Slackware Experience

Personal notes on using Slackware. Some old notes have been archived.

Do note that Slackware also has a great documentation site.

Getting Slackware

The official way to do this is, of course, to get it from slackware.com.

Personally, I have getslack, a bash script based on (more accurately, a trimmed-down version of) the excellent (he termed it *infamous*) mirror-slackware-current.sh by Alien Bob. When going down this path, the next step would be to prepare the installation media.

Slackware Installer ISO Image

I no longer need an ISO image (refer to USB installer below). But, I have my slack2iso script (also based on Alien Bob's script) that can help in creating one using the tree downloaded by `getslack`.

Slackware USB Installer

Alien Bob has provided a script to make/setup/configure a USB-based Slackware installation media. I wanted to do something simpler using the existing files in the Slackware tree that I mirrored using `getslack` (mentioned above). So, here is how I got that working.

1. Create a FAT32 partition
   - use `fdisk` and make sure it is bootable (bootable flag enabled)
   - use `mkdosfs` (e.g. `mkdosfs -F 32 /dev/sdb1`)
2. Use syslinux to provide bootloader
   - create a `/linux/boot/syslinux` folder on the USB
   - type

     ```
     syslinux -d /linux/boot/syslinux /dev/sdb1
     ```

     Note: On newer syslinux, use `-i` to indicate new installation

   - a file `ldlinux.sys` should appear in `/linux/boot/syslinux`
3. Copy boot facilities from Slackware tree to the media
   - copy a kernel from slackware tree to `/linux/boot` (I used `huge.s`)
   - copy `initrd.img` and `message.txt` to `/linux/boot`
   - copy `isolinux.cfg` to `/linux/boot/syslinux` as `syslinux.cfg`
   - edit `syslinux.cfg` accordingly (initrd, kernel params, etc.)
4. Copy slackware<64> in the Slackware tree (I used a shorter folder name like slack on the USB)

And… we're done! Now we have a simple Slackware USB Installer and install it on every computer we can get our hands on! 😎

Note: GPT Disks and EFI

Things moving to (U)EFI and GPT… slowly leaving legacy BIOS and MBR.

Instead of MBR, we use GPT partitioning scheme:

- supports bigger disk
- supports EFI booting (easier to maintain actually :p)

Partition codes are 2-bytes instead (only 1-byte on MBR's partition table). Among the common ones:

- EF00 (EFI System Partition): this is what EFI boot look for
  - format FAT32

    ```
    mkdosfs -F 32 -n MY1EFI /dev/sdxx
    ```

- 0700 (MS Basic Data): Windows Partition
  - format NTFS

    ```
    mkntfs -f -L MY1WIN /dev/sdxx
    ```

- 8300 (Linux filesystem): Linux Partition
  - format EXT4

    ```
    mkfs.ext4 -L MY1LIN /dev/sdxx
    ```

Once boot using EFI, `efibootmgr` tool can be used (available on Slackware 14.2)

- to create an entry labelled `Slackware` with loader file named `\efi\slackware\elilo.efi` located on first partition of first disk (`/dev/sda1`)

  ```
  efibootmgr -c -d /dev/sda -p 1 -L "Slackware" -l
  "\efi\slackware\elilo.efi"
  ```

- to delete an entry xxxx (bootnum)

  ```
  efibootmgr -b xxxx -B
  ```

- to re-order boot sequence

  ```
  efibootmgr -o xxxx,yyyy,zzzz
  ```

2025/01/01 15:54 · azman

Installing Slackware

Installation notes (i.e. packages, configs).

LastUpdated20250112

Basic Install

Using Slackware installer.

- official packages (`getslack`)
  - checkout my `getslack` config file
  - without kde (AND xfce if going DE-less)
  - `removepkg gnuchess xaos xsnow`
  - `removepkg joe nano vim-gvim slackpkg`
- setup/config
  - sample elilo.conf

    elilo.conf

    ```
    prompt
    #chooser=simple
    timeout=50
    default=Slack
    image=vmlinuz-huge
            label=SlackHuge
            read-only
            append="root=/dev/sda2 resume=/dev/sda4 vga=normal"
    image=vmlinuz
            label=Slack
            initrd=initrd.gz
            read-only
            append="root=/dev/sda2 resume=/dev/sda4 vga=normal"
    ```

  - make sure vim does not create backups (edit `/usr/share/vim/vimrc`)
    - or, run `vimstart` (from my1shell repo)
  - dmesg no longer allowed for user
    - append `rc.local` ← `echo 0 > /proc/sys/kernel/dmesg_restrict`
    - or, run `setup_slack` (from my1shell repo)
- additional packages (`getslackpack`)
  - checkout my `getslackpack` config file and repository list
  - (alien) openjdk libreoffice libreoffice-dict-en“
- additional packages (`getslackbuild`)
  - `slackware-xdm-theme`
  - `geany unrar`
  - `nss-mdns avahi libdaemon`

    - actually, scripts from slackbuilds.org (commonly used)

DE-less config

This is what I do for a `lean` (not necessarily minimal, but trimmed to my liking) installation.

