome/checkout/jhbuild/examples/sample.jhbuildrc ~/.config/jhbuildrc
```

Here is my `.jhbuildrc` modifications

```
# -*- mode: python -*-
--- /home/mohan/Devel/gnome/checkout/jhbuild/examples/sample.jhbuildrc 2015-01-17_
↪11:38:40.055187155 +0800
+++ /home/mohan/.config/jhbuildrc 2015-01-06 09:07:34.493161676 +0800
@@ -10,22 +10,25 @@
# jhbuild/defaults.jhbuildrc, but can be any file in the modulesets directory
# or a URL of a module set file on a web server.
# moduleset = 'gnome-apps-3.12'
-#
+moduleset = 'gnome-world'
+
# A list of the modules to build. Defaults to the GNOME core and tested apps.
# modules = [ 'meta-gnome-core', 'meta-gnome-apps-tested' ]

# Or to build the old GNOME 2.32:
# moduleset = 'gnome2/gnome-2.32'
# modules = ['meta-gnome-desktop']
+modules = ['gjs', 'farstream', 'gnome-online-accounts', 'polari', 'gnome-shell',
↪'glade', 'gnome-builder']

# what directory should the source be checked out to?
-checkoutroot = '~/jhbuild/checkout'
+checkoutroot = '~/Devel/gnome/checkout'

# the prefix to configure/install modules to (must have write access)
-prefix = '~/jhbuild/install'
+prefix = '~/Devel/gnome/runtime/usr'

# custom CFLAGS / environment pieces for the build
# os.environ['CFLAGS'] = '-Wall -g -O0'
+os.environ['PYTHON'] = '/usr/bin/python2'

# extra arguments to pass to all autogen.sh scripts
# to speed up builds of GNOME, try '--disable-static --disable-gtk-doc'
```

Here is the details of my changes,

moduleset

A moduleset is an xml file contains details (like modulename, version info, repo address, type of repo, how to build etc.,) of each and every project (we call it as module) needed for gnome environment. jhbuild maintains moduleset for each release, you can see the different modulesets inside your jhbuild directory (~/.Devel/gnome/jhbuild/modulesets). gnome-world moduleset represent generic set which will be updated for every release. So, I use gnome-world to make my environment as latest as possible.

modules

when we execute jhbuild build command, jhbuild will compile only the modules listed in this place. make sure you build what you want instead of building all the gnome projects (aka meta-gnome-desktop)

checkoutroot

place where jhbuild will clone the gnome repositories. set this to ~/Devel/gnome/checkout

prefix

place where jhbuild will put the compiled binaries, set this to ~/Devel/gnome/runtime/usr. If you notice, you will see that ~/Devel/gnome/runtime is the same path where we installed archlinux base,

so, we basically put the gnome binaries (libraries & executables) on top of archlinux, I'll tell you why we are doing this when I explain arch-chroot.

```
os.environ['PYTHON']
```

set this to `/usr/bin/python2` because, Archlinux's default python is python3 but most of gnome modules don't like python3 (especially `jhbuild`. see [here](#).)

12.3 Compiling

All set, its time to compile.

```
$ ~/Devel/gnome/runtime/usr/bin/jhbuild build
```

If `jhbuild` finds any missing libraries to compile the modules listed in `jhbuildrc`, it will throw error, here comes the need of arch-chroot.

12.4 Installing missing Dependencies

The reason we setup archlinux base in `~/Devel/gnome/runtime` is to install any missing libraries here from arch packages using arch-chroot and packman and make sure we don't disturb the current system. Let say, `jhbuild` throws that `spotread` is missing when compiling `colord` module. `spotread` executable is provided by `argyllcms` package and here is the simple way to fix the dependency issue,

```
$ cd ~/Devel/gnome
$ su
# arch-chroot runtime /usr/bin/bash
# pacman --sync --refresh
# pacman --sync --sysupgrade
# pacman --sync argyllcms
# exit
# chown -R <username>:<groupname> runtime
# exit
$
```

Now, we can start compiling again, I suggest to switch to `jhbuild` environment before we compile, because `jhbuild` will setup necessary environment variables (like `LD_LIBRARY_PATH` pointing to `~/Devel/gnome/runtime/usr/lib` etc.,) to make sure dependency libraries are resolved from `~/Devel/gnome/runtime/usr`.

```
$ ~/Devel/gnome/runtime/usr/bin/jhbuild shell
$ jhbuild build
$ exit
```

Now, to execute the latest compiled gnome-builder, here is the commandline

```
$ ~/Devel/gnome/runtime/usr/bin/jhbuild run gnome-builder
```

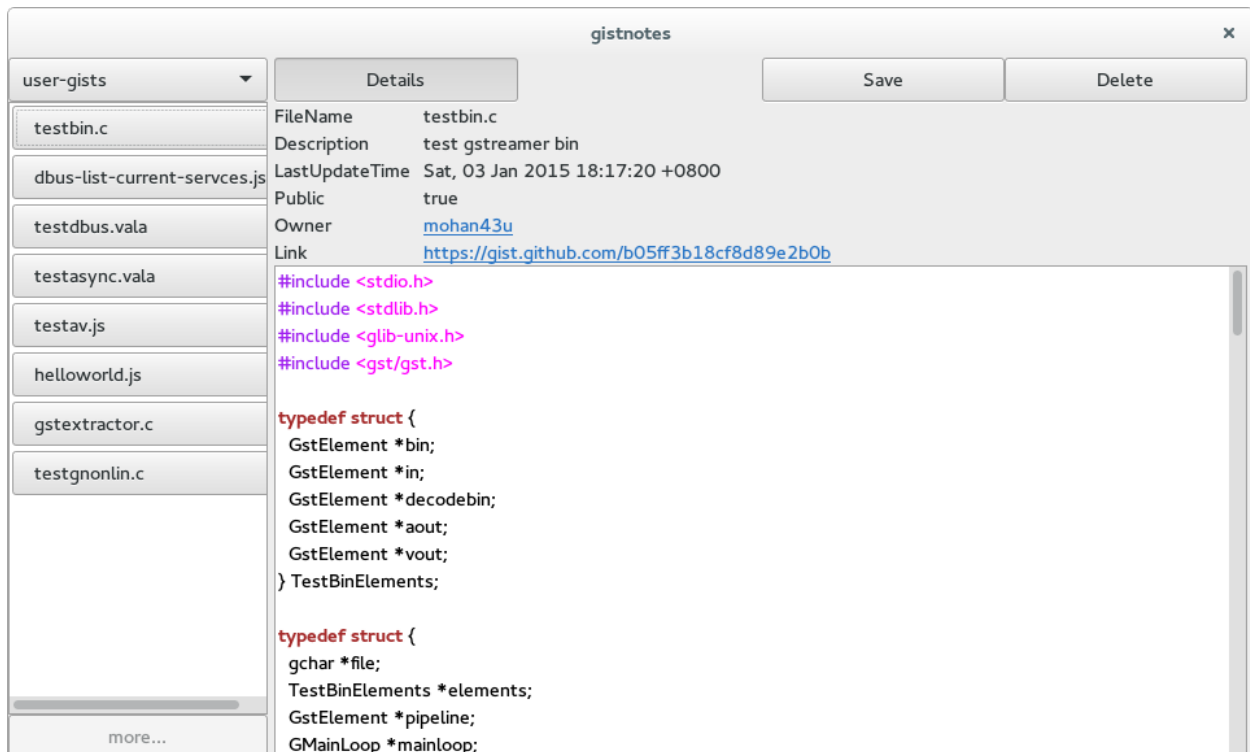
Thats it, you have your gnome development environment ready.

GISTNOTES - A GNOME SHELL EXTENSION

Updated: May 07, 2024

Well, I should probably say, its basically a gnome application disguised as a shell extension. written entirely using javascript.

Nothing much to say more than what the README say here, <https://github.com/mohan43u/gistnotes>



This is my first contribution to gnome. you can install it through <https://extensions.gnome.org/extension/917/gistnotes/>. Check it out, and provide feedback.

ISSUE WITH SABAYON

Updated: May 07, 2024

Long time since I blogged. Life is crazy mate (specially after you become a husband). Personals aside, I have been waiting to blog about my experience with Sabayon.

I was a user of sabyon for about two and half years. But now, I'm a Arch user and thats one damn thing I did right to kickstart my hunger for coding again. Well, sorry sabayon developers, you guys suck!! because you dont worry about your users anymore.

Nothing wrong with the distro (I admire the stability it provided), but your attitude towards the users suck!! everytime we report a bug, you people either say raise this bug to upstream (or) raise this bug in gentoo's bugzilla because they are the one who create the package.

If gentoo have a package and we ask it to enable it in sabayon overlay, you will say, we don't enable some package because of one user and one can use gentoo's overlay if they want. Dammit!! if I want to use gentoo's overlay, why do I need to install sabayon?

Anyway, goodbye. I'm now in a true rolling distro.

GNOMECONTINUOUS USING SYSTEMD-NSPAWN

Updated: May 07, 2024

OSTree

One of interesting projects from Gnome Community which tries to define a new way of packaging. I really don't know how exactly OSTree works, but it generates daily builds (qcow2 images) for Gnome. The OSTree repository which helps in building qcow2 images for Gnome is called [GnomeContinuous](#). Similarly, there is another OSTree repository which builds qcow2 images for Fedora is called [rpm-ostree](#).

Systemd-Nspawn

Someone in the vast internet call this tool as Poor Man's Virtualization. This is nothing but the good old chroot jail with modern walls ([namespaces](#)).

So, my experiment is to boot gnome-continus generated qcow2 image using systemd-nspawn to run a pure gnome OS parallely. First thing is to download the gnome-continuous qcow2 image.

Note: all commands in this post executed under root

```
# wget http://build.gnome.org/continuous/buildmaster/images/z/current/gnome-
↪continuous-x86_64-devel-debug-20140720.0.qcow2.gz
# gunzip gnome-continuous-x86_64-devel-debug-20140720.0.qcow2.gz
# ln -s gnome-continuous-x86_64-devel-debug-20140720.0.qcow2 gnome.qcow2
```

Network Block Device

In order to open-up qcow2 images, qemu provides a tool called `qemu-nbd`. this tool requires `nbd` kernel module to be loaded. So, load nbd driver as root,

```
# modprobe nbd
# qemu-nbd --connect /dev/nbd0 gnome.qcow2
```

Now, we have associated `/dev/nbd0` with qcow2 image. This image contains three partitions(boot, swap and root), we have to separte them into loop devices,

```
# parted /dev/nbd0 unit B print
Model: Unknown (unknown)
Disk /dev/nbd0: 8589934592B
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
```

(continues on next page)

(continued from previous page)

Number	Start	End	Size	Type	File system	Flags
1	32768B	209747967B	209715200B	primary	ext4	boot
2	209747968B	276856831B	67108864B	primary	linux-swap(v1)	
3	276856832B	8589934591B	8313077760B	primary	ext4	

```
# losetup -o 276856832 -f /dev/nbd0
# losetup -o 32768 -f /dev/nbd0
# losetup -j /dev/nbd0
/dev/loop0: [0005]:849 (/dev/nbd0), offset 276856832
/dev/loop1: [0005]:849 (/dev/nbd0), offset 32768
#
```

Now we have to get the ostree boot parameter from /dev/loop1 (which is the boot partition in qcow2 image)

```
#mkdir boot
#mount /dev/loop1 boot
#grep 'ostree=' boot/syslinux/syslinux.cfg
APPEND root=LABEL=gnostree-root quiet splash ostree=/ostree/boot.0/gnome-continuous/
↪f5a97103e9911b5a19ec154ecccfc6bd77f43e90e27a64adabf054e4db06c38b/0
#umount /dev/loop1
#rm boot
```

ostree boot parameter is important for ostree tools to operate properly, so we have to append this boot parameter into our host machine's /boot/grub/grub.cfg so that this boot parameter will be reflected in /proc/cmdline. Open your host machine's /boot/grub/grub.cfg and append ostree parameter like this,

```
linux /kernel-genkernel-x86_64-3.14.0-sabayon init=/sbin/init root=UUID=80b5a49b-
↪4156-49b3-ab4d-71d1de2dca36 resume=UUID=7550843f-1e8a-4c8e-b9a3-3c9907164726 ro rd.
↪auto rd.lvm.vg=vg_mohanlaptop rd.vconsole.keymap=us quiet splash ostree=/ostree/
↪boot.0/gnome-continuous/
↪f5a97103e9911b5a19ec154ecccfc6bd77f43e90e27a64adabf054e4db06c38b/0
initrd /initramfs-dracut-x86_64-3.14.0-sabayon
```

Once done, reboot your host machine and re-setup /dev/loop0 and /dev/loop1 as we did in above steps. We have to adjust root partition to stop plymouthd and modify default runlevel to multi-user instead of graphical,

```
# mkdir root
# mount /dev/loop1 root
# rm root/ostree/boot.0/gnome-continuous/
↪f5a97103e9911b5a19ec154ecccfc6bd77f43e90e27a64adabf054e4db06c38b/0/etc/systemd/
↪system/default.target
# ln -s root/ostree/boot.0/gnome-continuous/
↪f5a97103e9911b5a19ec154ecccfc6bd77f43e90e27a64adabf054e4db06c38b/0/usr/lib/systemd/
↪system/multi-user.target root/ostree/boot.0/gnome-continuous/
↪f5a97103e9911b5a19ec154ecccfc6bd77f43e90e27a64adabf054e4db06c38b/0/etc/systemd/
↪system/default.target
# mv root/ostree/boot.0/gnome-continuous/
↪f5a97103e9911b5a19ec154ecccfc6bd77f43e90e27a64adabf054e4db06c38b/0/usr/bin/
↪{plymouthd,plymouthd.disabled}
# umount /dev/loop1
# rm root
```

Now, time to setup the filesystems,

```
# mkdir root
# mount /dev/loop0 root
```

(continues on next page)

(continued from previous page)

```
# mount --bind root/ root/ostree/boot.0/gnome-continuous/
↪ f5a97103e9911b5a19ec154ecccfc6bd77f43e90e27a64adabf054e4db06c38b/0/sysroot/
# mkdir /dev/exported
# ln /dev/loop1 /dev/exported/
```

Now we are ready to chroot,

```
# systemd-nspawn --bind /dev/exported -b -D root/ostree/boot.0/gnome-continuous/
↪ f5a97103e9911b5a19ec154ecccfc6bd77f43e90e27a64adabf054e4db06c38b/0
```

this will boot the qcow2 image which we downloaded and show a login prompt, get inside as root and mount the boot partition we exported from host machine which will be available inside /dev/exported/loop1,

```
GNOME-OSTree 0+snapshot-20140717 qemux86-64 console
qemux86-64 login: root
Last login: Sat Jul 19 22:38:48 SGT 2014 on console
root@qemux86-64:~# mount /dev/exported/loop1 /boot
```

Now, we can start the gnome session by running a X in host's vt2 and point gnome-session inside chroot to connect to DISPLAY=:1

In Host Machine,

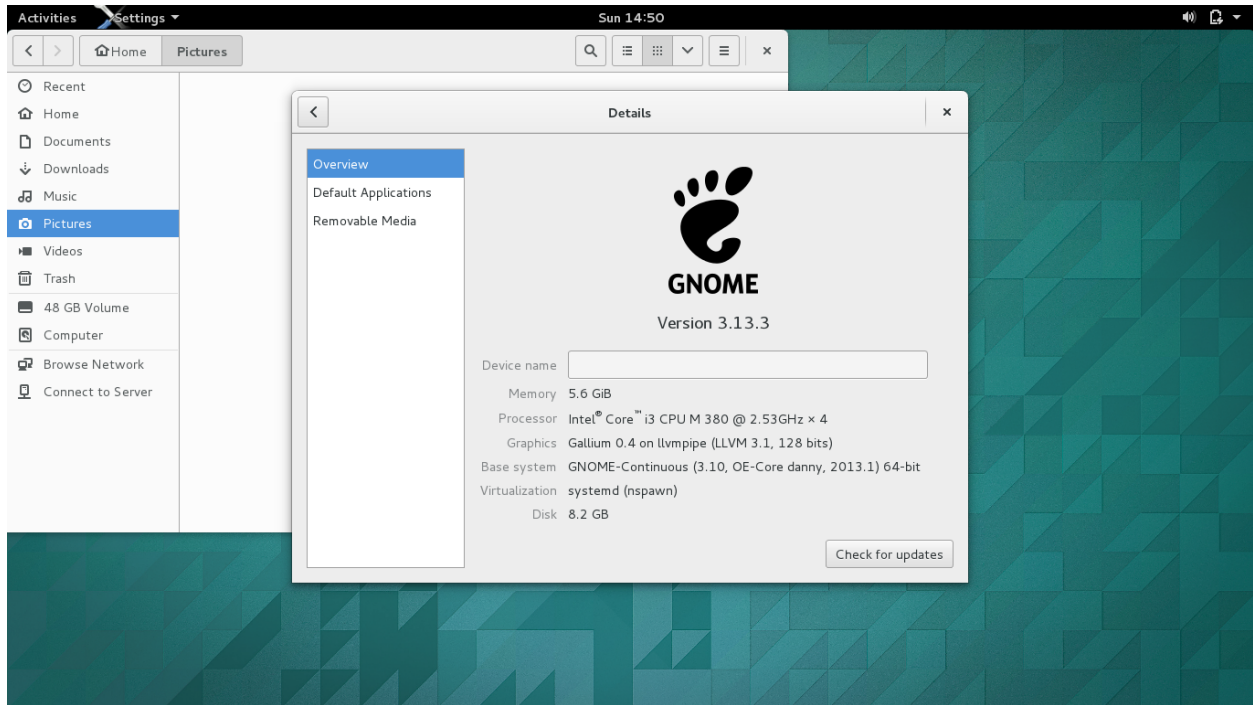
```
press ctrl-alt-f1 and login as root, then type this command
# X :1 vt2
```

Above step will start another X server in vt2 (you can switch between different X servers using ctrl-alt-f[1-9]).

Inside Systemd-Nspawn type,

```
root@qemux86-64:~# DISPLAY=:1 gnome-session
```

This will start gnome session in host machine's vt2 (ctrl-alt-f2) and show the latest gnome version built by OSTree which you downloaded as qcow2 image in the first step,



Also, you can verify which ostree version the current qcow2 image pointing to, open a gnome-terminal in the gnome-session started in vt2 (ctrl-alt-f2) and type the below command,

```
root@gemux86-64:~# ostree admin status
* gnome-continuous 9b441d1f14c4caea17eb2b50a778da4a08e31cbbdf1104edf2cd2cb40a4e3374.0
  origin refspeg: gnome-continuous:gnome-continuous/buildmaster/x86_64-devel-debug
root@gemux86-64:~#
```

you can also update to the latest version of gnome using following command,

```
root@gemux86-64:~# ostree admin upgrade
```

To stop the gnome-session, go to the terminal where you started it and press ctrl-c. It will stop the gnome-session on vt2 (ctrl-alt-f2) and give root prompt, now type following command to shutdown the systemd-nspawn machine

```
root@gemux86-64:~# systemctl poweroff
```

You will be back to host machine. Now, type following commands to peacefully disconnect everything,

```
# umount /dev/loop0
# umount /dev/loop0
# rm root
# rm -fr /dev/exported
# losetup -D
# qemu-nbd --disconnect /dev/nbd0
```

Thats it, the qcow2 image will be properly closed. Well, if you don't understand the steps in this post, comment here, I'll try to explain as much as possible.

SYSTEMCTL RESTART MYBLOG

Updated: May 07, 2024

Its been 1yr 3months since I published my last post, lot of changes happened in my life. Just a quick look at my past

Marrage

I got married on 06-Dec-2013 to Deepa (an introvert, but always want to be with her friends). Atlast, there is someone who will scold me when I don't eat/sleep/bath properly. I got a nice traveller now to travell with me rest of my life. My stomach is not burning when I go to multiplexes and malls after marrage :)

Moved to Pune

Moved to beautiful Pune on Oct-2013. Nice city. But, I'm missing [ilugc](#) meetings very much. I have to start going to lug meeting in pune.

GStreamer

Learnt [GStreamer](#) basics and wrote a module for [dhvani-tts](#) which can use gstreamer for playing sound.

Well, nice to be back to blogging after a long gap.

FETCHMAIL

Updated: May 07, 2024

As you already know from my previous [Postfix \(with smtp.gmail.com as relayhost\)](#) article, Any mail going through drunkenmonk.org domain will have my gmail address as sender. That's good. What about receiving mail from gmail?

In your mind, you are saying It's damn easy right? configure your evolution/thunderbird to point to gmail stupid? right? yeah, that's right. I can configure any mail client to point to gmail and get mails. But, I want my postfix to play a role.

Postfix is basically a smtp server, it can also act like smtp client. But, it can't work as a imap/pop3 client.

So, Here is the task, I need to get mails from gmail through pop3 and put it into my postfix so that all other clients connected with my postfix can get that mail. Here comes fetchmail.



Fetchmail is a beautiful program which will act like imap/pop3 client (just like evolution/thunderbird) and instead of saving it somewhere, fetchmail will put that mail into your local machine's mail server. So, all the mail clients connected with your local mail server will get your email from gmail.

That's enough of theory, let's get down and install fetchmail, first install the package

```
$ sudo apt install fetchmail
$
```

Now time to configure fetchmail, it uses /etc/fetchmailrc as its only configuration file. But, we need to be careful with it. Because, we are saving the username/password in plain-text, so the permission for /etc/fetchmail should be (0600) for root.

If you start configuring fetchmail by reading man fetchmail, you will switch off easily. That man page is such a damn brief and will confuse you easily. But, fetchmail config is really a easy one. It will take just three lines to configure it. Here are those three lines,

```
set daemon 120
set syslog
poll pop.gmail.com proto pop3 user "foo" pass "bar" to "stupid" ssl sslproto "TLS1"
```

Here, `set daemon 120` will make fetchmail to connect to gmail server every 2 minutes. `set syslog` will make fetchmail to put logs into syslog. `poll pop.gmail.com` instructs fetchmail to connect to pop.gmail.com server. `proto pop3` instructs fetchmail to use pop3 protocol. `user 'foo' pass 'bar' to 'stupid'` instructs fetchmail to use `foo` as username and `bar` as password to open gmail account in gmail server, fetch those mail from gmail server and put it into local machine's postfix as a mail for `stupid` user. `ssl` instructs fetchmail to use secure connection. `sslproto 'TLS1'` tells fetchmail to use STARTTLS for secure connection.

Thats it for configuration stuff, now its time to restart fetchmail

```
$ sudo eselect rc restart fetchmail
$
```

Now, you can see your mails are fetched from gmail and available to you through `mailx` command.

DOVECOT (WITH MBOX)

Updated: May 07, 2024

There are different ways to save a mail in *nix systems. The old baddy is `mbox` which saves all your email in one single file. The new one is `Maildir`.

Even though `mbox` have difficulties, traditional commandline mail clients like `mailx` works with `mbox` style mail files flawlessly. Also postfix which I setup in my [Postfix \(with smtp.gmail.com as relayhost\)](#) article by default creates single `mbox` mail file inside `/var/spool/mail/` directory for each user. So, I need a `imap+pop3` server which can operate with `mbox` like mail files. `Dovecot` is the choice.



First we need to install dovecot. I'm using sabayon, the instruction is specific to sabayon/gentoo,

```
$ sudo equo install net-mail/dovecot
```

Its time to setup dovecot. Before we do, we need to create self signed certificates to use it for `imaps` and `pop3s`. Lets first create those certificates.

