Using Slackware-current

This is actually NOT recommended for beginners. But, sometimes, the need to use the latest software is unavoidable and this COULD be a solution. Plus, this will add a LOT of COOL-points 8-)8-) 😎

Note: *I have removed a section on DE-less installation since my current Slackware installations ARE, in fact, DE-less.*

Note: *I have also removed a section on hijacking other Linux system - this, here, turned out to be VERY similar to what needed to be done.*

[LastUpdated20210620]

I need to use GTK3 version that is newer that the one on 14.2, so I tried the development version (**slackware64-current**). I have done the same once (pre-11), so I am aware that there can be some issues when doing this. I am happy to say that I AM writing this on a slackware64-current (15.0 beta?) installation on my laptop.

So, this is a little note to my future self (or anybody that may be find this useful **DISCLAIMER: Use this at your own risk!**). I am doing this while still using Devuan and I want to keep that for backup, in case things go wrong. (On a side note, the reason I use Devuan was because of the GTK3 version.) So, I have an extra partition that I have reformatted and prepared to download the stuffs I need.

- download official packages (`getslack`)
    - create download path: `<mount-path>/home/share/slackware`
    - create custom getslack config `.getslack`
    - set VERS=`current`
    - exclude kde & xfce
- setup EFI boot
    - `bzImage` in kernels/huge.s (rename to `vmlinuz`)
    - `initrd.img` in isolinux/ (this has the slackware setup)
- boot and run installation as usual
    - DO NOT format partition (packages are there!)
    - pick packages from mounted path
    - manually set kernel to boot (i use huge - generic needs initramfs)
- boot newly installed slackware
    - remove `gnuchess` and `xaos` packages
    - make sure vim does not create backups (edit `usr/share/vim/vimrc`)
    - allow `dmesg` for user
        - append etc/rc.d/rc.local ← echo 0 > /proc/sys/kernel/dmesg_restrict
    - just for personal reference, some useful info on using nmcli

```
nmcli r[adio] wifi
nmcli r[adio] wifi on

nmcli d[evice] wifi list
nmcli d[evice] wifi connect <ssid> password <pass> ifname <wlan0>
```

```
nmcli c[onnection] show
nmcli c[onnection] down <ssid>
nmcli c[onnection] up <ssid>
```

- customize `etc/xdg/user-dirs.defaults` (standard default paths)
- create user

From:

http://azman.unimap.edu.my/dokuwiki/ - **Azman @UniMAP**

Permanent link:

http://azman.unimap.edu.my/dokuwiki/doku.php?id=linux:slack_1current&rev=1736639179

Last update: **2025/01/12 07:46**