- setup acpi from my personal script
- additional packages (`getslackbuild`)
    - `dmenu slock st wmname`
    - `rox-filer pmount`
- custom dwm build
    - using my own build script (which has personalized patches)

Updating

To maintain:

- run slack-update
    - this actually runs 3 scripts (`getslack`,`getslackpack`,`getslackbuild`)
- run slackpatch (if required)
- run getslackbuild build -x -i (if required)

Sample configuration files for the above scripts are here.

2025/01/12 07:57 · azman

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 😎😎😎

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.*

Installing

[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
- get additional packages (`getslackpack`)
  - luckily, alienBob's repo 'supports' current
  - create custom getslackpack config `.getslackpack`
  - (alien) openjdk libreoffice libreoffice-dict-en“
- get additional packages (`getslackbuild`)
  - run as VERS=`14.2 getslackbuild fetch <pkg>`
  - pkgs: `dmenu geany rox-filer slackware-xdm-theme`
  - pkgs: `slock st wmname pmount unrar`
  - pkgs: `nss-mdns avahi libdaemon`
  - *note: rox-filer cannot be compiled, needed patching (this)*
  - i have gathered all the scripts from slackbuilds.org that i use and keep them here

- i want to use dwm
  - using my own custom build script (which has personalized patches)
  - my dwm xinitrc will run loginctl hibernate when battery<30% (→ what i need on my current laptop)

Updating

To maintain:

note: *my `libmy1slack` library will detect current when `etc/slackware-version` has '+' suffix. this sign will disappear when -current is near to a stable release.*

- run slack-update as usual
  - when `-current` going stable, use `SLACKVERS=current slack-update`
- run slack-current instead of `slackpatch`
  - when `-current` going stable, use `-f` switch
  - to see removed packages, use slackview (i.e. `SLACKVERS=current slackview find –alien`)
- update those installed using getslackbuild if needed

2025/01/01 16:06 · azman

Special Needs

Sometimes these can be useful.

Slackware in chroot

20110621 I want to have a 32-bit system running in chroot environment on my Slackware64. I've used such system on Debian using schroot…

20110906 I managed to do this as published here…

20120518 Minor change to the fstab entry for dev, which needs an rbind option so that the pty inside can be valid! Discussed here.

20120524 This is now part of my slackstuff collection (now known as my1shell)… in form of a script called slackroot.

20121031 The path to the chroot installation MUST ALL BE owned by root - or else, users will get a `Write failed: Broken pipe` error.

TODO *A how-to on creating 32-bit chroot on 64-bit Slackware using `slackroot` script.*

slack_chroot32.txt

```
- on my pure slack64 (maintained using getslack/getslackpack)
```

```
$ ARCH=i686 getslack

- create root filesystem using 32-bit packages
# slackroot /opt/chroot32 --arch x86 --desk -x

- copy user/group info from 64-bit system to chroot32
  = will maintain its own login info!
# preproot --init /opt/chroot32

- mount bind 'system' paths
# preproot /opt/chroot32

- ssh into system to use 32-bit chroot
# ssh user@127.0.0.1

- unmount bind 'system' paths
# preproot --done /opt/chroot32
```

2025/01/12 09:59 · azman

Slackware Multilib

Main references are here and here.

1. Basically, need to have C library and compiler capable of multilib. I use `getslackpack` to download required packages from Eric's multilib site. Install as instructed.
2. I already have a 32-bit Slackware tree downloaded using `getslack` which I use for my 32-bit chroot installation. I use `massconvert32.sh` script on this tree. The `massconvert32.sh` script can be used to update as well (built packages are not rebuilt). Install as instructed.
3. My `slackpatch` script has been updated to handle 'blacklisted' 64-bit versions and 'upgraded' compat32 packages

Update20180903 Update20250326

Read here. I now have a more specific script to get multilib stuff (previously part of getslackpack script),

- use getslack-multilib to download alien_bob's multilib stuff
  - compat32-tools glibc(&friends) gcc(&friends) [multilib packages]
  - compat32 library packages [32-bit packages]
- upgrade pure-64 glibc/gcc packages counterparts
  - upgradepkg –reinstall –install-new *.t?z
  - *note:* this includes compat32-tools package (helper scripts, noarch)
- install 32-bit layer support libraries
  - upgradepkg –install-new slackware64-compat32/*-compat32/*.t?z
  - obviously, can be used to upgrade as well

Configure slackpatch to check/ignore multilib stuff

dot-slackpatch

```
PKG_IGNORED="ffmpeg"
# ignoring these standard packages => multilib!
PKG_IGNORED="$PKG_IGNORED aaa_glibc-solibs"
PKG_IGNORED="$PKG_IGNORED gcc gcc-brig gcc-g++ gcc-gdc gcc-gfortran"
PKG_IGNORED="$PKG_IGNORED gcc-gnat gcc-go gcc-objc"
PKG_IGNORED="$PKG_IGNORED glibc glibc-i18n glibc-profile"
```

2025/01/12 10:01 · azman

Slackware Upgrade

For reference.