```
$ cd /etc/ssl/dovecot
$ sudo mkdir oldcerts
$ sudo mv * oldcerts
$ sudo openssl genrsa -out server.key.password -des3 1024
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
Enter pass phrase for server.key.password:
Verifying - Enter pass phrase for server.key.password:
$
```

We have to give `pass` phrase here, otherwise `openssl` command will not create key file. But, having `pass` phrase for key file is not good, because we need to provide this `pass` phrase everytime `dovecot` access this file. Here is the step to remove the `pass` phrase from `rsa` key file

```
$ sudo openssl rsa -in server.key.password -out server.key
Enter pass phrase for server.key.password:
writing RSA key
$
```

Now generate certificate request with the rsa key file

```
$ sudo openssl req -out server.csr -new -key server.key
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:TamilNadu
Locality Name (eg, city) []:Chennai
Organization Name (eg, company) [Internet Widgits Pty Ltd]:DrunkenMonk Private Limited
Organizational Unit Name (eg, section) []:Pattasarayam Generating Unit
Common Name (e.g. server FQDN or YOUR name) []:drunkenmonk.org
Email Address []:webmaster@drunkenmonk.org

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
$
```

We should not give password when creating certificate request. Ok, now time to create self signed certificate.

```
$ sudo openssl x509 -req -in server.csr -out server.crt -signkey server.key -days 365
Signature ok
subject=/C=IN/ST=TamilNadu/L=Chennai/O=DrunkenMonk Private Limited/OU=Pattasarayam
Generating Unit/CN=drunkenmonk.org/emailAddress=webmaster@drunkenmonk.org
Getting Private key
$
```

Now we have two files `/etc/ssl/dovecot/server.key` and `/etc/ssl/dovecot/server.crt` to use it for SSL. Let's configure dovecot now, we need to modify `/etc/dovecot/dovecot.conf` like this,

```
protocols = imap pop3
listen = *, ::
```

Configuration not ends with `/etc/dovecot/dovecot.conf`, it has different conf files for different purpose inside `/etc/dovecot/conf.d/`, let's modify one by one, Here is `/etc/dovecot/conf.d/10-auth.conf`

```
auth_mechanisms = plain login
!include auth-system.conf.ext
#!include auth-sql.conf.ext
#!include auth-ldap.conf.ext
#!include auth-passwdfile.conf.ext
#!include auth-checkpassword.conf.ext
#!include auth-vpopmail.conf.ext
#!include auth-static.conf.ext
```

If you want to permit only current machine's users and don't want to use ldap or other mechanisms, then make sure you comment all includes except `auth-system.conf.txt`.

Here is modifications inside `/etc/dovecot/conf.d/10-logging.conf`

```
log_path = syslog
syslog_facility = mail
```

Here is modifications inside /etc/dovecot/conf.d/10-mail.conf

```
mail_location = mbox:~/mail:INBOX=/var/mail/%u
```

Now /etc/dovecot/conf.d/10-ssl.conf

```
ssl = yes
ssl_cert = /etc/ssl/dovecot/server.crt
ssl_key = /etc/ssl/dovecot/server.key
```

Thats all from config stuff, lets restart dovecot.

```
$ sudo eselect rc restart dovecot
```

Now, we need to test if it is working, We need to connect with dovecot through imap port with TLS encryption

```
$ openssl s_client -connect localhost:143 -starttls imap
```

Above command will show lot of SSL stuffs, and then finally dovecot will say . OK, we need to start communication with dovecot from there,

```
. OK Pre-login capabilities listed, post-login capabilities have more.
a login mokka somepassword
* CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE IDLE SORT_
↳ SORT=DISPLAY THREAD=REFERENCES THREAD=REFS MULTIAPPEND UNSELECT CHILDREN NAMESPACE_
↳ UIDPLUS LIST-EXTENDED I18NLEVEL=1 CONDSTORE QRESYNC ESEARCH ESORT SEARCHRES WITHIN_
↳ CONTEXT=SEARCH LIST-STATUS SPECIAL-USE
a OK Logged in
b select inbox
* FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
* OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Draft \*)] Flags permitted.
* 0 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 1343790396] UIDs valid
* OK [UIDNEXT 320] Predicted next UID
* OK [NOMODSEQ] No permanent modsequences
b OK [READ-WRITE] Select completed.
c list "" *
* LIST (\HasNoChildren) "/" "INBOX"
c OK List completed.
```

The above session with dovecot shows that things are ok. You can now configure your Thunderbird or Evolution to use your machine for local emails. Have a nice day!!

POSTFIX (WITH SMTP.GMAIL.COM AS RELAYHOST)

Updated: May 07, 2024

Most of us forget that postfix not only acts like a mail-server, but it can act like a smtp-client (like thunderbird, evolution).

This post is all about configuring postfix to work as mail-server for my local domain (drunkenmonk.org) at the same time forward any outside domain mails to smtp.google.com with my gmail account credentials. If you are curious about drunkenmonk.org, read my previous articles about [Dnsmasq for Home User](#) and [Virtual Home Network \(with Dnsmasq, Bridge, TUN/TAP, Qemu\)](#). These two articles will explain how I setup my home network.

First things first, lets install postfix (Well I'm using sabayon)

```
$ sudo equo install mail-mta/postfix
```

BASIC MAIL SERVER

Postfix configuration is pretty simple and straight forward. You have to configure following parameters inside `/etc/postfix/main.cf`

```
mydomain = drunkenmonk.org
myorigin = $mydomain
inet_interfaces = all
mydestination = localhost, localhost.$mydomain, $myhostname, $mydomain, mail.
↳$mydomain
mynetworks_style = subnet
```

`mydomain` tells postfix that it should serve as mail server for `drunkenmonk.org` domain.

`myorigin` tells that postfix should append `drunkenmonk.org` to any incoming mail's address if the address don't have domain part. Which means, when you do `mail someuser` from commandline, postfix will append `drunkenmonk.org` so that To address will become `someuser@drunkenmonk.org`.

`inet_interfaces` tells postfix to listen on all network interfaces.

`mydestination` tells postfix that it should be the final destination for `localhost`, `localhost.drunkenmonk.org`, `mokka.drunkenmonk.org`, `drunkenmonk.org` and `mail.drunkenmonk.org`

`mynetworks_style` tells postfix that my local network is a subnet type.

Thats all needed for postfix to get start. We can restart postfix with this configurations to send/receive mails between local users. However we want a little bit more from postfix. Lets configure it to use STARTTLS for its communication with its clients

BASIC MAIL SERVER WITH STARTTLS

If you wonder what is **STARTTLS**, its a simple method to establish communication in a secure way. Your server application (like Postfix listening on port 25) says Hey client I have TLS support, If you want, we can communicate each other in TLS secure way within port 25 itself, If you don't want, we can continue with plain-text, what you say?, then your client (like Evolution or Thunderbird) will tell Ooh!! you offer TLS, then lets do TLS!!. So the packets going between server and client will be encrypted using TLS.

Before STARTTLS, people used different port for secure smtp which is smtps(port 465). But nowadays it is changing, the regular smtp port itself been used for both plain-text and SSL/TLS communication.

We have to configure Postfix to talk in SSL/TLS way, for that we need Self Signed SSL certificate. Lets create one,

```
$ cd /etc/ssl/postfix
$ sudo mkdir oldcerts
$ sudo mv * oldcerts
$ sudo openssl genrsa -out server.key.password -des3 1024
Generating RSA private key, 1024 bit long modulus
.....++++++
.....++++++
e is 65537 (0x10001)
Enter pass phrase for server.key.password:
Verifying - Enter pass phrase for server.key.password:
$
```

Here you have to give a pass phrase to generate RSA private key, but having a key with pass phrase is asking for trouble. You have to manually type this password everytime postfix access this key. The following command will remove pass phrase from a RSA key

```
$ sudo openssl rsa -in server.key.password -out server.key
Enter pass phrase for server.key.password:
writing RSA key
$
```

Now you have a RSA private key file server.key without pass phrase. Its time to create certificate request,

```
$ sudo openssl req -out server.csr -new -key server.key
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:TamilNadu
Locality Name (eg, city) []:Chennai
Organization Name (eg, company) [Internet Widgits Pty Ltd]:DrunkenMonk Pvt.
↳Ltd
Organizational Unit Name (eg, section) []: Pattasarayam Generating Unit
Common Name (e.g. server FQDN or YOUR name) []:MokkaPandi
Email Address []:webmaster@drunkenmonk.org

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```


Don't give any password when creating certificate request. Finally create a self signed ssl certificate using following command,

```
$ sudo openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
Signature ok
subject=/C=IN/ST=TamilNadu/L=Chennai/O=DrunkenMonk Pvt Ltd/OU= Pattasarayam
Generating Unit/CN=MokkaPandi/emailAddress=webmaster@drunkenmonk.org
Getting Private key
$
```

Ok, thats it. You will have two files, one is /etc/ssl/postfix/server.crt and another is /etc/ssl/postfix/server.key, Now configure postfix to use these two for SSL/TLS communication, Add the following lines at the end of /etc/postfix/main.cf

```
# TLS Support
smtpd_tls_cert_file = /etc/ssl/postfix/server.crt
smtpd_tls_key_file = /etc/ssl/postfix/server.key
smtpd_tls_security_level = may
```

Finally, start postfix

```
$ sudo eselect rc restart postfix
```

Check if postfix can operate with STARTTLS,

```
$ sudo openssl s_client -connect localhost:25 -starttls smtp
```

Above command will show the server certificate and all SSL stuffs. After this, whatever you send will be encrypted, now check if it can send mail,

```
250 DSN
EHLO drunkenmonk.org
250-mokka.drunkenmonk.org
250-PIPELINING
250-SIZE 10240000
250-VERFY
250-ETRN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
mail from: root@drunkenmonk.org
250 2.1.0 Ok
rcpt to: webmaster@drunkenmonk.org
250 2.1.5 Ok
data
354 End data with .
Subject: test mail
test mail
.
250 2.0.0 Ok: queued as E0E352C0091
quit
221 2.0.0 Bye
closed
```

Now see if root got a test mail

```
$ sudo mail
>N 8 root@drunkenmonk Tue Aug 7 16:50 14/533 test mail
&p
From root@drunkenmonk.org Tue Aug 7 16:50:23 2012
X-Original-To: webmaster@drunkenmonk.org
Subject: test mail
Date: Tue, 7 Aug 2012 16:49:48 +0530 (IST)
From: root@drunkenmonk.org

test mail

&q
$
```

We are not done yet. There is one more part. Lets make postfix to forward mails to gmail.

MAIL FORWARDER

Till now, postfix can send/receive mails within drunkenmonk.org domain. But if someone from your subnet want to send a mail to a person who have mail address outside drunkenmonk.org domain (let say stuid@yahoo.com), then your current postfix will try to contact mail server of yahoo.com. But, yahoo's mail server will reject your mail because drunkenmonk.org is not a valid registered domain.

But, If you have a gmail account, you can forward the mail first to smtp.gmail.com with your account credentials so that gmail will send your mail to stupid@yahoo.com with your gmail account credentials. The person who owns stupid@yahoo.com will receive a mail with from address as yours irrespective of whoever send within your local network.

To make postfix to know your credentials, we need to create a credential db with gmail account credentials, For that we need to add following line at the end of /etc/postfix/saslpass file

```
# run following command to generate hash password db
# that can be used inside postfix's main.cf
#
# command:
# postmap hash:saslpass
#
#file content format:
#remotehost username:password
smtp.gmail.com foouser:barpassword
```

Now run the following command to create saslpass.db

```
$ cd /etc/postfix
$ sudo postmap hash:saslpass
```

Now you can see saslpass.db created as a Berkeley DB file. We can now use this file inside main.cf to inform postfix about your gmail account credential. Before doing that, please remove /etc/postfix/saslpass file or change its access to 0600 for root. Because your credentials are in plaintext. We need to add following lines at the end of main.cf

```
# gmail
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/saslpass
smtp_sasl_security_options = noanonymous
smtp_tls_security_level = encrypt
smtp_tls_mandatory_ciphers = high
```

(continues on next page)

(continued from previous page)

```
smtp_tls_verify_cert_match = hostname, nexthop, dot-nexthop
relayhost = smtp.gmail.com:587
```

Here `hash:/etc/postfix/saslpass` actually makes postfix to access `/etc/postfix/saslpass.db` file. You can remove `/etc/postfix/saslpass` which have your gmail password as plaintext. It will not affect postfix and postfix doesn't need it.

Thats it, we are done. Once you restart postfix, It becomes capable of sending mail to outside world but with your gmail account's credentials. Thanks for reading all the way down to this line. Have a nice day!!

VIRTUAL HOME NETWORK (WITH DNSMASQ , BRIDGE, TUN/TAP, QEMU)

Updated: May 07, 2024

As I said in my [Dnsmasq for Home User](#) article, this post will explain how we can use bridge interface to configure networks inside virtual hosts and much more about virtualization.

First we need to make sure our machine is capable of kvm virtualization, see if you get any output for below command

```
$ grep -E 'vmx|svm' /proc/cpuinfo
```

Most modern processors supports Hardware assisted Virtualization. If you don't get any output for above command, It means you don't have a processor capable of providing [Hardware Assisted Virtualization](#). Qemu-kvm will not work, but you can use Qemu without kvm or some other virtualization applications like [virtualbox](#) etc. Also skip [LOADING KERNEL MODULES](#) section of this article if you don't have kvm support.

LOADING KERNEL MODULES

If your processor supports kvm, then we have to load following kernel modules and make sure they load automatically whenever we restart our system. In gentoo/sabayon, we need to modify `modules=` line in `/etc/conf.d/modules` file as

```
modules="kvm-intel tun"
```

Here `kvm-intel` is the kernel module for virtualization and `tun` is the kernel module for TUN/TAP devices. For, AMD machines, we need to load `kvm-amd` instead of `kvm-intel`. Now we need to load these modules for the current run, Here is the commands which will load `kvm-intel` and `tun` drivers into kernel

```
$ sudo modprobe kvm-intel tun
$
```

INSTALLING PACKAGES

Next is to install required packages, Here is sabayon command to install required packages

```
$ sudo equo install app-emulation/qemu-kvm net-misc/bridge-utils net-dns/
↪ dnsmasq sys-apps/usermode-utils
$
```

`qemu-kvm` is for virtualization, `bridge-utils` is to get `brctl` command, `dnsmasq` is to handle DNS and DHCP requests from guests, `usermode-utils` is to get `tunctl` command.

CREATING INTERFACES

Its time to create interfaces, first we need to create the bridge

```
$ sudo brctl addbr br0
```

Assign IP address to the bridge,

```
$ sudo ifconfig br0 192.168.2.1 netmask 255.255.255.0 up
```

Next, create a tap0 interface, following command will create tap0 pseudo slave interface from /dev/net/tun master interface

```
$ sudo tuncctl -t tap0
```

Now, we need to hook the tap0 interface with the bridge br0

```
$ sudo brctl addif br0 tap0
```

Bring up the tap0

```
$ sudo ifconfig tap0 up
```

FIREWALL RULES AND MASQUERADING

If you read my previous ref:[dnsmasq-for-home-user](#), I have dnsmasq configured to serve DHCP requests coming from br0. Hooking tap0 into br0 will make the guest send DHCP requests via tap0->br0 and my host machine's dnsmasq process will serve ip address to the guest. We have to make sure iptables dont block BROADCAST packets as well as open the port 53 and 64 so that dnsmasq will get the DHCP and BOOTP packets. Make sure your iptables contains following lines,

```
$ sudo iptables -t filter -L | grep -E 'BROADCAST|domain|bootp'
ACCEPT      all  --  anywhere                    anywhere                    ADDRTYPE match_
->src-type BROADCAST
ACCEPT      udp  --  anywhere                    anywhere                    udp dpt:bootps
ACCEPT      udp  --  anywhere                    anywhere                    udp dpt:domain
ACCEPT      tcp  --  anywhere                    anywhere                    tcp_
->dpt:domainflags: FIN, SYN, RST, ACK/SYN
ACCEPT      tcp  --  anywhere                    anywhere                    tcp_
->dpt:bootpsflags: FIN, SYN, RST, ACK/SYN
$
```

Now we need to make sure MASQUERADING is enabled and let your host system forward the packets. Also make these settings permanent so that reboot dont break the settings,

```
$ sudo iptables -t nat -A POSTROUTING -j MASQUERADE
$ sudo sysctl net.ipv4.ip_forward=1
$ sudo /etc/init.d/iptables save
$ sudo /etc/init.d/iptables reload
$ sudo sed -i 's/^.*net.ipv4.ip_forward.*$/net.ipv4.ip_forward = 1/g' /etc/
->sysctl.conf
```

Check these settings are ok,

```
$ sudo iptables -t nat -L
Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination
MASQUERADE all  --  anywhere              anywhere
$ sudo sysctl -a | grep ipv4.ip_forward
```

(continues on next page)

(continued from previous page)

```
net.ipv4.ip_forward = 1
$
```

STARTING QEMU GUEST

Finally, its time to start virtual guest, first we need to create disk image, Here is the command to create a 5GB raw disk image for qemu-kvm

```
$ sudo qemu-img create -f raw Debian.img 5G
$
```

Now download debian netinstall image for amd64 architecture

```
$ sudo wget http://ftp.nl.debian.org/debian/dists/squeeze/main/installer-
→amd64/current/images/netboot/gtk/mini.iso
$
```

Atlast, start the guest with with tap0 networking

```
$ qemu-kvm -cpu kvm64 -drive file=Debian.img,if=virtio -cdrom mini.iso -boot_
→order=dc -m 512 -soundhw sb16 -name "Debian" -net nic,model=virtio -net_
→tap,ifname="tap0",script=no,downscript=no
```

Thats it, your qemu guest will automatically get network settings from host machine. You can also verify that dnsmasq served a DHCP request via syslog

```
$ sudo grep 'dnsmasq-dhcp' /var/log/messages
dnsmasq-dhcp[9810]: DHCPDISCOVER(br0) 192.168.2.89 52:54:00:12:34:56
dnsmasq-dhcp[9810]: DHCPOFFER(br0) 192.168.2.89 52:54:00:12:34:56
dnsmasq-dhcp[9810]: DHCPREQUEST(br0) 192.168.2.89 52:54:00:12:34:56
dnsmasq-dhcp[9810]: DHCPACK(br0) 192.168.2.89 52:54:00:12:34:56 virt0
$
```

Well, if you still reading this article and not get bored, you must be a *nix admin. Have a great day!!

DNSMASQ FOR HOME USER

Updated: May 07, 2024

Every BSNL broadband subscribers know that BSNL's default nameservers are one of the worst in response times. Even if you have connectivity exceeding 1gbps, If your nameservers are not good, then your internet experience will not be good.

So, I decided to use google's nameservers. I could have configure NetworkManager not to get nameservers through DHCP instead ask it to use google's nameservers, but I thought of giving dnmasq a try.



For those who don't know what is mean by [dnsmasq](#), it is a dns forwarder which can handle DNS, DHCP and BOOTP requests. It means, it can act like [bind](#), [dhcpd](#) and [tftpd](#).

Configuring dnsmasq is pretty simple and straight forward. We have to modify two files `/etc/hosts` and `/etc/dnsmasq.conf`. Here is my `/etc/hosts` file

```
127.0.0.1    mokka.drunkenmonk.org mokka localhost
::1         mokka.drunkenmonk.org mokka localhost
192.168.2.1  bridge.mokka.drunkenmonk.org
```

The first line says `127.0.0.1` is the ip address for `mokka.drunkenmonk.org` and also for `mokka` and `localhost` hostnames. I other words, `mokka`, `localhost` and `mokka.drunkenmonk.org` will resolve to `127.0.0.1` ip address.

`mokka.drunkenmonk.org` is a cononical name which contains two parts, `mokka` as hostname and `drunkenmonk.org` as domainname. So, by putting `drunkenmonk.org`, I'm also setting domainname for my system.

I really don't know the proper way to configure domainname in my sabayon (in other words gentoo). From googling, I came to know that putting domain `drunkenmonk.org` into `/etc/resolv.conf` is the right way to set domainname in a linux system. But, `/etc/resolv.conf` is such dynamic nowadays, we can't be sure who will modify it later and I never see a router providing domainname in DHCP response (Not in BSNL's ADSL routers for sure). So I endup putting my domainname in `/etc/hosts` file.

The second line is same like the first one, but for ipv6.

The third line says `192.168.2.1` is the ip address for `bridge.mokka.drunkenmonk.org`. I'll come to this later. Now, take a look at my `/etc/dnsmasq.conf`

```
domain-needed
bogus-priv
no-resolv
```

(continues on next page)

(continued from previous page)

```

server=8.8.8.8
server=8.8.4.4
local=/drunkenmonk.org/
domain=drunkenmonk.org
dhcp-range=interface:br0,192.168.2.2,192.168.2.254,255.255.255.0,1d
mx-host=drunkenmonk.org,mail.drunkenmonk.org,30
cname=drunkenmonk.org,bridge.mokka.drunkenmonk.org
cname=www.drunkenmonk.org,bridge.mokka.drunkenmonk.org
cname=mail.drunkenmonk.org,bridge.mokka.drunkenmonk.org

```

domain-needed instructs dnsmasq to never forward DNS queries which don't have domain part. It means, If you query for nslookup google dnsmasq will never forward it to upstream server (In my case, to google's nameservers).

bogus-priv tells dnsmasq never forward reverse-lookup queries which have local subnet's ip range to upstream. Which means nslookup 10.0.0.1 will not be forwarded to upstream instead dnsmasq will try to resolve itself. If it doesn't find hostname for 10.0.0.1 in /etc/hosts or its dhcp leases, then it will send back no such domain response.

no-resolv says dnsmasq will not read /etc/resolv.conf to get upstream nameservers. Normally dnsmasq will read /etc/resolv.conf file to get upstream nameservers.

server=8.8.8.8 and server=8.8.4.4 instructs dnsmasq to use google's 8.8.8.8 and 8.8.4.4 as primary and secondary DNS nameservers (or in other words upstream nameservers).

local=/drunkenmonk.org/ says that DNS queries with hostnames like mokka.drunkenmonk.org should not be forwarded to upstream instead dnsmasq should resolve it from /etc/hosts or from its dhcp leases.

domain=drunkenmonk.org tells dnsmasq to send domain=drunkenmonk.org in DHCP response so that machines configured through DHCP will come under drunkenmonk.org domain.

dhcp-range=interface:br0,192.168.2.2,192.168.2.254,255.255.255.0,1d instructs dnsmasq to allocate ip address between 192.168.2.2 and 192.168.2.254 with netmask 255.255.255.0 for 1 day only to the DHCP requests coming from br0 interface. Which means, machines connected through br0 interface and asking for DHCP response will get ip address between 192.168.2.2 and 192.168.2.254 for 1 day. dnsmasq will not send response to requests coming from interfaces other than br0.

mx-host=drunkenmonk.org,mail.drunkenmonk.org,30 tells dnsmasq to send mail.drunkenmonk.org as MX response for drunkenmonk.org domain. Which means, when you do nslookup -q=MX drunkenmonk.org it will give mail.drunkenmonk.org as response.

cname lines are like aliases. means when you do nslookup www.drunkenmonk.org it will resolve to bridge.mokka.drunkenmonk.org which resolves to 192.168.2.1 according to /etc/hosts file. So dnsmasq will send 192.168.2.1 as response.

That's all from configuration stuff. These changes will take effect once you restart dnsmasq service. But this does not mean that all your DNS queries will go through your machine's dnsmasq daemon, If you have nameserver entry in /etc/resolv.conf, your machine's DNS queries will go to /etc/resolv.conf nameservers. So, there is no use of using dnsmasq. You have to make sure you are not creating entries in /etc/resolv.conf. If you are a home user, make sure you select Automatic (DHCP) Address only in Method drop-down list inside ipv4 settings tab in NetworkManger.

You can check that you are getting response from dnsmasq with below commands,

```

$ dig ANY drunkenmonk.org
; <> DiG 9.9.1-P2 <> ANY drunkenmonk.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39480
;; flags: qr aa rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

```

(continues on next page)

(continued from previous page)

```
;; QUESTION SECTION:
;drunkenmonk.org.          IN      ANY
;; ANSWER SECTION:
drunkenmonk.org.          0      IN      CNAME    bridge.mokka.drunkenmonk.org.
bridge.mokka.drunkenmonk.org. 0 IN A      192.168.2.1
;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Aug 6 00:15:16 2012
;; MSG SIZE rcvd: 99
```

The line contains **SERVER:** tell us that our DNS query is served by 127.0.0.1, means our local machine. Here is the command to verify MX record,

```
$ dig MX drunkenmonk.org
; <> DiG 9.9.1-P2 <> MX drunkenmonk.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14301
;; flags: qr aa rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;drunkenmonk.org.          IN      MX
;; ANSWER SECTION:
drunkenmonk.org.          0      IN      MX      30 mail.drunkenmonk.org.
;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Aug 6 00:16:38 2012
;; MSG SIZE rcvd: 71
```

If you are not introduced to [Linux's bridge interface](#), you may wonder what the hell is br0. Well, it is a virtual device in Linux which act like a physical network device (like eth0). I have particularly configured my dnsmasq to serve DHCP requests to whoever connect through my bridge interface. Howto use this br0 effectively to configure networking in my Qemu virtual machines is going to be my next *Virtual Home Network (with Dnsmasq, Bridge, TUN/TAP, Qemu)* article. Have a nice day!!

GNU GETTEXT - YET ANOTHER TUTORIAL

Updated: May 07, 2024

Well, developing a simple C program is easy, but developing it in a internationalized way (Yeah!! all those `l10n`, `m17n` and `i18n` thingy) is not so easy unless you understand Autotools. However, understanding it may take some time (atleast it took some time for me). In this post, I'm trying to explain how I learnt it. It may be wrong way, but atleast I can recollect what I did today in future.

Lets just create a simple C project. Obviously without any doubt, it should be called `helloworld`. Lets just create the directory tree first.

```
$ mkdir -p helloworld/{src,man}
```

Switch to `helloworld/src` and create two files `helloworld.h` and `helloworld.c`

```
/* helloworld/src/helloworld.h */
#ifndef __HELLOWORLD__
#define __HELLOWORLD__

#define _GNU_SOURCE

#include <stdio.h>
#include <stdlib.h>
#include <libintl.h>
#include <locale.h>

#define _(STRING) gettext(STRING)

#endif
```

We have to include `libintl.h` to get `bindtextdomain(3)`, `textdomain(3)` and `gettext(3)` functions. We have to include `locale.h` to get `setlocale(3)` function, let see why we need these functions,

`setlocale()`

Every glibc executable starts with the default locale called `C`. We use `setlocale()` function to switch to different locale, this function takes two parameters `category` and `locale`, `category` indicates which locale variable we want to change and `locale` contains what is the new value, If `locale` is `""`, `setlocale()` will get the value from the corresponding environment variable (see man page for more details).

`bindtextdomain()`

The gettext framework works as follows,

- Get all the english output strings from the sources and generate a `.pot` file

- Translate the english strings in the .pot file in different language and create .po file for each language
- Generate .gmo binary files from the .po file for each language
- Make the executable read corresponding translation from the .gmo file according to the locale settings each time it wants to print a message

To do the last step, we have to specify where the .gmo files are available. For that purpose, we use `bindtextdomain()`, it takes two arguments, `domainname` and `dirname`. `domainname` is the name we choose to group all our .gmo files under one place. Most of the time, we use the name of our project as `domainname`. `dirname` is the common directory where different project's .gmo files were placed. Usually it is `/usr/share/locale`.

`textdomain()`

we have to set the `textdomain` so that executable will get the translated messages from the .gmo files correctly. `textdomain()` takes only one argument `domainname` which is the name of our project.

`gettext()`

Finally we have to wrap every output string to make them pass through `gettext()` so that it can catch the correct translated string from .gmo files. We defined a macro `_()` alias to `gettext()` because we are lazy(aren't we!?) to type `gettext()` everytime.

So, here is the `helloworld.c`

```
/* helloworld/src/helloworld.c */
#include <helloworld.h>

int main(int argc, char *argv[])
{
    setlocale(LC_ALL, "");

#ifdef ENABLE_NLS
    bindtextdomain(PACKAGE, LOCALEDIR);
    textdomain(PACKAGE);
#endif

    printf(_("hello world\n"));

    return (0);
}
```

Now, we have to replace `PACKAGE` and `LOCALEDIR` macros to the real values. Here comes autotools, automake can give real value to `PACKAGE` at compile time and automake also have a way to define `LOCALEDIR` at compile time, Lets do autotools by creating following files,

```
# helloworld/src/Makefile.am
bin_PROGRAMS = helloworld
helloworld_SOURCES = helloworld.c helloworld.h
DEFS += -DLOCALEDIR=\"$(localedir)\"
```

```
# helloworld/man/helloworld.1
helloworld :) !!! check after sometime
to see the real man page
```

```
# helloworld/man/Makefile.am
dist_man_MANS = helloworld.1
```

```
# helloworld/Makefile.am
SUBDIRS = src man
```

We need to run `autoscan` to generate `configure.scan` file. Rename `configure.scan` to `configure.ac` and edit that file according to the project's need. I can't explain all the `autoconf` macros within this blog post, see the end of this blog post to get the links for further reading.

```
$ cd helloworld
$ autoscan
```

Here is the customized `configure.ac` file,

```
# helloworld/configure.ac
#                                     -*- Autoconf -*-
# Process this file with autoconf to produce a configure script.

AC_INIT([helloworld], [0.1], [mokka at comedysite dot com])
AC_CONFIG_SRCDIR([src/helloworld.c])

# Automake init
AM_INIT_AUTOMAKE([foreign -Wall])

# Checks for programs.
AC_PROG_CC
AM_PROG_CC_C_O

# Gettext init
AM_GNU_GETTEXT_VERSION([0.18])
AM_GNU_GETTEXT([external])

# Checks for libraries.

# Checks for header files.
AC_CHECK_HEADERS([libintl.h locale.h stdlib.h])

# Checks for typedefs, structures, and compiler characteristics.

# Checks for library functions.
AC_CHECK_FUNCS([setlocale])

AC_CONFIG_FILES([Makefile
                 man/Makefile
                 src/Makefile])
AC_OUTPUT
```

Now we have to run `gettextize` under `helloworld` directory to put `gettext` settings into `configure.ac` and `Makefile.am`.

```
$ cd helloworld
$ gettextize
```

If things go well, you can see `helloworld/po` directory and modifications into `configure.ac` and `Makefile.am`. Now we can run `autoreconf` to finish `autotools` procedure.

```
$ cd helloworld
$ autoreconf --force --install --verbose
```

Now, switch to `helloworld/po` and rename `Makevars.template` to `Makevars`. Inside `Makevars` file, you

may have to give inputs to some variables, may be atleast to MSGID_BUGS_ADDRESS, Here is a way to add your email address to that variable

```
$ cd helloworld/po
$ mv Makevars.template Makevars
$ sed -i '/^MSGID/s/$/mokka at comedytime dot com/g' Makevars
```

Now, we need to add the source filenames to POTFILES.in, Here a way,

```
$ cd helloworld/po
$ find ../src -name '*.c' -o -name '*.h' | sed 's/\.\/\.\.\/g' >> POTFILES.in
```

Time to compile,

```
$ cd helloworld
$ ./configure
$ make
```

You can see PACKAGE, LOCALEDIR macro definitions when make compile helloworld.c. As a programmer, your job is almost done.

Now switch yourself as a translator. go to helloworld/po directory and generate a po file for your language using msginit, you have to provide locale using -l option. You should know the [languagecode](#) and [countrycode](#) to construct locale string. msginit will ask for your email-id to put yourself into the translators list. Here I'm translating for Tamil (ta_IN.utf8).

```
$ cd helloworld/po
$ msginit -i helloworld.pot -o ta.po -l ta_IN.utf8
```

I edited ta.po file with gedit+ibus, translated the word hello world\n to `ஹெல்லோ வர்ல்ட்\n`. Now, I have to add my language to LINGUAS file. LINGUAS file contains languagecodes which have corresponding translated .po file inside helloworld/po directory.

```
# helloworld/po/LINGUAS
ta
```

Now its time to generate binary .gmo file. Before that, We have to re-run autoreconf to regenerate the helloworld/po/Makefile.in, because we updated LINGUAS file.

```
$ cd helloworld
$ make distclean
$ autoreconf --force --install --verbose
$ cd po
$ make update-gmo
rm -f ta.gmo && /usr/bin/gmsgfmt -c --statistics --verbose -o ta.gmo ta.po
ta.po: 1 translated message.
$
```

If your translation don't have any errors, you will see 1 translated message. Few more steps to achieve our goal, that is, creating distribution tarball and install our program to see the result.

```
$ cd helloworld
$ make distclean
$ make dist-bzip2
$ mkdir -p /tmp/buildir
$ mv helloworld-0.1.tar.bz2 /tmp/buildir
$ cd /tmp/buildir
```

(continues on next page)

(continued from previous page)

```
$ tar xvjf helloworld-0.1.tar.bz2
$ cd helloworld-0.1
$ ./configure --prefix="/tmp/destdir"
$ make install
$ LANG="ta_IN.utf8" /tmp/destdir/bin/helloworld
???????
$ /tmp/destdir/bin/helloworld
hello world
$
```

Thats it. My helloworld program can say வணக்கம் now. You can also make it to speak your favourite language!!

References

- [autoconf](#)
- [automake](#)
- [libtool](#)
- [gettext](#)
- [make](#)

There is another beautiful tutorial for gettext available at oriya.sarovar.org.

PCAPSTREAMER - A PACKET DUMPER

Updated: May 07, 2024

Hi guys,

Happy Holidays!!

This year is about to finish, thinking about this year, lot of things happened, love, break-up, home, health, mother, father, work and more importantly passion. Well, all is well and life is moving ahead. Still I'm travelling alone, in my own path. (What the hell am I, this is suppose to be technical post, shit!! crap philosophy!!)

I got some free time and spent that time learning [libpcap](#). For those who don't know, it is used in most of the network monitoring/capturing tools in *nix world. Very powerful.

The [tcpdump\(1\)](#) command is one such tool which uses libpcap (actually they are the one who created libpcap from tcpdump) to dump information about packets. It has a robust filtering mechanism to narrow down packet capturing to specific packets.

While trying to understand filter expressions in tcpdump, I got an Idea, I thought why not just convert the bytes in packets to strings and print them in stdout, this way, we can see the exact bytes, so further processing can be done by other unix tools (like awk, perl etc.,).

So, I just wrote a tool called [pcapstreamer](#) to capture packets from linux's any pseudo-interface. Its very simple tool, you need to run this tool as root user. It just dump packets, that's all. Here is an example, this shows one packet dumped into stdout.

```
$ sudo ./pcapstreamer
[cl:76 l:76 t:20111226085033.641612] 00000000 00000000 00000011 00000100 00000000_
→00000110 00000000 00000000 00000000 00000000 00000000 00000000 00000000_
→00001000 00000000 01000101 00000000 00000000 00111100 01000111 10001100 01000000_
→00000000 01000000 00000110 11110101 00101101 01111111 00000000 00000000 00000001_
→01111111 00000000 00000000 00000001 11100101 01100011 00010101 10110011 01111000_
→00011101 00100110 01010100 00000000 00000000 00000000 00000000 10100000 00000010_
→10000000 00011000 11111110 00110000 00000000 00000000 00000010 00000100 01000000_
→00001100 00000100 00000010 00001000 00001010 00000000 10100010 01100001 11011000_
→00000000 00000000 00000000 00000000 00000001 00000011 00000011 00000101
```

Here cl:76 and l:76 indicates captured length, t:20111226085033.641612 indicates timestamp in localtime. Other strings are just pure raw packet.

Linux Cooked Header

To understand first 16 bytes, we need to understand [Linux Cooked Header](#). First 2 bytes 00000000 00000000 or 0x00 represents that this is an incoming packet. To understand the next 2 bytes, we need to refer [linux's ARPHRD_](#). 3rd and 4th bytes 00000011 00000100 or decimal 772 indicates that this packet is coming into loopback interface. 5th and 6th bytes 00000000 00000110 or 0x0006

indicates the length of link-level address, the next 8 bytes (7th byte to 14th byte) represents the link-level address, however we should take only the next 6 bytes as link-level address, two more bytes (13th and 14th) are padded with zero. 15th and 16th bytes 00000000 00001000 or 0x0008 represents `ethertype` as `ip`, this tells us that this is an `ip` packet. This ends the link-level header (data-link layer in OSI). We are now moving to `ip` header (network layer in OSI)

IP Header

To understand details from 17th byte to 36th byte, we need to refer `IP Header`. Higher order 4 bits in 17th byte 0100 or 0x4 indicates that this `ip` packet is an `ipv4` packet. Lower order 4 bits in 17th byte 0101 or 0x5 indicates `IHL` (Internet Header Length) usually this defaults to 5. 18th bytes represents `differentiated services` usually 0. 19th and 20th bytes 00000000 00111100 or 0x003c or `Decimal 60` represents remaining bytes count (`CaptureLength` minus `Linux-Cooked-Header` length). 21st and 22nd bytes 01000111 10001100 indicates identification. Higher order 3 bits in 23rd and 24th bytes 010 indicates that this packet is not fragmented, remaining 13 bits indicates fragment offset. 25th byte 01000000 or 0x40 or `Decimal 64` indicates `TTL` value. 26th byte 00000110 or 0x06 indicates that this is a `tcp` packet. 27th and 28th packets indicates `Header Checksum`. 29th to 32nd bytes indicates source `ip` address (127.0.0.1) and 33rd to 36th byte indicates destination `ip` address (127.0.0.1). This ends the `ip` header, we are now moving to `tcp` header (Transport layer in OSI).

TCP Header

To understand details from 37th byte to 76th byte, we need to refer `TCP Header`. 37th and 38th bytes 11100101 01100011 or `decimal 58723` indicates the source port number. 39th and 40th bytes 00010101 10110011 or `decimal 5555` indicates destination port number (means incoming packet is trying to connect port 5555). 41st to 44th byte indicates sequence number and 45th to 48th byte indicates sequence acknowledgement number. Higher order 4 bits in 49th and 50th byte 1010 or `Decimal 10` indicates `Data offset`, means there are $10*4=40$ bytes in `TCP` header. Next 3 higher order bits are reserved in 49th byte 50th byte. Next 3 bits indicates `ECN`. Next 6 bits 000010 or 0x02 indicates that `SYN` flag was set in `Control` bits. 51st and 52 bytes indicates window size, means the sender is willing to accept 10000000 00011000 or `decimal 32792` bytes in the response packet. 53rd and 54th bytes indicates checksum. 55th and 56th bytes indicates `Urgent pointer`, usually 0.

Options

Inside `TCP` header, bytes 57 to 76 contains value based on 50th byte(`Data Offset`). In this particular packet, 50th byte has (0xa), which means, `TCP` header in this packet contains totally 40bytes. Mandatory `TCP` fields (from 37th byte to 56th byte) are already discussed, but we have 20 more bytes to decode, these bytes are represented as `Options` in `TCP` header. They may occur or they may not occur in a `TCP` packet. Mostly they occur in `SYN` packet.

Here, 57th byte (0x02) represents option-kind, 58th byte represents option-length (0x04). Both 57th and 58th bytes represents that 59th and 60th bytes 01000000 00001100 or 0x400c or `Decimal 16396` indicates `Maximum Segment Size`. 61st byte (0x04) represents option-kind, 62nd byte (0x02) represents option-length, both bytes represents `SACK permitted`. 63rd byte (0x08) represents option-kind, 64th byte (0x0a) represents option-length, both bytes indicates that from 65th byte to 68th byte contains `TSVal` and from 69th byte to 72nd byte contains `TSecr`. 73rd byte 0x01 indicates option-kind as `No-Operation`, 74th byte (0x03) indicates option-kind, 75 byte indicates (0x03) option-length, both bytes indicates that 76th byte (0x05) contains `WSOpt (Window Scale Option)`, which means, the host which sent this packet can accept upto $32792 * (2^5)$ or $(\text{window size}[\text{byte51\&52}] * (2^{\text{wsopt}[\text{byte76}]})$ before sending `ACK`.

pcapstreamer with awk

To display only `ICMP` packets, we can use the following commandline

```
$ sudo ./pcapstreamer 2>/dev/null | awk '{if($26 ~ "00000001"){print $0;}}'
```

To display only `SYN` packets, we can use the following commandline

```
$ sudo ./pcapstreamer 2>/dev/null | awk '{ctrlbytes=$49$50; if(ctrlbytes ~ "^  
↪.....000010...$"){print $0;}}'
```

I hope this utility may be useful for newbies like me to learn networking. Have a great new year.

SYSTEMD FOR SIMPLE BACKUP

Updated: May 07, 2024

Hi,

Nice to get back, Recently switched to F15. Wow!! my userland changed heavily. Previously, its very simple things like,

- SysVinit for booting
- Udev for devices
- pppconfig for Network
- Xfce for GUI
- ALSA for sound
- ffmpeg & mplayer for video

Now,

- Systemd for booting, daemons
- Udev, Dbus, UDisks for devices
- ModemManager+NetworkManager for Network
- Gnome3 for GUI
- PulseAudio+ALSA for sound
- GStreamer+totem for video

These Technologies are very interesting to learn. I'm not going to explain about each. But, I went through systemd and came up with a solution for my backup problem.

Backup Problem

While in its last phase, my previous laptop taught me the importance of external backup disks. So, I bought an external 512GB Segate, and backed-up /home/\${HOME} tree. It helped me to quickly get back my files to F15. However, one problem is, maintaining my backup. I thought If I plug my external drive, someone should automatically copy all the new files resides in my current /home/\${HOME} in F15 to that drive. There are lot of ways to do it. I can think of two main ways,

- Write an udev rule to call a script which will mount that external backup partition and rsync /home/\${HOME} in F15 to that external partition. Pros: Fairly straight forward, udevrule-file+rsyncscript will do it. Cons: No control, this will copy every time udev detects that external drive. If there is lot of files to copy, then this will make a mess.

- Use systemd to call a script whenever that external backup partition gets mounted. **Pros**: You have full control, create a backup.service for systemd, create backup.bash to rsync /home/\${HOME} to backup disk. Enable that service in systemd to automatically do rsync, or just load that service to systemd and only start that service If you want to backup your files. **Cons**: Need to pass one more layer to run the actual rsync script.

So, I took-up systemd

There are lot of ways you can trigger your service in systemd. Also, you can depend on another systemd unit to trigger your service. In this case, I depend on a mount unit to trigger my backup.service. This service, in-turn, will trigger backup.sh script.

To define a systemd unit, you need to know what type of unit you want to create. Currently there are 10 types of units systemd can understand [read systemd.unit(5)]. For backup job, I used two type units, one is systemd.mount(5) and systemd.service(5). If you go through systemd.mount(5) you will understand that systemd will automatically load this units whenever systemd saw a block device. So, systemd will automatically provide media-ExternalBackupPartition.mount whenever I insert my external hard disk. I only need to define the next unit, backup.service for systemd.

Note: There is a story behind the name media-ExternalBackupPartition. I'll tell you at the end of this post. Lets just continue with systemd for now.