Upgrading 14.1 to 14.2

A bash shell script I used to upgrade from 14.1 to 14.2.

upgrade_14-1_to_14-2.sh

```bash
#!/bin/bash
# - upgrade 14.1 to 14.2
SLACKVERS="14.2"

# setup path to slackware tree
SLACKROOT=${SLACKROOT:="$(pwd)"}
[ -z "$SLACKARCH" ] && [ -n "$ARCH" ] && SLACKARCH=$ARCH
SLACKARCH=${SLACKARCH:="$(uname -m)"}
SLACKSUFX=${SLACKSUFX:=""}
[ "$SLACKARCH" == "x86_64" ] && SLACKSUFX="64"
SLACKFULL=${SLACKFULL:="slackware${SLACKSUFX}"}
[ -z "$SLACKVERS" ] && [ -n "$RELEASE" ] && SLACKVERS=$RELEASE
SLACKVERS=${SLACKVERS:="current"}
SLACKRELS=${SLACKFULL}-${SLACKVERS}
SLACKPATH=${SLACKROOT}/${SLACKRELS}

# step 1
upgradepkg ${SLACKPATH}/${SLACKFULL}/a/glibc-solibs-*.txz

# step 2
upgradepkg ${SLACKPATH}/${SLACKFULL}/a/pkgtools-*.txz
upgradepkg ${SLACKPATH}/${SLACKFULL}/a/tar-*.txz
upgradepkg ${SLACKPATH}/${SLACKFULL}/a/xz-*.txz
upgradepkg ${SLACKPATH}/${SLACKFULL}/a/findutils-*.txz
```

```
# step 3
for dir in a ap d k l n t tcl x xap xfce ; do
    do_path=${SLACKPATH}/${SLACKFULL}/$dir
    [ ! -d $do_path ] && continue
    ( cd $do_path ; upgradepkg --install-new *.t?z )
done

# step 4
removepkg ConsoleKit apmd bluez-hcidump cxxlibs foomatic-filters \
  gnome-icon-theme imlib kdeadmin kdenetwork kdesdk kdetoys kwallet \
  lesstif libelf libjpeg libxfcegui4 networkmanagement obex-data-server
\
  obexfs open-cobol oxygen-gtk3 phonon-mplayer phonon-xine pil portmap
\
  procps qca-cyrus-sasl qca-gnupg qca-ossl udev xchat xf86-input-aiptek
\
  xf86-video-modesetting xfce4-mixer xfce4-volumed xfwm4-themes

# step 5
# - run chknew

# step 7
# - check boot config
```

2025/01/12 10:15 · azman

Extra Notes

Some things to note…

20210620 Slackware's CPU frequency scaling works (checkout `rc.cpufreq`) - just to remind myself, no need to look into this!

Admin-stuff for non-root user

To allow non-root users basic admin (poweroff,reboot,etc.), add them to `power` group and insert the following to `sudoers`.

```
%power ALL=(ALL) NOPASSWD:/sbin/poweroff
%power ALL=(ALL) NOPASSWD:/sbin/reboot
%power ALL=(ALL) NOPASSWD:/usr/sbin/pm-suspend
%power ALL=(ALL) NOPASSWD:/usr/sbin/pm-hibernate
%power ALL=(ALL) NOPASSWD:/usr/sbin/pm-powersave
```

Note: *The pm-* binaries (pm-utils) are no longer available on Slackware 15.0*

Note20250112: *use loginctl on Slackware 15.0 - no need to add sudoers*

Multicast DNS @ Zeroconf

- install nss-mdns (requires avahi, which requires libdaemon)
- save nss config (in etc)
  - `mv nsswitch.conf nsswitch.conf-orig`
- use provided nss config
  - `cp nsswitch.conf-mdns nsswitch.conf`
- run daemon at startup `rc.avahidaemon` and `rc.avahidnsconfd`

2025/01/12 10:09 · azman

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

Permanent link:
http://azman.unimap.edu.my/dokuwiki/doku.php?id=linux:slackware&rev=1736648579

Last update: **2025/01/12 10:22**