To define a unit for systemd, you need to create a file as /etc/systemd/system/name.unit; (for my backup problem, unit file is /etc/systemd/system/backup.service). Here name, may be anything relevant to your job and unit must be one of service, socket, device, mount, automount, swap, target, path, timer and snapshot. Read systemd.unit(5) man pages for more precise information.

Unit Class

Unit definition files contains information in .ini format. One of the class [Unit] must exist in every unit file. Here is the [Unit] class for backup.service

```
[Unit]
Description='sync my files with external backup drive'
Requires=media-ExternalBackupPartition.mount
After=media-ExternalBackupPartition.mount
```

Here Description is a general description for your service. We need to put the required unit which will trigger this new unit in Requires field. Systemd will run this unit once all the units in Requires fields satisfied. We can call these Requires units as parent units, and your unit as child. However, systemd will not wait for parent units to complete to run child unit. We need to explicitly ask systemd to wait for parent units to complete, For this purpose we have After field. If you define which parent unit needs to be completed before your child unit could run, you need to mention it in After field.

For my backup.service child unit, media-ExternalBackupPartition.mount unit must be satisfied in systemd. That means, my external HD partition must be mounted inside /media/ExternalBackupPartition path. Also, Using After= field, I instructed systemd, not to start this child unit before mount-ExternalBackupPartition.mount finishes.

Service Class

Now we need to define what to do once the requirements satisfies. For that purpose, we need to define [Service] class, Here is the service class for backup.service,

```
[Service]
Type=simple
ExecStart=/home/mohan/Development/scripts/backup.sh
```


This [Service] class is specific to systemd.service units. Here, ExecStart=/home/mohan/Development/scripts/backup.sh asks systemd to run backup.sh whenever backup.service satisfies. You can do lot of customization to setup the execution environment before start running any commands, such as log redirection, demonizing etc., there are lot of fields to use in [Service] class, but I simply used Type=simple to tell systemd, that no need to do any change in execution environment. The script backup.sh will take care of all the redirection within itself.

Install Class

In SysV init system, we use chkconfig (redhat/fedora) or update-rc.d (debian) to enable or disable a service. In systemd, we use this [Install] class to enable or disable our backup.service unit so that it will work even after a restart. Here is install class for backup.service,

```
[Install]
WantedBy=media-ExternalBackupPartition.mount
```

In systemd, enabling a service means, adding a symlink to /etc/systemd/system/name.service.wants/ directory. disabling a service means, removing that symlink. systemctl command can do this add/remove symlink automatically when we call it with systemctl enable backup.service or systemctl disable backup.service, but we need to say the parent unit name, thats why we have this [Install] class. Simply, we need to mention that parent unit in WantedBy= field.

Execution

Once the unit files are ready, we need to enable them into systemd. For backup.service, I executed following commands to setup the service

```
$ sudo cp ~/backup.service /etc/systemd/system/backup.service
$ sudo systemctl daemon-reload
```

I just copied backup.service unit file to systemd's location and asked systemd to reload unit definitions. Now we can check if things loaded properly or not using following command,

```
$ sudo systemctl status backup.service
backup.service - 'sync my files with external backup drive'
   Loaded: loaded (/etc/systemd/system/backup.service)
   Active: inactive (dead)
     CGroup: name=systemd:/system/backup.service

$
```

systemd will say Loaded : error if it can't understand any definition in backup.service or if it can't satisfy the definitions. Otherwise we can start this service using following command, we need to make sure the final rsync script backup.sh exists in the location pointed by ExecStart= field.

```
$ sudo systemctl start backup.service
```

This will start syncing new files to External backup HD drive, only when It is plugged-in and mounted. Otherwise, the service will fail. you can check the status again using systemctl status. Once you checked that the service is working as intended, we can enable this service (I mean, creating symlinks) using following command,

```
$ sudo systemctl enable backup.service
```

We can verify the symlink as below to make sure parent-child linking is done correctly.

```
$ ls -l /etc/systemd/system/media-ExternalBackupPartition.mount.wants/backup.
→service
lrwxrwxrwx 1 root root 34 Nov  9 02:08 /etc/systemd/system/media-
```

(continues on next page)

(continued from previous page)

```
↳ ExternalBackupPartition.mount.wants/backup.service -> /etc/systemd/system/  
↳ backup.service
```

If we don't want to copy automatically, we can disable it using below command,

```
$ sudo systemctl disable backup.service
```

Even If the service is disabled, you can start/stop the service. Systemd will recognize the backup.service, check it's dependencies and execute backup.sh correctly.

media-ExternalBackupPartition.mount

As I already said, systemd will automatically create units using udev, so when my external HD plugs-in, udev will tell to udisks that a new partition is available, then udisks will mount that partition inside /media/uuid location, then systemd will create a unit as media-uuid.mount (systemd uses - instead of / for path separation). But specifying Requires=media-uuid.mount inside backup.service file is not working. Thus, I used a simple udev rule to rename udisk's mount path, here is the rule file,

```
$ cat 99-rename-udisk-mountpoint.rules  
ENV{ID_FS_UUID}=="251c683d-bce0-489c-aab5-f684a9a1f3b2", ENV{ID_FS_UUID}=  
↳ "ExternalBackupPartition"  
$ sudo cp ~/99-rename-udisk-mountpoint.rules /etc/udev/rules.d
```

This above two commands, will modify udisk's mount path to /media/ExternalBackupPartition instead of /media/251c683d-bce0-489c-aab5-f684a9a1f3b2, thus systemd will automatically create media-ExternalBackupPartition.mount instead of media-251c683d-bce0-489c-aab5-f684a9a1f3b2.mount

Finally, Thanks for reaching this line. I hope this long boring article will help you to understand something about systemd.

CREATING FEDORA DVD REPO

Updated: May 07, 2024

Hi,

Its almost 1 year since my last post, what happens to me? It a 1 year story to say. But in short, my laptop became older and eventually went down. Graphics card gone. It took a while for me to buy another new one. Here I am.

In between lot thing happened, Something I just wanted to post, but now I can't remember them. As a fresh start, This is what happened when I got my new Samsung Notebook. Ordered through FlipKart and got it by last week.

I know that I can't get Internet because I cancelled my netconnect+ subscription way back when my previous lappy started giving troubles. Thus, My beloved debian is out of reach. So, I decided to get Fedora or Mint DVD from someone and Install it. Thanks to Srini, Amachu and his Gang, for providing Fedora DVD to me. Installation went smoothly and for the first time, I'm in Gnome3,



Fig. 1: Fedora 15 Desktop

Installation went smoothly and to my surprise, no more tinkering on mobile modem. Thanks to those guys who created profile for Aircel. At first, bluetooth daemon didn't start, googled and fixed it.

While playing with Gnome3, I thought of using installation DVD as my repo, because I have limited network with very little speed, It would be better to use existing DVD as my repo. So, that starts my hunger and this is how I ended.

1. Inserted my DVD and monitored how my DVD is mounting. Seems `org.freedesktop.UDisks` takes care of mounting and it always mount my DVD inside `/media` folder taking mount directory name from `ENV{ID_FS_LABEL_ENC}` Udev environment variable. I came to know that yum is facing some trouble If I have a `fedora-dvd.repo` file like below under `/etc/yum/yum.repos.d` directory,

```
[fedora-dvd]
name=Fedora DVD $releasever - $basearch
baseurl=file:///media/Fedora 15 i386 DVD
enabled=1
gpgcheck=1
gpgkey=file:///media/Fedora 15 i386 DVD/RPM-GPG-KEY-fedora-$basearch
```

So, it seems yum cannot handle white-space in URI. Then, I decided to write a Udev rule to rename `Fedora 15 i386 DVD` to `Fedora15i386DVD`, So that `org.freedesktop.UDisks` will mount it as `file:///media/Fedora15i386DVD`. Here is the rule file

```
$ cat /etc/udev/rules.d/99-rename-fedora-dvd.rules
ENV{ID_FS_LABEL_ENC}=="Fedora\x2015\x20i386\x20DVD", ENV{ID_FS_LABEL_ENC}=
↳ "Fedora15i386DVD"
$
```

Then, I recreated `fedora-dvd.repo` as below

```
$ cat /etc/yum.repos.d/fedora-dvd.repo
[fedora-dvd]
name=Fedora DVD $releasever - $basearch
baseurl=file:///media/Fedora15i386DVD
enabled=1
gpgcheck=1
gpgkey=file:///media/Fedora15i386DVD/RPM-GPG-KEY-fedora-$basearch
$
```

One of the main reason to add this `Fedora15i386DVD` repo is to install development tools without internet. In a way it is more useful than a plain LiveCD.

Don't forget to insert Fedora DVD If you follow this setup and want to install something from Fedora DVD. A little google tells me that there are lot of ways to achieve this DVD-to-Repo thing in Fedora, I'm relatively new to Fedora, I will learn as I move on.

HOWTO BAKE PERSISTENT LIVE USB - DEBIAN WAY (LIVE-BUILD)

Updated: May 07, 2024

Hey guys!! wish you happy festive days!!

For long time I was thinking about creating my own system. Recently, during my experiment with `mic2`, I came to know about `live-build` package.

`mic2` is a package which contains tools to create Meego images. I'm yet to create my Meego Image. During Meego image creation, I had lot to do with my USB pendrive. My pendrive previously had puppy for system rescue purpose. After spending much time with Meego without any success, I thought of creating my rescue USB by downloading puppy. But decided to give `live-build` a try.

First we need to install `live-build`

```
$ sudo apt-get install live-build cdebootstrap
```

`live-build` is a set of scripts to build customized debian iso/usb-hdd images. I'm not going to explain each scripts, instead, just lets start building a Debian Persistant Live USB with `rescue` packages from Testing repository for `i386` architecture. My host system is Debian Testing (amd64) Here is the steps,

Create a directory and go inside

```
$ mkdir live-build-base && cd live-build-base
```

Execute below command

```
$ sudo lb config --architecture i386 --binary-images usb-hdd --binary-filesystem_
↳fat32 --bootappend-live ``persistent`` --bootstrap cdebootstrap --distribution_
↳testing --mirror-binary http://ftp.de.debian.org/debian/ --packages ``usb-
↳modeswitch`` --packages-lists ``rescue``
```

This tells `--architecture` for the live system should be `i386`, you can give any of debian ports like `amd64`, `sparc`, `powerpc`, `armel`, `hurd-i386`, `kfreebsd-gnu` etc., to this option. But careful, we are not using `qemu` to build the image. we are using `cdebootstrap`, thus, an `amd64` host system can bake `amd64`, `i386`, `hurd-i386`, `kfreebsd-gnu` images, but it cannot bake `armel`, `sparc` or `powerpc`.

The next option `--binary-images` tells what kind of image to create, like `iso` or `iso-hybrid` or `usb-hdd` etc., (see `man lb_config`). Next option `--binary-filesystem` tells the format of the final root filesystem.

Next one `--bootappend-live` passes options to `live-initramfs` which is the main tool to create live debian system, see `man live-initramfs` to know about its options. `persistent` informs `live-initramfs` to generate `initrd` image which will search for partitions with label `live-rw` or `home-rw` at boot time, if `initrd` finds, it will instruct `aufs` to stage that partition as a `rw` layer for root filesystem.

--bootstrap cdebootstrap will use cdebootstrap while creating the image. you should install cdebootstrap in your host system first.

--distribution testing indicates which repo to download packages, you can give stable, unstable, testing, experimental, lenny, squeeze, sid and wheezy.

--mirror-binary points to the repo URL

--packages usb-modeswitch will additionally install usb-modeswitch package. I need this, because I**m using a netconnect datacard for internet access.

--packages-list rescue will get packages from /usr/share/live/build/lists/rescue file. This file is preconfigured to install specific packages for specific purpose, we can give rescue, gnome, kde, gnustep, lxde, studio etc., each have a corresponding file inside /usr/share/live/build/lists directory in your host system.

Above options prepares live-build-base directory to build your customized debian. Now, its time to build, execute the below command,

```
$ sudo lb build
```

Once started, lb build will take some time to bake the required image. Finally, you will get build.img with all your customized packages installed. Next step should be dd-ing the image to your USB pendrive. Insert it, and execute the following command,

```
$ sudo dd if=build.img of=/dev/sdX bs=4096
```

Here, /dev/sdX indicates your loaded pendrive name, it may be /dev/sdb or /dev/sdc. Once dd finishes, reboot your system and boot from your USB pendrive. You will be inside Debian Live !!!!

Next, we need to create live-rw partition in your pendrive for persistent storage (means automatically saving the changes you do in your live system). Just create a partition with label as live-rw from the free space left in your pendrive using gparted or fdisk. Then boot your machine from USB drive, once you are inside Debian Live, you can see the new partition mounded as aufs rw layer for root filesystem in /live/cow.

During lb build phase, if any error occurs, or you decide to change your configuration, remove all directories except cache from live-build-base directory. Then remove .stage directory and start lb config with your options again from live-build-base directory. This will reduce package download time, but if you are building for different architecture, then its better to delete cache also. Here is the screenshot of my image running under QEMU,



Thats it!! hope you will bake a better Debian Live cake than me!!!

MILANA (2007) - FROM A TAMIL FAN

Updated: May 07, 2024

Disclaimer: By no means this is a review. This is how I came to know about this movie and enjoyed.

One of my best friend (he is from mysore) had that song as his ringtone at that time when milana was released. Its so catchy, that whenever I was bored, I would go to his desk and start listening to that song. Yes, I'm talking about Ninnindale..

Now, I'm in different place and my friend is in different place. recently I spoke with him, suddenly my mind (What!!!!?? isn't my heart??) thought just get that song and have it in my mobile. Also I saw lot of positive reviews and massive statistics (100 days.. 200.. days in Bengaluru PVR.. don't know the exact statistics mate!!). Even though my relatives are in Bengaluru, I didn't get a chance to watch a kannada movie. This is the one which I watched for the first time.

His sharp face and uncombed hair (Is it his trademark appearance?) first gave some different opinion about Puneet Rajkumar, But his performance as Akash, showed how good he is, as a Actor. Be it as a Son, a Mirchi RJ, Husband, Friend, Fighter, Dancer everything, shows the capabilities of a mass hero(is he already? I see he is called Power Star, but nowadays every star have a subname!!). I hope he should not develop any style pattern like many stars in south. He should act for directors.

Like Akash, his wife Anjali (Parvathi menon) who ask divorce in her first night, steal some important scenes as a good Actress (Yaepulla Maari, neeya idhu, romba alaga iruka pulla). Her cute face expressions especially when she get angry because of akash's Bred & Jam flirt, indicates a good chemistry between both (is there any gosip in KFI about them? I came to know that they had another big hit, Prithvi..). Later at the end, when she suffer because Akash didn't realize her love, makes a real beautiful love story.

On the whole, This movie shows how good a family, friendship and love can bring happiness to a person's life. Really!! Its a great movie to watch. Mano Moorthy's Songs and Background Music will capture anyone's heart!! I enjoyed every frame!!! Thanks to Director Prakash and the whole team for this wonderful movie.

Somebody pleeeassseee!!!! bring this movie to Kollywood!!!!

RELIANCE NETCONNECT BROADBAND+ ON DEBIAN

Updated: May 07, 2024

Hey guys, I'm back after a loooooonnnnnnggggggg.... gap... :-)

Well, living without internet is something different. I experienced how a mobile with 2-12 KB/s will serve a person. You will get very much happy if it touches 12 KB/s.

But now..., Wow !!! I'm connected again !!! with good speed... here is the story.

Recently I moved from suburb to inner city. So, its time to get all my regular activities back into track. For that to happen, I need a good internet connection, like the one which I enjoyed from BSNL Broadband. But now, the problem is, I can't get a wired broadband as I'm currently staying in a mansion. So only choice is, to go for USB modem. Already I had a bitter experience with Photon+, so I considered other offers and found some +ve reviews about Netconnect. So, I decided to go for it.

I called one of the reseller in nungambakkam, within 4 hours a guy(lokes) appeared before me with the Card (Huawei Technologies Co., Ltd. EC1260 Wireless Data Modem HSD USB Card) and some forms. We discussed about support for linux. He said that he heard from someone that the card was working in linux (I think I'm the first linux customer to him). I googled a lot, There are lot of ubuntu stuffs related to netconnect. So I'm pretty confident that it will work in linux. Applied for it and got the card.

Tinkering starts here, At first, when I plugged it into my laptop, debian realized it as a scsi storage and loaded it as one of my DVD drive (/dev/sr0). Then I came to know that it was a switchable USB device. Obviously, *nix guys always have a solution for a popular problem. Here is one of the popular problem and the solutions is,

```
# aptitude install usb-modeswitch
```

usb-modeswitch takes care of switching these kind of devices to USB modems so that udev will load proper modules into kernel. Once you installed usb-modeswitch, you can see the following messages appear in /var/log/syslog when you plug-in the device,

```
New USB device found, idVendor=12d1, idProduct=1446
New USB device strings: Mfr=1, Product=2, SerialNumber=4
Product: HUAWEI Mobile
Manufacturer: HUAWEI TECHNOLOGIES
SerialNumber: 00000000000000000000
configuration #1 chosen from 1 choice
kernel: [10337.323907] scsi8 : SCSI emulation for USB Mass Storage devices
kernel: [10337.367370] usb-storage: device found at 4
kernel: [10337.367379] usb-storage: waiting for device to settle before scanning
usb_modeswitch: switching 12d1:1446 (HUAWEI TECHNOLOGIES: HUAWEI Mobile)
USB disconnect, address 4
new full speed USB device using ohci_hcd and address 5
```

(continues on next page)

(continued from previous page)

```

New USB device found, idVendor=12d1, idProduct=140b
New USB device strings: Mfr=1, Product=2, SerialNumber=4
Product: HUAWEI Mobile
Manufacturer: HUAWEI TECHNOLOGIES
SerialNumber: 00000000000000000000
configuration #1 chosen from 1 choice
GSM modem (1-port) converter detected
GSM modem (1-port) converter now attached to ttyUSB0
GSM modem (1-port) converter detected
GSM modem (1-port) converter now attached to ttyUSB1
GSM modem (1-port) converter detected
GSM modem (1-port) converter now attached to ttyUSB2
registered new interface driver option
v0.7.2:USB Driver for GSM modems

```

Now its time to dial, since I'm using Debian, I have two options, one is wvdial and the other one is pppconfig + pon + poff. I'm already a fan of pppconfig. So decided to create a profile using pppconfig. Here is my peers file,

```

# This optionfile was generated by pppconfig 2.3.18.
#
#
hide-password
noauth
connect "/usr/sbin/chat -v -f /etc/chatscripts/netconnect"
debug
/dev/ttyUSB0
115200
defaultroute
noipdefault
user "xxxxxxxxxx" # fill your MDN number without the leading 0 (10 digit)
remotename netconnect
ipparam netconnect

usepeerdns
novjccomp

```

and here is my chatscript

```

# This chatfile was generated by pppconfig 2.3.18.
# Please do not delete any of the comments. Pppconfig needs them.
#
# ispauth PAP
# abortstring
ABORT BUSY ABORT 'NO CARRIER' ABORT VOICE ABORT 'NO DIALTONE' ABORT 'NO DIAL TONE'
↵ABORT 'NO ANSWER' ABORT DELAYED
# modemininit
'' ATZ
# ispsnumber
OK-AT-OK "ATDT#777"
# ispconnect
CONNECT \d\c
# prelogin

# ispname
# isppassword
# postlogin

```

(continues on next page)

(continued from previous page)

```
# end of pppconfig stuff
```

Here is how you can generate your settings files using `pppconfig`

- Give a good name to your PPP profile (from here, I'll refer this name as `profilename`)
- Choose dynamic DNS
- Choose PAP Authentication
- Give your MDN number without the leading 0 (10 digit) as your Username
- Give Password same as Username
- Choose default speed (115200)
- Choose Tone dialing method
- Give #777 as the Phone Number to Dial ISP.
- Choose No for modem detection and give `/dev/ttyUSB0` as your modem
- Verify your settings and save. These settings will be saved in `/etc/ppp/peers/profilename` and `/etc/chatscripts/profilename`

Once you are done with the above steps, all you need to do is, start `pppd` using `pon`

```
[root@Debian ~] # pon profilename
```

Here comes the trickiest part in my experiment. One of the worst thing about Netconnect is, Reliance server will never respond to your packets untill you activate the device from "F`*in W`dows !!!!!". If your device is not activated, your `pppd` with cry and die with the following messages.

```
pppd 2.4.4 started by root, uid 0
Serial connection established.
using channel 4
Using interface ppp1
Connect: ppp0 /dev/ttyUSB0
sent [LCP ConfReq id=0x1  ]
sent [LCP ConfReq id=0x1  ]
sent [LCP ConfReq id=0x1  ]
sent [LCP ConfReq id=0x1  ]
sent [LCP ConfReq id=0x1  ]
sent [LCP ConfReq id=0x1  ]
sent [LCP ConfReq id=0x1  ]
sent [LCP ConfReq id=0x1  ]
sent [LCP ConfReq id=0x1  ]
sent [LCP ConfReq id=0x1  ]
sent [LCP ConfReq id=0x1  ]
LCP: timeout sending Config-Requests
Connection terminated.
Receive serial link is not 8-bit clean:
Problem: all had bit 7 set to 0
Hangup (SIGHUP)
Modem hangup
Exit.
```

Thankfully one of my new friend who is running a browsing center nearby helped me to activate the device from his windows machine. There are some steps to be followed to activate in Windows, Call the Reliance guys, they are very much familiar with the procedure.

I'm ready to connect with internet !!! I plugged the device into my laptop and started pppd. Reliance server responded nicely. Finally!!!! I got the connection. Here is the logs from /var/log/syslog,

```
pppd[32557]: pppd 2.4.4 started by root, uid 0
chat[32563]: abort on (BUSY)
chat[32563]: abort on (NO CARRIER)
chat[32563]: abort on (VOICE)
chat[32563]: abort on (NO DIALTONE)
chat[32563]: abort on (NO DIAL TONE)
chat[32563]: abort on (NO ANSWER)
chat[32563]: abort on (DELAYED)
chat[32563]: send (ATZ^M)
chat[32563]: expect (OK)
chat[32563]: ATZ^M^M
chat[32563]: OK
chat[32563]: -- got it
chat[32563]: send (ATDT#777^M)
chat[32563]: expect (CONNECT)
chat[32563]: ^M
chat[32563]: ATDT#777^M^M
chat[32563]: CONNECT
chat[32563]: -- got it
chat[32563]: send (\d)
pppd[32557]: Serial connection established.
pppd[32557]: using channel 4
pppd[32557]: Using interface ppp0
pppd[32557]: Connect: ppp0 /dev/ttyUSB0
pppd[32557]: sent [LCP ConfReq id=0x1      ]
pppd[32557]: rcvd [LCP ConfAck id=0x1      ]
pppd[32557]: rcvd [LCP ConfReq id=0x2      ]
pppd[32557]: sent [LCP ConfAck id=0x2      ]
pppd[32557]: sent [LCP EchoReq id=0x0 magic=0x2d94925b]
pppd[32557]: sent [PAP AuthReq id=0x1 user="xxxxxxxxxx" password=]
pppd[32557]: rcvd [LCP EchoRep id=0x0 magic=0xcd908d56]
pppd[32557]: rcvd [PAP AuthAck id=0x1 ""]
pppd[32557]: PAP authentication succeeded
pppd[32557]: sent [CCP ConfReq id=0x1      ]
pppd[32557]: sent [IPCP ConfReq id=0x1      ]
pppd[32557]: rcvd [IPCP ConfReq id=0x1      ]
pppd[32557]: sent [IPCP ConfAck id=0x1      ]
pppd[32557]: rcvd [LCP ProtRej id=0x1 80 fd 01 01 00 0f 1a 04 78 00 18 04 78 00 15 03_
↪2f]
pppd[32557]: Protocol-Reject for 'Compression Control Protocol' (0x80fd) received
pppd[32557]: rcvd [IPCP ConfNak id=0x1      ]
pppd[32557]: sent [IPCP ConfReq id=0x2      ]
pppd[32557]: rcvd [IPCP ConfAck id=0x2      ]
pppd[32557]: Cannot determine ethernet address for proxy ARP
pppd[32557]: local  IP address 115.184.98.253
pppd[32557]: remote IP address 220.224.141.145
pppd[32557]: primary  DNS address 202.138.103.190
pppd[32557]: secondary DNS address 202.138.117.60
pppd[32557]: Script /etc/ppp/ip-up started (pid 32583)
pppd[32557]: Script /etc/ppp/ip-up finished (pid 32583), status = 0x0
```

To stop the connection, you need to issue this command,

```
[root@Debian ~] # poff profilename
```

This is my personal request to service providers, Please provide your services to all platforms.

Anyway, the struggle seems worthy, My machine never enjoyed 100 KB/s speed in its lifetime :-)

Speedtest.net - The Global Broadband Speed Test - Iceweasel

File Edit View History Bookmarks Tools Help

http://www.speedtest.net/

Speedtest.net - The Global Broadband Speed Test

SPEEDTEST.NET™

World Results Your Results Your Summary

settings help about contact

PINGTEST.NET™

View Upload Speed

DOWNLOAD SPEED

MP3 File (5 MB) 1 min

Video Clip (35 MB) 8 min

Movie (800 MB) 188 min

Your Result

ISP Average

Test Again

New Server

SPEEDTEST.NET

7/27/2010 4:15 PM GMT

DOWNLOAD 0.57 Mb/s

UPLOAD 0.20 Mb/s

PING 264 ms

ISP: RELIANCE COMMUNICATIONS ★★★★★ 2.8/5

SERVER: Colombo

DISTANCE: ~ 450 mi

Copy Forum Link Copy Direct Link

115.242.216.11 RELIANCE COMMUNICATIONS! ★★★★★ Rate Your ISP

OOKLA

What is Speedtest.net?

Use Speedtest.net to test the speed of your Internet connection. See if you are getting what you pay for and share your results with others!

Net Index by Ookla

View a cumulative history of Speedtest.net results, and analyze how your speed compares to others in the world.

Transferring data from ads.ookla.com...

Apache/2.2.3 69.17.117.207 FoxyProxy: Disabled

Terminal - squeeze@D... Speedtest.net - The Gl...

Have a nice day !!!!

CAPTURE SCREEN AND MIC INPUT USING FFMPEG AND ALSA

Updated: May 07, 2024



Recently, interesting commands appeared in [commandlinefu](#) about x11grab using ffmpeg. All are very interesting commands and they did what I expected in my Debian. But only one thing is missing. While creating a screencast, If I want to speak, that should appear in the capturing stream.

Thanks to alsa and ffmpeg, they have a way to capture my laptop's in-built mic input. So I customized the existing commands which appeared in [commandlinefu](#) and created a small script.

```
#!/bin/sh
# simple command to capture your screen

OUTPUT="${1:?[usage] screencast.sh OUTPUTFILE}"

getgeometry()
{
    xwininfo -root |
    grep 'geometry' |
    awk '{print $2;}'
}

captureX11()
{
    ffmpeg -f alsa -itsoffset 00:00:02.000 -ac 2 -i hw:0,0 \
        -f x11grab -s $(getgeometry) -r 10 -i :0.0 \
        -sameq -f mp4 \
        -s wvga \
        -y "${OUTPUT}"
}

captureX11
```

Finally I created my first screencast using `screencast.sh` script. Here is the video

Atlast, I don't want to install one more app to create screencast. FFmpeg is with me !!!

AIRCEL GPRS WITH LINUX AND BLUETOOTH

Updated: May 07, 2024

Ah!! finally, I bought a new mobile. its Nokia 2730c,

But my struggle started here with my service provider, My main reason to bought this particular phone is to get Internet access at my home. So that I can #join at anytime. But it was a tough ask with Aircel.

At first, when I inserted my SIM card, lot of activity happens and I got a message from Aircel saying to save a settings message. After that, I received a settings message saying Pocket Internet. I already know that there is a service provided by Aircel called Pocket Internet to access internet in mobile, But when I searched more about Pocket Internet, I came to know that I can't connect to my PC and browse, because Pocket Internet is a WAP service.

So I called CustCare, but as I expected, those guys are dump about what I'm talking, First I asked whether I can connect my PC and use Pocket Internet for browsing? but no clear answers from them. After 2 calls, I called one more time, this time, CustCare person informed that I need to configure my phone to access Aircel GPRS, he also informed that if your phone is capable, the settings would have already came.

I was shocked, because my Nokia 2730c is a 3G capable mobile, but I didn't receive Aircel GPRS settings, only I received Pocket Internet settings. I was nearly at the edge to change my provider. Whenever I call them, they are not at all listening to my words, they simply saying that my mobile is not capable. I got frustrated and stopped calling them, after 2 days, I called them thinking that it would be my last call, thankfully that person responded me in a positive manner, they already have my call history and he is able to understand what my problem is, The exact problem is simple,

Aircel didn't have settings for Nokia 2730c, because its a new model

So, I didn't receive Aircel GPRS settings when I inserted my SIM. Thanks to that person, he raised a ticket to their technical team, and they rectified this issue next day itself. Day after that day, I received Aircel GPRS settings. and connected to Nokia OVI store using Aircel GPRS Access Point in my mobile.

First phase of this struggle is over, here comes the next,

It would have been a simple task if I'm a windows user, connect my mobile to laptop, use PC suite to dial-up and access internet, thats all. But, Its a task up to me to get the ppp settings from Aircel. Thankfully [Ubuntu guys have a very good writeup for aircel](#). It makes easy for me to use pppconfig and create /etc/ppp/peers/aircel file and etc/chatscripts/aircel file. Here is the steps,

```
root@Debian:/etc/ppp/peers# aptitude install pppconfig
root@Debian:/etc/ppp/peers# pppconfig aircel
```

```
select "Create a Connection named aircel" and select "Ok"
select "Dynamic DNS" and select "Ok"
select "PAP" and select "Ok"
give username as "Aircel" and select "Ok"
give password as "aircel" and select "Ok"
```

(continues on next page)

(continued from previous page)

```

pppconfig will give speed of your modem as 115200, use this default
Select "Tone" and select "Ok"
give phone number as *99***1#
Next step will be giving the modem name, I'm going to use /dev/rfcomm0 (bluetooth
↳serial device) so I selected "No" and gave '/dev/rfcomm0' as my modem name.
Finally select "Finished Write files and return to main menu", two files will be
↳created, only will be '/etc/ppp/peers/aircel' and another one is '/etc/chatscripts/
↳aircel'
select "Quit"

```

Now, I need to modify the /etc/chatscripts/aircel file and insert the following lines after ATZ init line. These two lines specifies Aircel's APN aka Access Point Name which is aircelgprs.pr, I think if you are using Aircel's postpaid service, then you need to give aircelgprs.po. But check your mobile's Aircel GPRS settings first, it will clearly tell Access Point Name.

```

' ' "ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0"
' ' AT+CGDCONT=1,"IP","aircelgprs.pr"

```

Here is the final /etc/chatscripts/aircel

```

# This chatfile was generated by pppconfig 2.3.18.
# Please do not delete any of the comments. Pppconfig needs them.
#
# ispauth PAP
# abortstring
ABORT BUSY ABORT 'NO CARRIER' ABORT VOICE ABORT 'NO DIALTONE' ABORT 'NO DIAL TONE'
↳ABORT 'NO ANSWER' ABORT DELAYED
# modeminit
' ' "ATZ"
' ' "ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0"
' ' AT+CGDCONT=1,"IP","aircelgprs.pr"
# ispsnumber
OK-AT-OK "ATDT*99***1#"
# ispconnect
CONNECT \d\c
# prelogin

# ispname
# isppassword
# postlogin

# end of pppconfig stuff

```

And here is the final /etc/ppp/peers/aircel

```

# This optionfile was generated by pppconfig 2.3.18.
#
#
hide-password
noauth
connect "/usr/sbin/chat -v -f /etc/chatscripts/aircel"
debug
/dev/rfcomm0
115200
defaultroute
noipdefault

```

(continues on next page)

(continued from previous page)

```
user "Aircel"
remotename aircel
ipparam aircel
usepeerdns
```

Now, if you connected your mobile through wire, your mobile will be accessible through any one of /dev/ttyS* or /dev/ttyUSB* devices, you can start pon aircel and get connected to net. But I'm going to setup bluetooth, so here is the steps,

```
root@Debian:/etc/ppp/peers# hcitool scan
root@Debian:/etc/ppp/peers# BDADDR="[type your mobile's bluetooth address here, it_
↳will be displayed as the output of above command]"
root@Debian:/etc/ppp/peers# hcitool cc "${BDADDR}"; hcitool auth "${BDADDR}"
```

Now, your authentication-agent will ask for pin and all, go through it and create a pair. Next step will be detecting the channel to setup a rfcomm bind,

```
root@Debian:/etc/ppp/peers# sdptool browse "${BDADDR}" | less
```

In the above command, there will be lot of services your mobile will provide to you, each will have separate rec handle, name etc., some of them like obex push, obex file transfer, Headset Gateway, Phonebook access etc. You need to see Dial-up Networking and note down the RFCOMM channel number. In my mobile, the RFCOMM channel number is 1. So here is the step to bind your mobile's RFCOMM channel with /dev/rfcomm0 device

```
root@Debian:/etc/ppp/peers# CHANNEL="[Type your mobile's Dial-up service's RFCOMM_
↳channel number]"
root@Debian:/etc/ppp/peers# rfcomm bind rfcomm0 "${BDADDR}" "${CHANNEL}"
```

bind will create a communication line between your mobile and /dev/rfcomm0 device. But it will not connect to your mobile until it is opened for reading/writing. You can check current rfcomm devices using following command

```
root@Debian:/etc/ppp/peers# rfcomm
rfcomm0: 3C:F7:2A:46:19:AB channel 1 clean
root@Debian:/etc/ppp/peers#
```

Now everything was setup, all you need to do is, type the following command

```
root@Debian:/etc/ppp/peers# pon aircel
```

it will start pppd and try to connect, but it will not throw any information in your terminal, you need to watch the activities through tail -f /var/log/syslog. If everything goes fine, it will establish ppp connection to Aircel. Here is the whole log

```
Mar 23 02:02:00 Debian bluetoothd[2072]: link_key_request (sba=00:1A:6B:E3:80:D8, _
↳dba=3C:F7:2A:46:19:AB)
Mar 23 02:02:01 Debian chat[31061]: abort on (BUSY)
Mar 23 02:02:01 Debian chat[31061]: abort on (NO CARRIER)
Mar 23 02:02:01 Debian chat[31061]: abort on (VOICE)
Mar 23 02:02:01 Debian chat[31061]: abort on (NO DIALTONE)
Mar 23 02:02:01 Debian chat[31061]: abort on (NO DIAL TONE)
Mar 23 02:02:01 Debian chat[31061]: abort on (NO ANSWER)
Mar 23 02:02:01 Debian chat[31061]: abort on (DELAYED)
Mar 23 02:02:01 Debian chat[31061]: send (ATZ^M)
Mar 23 02:02:01 Debian chat[31061]: send (ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0^M)
Mar 23 02:02:02 Debian chat[31061]: send (AT+CGDCONT=1,"IP","aircelgprs.pr"^M)
```

(continues on next page)

(continued from previous page)

```

Mar 23 02:02:02 Debian chat[31061]: expect (OK)
Mar 23 02:02:02 Debian chat[31061]: ATZ^M^M
Mar 23 02:02:02 Debian chat[31061]: OK
Mar 23 02:02:02 Debian chat[31061]: -- got it
Mar 23 02:02:02 Debian chat[31061]: send (ATDT*99***1#^M)
Mar 23 02:02:02 Debian chat[31061]: expect (CONNECT)
Mar 23 02:02:02 Debian chat[31061]: ^M
Mar 23 02:02:02 Debian chat[31061]: ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0^M^M
Mar 23 02:02:02 Debian chat[31061]: OK^M
Mar 23 02:02:02 Debian chat[31061]: AT+CGDCONT=1,"IP","aircelgprs.pr"^M^M
Mar 23 02:02:02 Debian chat[31061]: OK^M
Mar 23 02:02:04 Debian chat[31061]: ATDT*99***1#^M^M
Mar 23 02:02:04 Debian chat[31061]: CONNECT
Mar 23 02:02:04 Debian chat[31061]: -- got it
Mar 23 02:02:04 Debian chat[31061]: send (\d)
Mar 23 02:02:05 Debian pppd[31042]: Serial connection established.
Mar 23 02:02:05 Debian pppd[31042]: using channel 9
Mar 23 02:02:05 Debian pppd[31042]: Using interface ppp0
Mar 23 02:02:05 Debian pppd[31042]: Connect: ppp0 <--> /dev/rfcomm0
Mar 23 02:02:05 Debian pppd[31042]: rcvd [LCP ConfReq id=0x0 <auth pap> <mru 1500>
↳<asynctest 0xa0000>]
Mar 23 02:02:05 Debian pppd[31042]: sent [LCP ConfReq id=0x1 <asynctest 0x0> <magic_
↳0xcd88324c> <pcomp> <accomp>]
Mar 23 02:02:05 Debian pppd[31042]: sent [LCP ConfAck id=0x0 <auth pap> <mru 1500>
↳<asynctest 0xa0000>]
Mar 23 02:02:05 Debian pppd[31042]: rcvd [LCP ConfRej id=0x1 <magic 0xcd88324c>
↳<pcomp> <accomp>]
Mar 23 02:02:05 Debian pppd[31042]: sent [LCP ConfReq id=0x2 <asynctest 0x0>]
Mar 23 02:02:05 Debian pppd[31042]: rcvd [LCP ConfAck id=0x2 <asynctest 0x0>]
Mar 23 02:02:05 Debian pppd[31042]: sent [LCP EchoReq id=0x0 magic=0x0]
Mar 23 02:02:05 Debian pppd[31042]: sent [PAP AuthReq id=0x1 user="Aircel" password=
↳<hidden>]
Mar 23 02:02:05 Debian pppd[31042]: rcvd [LCP EchoRep id=0x0 magic=0x0]
Mar 23 02:02:05 Debian pppd[31042]: rcvd [PAP AuthAck id=0x1 ""]
Mar 23 02:02:05 Debian pppd[31042]: PAP authentication succeeded
Mar 23 02:02:05 Debian pppd[31042]: sent [CCP ConfReq id=0x1 <deflate 15> <deflate(old
↳#) 15> <bsd v1 15>]
Mar 23 02:02:05 Debian pppd[31042]: sent [IPCP ConfReq id=0x1 <compress VJ 0f 01>
↳<addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns3 0.0.0.0>]
Mar 23 02:02:05 Debian pppd[31042]: rcvd [IPCP ConfReq id=0x0 <addr 10.6.6.6>]
Mar 23 02:02:05 Debian pppd[31042]: sent [IPCP ConfAck id=0x0 <addr 10.6.6.6>]
Mar 23 02:02:05 Debian pppd[31042]: rcvd [LCP ProtRej id=0x0 80 fd 01 01 00 0f 1a 04_
↳78 00 18 04 78 00 15 03 2f]
Mar 23 02:02:05 Debian pppd[31042]: Protocol-Reject for 'Compression Control Protocol
↳' (0x80fd) received
Mar 23 02:02:06 Debian pppd[31042]: rcvd [IPCP ConfRej id=0x1 <compress VJ 0f 01>]
Mar 23 02:02:06 Debian pppd[31042]: sent [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns1_
↳0.0.0.0> <ms-dns3 0.0.0.0>]
Mar 23 02:02:06 Debian pppd[31042]: rcvd [IPCP ConfNak id=0x2 <addr 10.1.6.46> <ms-
↳dns1 202.148.202.3> <ms-dns3 202.148.202.4>]
Mar 23 02:02:06 Debian pppd[31042]: sent [IPCP ConfReq id=0x3 <addr 10.1.6.46> <ms-
↳dns1 202.148.202.3> <ms-dns3 202.148.202.4>]
Mar 23 02:02:06 Debian pppd[31042]: rcvd [IPCP ConfAck id=0x3 <addr 10.1.6.46> <ms-
↳dns1 202.148.202.3> <ms-dns3 202.148.202.4>]
Mar 23 02:02:06 Debian pppd[31042]: Cannot determine ethernet address for proxy ARP
Mar 23 02:02:06 Debian pppd[31042]: local IP address 10.1.6.46
Mar 23 02:02:06 Debian pppd[31042]: remote IP address 10.6.6.6

```

(continues on next page)

(continued from previous page)

```
Mar 23 02:02:06 Debian pppd[31042]: primary   DNS address 202.148.202.3
Mar 23 02:02:06 Debian pppd[31042]: secondary DNS address 202.148.202.4
Mar 23 02:02:06 Debian pppd[31042]: Script /etc/ppp/ip-up started (pid 31085)
Mar 23 02:02:07 Debian pppd[31042]: Script /etc/ppp/ip-up finished (pid 31085),↵
↵status = 0x0
Mar 23 02:07:05 Debian pppd[31042]: Terminating on signal 15
Mar 23 02:07:05 Debian pppd[31042]: Connect time 5.0 minutes.
Mar 23 02:07:05 Debian pppd[31042]: Sent 70568 bytes, received 287622 bytes.
Mar 23 02:07:05 Debian pppd[31042]: Script /etc/ppp/ip-down started (pid 32324)
Mar 23 02:07:05 Debian pppd[31042]: sent [LCP TermReq id=0x3 "User request"]
Mar 23 02:07:05 Debian pppd[31042]: rcvd [LCP TermAck id=0x3]
Mar 23 02:07:05 Debian pppd[31042]: Connection terminated.
Mar 23 02:07:05 Debian pppd[31042]: Script /etc/ppp/ip-down finished (pid 32324),↵
↵status = 0x0
Mar 23 02:07:05 Debian pppd[31042]: Exit.
```

To stop your pppd, type the following command

```
root@Debian:/etc/ppp/peers# poff aircel
```

Finally, once you finished with pppd, release the rfcomm0 bind using the following command,

```
root@Debian:/etc/ppp/peers# rfcomm release rfcomm0
```

Thats it. Enjoy old style pppd :) Always old tech rules !!!

WIRELESS AD-HOC NETWORK - LINUX

Updated: May 07, 2024

Long time back, I actually did peer-to-peer with RJ232 connector using mine and my friend's laptops. Its really simple and straight forward. Choose a local subnet(probably 192.168.2.0/24), assign 192.168.2.1 to one machine and assign 192.168.2.2 to another machine. Thats all, connection established and I used `nc` to transfer files.

This time, its the same old thing, but without RJ232 wire. Also I just configured my system as a gateway so that my friend can access Internet through my laptop.

Before starting, I conformed that my wireless is working,

```
$ su -c 'iwconfig'
lo          no wireless extensions.

eth1        no wireless extensions.

eth3        IEEE 802.11bg  ESSID:""
            Mode:Managed  Frequency:2.412 GHz  Access Point: Not-Associated
            Bit Rate:54 Mb/s   Tx-Power:off
            Retry min limit:7   RTS thr:off   Fragment thr:off
            Power Managementmode:All packets received
            Link Quality=5/5  Signal level=0 dBm  Noise level=0 dBm
            Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
            Tx excessive retries:0  Invalid misc:0  Missed beacon:0

pan0        no wireless extensions.

eth0        no wireless extensions.
```

First I announced a Wireless Ad-Hoc with a good name(ESSID Dark), here, there is something called WEP, but I'm still not able to set WEP passphrase through `iwconfig`. If someone had success through `iwconfig` please help me to learn. The below ad-hoc network will not use WEP or any authentication.

```
$ su -c 'iwconfig eth3 essid 'Dark' mode ad-hoc channel 1'
```

waited for 2 - 10 seconds, and then tried to connect to that Dark network.

```
$ su -c 'iwconfig eth3 essid 'Dark' mode ad-hoc channel 1'
```

I tried the above command more than one time till I'm associated with a Cell. If you get a mac address in Cell, it means you are now connected to the Dark Network.

```
$ su -c 'iwconfig eth3'
eth3      IEEE 802.11bg  ESSID:"Dark"
          Mode:Ad-Hoc  Frequency:2.412 GHz  Cell: F6:A5:B9:E6:37:E3
          Bit Rate=54 Mb/s   Tx-Power:32 dBm
          Retry min limit:7   RTS thr:off   Fragment thr:off
          Power Managementmode:All packets received
          Link Quality=5/5   Signal level=-57 dBm  Noise level=-96 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0  Missed beacon:0
```

Now, my link-layer is configured, next is to configure the ip layer,

```
$ su -c 'ifconfig eth3 192.168.2.1 netmask 255.255.255.0'
$ su -c 'ifconfig eth3; route'
eth3      Link encap:Ethernet  HWaddr 00:1a:73:9f:d3:47
          inet addr:192.168.2.1  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::21a:73ff:fe9f:d347/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:2
          TX packets:0 errors:6 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:19

Kernel IP routing table
Destination    Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.2.0    *              255.255.255.0   U        0      0        0 eth3
192.168.1.0    *              255.255.255.0   U        0      0        0 eth0
link-local     *              255.255.0.0     U       1000    0        0 eth0
default        192.168.1.1    0.0.0.0         UG        0      0        0 eth0
```

Now I configured IP layer, the next part is to configure my debian to route packets,

```
$ su -c 'sysctl -w net.ipv4.ip_forward=1'
$ su -c 'iptables --verbose --table filter --policy FORWARD ACCEPT'
$ su -c 'iptables --verbose --table nat --append POSTROUTING --jump MASQUERADE'
```

Note, that the above steps are just for current running kernel, if I want my changes to be permanent, I need to edit `/etc/sysctl.conf` and I need to save my current iptables using `iptables-save/iptables-restore`.

Thats all from the gateway machine(that is, my laptop), now I need to add my friend's laptop to this Dark Network.

```
$ sudo iwconfig wlan0 essid "Dark" mode ad-hoc channel 1
$ sudo ifconfig wlan0 192.168.2.2 netmask 255.255.255.0
$ sudo route add default gw 192.168.2.1
$ sudo echo "nameserver 192.168.1.1" >> /etc/resolv.conf
```

Thats all from my friend's machine, now he just opened firefox and happily browsed from the balcony. No more wires.

Whenever my friends come to my room, they have some bit of tough time accessing internet, because most of the time, my laptop will be connected with internet through ADSL. Obviously they have shyness to ask me for stop my surfing. But now, I can happily share my internet connection with my friends. I don't need to buy a wireless router. My laptop itself with the ADSL router doing what a wireless router can do.

Thanks linux.

SUPPORT FOR NVIDIA GEFORCE GRAPHICS CARD - DEBIAN - TESTING

Updated: May 07, 2024

As you can see in my previous article, this is about getting my nVidia card to work with Debian.

By default Xorg provides `nv` driver. Its unfortunate that nVidia not even working with Xguys for basic compatibility for `nv` driver. As a result, the default `nv` driver is not usable in my machine.

I'm not left alone here. Debian have a package called `nvidia-glx` which will install full support for nvidia cards with proprietary nvidia driver.

```
$ su -c'aptitude install nvidia-glx'
```

But currently, this is not working in Debian Testing. I got a bug, [last x11 update breaks things with nvidia-glx](#).

Now also I'm not left alone, using `module-assistant` I can still get the nvidia driver.

```
$ su -c'aptitude install module-assistant'
$ su -c'module-assistant prepare'
$ su -c'module-assistant'
```

After the above steps I will get ncurses interface for `module-assistant`, choose `nvidia` kernel in the wizard and compile the `nvidia-kernel` package. If everything goes right, I will get `nvidia` driver in `lib/modules/$(uname -r)/kernel/drivers/video/nvidia.ko`. Also my `/etc/x11/xorg.conf` file will be modified to use `nvidia` driver. Thats all, I can now start X session.

Unfortunately, this method is also not working currently, [while compiling I'm getting this error](#).

So there is no other way for me then to go for nvidia's own builder. You can download the driver+installer [from this location](#).. Now we need to make our system capable for compilation.

```
$ su -c'aptitude install module-assistant'
$ su -c'module-assistant prepare'
```

Then, the next step is just run the nvidia's installer

```
$ su -c'chmod 755 NVIDIA-Linux-x86_64-190.53-pkg2.run'
$ su -c'./NVIDIA-Linux-x86_64-190.53-pkg2.run'
$ su -c'aptitude install nvidia-kernel-common'
$ su -c'depmod -aeF /boot/System.map-$(uname -r) '
$ su -c'update-initramfs -u'
```

The above step will automatically compile and install the `nvidia.ko` driver in `/lib/modules/$(uname -r)/kernel/drivers/video/nvidia.ko`. Also my `/etc/x11/xorg.conf` file will be modified to use `nvidia` driver. That's all, I can now start X session.

SUPPORT FOR BROADCOM WIRELESS DEVICES - DEBIAN - TESTING

Updated: May 07, 2024

Got some time to post this one, I thought I should log my experience, installing wireless support for my broadcom PCI card in debian, so that next time, I will not miss the methods.

Debian provides three way to get support for BCM43XX devices. One is the reverse engineered `b43xx` driver coming with the default kernel image. Its opensource and mainly, it works. To get it work, first we need to check whether our device have support from this driver,

```
squeeze@Debian:~$ lspci | grep BCM
03:00.0 Network controller: Broadcom Corporation BCM4311 802.11b/g WLAN (rev 02)
squeeze@Debian:~$
```

My card have support for b43, I just checked with the list of [PCI cards they provided in their site](#). Now all I need to do is, install the firmware,

```
$ su -c 'aptitude install b43-fwcutter'
```

This will automatically install the firmware. Thus, in your next boot, you will get `wlan(x)` interface provided by b43 opensource driver.

If your card is not supported by opensource b43xx driver, still debian provides a good way, its `broadcom-sta-common` package. This will install the proprietary `wl` driver in a way that it will perfectly sync with debian packaging system.

Before starting this, don't forget to have your backports ready, because debian testing repos have the habit of removing important packages, I had an experience that I was not able to find `kernel-headers-2.6.30-amd64` in testing repo, after some googling, I came to know that its available in backports.

```
$ su -c 'echo "deb http://www.backports.org/debian/ lenny-backports main non-free_
↪contrib" >> /etc/apt/sources.list.d/backports.sources.list'
$ su -c 'echo "deb-src http://www.backports.org/debian/ lenny-backports main non-free_
↪contrib" >> /etc/apt/sources.list.d/backports.sources.list'
$ su -c 'aptitude update'
$ su -c 'aptitude install debian-backports-keyring'
```

Now make sure `module-assistant` is fully prepared and ready to compile source codes for you.

```
$ su -c 'module-assistant prepare'
```

Now we can start our proceedings,

```
$ su -c'aptitude install broadcom-sta-common'
```

Now, we need to compile the proprietary driver using `module-assistant` command. This command will download packages to compile source codes on behalf of you using GNU Tools, Debian Build tools etc. Its ncurses based interface will clearly tell you what its doing. select `broadcom-sta` in the wizard, `module-assistant` will compile and install `wl` driver in this path `/lib/modules/$(uname -r)/kernel/drivers/net/wireless/wl.ko`

```
$ su -c'module-assistant'
```

Thus, in your next boot, you will get `eth(x)` provided by `wl.ko` proprietary driver. This method is called The Debian Way.

Final method is so simple, [Download the driver from broadcom's website](#). Prepare your system to compile source codes. You can do this by installing `gcc`, `autotools`, `libtool`, `build-essential` etc., or just install and prepare `module-assistance` so that your system will be fully capable to compile source codes.

```
$ su -c'aptitude install module-assistant'
$ su -c'module-assistant prepare'
```

Now just extract, make and make install,

```
$ su -c'tar xvfz hybrid-portsrc-x86_64-v5.10.91.9.3.tar.gz'
$ su -c'make'
$ su -c'make install'
$ su -c'echo "blacklist b44" >> /etc/modprobe.d/wl.conf'
$ su -c'echo "blacklist b43legacy" >> /etc/modprobe.d/wl.conf'
$ su -c'echo "blacklist b43" >> /etc/modprobe.d/wl.conf'
$ su -c'echo "blacklist ssb" >> /etc/modprobe.d/wl.conf'
$ su -c'depmod -aeF /boot/System.map-$(uname -r) '
$ su -c'update-initramfs -u'
```

Finally you can see `wl` driver in this path `/lib/modules/$(uname -r)/kernel/drivers/net/wireless/wl.ko`. In next boot you will get `eth(x)` wireless interface provided by proprietary `wl` driver. In this method, debian will not update this driver for the next kernel updates, you need to do it on your own. This is the main drawback in the final method.

SAKLESHPUR TREKKING - A THRILLING STORY

Updated: May 07, 2024

Aah!! at last I got some time to blog, I was stuck up nearly two month doing RHCE and i'm happy that I did it in first shot itself. In the mean time, one of unforgettable moment happened in my life.

Dec 17, A serious discussion happened in my team, everybody want to go to their home inviting team members as their guest saying that there are nice places to go nearby. I also invited colleagues to vellore (What?? are you cracy?? - I know... I know.. a big laugh around my teammates..). Srinath surfed like anything to find a new, unpopular, best place in karnataka. Finally he saw a blog about Sakleshpur. Nobody knows this untill that night.

Dec 18, We decided to go wherever in karnataka, on that night, srinath arranged a car, took lot of maps routing to sakleshpur from chennai, everyone finished their work, at 11.30, I started my journey with srinath. After we reached our office, we took manju, karthik, saravanan and sridhar heading to sakleshpur.

Dec 19, 6:00 am, We are at banglore - tumkur outer ring road, wow!! top class roads!!

Dec 19, 11.00 am, our first stop for morning breakfast, moving towards hassan,

Dec 19, 12.00 pm, our second stop, just to see a village which impressed saravanan,

Dec 19, 2.00 pm, we are at outer side of hassan, moving towards sakleshpur, one of the best journey in my life, full of greenery.

Dec 19, 3.00 pm, we reached sakleshpur, booked a room, everyone want to take a nap, but no time for us, heading to the Green Track immediately. Till now we don't know what kind of place that is.

We took auto from sakleshpur to dhonigal, after giving money to that driver, he advised to be careful, trains will come at any time and you will not have space to stand aside every-time. Also he asked us to watch out for elephants, jaguars and forest beasts, because previously two men went inside this deep forest and came back as skeletons !!!!!!!!!!!!!!! :0

After this initial fear, we started our treking, its a small mountain climb, we reached the track, its Dhonigal railway station, darkness taking raise !!

Even though its risky, we had no other choice now, heading towards Yedakumeri railway station. Fully covered by trees, everywhere is green, just amazing!! and thrilling !!

Dec 19, 6.00 pm, its almost dark, we heared some kind of sound, its a train!! heading towards us, here we got a place to step aside and captured that train in our cameras

Dec 19, 6.30 pm, its fully dark, we stuck in a middle of deep forest, everybody got some kind of uncertainty and discussed about dropping the plan and back to sakleshpur, but there is no choice for us. Even going back to sakeleshpur is as dangerous as going to Yedakumeri.

At that time, we saw someone coming towards us. Manju asked that person who are you?, because he only know kannada, with initial fearness solved, that person is the one god sent to save us. It seems he is heading towards his home near dhonigal railway station, and scolded us for unplanned journey. He called his friend who is staying in one of the



Fig. 1: Karthk, Saravanan, Srinath, Sridhar and Manju



Fig. 2: Bangalore - Tumkur outer ring road



Fig. 3: First stop for breakfast with this beautiful baby!!



Fig. 4: villages - True India !!



Fig. 5: Hassan to Sakleshpur



Fig. 6: Donigal railway station



Fig. 7: no wonder why its called green track!!



Fig. 8: First train !!

railway workers quarters and explained the situation. His friend is also helpful that he promised to come here and take us to his home, also asked this person to stay until he come.

As he said, his friend came and took us to his home. Near his home, there is another abandoned home, he broke those locks and gave it to us saying If somebody asks!! I will tell that its thieves work!! they broken this locks. We thanked him a lot. at last our fears came to somewhat minimal.

Dec 19, 7.30 pm, sridhar's airtel got some thin signal, everybody started to call and inform to their beloved ones, I took it and sent a message to kesavan Machi, mohan here, I'm in middle of deep forest situated between dhonigal - yedakumari, If i'm not calling to you before morning 10.00 tomorrow, please inform to police, also tell to my father, please dont panic, don't tell it now to my father, I'm safe here!! . Kesav behaved as I said.

After this, everyone got some relax, talked freely and cracked jokes !! thinks came normal !! and we started a camp fire outside that home. Its full of dark we don't know how much beautiful that place is until morning.

Dec 20, 6.30 am, After amazed by the place we slept, everyone went to take bath even in that freezing temperature. At last we thanked everyone in that place and started our journey to Yedakumari. Also we saw few people (most of them are railway contract workers) using that track for transporting. While walking we asked another person how long is Yedakumari? he said another 10km, we already passed 7km, he also said, there is one small station nearby within 4km from the current place. So we decided to stop in that small station and back to dhonigal. Everything went as we planned this day and we witnessed a beautiful paradise !!!

I'm a person who didn't get opportunity to witness nature's gifts to human, but after visited this place, I just want to spend my life there, people living there love their world, money is not much important to them (we learn that when we gave money for staying one night, that person refused).

There is something which we are gifted, we should protect that gifts till our time and give it to our descendants.

Finally, thanks to hyundai for producing quality product. I love that car which we traveled, its as perfect and gave no trouble in our journey. Even riding it is as enjoyable as nature.



Fig. 9: Walking in a railway bridge!!



Fig. 10: Savior!!!



Fig. 11: Helping hands!!



Fig. 12: Camp Fire!!!



Fig. 13: nearby railway workers house



Fig. 14: First Tunnel



Fig. 15: Second Train - at this time we are inside a tunnel - run!! run!! - amazing moments :) !!



Fig. 16: Longest and Deepest bridge we crossed !! - legs will defenately shake!!



Fig. 17: Mountain View!!

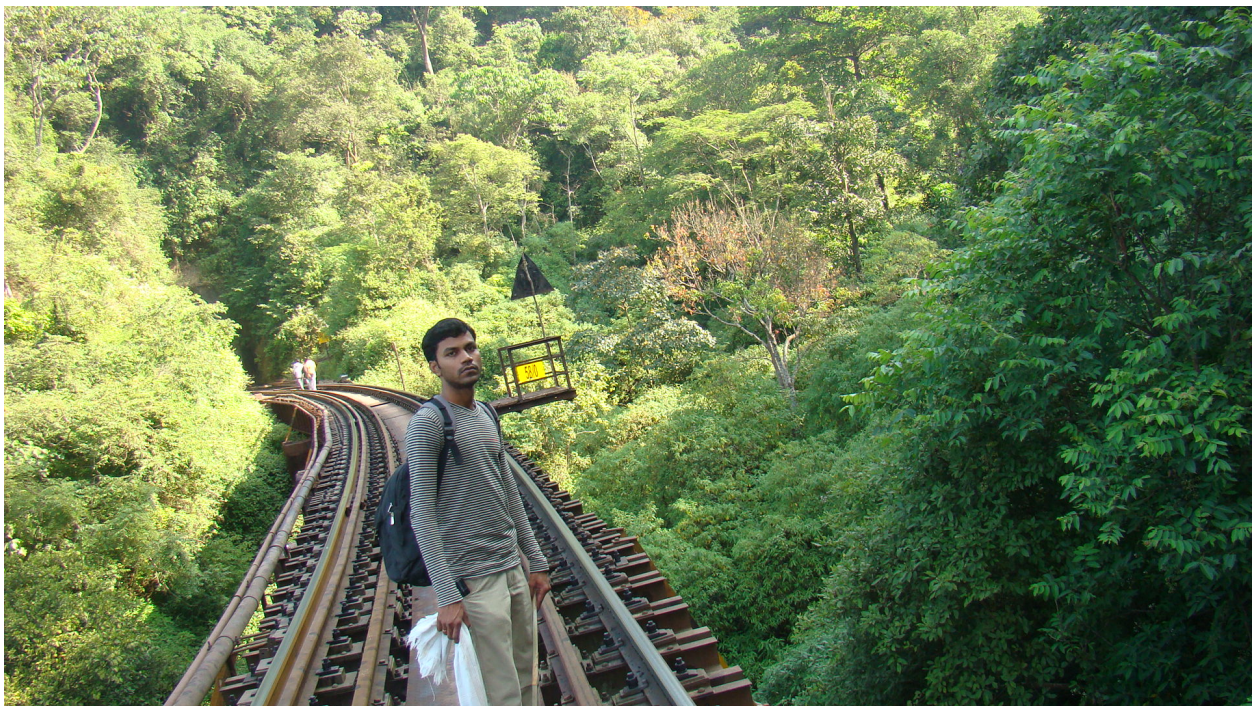


Fig. 18: Did you see any rocks here??



Fig. 19: We reached The Next Small Station!!!



Fig. 20: Back to pavilion - Sakleshpur hotel!!

Dec 21, 7.30 am, we are here!! chennai !!. with lot of energy!! happiness!! and full of beautiful memories!! :)



Fig. 21: Thanks Santro !!

Here is the map

HOWTO SWITCH FROM UBUNTU TO DEBIAN - MY EXPERIENCE - PART II

Updated: May 07, 2024

continued...

Once I get the base system installed, I tried to hang around pure commandline for sometime, I installed the following package to hang around,

```
$ su -c 'apt-get install elinks'
```

Before going into aptitude for the final phase of my installation, I configured contrib nonfree and multimedia repos by creating two more files in `/etc/apt/sources.list.d` directory as below,

```
$ cat /etc/apt/sources.list.d/nonfree.sources.list
deb http://ftp.iitm.ac.in/debian/ lenny contrib non-free
deb-src http://ftp.iitm.ac.in/debian/ lenny contrib non-free

deb http://security.debian.org/ lenny/updates contrib non-free
deb-src http://security.debian.org/ lenny/updates contrib non-free

deb http://volatile.debian.org/debian-volatile lenny/volatile contrib non-free
deb-src http://volatile.debian.org/debian-volatile lenny/volatile contrib non-free
$
```

and

```
$ cat /etc/apt/sources.list.d/multimedia.sources.list
deb http://mirror.optus.net/debian-multimedia/ stable main
deb-src http://mirror.optus.net/debian-multimedia/ stable main
$
```

To use multimedia repo, we also need to install the gpg-key package using the following command,

```
$ wget http://www.debian-multimedia.org/pool/main/d/debian-multimedia-keyring/debian-
↳multimedia-keyring_2008.10.16_all.deb
$ su -c 'dpkg -i debian-multimedia-keyring_2008.10.16_all.deb'
```

Debian multimedia website provides more details, check this [link](#)

At the end, I started aptitude and installed all my must needed packages(Xfce,built-essential, gcc, vim-full, nvidia-glx, flash-10, mplayer, ffmpeg, w64codecs etc).

I don't want to run any services in my laptop, so I disabled all the default services in lenny (like `exim4`, `avahi`, `gdm` etc.). The base system runs with just ~34MB. Once I `startx`, it goes to ~140MB. If I start `icewesel`, it goes to ~314MB. I never thought I can run a full fledged system with this kind of less memory. No pause on Boot (just taking 10-12 sec to print login prompt)

Now, I'm really happy. I never expected the switch is this much smooth. Thanks to Debian.

HOWTO SWITCH FROM UBUNTU TO DEBIAN - MY EXPERIENCE - PART I

Updated: May 07, 2024

After long time I'm posting this. In this time, I got a fantastic experience of switching myself from ubuntu to debian. Reasons? may be silly but here they are,

1. From 8.10, I had a tough time with all the kernel's ubuntu shipped and updated into my laptop, I was hit by a kernel [acpi bug](#) which still exists(in i686, not in x86_64). It pauses boot process when my laptop runs in battery and I need to press until init starts.
2. Nowadays, I hate the idea of linux for everyone. because nix systems are not for everyone. its for the one who like to learn computing.
3. My laptop lost DVD drive.

So I was in a mood to switch myself to something else. While surfing, I came to the debian website, their Installation guide explains howto install debian from a [USB](#). Mine is Athlon64, so this time I went for x86_64. First I prepared my pendrive. For that we need to partition it using fdisk

```
$ sudo fdisk /dev/sdb
```

You can do this by running system->administration->partition-editor in Ubuntu. Once you created a new partition in your pendrive(/dev/sdb1), you need to format the partition as FAT32 using the following command

```
$ sudo mkfs -t vfat /dev/sdb1
```

Once you formatted, the next step is to install syslinux into it, If your Ubuntu don't have syslinux command, install syslinux package

```
$ sudo apt-get install syslinux
$ sudo syslinux /dev/sdb1
```

Now its time to download the installation files, I copied all the files from the [following location](#) and copied into /dev/sdb1. Finally the drive will look like this,

```
$ find /media/disk
.
./gtk
./gtk/vmlinuz
./gtk/initrd.gz
./syslinux.cfg
./vmlinuz
```

(continues on next page)

(continued from previous page)

```
./initrd.gz
./ldlinux.sys
./MANIFEST
./boot.img.gz
```

And it is capable of booting using `syslinux` bootloader. To use Graphical Installer, we need to modify `syslinux.cfg` to look like this

```
default gtk/vmlinuz
append initrd=gtk/initrd.gz video=vesa:ywrap,mtrr vga=788
```

Once your pendrive is ready, unmount it and reboot the system, change the Boot order(In Compaq, press F9) and select the USB drive. The debian-installer will start asking questions, there on, its an easy ride. I didn't install Gnome/Kde when the installer asked to choose one. I was interested in Xfce, so I postponed it.

In next post, I'll tell you how I installed the 3rd layer(desktop).

will continue..

STARTING ANOTHER X SERVER WITH DIFFERENT LOCALE - UBUNTU

Updated: May 07, 2024

While playing with `quemu+lfslivecd` under ubuntu, I came to know about the following commands which sets the locale environment for a linux system(atleast for `lfslivecd`). Using the following steps, I'll share the commands and start a new X server with the modified environment.

1. Press `ctrl+alt+F1` to go to virtual terminal 1 or `tty1` (To switch back to current gnome session, use `ctrl+alt+f7`). Login to that terminal with different userid then the current userid who already logged into current gnome session.
2. type the following command to set the timezone. you can see the timezone data file in `/usr/share/zoneinfo/posix/{continent}/{country}` where `{continent}` is the name of your continent like Europe, Asia etc., and `{country}` is your country name. This file should be pointed by `TZ` environment variable. The below command will change Timezone to Europe/Spain.

```
$ export TZ="Europe/Spain"
$ date
```

The `date` command will show the date according to the new timezone.

3. `localedef` command is used to create binary definitions in `/usr/lib/locale` directory from `/usr/share/i18n/` directory, the `i18n` directory contains `charmaps` directory which contains encodings(like UTF-8 etc). `i18n` also contains `locales` directory which contains definitions for locales. `localedef` will use these two directories to generate a definition for particular language termed in `{LANG}_{COUNTRY}.{ENCODING}` (eg: `es_ES.utf8` - means language code `es` (spanish), country `ES` (spain) and encoding in `utf8`). The following command will compile language definition for spanish language in `utf8` encoding.

```
$ sudo localedef -f UTF-8 -i es_ES --no-archive 'es_ES.utf8'
$ locale -a | grep 'es_ES.utf8'
```

`locale -a` will show whether language definition added or not.

4. Now we are ready to change the `LANG` variable, this variable is used by all `glibc`'s language aware applications to switch their output to a particular language, the output strings for a particular application for a particular language should be available in `{application}.mo` file inside `/usr/share/locale/{LANG}/LC_MESSAGES` directory(Distros provide `.mo` files through language specific packages). Now switch the `glibc` applications to spanish using the following command,

```
$ export LANG="es_ES.utf8"
$ locale
```

`locale` command will tell the current language.

5. Finally start a new X server using the following command with modified locale. You can switch back to old X server using `ctrl+alt+F7`. To switch to new X server, use `ctrl+alt+F9` (or `ctrl+alt+F10` if it started in `tty10` virtual terminal)

```
$ startx -- :1 -br -audit 0 -nolisten tcp
```

At first, it will ask to change the name of the directories to suite the new language, don't rename unless you always going to work in the new locale environment.

6. Once you done, Logout from the New X server(new gnome session), it will stop the newly started X server and put you in `tty1` virtual terminal. `exit` command will exit you from the `tty1` terminal. Finally, switch back to old X server's gnome session using `ctrl+alt+F7`.

Thats all.

QEMU NETWORKING (TUN/TAP)

Updated: May 07, 2024

Setting up networking in Qemu instance is very simple if you are using `user` mode networking stack. And most of us don't care about Full fledged networking while using Qemu Guests. But configuring Network in Qemu with full access requires TUN/TAP configuration.

Here is some simple steps which I did to configure TUN/TAP in Ubuntu 9.04,

Inside Ubuntu(Host), type the following commands,

```
$ sudo apt-get install qemu qemu-common qemu-source
$ wget ftp://ftp.osuosl.org/pub/lfs-livecd/lfslivecd-x86-6.3-r2145-min.iso
$ sudo qemu -cdrom lfslivecd-x86-6.3-r2145-min.iso -boot d -kernel-kqemu -m 512 -net_
→ nic -net tap &
$ sudo ifconfig tap0 192.168.10.1
$ sudo sysctl -w net.ipv4.ip_forward=1
$ sudo iptables -t nat -A POSTROUTING -j MASQUERADE
$ sudo iptables -t filter -P FORWARD ACCEPT
```

Inside lfslivecd(Guest) type the following commands,

```
$ ifconfig eth0 192.168.10.2
$ route add default gw 192.168.10.1
$ cat > /etc/resolv.conf <<EOF
nameserver 208.67.222.222
nameserver 208.67.220.220
EOF
$
```

Ofcourse, you can use any distro as a guest in qemu. Enjoy Qemuing.

SHARE YOUR QEMU GUEST WITH YOUR FRIENDS

Updated: May 07, 2024

Don't know about the how useful it will be, but I found some way to enable my friends to access my Qemu Guest Instance running LFSLiveCD in Ubuntu Host using `ssh`. Here is the steps I did,

1. Download Qemu Packages

```
$ sudo apt-get install qemu kgemu-common kgemu-source
```

2. Download LFSLiveCD from [here](#)

3. Now start the LFSLiveCD guest using the following command

```
$ qemu -cdrom lfslivecd-x86-6.3-r2145-min.iso -boot d -m 512 -net nic -net user -  
→redir tcp:35000::22
```

4. Now Qemu will start LFSLiveCD with user mode network stack, that is, from LFSLiveCD, ip 10.0.2.2 will be the Default gateway and DHCP server. ip 10.0.2.3 will be the DNS server and the LFSLiveCD will get 10.0.2.15. If you look `netstat` output in Host(here Ubuntu), there will be a line which tells port 35000 is listening. Qemu process is listening in that port, it will redirect connections from Host(Ubuntu's) 35000 port to Guest(LFSLiveCD's) 22(ssh) port. All we need to do is, start `ssh` service in LFSLiveCD Guest. Type the following command in Qemu.

```
[root:]# cd /etc/rc.d/init.d  
[root:]# ./syslogd start  
[root:]# ./sshd start  
[root:]# passwd  
New Password: [newrootpassword]  
Retype New Password: [newrootpassword]  
[root:]# tail -f /var/log/*.log
```

Now you have `sshd` running in the guest, and you also changed the `root` password to a known password. All you need to do is, ask your friends to `ssh` to your Host's(Ubuntu) ip address but with 35000 port instead of default 22 port. And ask them to login as `root` as userid and `[newrootpassword]` as password. Your friends will be so much happy to see they are became `root`. In the mean while you can also login to your guest with `ssh root@127.0.0.1 -p 35000`. If you want, add a new user and share that userid and password to your friends, so that they can have less power.

Instead of Ubuntu Host, We can do this in Windows XP, as a Normal powerless user. The below thing you need to do,

1. Download QemuManager [Zipfile Edition](#) from [here](#)
2. Download LFSLiveCD from [here](#)
3. Configure LFSLiveCD inside QemuManager(follow this [Guide](#)).

4. While configuring Network, DoubleClick NetWork Card 1, GoTo Port Redirectors tab, type 35000 in Port Visible to Host and 22 in Port Visible On Guest and click Add button.
5. Start LFSLiveCD Qemu Guest. and Do all changes I explained above to start sshd inside LFSLiveCD.
6. Download [Putty.exe from here](#)
7. Start Putty.exe with Host as 127.0.0.1 and Port as 35000 and Connection Type as ssh.
8. Login with root as username and [newrootpassword] as your password.
9. If you want to share your Qemu Guest, with your windows friends who's machines and your's machine are in same subnet, then ask them to start putty.exe with host as your machine's ip address, port as 35000, username as root and password as [newrootpassword]. You can also create seperate user in your Guest(LFSLiveCD) and share with your friends.
10. Enjoy Virtualization.

I also tried this method in my friend's Vista laptop, but it failed!! First, it is not allowing to run QemuManager.exe as a normal user, Second, it is giving warning while opening 35000 as listening port, Finally, putty.exe is not at all connecting to Guest Instance, even though sshd is running in the LFSLiveCD Guest. If anyone have success story, kindly share.

Finally Thanks to my company for blocking access to the linux learning server, which make me to go for alternate methods and finally endup in the Windows XP method.

UPSTART, UPDATE-IDENTICA AND MY BLOG

Updated: May 07, 2024

Hi friends,

Curl is one of the fantastic tool I'm using day to day. and Upstart is one good upgrade to the long standing `rc.*` method to start and stop services. While learning TwitterAPI to interact with `identi.ca`, I got an idea to update my laptop's CPU usage to the `identi.ca` feed whenever my system starts and attach that feed into my blog. Well, It developed into something usable now. here is the script

```
#!/bin/bash
# update-identica - script to update my identi.ca feed

USERID=""
PASSWORD=""
BASEURL="http://identi.ca/api"
UPDATEAPIURL="/statuses/update.xml"
UPDATEURL="${BASEURL}${UPDATEAPIURL}"
MAXMESSAGES="10"
LOGFILE="/var/log/update-identica.log"
LOGFILESIZE=$(du -sb "${LOGFILE}" | awk '{print $1;}')
MAXLOGFILESIZE=$((1024 * 1024))
test "${LOGFILESIZE}" -gt "${MAXLOGFILESIZE}" && rm "${LOGFILE}"
INTERVAL=$((5 * 60))

destroymessages()
{
    EXTRAMESAGES=$((MESSAGECOUNT - MAXMESSAGES))
    EXTRAMESAGEIDS=$(curl -u "${USERID}:${PASSWORD}" "${BASEURL}/statuses/user_
→ timeline.xml" 2>/dev/null |
    grep '[0-9]\{7\}' | cut -d'>' -f2 | tee -a "${LOGFILE}")
}

while true
do
    MESSAGE="status="
    MESSAGE="${MESSAGE}[cpu:$(vmstat 1 2 | tail -1 | awk '{print $13+$14;}')%]"
    MESSAGE="${MESSAGE} [uptime:$(uptime | cut -d',' -f1 | tr -d ' ')]"

    curl -u "${USERID}:${PASSWORD}" -d "${MESSAGE}" "${UPDATEURL}" 2>/dev/null | tee
→ -a "${LOGFILE}"
    MESSAGECOUNT=$(grep ' ' -f2 | cut -d'<' -f1)
    test "${MESSAGECOUNT}" -gt "${MAXMESSAGES}" && destroymessages
```

(continues on next page)

(continued from previous page)

```
sleep '${INTERVAL}'  
done
```

Also call this script automatically by creating a event file /etc/events.d/update-identica

```
# update-identica  
# a small program to update this laptop's feed in  
# http://identi.ca  
  
start on update-identica-event  
start on runlevel 2  
stop on runlevel [!2]  
exec /usr/bin/update-identica
```

Take a look at the feed titled My laptop's health in this blog, it is the feed which my laptop is updating every 5 minutes.

If you are interested in WebAPIs and accessing through curl, then here is link, `lucanica (identi.ca's software)` also using the same REST methods to give web services.

<http://apiwiki.twitter.com/Twitter-API-Documentation>

LDD2CHROOT - A SCRIPT TO EXTRACT PACKAGES

Updated: May 07, 2024

At last I got some time to blog, and I did something with `chroot`. Here is a script which will extract ldd dependency library packages(.debs) to `chroot` directory. Here is the steps it follows,

1. For the given command, get the shared libraries.
2. Then get the packages which will give the shared libraries.
3. and download those shared libraries and extract in the given `chroot` directory.

I hope it will be useful to package maintainers.

```
#!/bin/bash
# ldd2chroot - A script to extract packages
# Author: mohan43u

USAGE="[usage]
      ldd2chroot [-i chrootdir] [-p] [-h] cmd_or_pkg [...]"
HELP="${USAGE}"
[description]
      -i chrootdir      Install pkgs in 'chrootdir'
      -p                'cmd_or_pkg' will be package name
      -h                Print this help
      cmd_or_pkg        Command name or package name(only for -p)"

executeldd()
{
  ldd $(which "${1}")
}

getsharedlibs()
{
  tr '\t' ' ' |
  tr -s ' ' '\n' |
  grep '^/'
}

getlibpkgs()
{
  while read LIB; do dpkg -S "${LIB}"; done
}
```

(continues on next page)

(continued from previous page)

```

getpkgs()
{
cut -d':' -f1 | sort | uniq
}

extractpkg()
{
for PKG
do
    echo "[starting apt-get for ${PKG}]"
    apt-get clean
    apt-get --reinstall --download-only --yes install "${PKG}"
    DEBFILE=$(find /var/cache/apt/archives -name '*.deb')
    cp -p ${DEBFILE} $(pwd)
    test ! -z "${CHROOTDIR}" && echo -e "[extracting ${DEBFILE} to ${CHROOTDIR}]" &&
    dpkg -X "${DEBFILE}" "${CHROOTDIR}"
done
}

while getopts 'i:ph' OPTIONS
do
    case "${OPTIONS}" in
        i) CHROOTDIR="${OPTARG}";;
        p) PKGARGS="1";;
        h) echo -e "${HELP}" && exit 0;;
        \?) echo -e "${USAGE}" && exit 1;;
    esac
done
shift $((OPTIND - 1))
test -z "${@}" && echo -e "${USAGE}" && exit 1

for ARG
do
    if test -z "${PKGARGS}"
    then
        LDDOUTPUT=$(executeldd "${ARG}")
        echo -e "[ldd]\n${LDDOUTPUT}"
        SHAREDLIBS=$(echo -e "${LDDOUTPUT}" | getsharedlibs)
        SHAREDLIBS="${which ${ARG}}\n${SHAREDLIBS}"
        echo -e "[sharedlibs]\n${SHAREDLIBS}"
        LIBPKGS=$(echo -e "${SHAREDLIBS}" | getlibpkgs)
        echo -e "[required shared library packages]\n${LIBPKGS}"
        PKGS=$(echo -e "${LIBPKGS}" | getpkgs)
        echo -e "[required pkgs for apt-get]\n${PKGS}"
    else
        PKGS="${ARG}"
    fi

    extractpkg ${PKGS}
done

```

If someone improving this, let me know, so that I can also use the changes.

URLENCODER

Updated: May 07, 2024

Hi friends,

Previously I had a post regarding url encoding and decoding, This is much improved version of that script. To know more about urlencoding, try [this link](#)

```
#!/bin/bash
# urlencode - program to urlencode/urldecode
# Author: Mohan Raman
# License: Its in public domain.

USAGE="[USAGE]
        urlencode [-e|-d] [-h]
"

HELP="${USAGE}
[DESCRIPTION]
        -e          URLEncode standard input (default)
        -d          URLDecode standard input

[EXAMPLE]
        $ urlencode
        http://www.google.co.in
        http%3A%2F%2Fwww.google.co.in

        $ urlencode -d
        http%3A%2F%2Fwww.google.co.in
        http://www.google.co.in
"

encode()
{
sed -e's/./&\n/g' -e's/ /%20/g' | grep -v '^$' | while read CHAR; do test "${CHAR}" =
↪ "%20" && echo "${CHAR}" || echo "${CHAR}" | grep -E '[-:alnum:~!*.,'\'"()]\[\]|\]'
↪ || echo -n "${CHAR}" | od -t x1 | tr ' ' '\n' | grep '^[[:alnum:]]\{2\}$' | tr '[a-
↪ z]' '[A-Z]' | sed -e's/^/%/g'; done | sed -e's/%20+/g' | tr -d '\n'; echo
}

decode()
{
sed -n -e's/%\([0-9A-F][0-9A-F]\)\(\|x\1/g' -e's/+ /g' -e's/./echo -e "&/g' -ep | "$
↪ {SHELL}"
```

(continues on next page)

(continued from previous page)

```
}

DECODE="0"

while getopts "edh" OPTION
do
    case "${OPTION}" in
        d) DECODE="1";;
        h) echo "${HELP}"; exit 0;;
        \?) echo "${USAGE}"; exit 1;;
        esac
    done

test "${DECODE}" = "0" && encode
test "${DECODE}" = "1" && decode
```


FUNCTIONS FROM HEADER FILES

Updated: May 07, 2024

I always think if we have a reference to all the C/C++ functions present in our system, it will be so much easier to write C programs with those functions. I posted this question to our ilugc, they introduced `ctags` to me.

`ctags` is a simple utility which will produce a `ctags` file which contains all definitions of macros, functions, structures, etc., If you want to know more about it, try `man ctags`.

The `ctags` output file will be used by editors like `vi`, `emacs` to help developers to know about the syntax and function definition.

I thought, It would be good if we can able to `grep` for functions, macros, from the `ctags` file. Here is a simple script to achieve this task. Don't forget to run,

```
$ ctags --verbose --recurse /usr/include
```

before executing this script. If you know better way to get reference to functions in our system, kindly share with me.

```
#!/bin/sh
#
# Description:
# program to search for functions, macros, definition, etc.,
# in a 'ctags' file.
#
# Author: Mohan Raman
# License: Its in public domain, use it whatever way you want

USAGE="[USAGE]
    ctagsfilter.sh [-n tagname] [-t tagtype] [-f headerfilename]
                [-h help] ctagsfile1 [ctagsfile2 ...]
"
HELP="${USAGE}
[DESCRIPTION]
    -n tagname           tag name to search
    -f headerfilename    header file to search
    -t tagtype           tag type, anyone of listed below
                        c      class
                        d      defined macro
                        e      enum/enum member
                        f      function
                        g      global
                        m      member of struct/class/enum/union
                        n      namespace
```

(continues on next page)

(continued from previous page)

```

                                s      struct
                                t      typedef
                                u      union
                                v      global variable
    -h                          print this help
"

while getopts "n:t:f:h" OPTIONS
do
    case "${OPTIONS}" in
        n) TAGNAME="${OPTARG}";;
        t) TAGTYPE="${OPTARG}";;
        f) HEADER="${OPTARG}";;
        h) echo "${HELP}" && exit 0;;
        \?) echo "${USAGE}" && exit 1;;
        esac
    done

shift $((OPTIND - 1))
test "${#}" -eq 0 && echo "${USAGE}" && exit 1

for FILE
do
    OUTPUTBUFFER=`cat "${FILE}"`

    if test ! -z "${TAGNAME}"
    then
        OUTPUTBUFFER=`echo "${OUTPUTBUFFER}" |
            awk -F' ' "\$1 ~ /${TAGNAME}/{print \$0;}"`
    fi

    if test ! -z "${HEADER}"
    then
        OUTPUTBUFFER=`echo "${OUTPUTBUFFER}" |
            awk -F' ' "\$2 ~ /${HEADER}/{print \$0;}"`
    fi

    if test ! -z "${TAGTYPE}"
    then
        OUTPUTBUFFER=`echo "${OUTPUTBUFFER}" |
            grep ";\"      ${TAGTYPE}"`
    fi

    echo "${OUTPUTBUFFER}"
done

```

‘LOCALTIME()’ LOVES ‘05:30:00’

Updated: May 07, 2024

Got some problem when I'm converting elapsed time `time_t` to `struct tm` for printing time in `%H:%M:%S` format. Here is the code,

```
/* code to learn how 'localtime()' works */

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(int argc, char **argv)
{
    struct tm time1;
    struct tm *time2;
    time_t time;
    char timestring[20];

    time2 = (struct tm *)malloc(sizeof(struct tm));

    /* setting 0 seconds to time */
    time = (time_t) 0;
    localtime_r(&time, &time1);

    time1.tm_hour = time1.tm_hour - 5;
    time1.tm_min = time1.tm_min - 30;

    strftime(timestring,
              20,
              "%H:%M:%S",
              &time1);
    printf("Time1: %s\n",
           timestring);

    /* setting 1 minute 10 seconds to time */
    time = (time_t) 70;
    time2 = localtime(&time);

    time2->tm_hour = time2->tm_hour - 5;
    time2->tm_min = time2->tm_min - 30;

    strftime(timestring,
              20,
```

(continues on next page)

(continued from previous page)

```
        "%H:%M:%S",
        time2);
    printf("Time2: %s\n",
        timestring);

    free(time2);
    return(0);
}
```

And the output is,

```
Time1: 05:30:00
Time2: 05:31:10
```

I don't know why its happening, `localtime()` creates `struct tm` structure which by default have 05:30:00 as its starting time. I need to manually subtract the 5 from `tm_hour` and 3 from `tm_min` to get what I want.

```
time1.tm_hour = time1.tm_hour - 5;
time1.tm_min = time1.tm_min - 30;

time2->tm_hour = time2->tm_hour - 5;
time2->tm_min = time2->tm_min - 30;
```

Here is the updated output,

```
Time1: 00:00:00
Time2: 00:01:10
```

Also `localtime()` sets default values to all the variables in the result `struct tm` structure. google not helping to find the reason.

DHARMASTALA - WHERE YOU SEE YOURSELF

Updated: May 07, 2024

Hi friends,

Its my grandma's wish, that I should wipe my hair to god Manjunatha, who is there in dharmastala. My family planning to go and finish the pilgrimage, but time didn't allowed us, last week, we finally decided and started our journey.

Last Wednesday morning 3.30 am, we started, but there is no bus to banglore from my village pallikonda until 5.00 am.(Actually every bus to Bangalore will cross my village). So we went to Vellore and the boarded bus to Bangalore. Afternoon 11.55, we reached majestic.

The we had lunch, then we boarded Dharmastala but at 1.00 pm, after a long journey, we reached hassan at 5.45 pm, hassan is a beautiful and well planned city, which is center to mangaluru and Bengaluru(sorry, Bangalore was changed to bengaluru). The real journey started after hassan, full of forest, full of waterfalls and kerala style houses gives so much pleasure to our eyes.

Also the way the drivers drive the bus is simply fantastic, my dirver constantly driving the bus with 60km even in the danger curves, I really enjoyed his driving. If you wana roller coster ride, go to dharmastala, you will definitely enjoy the bus ride.

We reached that beautiful temple at 9.55 pm, then as usual, I booked a dharma room and we had dinner there. Morning 4.00 am, me and my father went to wipe my hair, it took nearly 3 hours to finish that proccess. Then our family went inside temple at 9.30 am, and came out 11.45 am. Then we went to Dharma Saapadu, means afternoon meals. Its just rice with rasam, but it tastes like amirdham. I already know about the afternoon meals, but I felt the sweetness of the meals at realtime. It was simply superb.

We started our return at 2.00 pm afternoon, and reached home at 4.00 am. All in all, it was a beautiful time, I spend with my family, after my college studies.

UTF-8 CODES IN TERMINAL

Updated: May 07, 2024

Last week is fantastic for me, hope you also enjoyed your previous week. Sometimes I realize that, reading man pages and html pages in `/usr/share/doc` gives us more information, that we never get from google. Last Saturday, I read `utf8`, `Unicode` and `console_codes` man pages one more time to refresh my mind. Also I came up with two scripts which will do some quick works converting tagged unicodes to `utf8` codes and it will display it in terminal. These script only works if you have `/usr/share/i18n/charmaps/UTF-8.gz` file. Here is the scripts.

46.1 unicode2utf8.bash

```
#!/bin/bash

UTF8FILE="/usr/share/i18n/charmaps/UTF-8.gz"
BUFFER=`cat`
BUFFER=`echo "${BUFFER}" | tr ' [a-z]' '[A-Z]'`
BUFFER=`echo "${BUFFER}" | tr '\n' ' '`
BUFFER=`echo "${BUFFER}" | tr '\t' ' '`
BUFFER=`echo "${BUFFER}" | tr -s ' '`
BUFFER=`echo "${BUFFER}" | sed -e 's/>/> /g'`
UTF8BUFFER=""

for UNICODE in ${BUFFER}
do
    UTF8BUFFER="${UTF8BUFFER}"`gunzip -c "${UTF8FILE}" |
    grep "${UNICODE}" |
    awk '{print $2;}'`
done

echo -e "\x1b%G${UTF8BUFFER} // \/\ // \/\ \x1b%@"
```

This script will take tagged Unicode as standard input and display the resolved glyph in standard output. Here is an example screenshot.

```

mokka@mohan-laptop: ~/Development/ShellScripts
File Edit View Terminal Tabs Help
mokka@mohan-laptop:~/Development/ShellScripts$ ./unicode2utf8.bash
<U0B85>
<U000A>
<U0B86>
<U000A>
<U0B87>
<U000A>
அ
ஆ
இ
mokka@mohan-laptop:~/Development/ShellScripts$

```

46.2 unicodes.bash

```

#!/bin/bash
UTF8FILE="/usr/share/i18n/charmaps/UTF-8.gz"

for LANGUAGE
do
    LANGUAGE=`echo "${LANGUAGE}" | tr 'a-z' 'A-Z'`

    gunzip -c "${UTF8FILE}" |
    awk "\$3 ~ /${LANGUAGE}/{print \$0;}" |
    (
        while read UNICODE UTF8CODE DESCRIPTION
        do
            echo -n -e "${UNICODE}\\t"
            echo -n -e "\\x1b%G${UTF8CODE//\\/\\/}\\x1b%@"
            echo -e "\\t${DESCRIPTION}"
        done
    )
done

```

This script will be very interesting to you. If you give a grep pattern matching your language, say `tam` for tamil, this script will fetch all the unicode details about the whole language. Take a look at the screenshot.


```

mokka@mohan-laptop: ~/Development/ShellScripts
File Edit View Terminal Tabs Help
mokka@mohan-laptop:~/Development/ShellScripts$ ./unicodes.bash tam
<U0B82> * TAMIL SIGN ANUSVARA
<U0B83> . TAMIL SIGN VISARGA
<U0B85> . TAMIL LETTER A
<U0B86> . TAMIL LETTER AA
<U0B87> . TAMIL LETTER I
<U0B88> . TAMIL LETTER II
<U0B89> . TAMIL LETTER U
<U0B8A> . TAMIL LETTER UU
<U0B8E> . TAMIL LETTER E
<U0B8F> . TAMIL LETTER EE
<U0B90> . TAMIL LETTER AI
<U0B92> . TAMIL LETTER O
<U0B93> . TAMIL LETTER OO
<U0B94> . TAMIL LETTER AU
<U0B95> . TAMIL LETTER KA
<U0B99> . TAMIL LETTER NGA
<U0B9A> . TAMIL LETTER CA
<U0B9C> . TAMIL LETTER JA
<U0B9E> . TAMIL LETTER NYA
<U0B9F> . TAMIL LETTER TTA
<U0BA3> . TAMIL LETTER NNA
<U0BA4> . TAMIL LETTER TA
<U0BA8> . TAMIL LETTER NA
<U0BA9> . TAMIL LETTER NNNA
<U0BAA> . TAMIL LETTER PA
<U0BAE> . TAMIL LETTER MA
<U0BAF> . TAMIL LETTER YA
<U0BB0> . TAMIL LETTER RA
<U0BB1> . TAMIL LETTER RRA
<U0BB2> . TAMIL LETTER LA
<U0BB3> . TAMIL LETTER LLA
<U0BB4> . TAMIL LETTER LLLA
<U0BB5> . TAMIL LETTER VA
<U0BB6> . TAMIL LETTER SHA
<U0BB7> . TAMIL LETTER SSA
<U0BB8> . TAMIL LETTER SA
<U0BB9> . TAMIL LETTER HA
<U0BBE> . TAMIL VOWEL SIGN AA
<U0BBF> . TAMIL VOWEL SIGN I
<U0BC0> . TAMIL VOWEL SIGN II

```

I actually intend to write my own algorithm to convert unicode to utf8, but I'm started learning one art called don't invent the wheel again. So I used that file to convert unicode to utf8.

And one more thing, there is a quick way you can type your language characters in console using CTRL+SHIFT+U then giving unicode. For example,

CTRL+SHIFT+U0B85 will display அ in console.

HOWTO ADD LICENSE TO SOURCEFILES AUTOMATICALLY(VI)

Updated: May 07, 2024

Hi guys,

After 2 hours of learning, I finally wrote a .vimrc file which will add contents of my license file. Here is my license file,

```
/*
 * <currentfilename>
 * Copyright (C) <date> Mohan Raman
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 2 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 */
```

and here is my .vimrc file

```
" Set column to 80
set columns=80

" Insert License

let licensefile = $HOME."/Development/License/GPL_v2.txt"
let currentfile = expand("%")

if filereadable(licensefile)

    let license = system("grep Copyright ".currentfile)

    if (!len(license))

        execute "0r ".licensefile
```

(continues on next page)

(continued from previous page)

```

let startline = 1
let endline = system("wc -l ".licensefile." | tr -s ' ' | cut -d' ' -f1")
let endline = substitute(endline, "\n", "", "g")
let linerange = startline.",".endline

let command = linerange."s/<currentfilename>/".currentfile."/g"
execute command

let currentyear = strftime("%Y")
let command = linerange."s/<date>/".currentyear."/g"
execute command

endif
endif

```

First it will set column to 80. Then It will search for license file. If the file present, then It will check whether the present file already contains Copyright line or not. If the present fine contains Copyright, then it will not do anything, else, it will read the license file contents into the current file.

It also adds the filename and the current year automatically. Hope this small .vimrc will fit into your needs.

Screenshot of my newfile.c file which is processed by new .vimrc

[illegible]

MAIL WITH ATTACHMENTS (UNIX SHELLSCRIPT)

Updated: May 07, 2024

Lot of us already know howto send attachments in mails using shellscrip. Lot of us already wrote these kind of scrip. Here is one more scrip to do the same thing. I already added all the details in the scrip itself. It is self explainable. If somebody wants, take it.

```
#!/bin/sh
# generatemail.sh - script to send mail with multiple attachments.
# Wrote By: me! mohan43u.
# Date: Thu Mar 06 11:56:12 IST 2009
# License: Its in public domain. Use it whatever way you can.

USAGE="USAGE:

    generatemail.sh [-t tolist] [-s subject] [-e encodetype]
                    [-h] ATTACH_FILE1 ATTACH_FILE2 ...
"
HELP="${USAGE}"
OPTIONS:
    -t tolist      comma seperated e-mail addresses
                  (Default: current userid).

    -s subject     Subject line (Default: 'test mail from generatemail.sh
                  script')

    -e encodetype  Encoding type (x-uencode/base64). only GNU uuencode
                  can encode in base64. If you are using this script
                  in Old school unixes(solaris, HP-UX, AIX, etc) use
                  only 'x-uencode'. Linux can handle both types
                  (Default: x-uencode).

    -h            Print this Help text

This script will read standard input as a mail body and takes filenames as
arguments to generate a preformatted mail content in satndard output. you can
simply pipe the output to any MTA (sendmail, postfix etc.,) to send a mail
with attachments.

Eg:
    $ generatemail.sh -t foo@bar.com -s 'mail with attachments' file1 \
      file2 | /usr/sbin/sendmail -v -i -t
    hi foo,
```

(continues on next page)

(continued from previous page)

```

        I attached file1 and file2 with this mail. Take a look at it.

Thanks,
ben.
[CTRL-D]
$

This above command line will attach file1 and file2 and send a mail
to foo@bar.com. The files file1 and file2 will be encoded in
'x-uencode' and attached with this mail.
"

TO="${USER}"
SUBJECT="testmail from generatemail.sh script"
BOUNDARY="`date +%G%m%d%H%M%S`"
ENCODE_TYPE="x-uencode"

while getopts 't:s:e:h' OPTIONS
do
    case "${OPTIONS}" in
        t) TO="${OPTARG}";;
        s) SUBJECT="${OPTARG}";;
        e) ENCODE_TYPE="${OPTARG}";;
        h) echo "${HELP}" && exit 0;;
        \?) echo "${USAGE}" && exit 1;;
        esac
    done

SHIFT_COUNT=`expr "${OPTIND}" - 1`
shift "${SHIFT_COUNT}"

header()
{
    echo "To: ${TO}"
    echo "Subject: ${SUBJECT}"
    echo "Mime-Version: 1.0"
    echo "Content-Type: multipart/mixed; boundary=\"${BOUNDARY}\""
    echo "Content-Disposition: inline"
    echo "User-Agent: generatemail.sh 0.1"
}

create_octet_part()
{
    BASENAME=`basename "${1}"`
    FILE="${1}"

    if test "${ENCODE_TYPE}" = "base64"
    then
        ENCODED_MESSAGE=`uencode -m "${FILE}" "${BASENAME}" \
        | egrep -v '^begin|^='`
    else
        ENCODED_MESSAGE=`uencode "${FILE}" "${BASENAME}"`
    fi

    echo "Content-Type: application/octet-stream"
    echo "Content-Disposition: attachment; filename=\"${BASENAME}\""
    echo "Content-Transfer-Encoding: ${ENCODE_TYPE}"
    echo

```

(continues on next page)

(continued from previous page)

```
    echo "${ENCODED_MESSAGE}"
    echo
}
create_text_part ()
{
    echo "Content-type: text/plain; charset=utf-8"
    echo "Content-Disposition: inline"
}

# Main
BODY=""`cat`"
header
echo
echo "--${BOUNDARY}"
create_text_part
echo
echo "${BODY}"
echo
for FILENAME
do
    echo "--${BOUNDARY}"
    create_octet_part "${FILENAME}"
done
echo "--${BOUNDARY}--"
```


FOSSCONF2009(UPDATED)

Updated: May 07, 2024

Hi everyone,

Last week went with full of work pressure, No time to do anything for linux. With little time, I prepared for the FossConf 2009.

While preparing, I learnt a lot from gcc, autotools and lot others. Now I have some really good exposure about gcc and its tools. I feel very happy about going to madurai.

I already told to my PL about getting leave on friday, he also accepted it. I had some deep sleep on friday afternoon. Then I started my journey at 5pm. Exactly 8.05pm the bus started from Koyambedu Bus terminal.

On saturday morning 4am, I reached Trichy, then after 3 hours, I reached Maatuthavani Bus terminus in madurai. After some little journey in TownBus, I reached Thiyagaraja Engineering College, near Thiruparangundram, Madurai.

One of the student received me and gave me a seperate room. While going to the college, someone called me, What a surprise, Amachu standing near the Thiruparangundram bus stop, I went near to him, He asked about my journey and suggested me to call Bharathi and inform him.

Once I went into the room, I felt tired and sleepy, but somehow I was awoken and had a small trial on my presentation. Then it became boring, I started to look around the confrence.

Man!! lot of school students, participating and enjoying, Baskar(linuXpert) had full of attention towards those students. I can even take these students if they don't want to study Higher degrees, I can give the same salary a BE guy can get, to these students, b'se they worth equal, he said to me.

Then I saw Amachu's Ubuntu-tam stall, So I felt that I can be a part of Ubuntu-tam. I was a person after Amachu who is helping him a lot. I introduced myself to him, by surprise, he is padu, the same padmanaban from pollachi. After sometime, amachu went away from the stall, me and padu handled students and visitors to our stall. Sametime, padu shared his experience with me, and I shared my happiness with him. One great thing is, we both are Puppy Lovers.

The afternoon 2pm I started my session, I felt disappointed about the awareness of the students, especially BE students about gcc, and its tools. So I quickly finished my session and Bharathi took from where I finished.

We both finished the sessions. Amachu asked me to give company while returning to chennai. We both started Journy at 5pm. On the way we met lot of persons from Yahoo!!, Kerala University, DrunkenMan etc. We boarded SETC Airbus. The Bus started exactly 6.45pm from Maatudhavani Bus Terminus. While travelling, me and amachu exchanged our views about FOSS, Tamil Culture, Upper Middle Class, Respecting others culture, culture clashes in Metropolitan cities like mumbai, bangaluru. Amachu said that he also expects this kind of clash in Chennai.

At last, at 5.45am, we reached thambaram. All in all, this weekend was a meaningful to me. I met lot of FOSS enthusiasts and Also I taught something to the students.

After all the good things, I came to hear about NRCFOSS and their .ppt presentation. A lot of discussion going on in ILUGC mailing list.

In my openion, Whatever the reason is, if an organization is named for giving awareness about FOSS, then why the people working on NRCFOSS not using FOSS tools? How they will make awareness to others? And that person KG should change his attitude when posting in mailing list, his response to Arun SAG mail is not at all acceptable.

Anyway, here is my presentation. Hope this will help someone to program using C in Linux.

KESAVAN'S BIRTHDAY WITH FFMPEG AND MENCODER

Updated: May 07, 2024

Today is kesavan's birthday, shankar reminded us about this day before yesterday by telling that kesavan's birthday is on Feb 19. So I bags a cake and went to room. I also called stalin to come to room.

I reached home at 11pm, within few minutes, stalin also came. But kesavan and santhosh didn't come as per the timing. But we waited.

At 2am, santhosh and kesavan reached the room, and we wished kesavan with full of bumps, and beat him everywhere. He became frustrated and asked us to stop. We finally stopped and he asked why you are beating me?. We said today is your birthday, and he said, Ada nadhari payabulligala, Innaiku illada nalaiku thaanda nan porandha naal.

What a mess, But we said, Parava illada oru naal munnadi adikarathula onnum thappu illa. Then we started to prepare for the celebration. Santhosh started to capture that moments in stalin's mobile(Its a Nokia 3500 classic).

Everybody enjoyed those moments and we captured that moment in mobile. Then kesavan, santhosh started to sleep, But something telling to me to do something with the video.

I transfered that video to my system and started to convert it to mpeg using ffmpeg, but it didn't work, because ffmpeg don't know howto decode smr audio stream. Thankfully mencoder knows howto decode smr audio. I used mencoder to convert that 3gp to mpeg4.

```
$ mencoder -oac lavc -ovc lavc -lavcopts acodec=libmp3lame:vcodec=mpeg4 -af_
↳scale=:~::~:qpall -o Video000.mp4 Video000.3gp
```

Converted that video to mp4 format. Now its time for ffmpeg to convert it into normal mpeg, because In my office, that windows media player don't know how to play mp4. So I decided to convert it to mpeg then to avi. Here is the commands,

```
$ ffmpeg -i Video000.mp4 -ar 44100 -ab 128k -acodec libmp3lame -vcodec mpeg2video -s_
↳qcif Video000.mpg
$ ffmpeg -i Video000.mpg -ar 44100 -ab 128k -acodec wma2 -vcodec msmpeg4 -s qcif_
↳Kesavan_Birthday.avi
```

everything went fine and we uploaded mpeg video to orkut through youtube and we mail .avi video to all our ignite friends.

That night is one dam good night for me. Here is the video,

THUNDERBIRD PET PACKAGE

Updated: May 07, 2024

Hi,

One more post, I compiled thunderbird 2.0.0.19 and created pet package using my new createpet script. That script is improving little by little. And it is doing good job so far.

I really stuckup with the .files file. First I taught it should be present with a pet package. But after seeing gcalculator.pet package, I realize that pet package manager is automatically creating that .files file. So I commented out and updated the new script.

I don't know whether it is good or bad. But I'm really improving it and it is paying off. Also the another script createsfs is going to be useful, because seniors are talking to automount .sfs files on the fly. It will be really interesting if puppy get that advantage over other distros.

Talking about thunderbird, boy!! it the longest compilation my AMD Athlon 1.8GHz machine did. The final pet size is 16MB. But with all headers and libraries.

So here is the link to get the latest thunderbird for puppy.

<http://www.murga-linux.com/puppy/viewtopic.php?t=38542>

FOSSCONF2009

Updated: May 07, 2024

Hi everyone,

Here is another festival in madurai, FOSSCONF2009 going to take place at madurai TEC(Thiagarajar Engineering College). Last year, It happened in MIT chennai. I also presented an introduction about my favourite puppylinux.

This year, I'm going to take C under Linux. Its about howto compile C programs under linux. What are the tools available under linux to create Softwares under linux. Why we need Autotools and lot more.

I think it will take atleast 2 hours to introduce C one more time to them, because at this time, they will be practicing some 4th gen language. Its a tough ask, lets hope students will understand my english.

INDIAN TAMIL MOVIE ALL BGM

Updated: May 07, 2024

Hi everyone,

Here is another post with BGM's of INDIAN movie directed by shankar, casting Kamal Haasan, Manisha koirala, Urmila, Koundamani, matrum palar. A R Rahman established himself as the king of tamil music during this time. Enjoy his fantastic background scores.

These files are created using firefox's DownloadHelper plugin and mplayer. Most of the BGM's were flash contents. I simple play them, while playing, I'll copy the original flv file from /tmp directory or using DownloadHelper, then I'll go to the command prompt and convert that flv file to mp3 like this,

```
$ mplayer -dumpaudio -dumpfile 19SenapathyTheme.mp3 19SenapathyTheme.flv
```

Thats all. you got your mp3 file back from flv container.

But I'm really missing one BGM played when Senathipathy escaped from plane fire and saying `If you currrupt, Senathipathi will come back!!`. I'm searching for that BGM at the end. If someone know the link, kindly post here.

Updated (18/05/2009):

Hey, I got that final BGM, and 2 more extra, Man.. I searched for more than a year for this BGM(I-will-never-die.mp3), finally I got it.

Updated (22/06/2015):

Lost all the files which the above links pointing to. I have re-generated BGMs again (may not be the same ones as previous), here is the new links.

[Indian.clip0.mp3](#)
[Indian.clip1.mp3](#)
[Indian.clip2.mp3](#)
[Indian.clip3.mp3](#)
[Indian.clip4.mp3](#)
[Indian.clip5.mp3](#)
[Indian.clip6.mp3](#)
[Indian.clip7.mp3](#)
[Indian.clip8.mp3](#)
[Indian.clip9.mp3](#)

Enjoy..

CREATEPET SCRIPT

Updated: May 07, 2024

After rearranging my partitions and installing puppy, I got tha hunger to build .pet packages one more time(Ya, I really enjoyed compiling and building pet packages using puppy through my pendrive). One mistake I did previously is, I didn't put puppy into my hard disk. Yes, Its really a big mistake I did. It prevent me to interact more with puppy and diverted me to somewhere else.

But from last week, I'm always boot straight to puppy, from my experience, I would suggest, If you want real geekness, like how gentoo guys do, or how arc linux guys do, or how slackware guys do, then install puppy into your hddisk. Puppy is a perfect platform to do stuffs like compiling packages, hunting for dependencies, rebuilding, packing stuffs.

While using puppylinux, I felt that I'm missing my favourite browser, yes, firefox. So I download the source and started compiling, knowing that it will ask for different libraries, I fullfilled all the gaps by downloading and compiling the dependencies and finally got firefox 3.0.6 running in puppy 4.11.

To create packages, puppy have two scripts, one is new2dir and dir2pet. I used them on beginning, and lost interest with them. They are not giving enough flexibility to create .pet packages, So I went through dir2pet package and noted down what all files that script creating.

Then I started my own script to create .pet package from DESTDIR directory. After some initial testing I started to use createpet to create .pet packages. Here are the steps you need to do to create .pet package from source tarball using createpet script,

```
$ tar xvzf package-version.tar.gz
$ cd package-version
$ ./configure --prefix="/usr" --sysconfdir="/etc" --localstatedir="/var"
$ make all
$ make install DESTDIR="`pwd`-i686"
$ cd ..
$ createpet package-version-i686
```

or if you want a menu entry, then

```
$ createpet -m package-version-i686
```

Thats all, createpet will ask you some question. and finally your pet package will be ready with the name package-version-i686.pet. I also posted this script in puppylinux forum and people started to download. I'm waiting for someone to suggest something about the script.

You can also try this script and suggest me some improvements.

<http://www.murga-linux.com/puppy/viewtopic.php?t=38311>

MOUNAM PESIYADHE BGM

Updated: May 07, 2024

Ooh I love this music, Yuvan you are marvalous!! This is one of the best (I can say it is the best) from you!! keep rocking!!

Today suddenly I heared this BGM music one more time from Mounam Pasiyadhae, an unforgettable film I ever watched, the background music will take anyone to heaven. I have all the songs of Mounam Pasiyadhae, but don't able to get this BGM music.

sometimes back, I learnt howto use ffmpeg, ffmpeg2theora, mplayer and mencoder, so I decided to download entire movie and do some stuff to convert that music into mp3.

While downloading the first part, my temper is out, I desperately want to convert that amazing background music immediately, first I played the downloading movie part with mplayer, enabling (press key o) OSD output. mplayer started to play the movie, man!!! what a music!! I will hear this 2min music for my lifetime :, OSD showed 00:02:12 as its passed time, I decided to use mplayer with -dumpaudio -dumpfile and -endpos 00:02:12 to dump audio only for that duration, but mplayer did differently. It dumped the whole audio upto the downloaded part of the movie. Damn!!

So its fallback time, I decided to go to friendly beast ffmpeg, here is the command I did,

```
$ ffmpeg -i TamilBlast.Com_Mounam_Part\ I.avi -t 00:02:12 -f mp3 -ab 128k Mounam_
↳Pasiyadhae_BGM.mp3
```

Thats all, I got the perfect duration of that music. All ends well and I'm writting this post while hearing this BGM music, thana.. naa.. naa.. naananaa.. naaa.. na!!!

[Mounam Pasiyadhae BGM.mp3](#)

DIVX_411 ON PUPPYLINUX FULL INSTALL

Updated: May 07, 2024

Day before yesterday, I installed my puppy 4.11 into my hddrive. But the thing is, I don't know where to put divx_411.sfs file and make puppy load when it boots.

So I decided to extract divx_411 and copy it into my filesystem, here are the steps.

at /root directory,

```
$ mkdir divx_411_directory
$ mount -o loop -t squashfs divx_411.sfs divx_411_directory
$ cp -Rv divx_411_directory/* /
$ umount divx_411.sfs
$ rmdir divx_411_directory
```

Thats all, I got gcc back into my puppy. Need to build firefox-3.0 from source, I'll post it once I finish.

ABOUT ME

Nothing special, just another FOSS enthusiast

BIBLIOGRAPHY

[CIT01] example reference

INDEX

E

environment variable
 nspawncmdline, 40, 41
 tmuxcmdline, 40

N

nspawncmdline, 40

T

tmuxcmdline, 40