Руним дотфайлами. Как быстро повысить комфорт в любом Linux или WSL

Автор: Ivan Gufan
Опубликовано: 2026-01-12 · Последнее изменение: 2026-01-12 · Время на прочтение: 12 мин

Содержание статьи

- Дотфайлы
- Zsh для Zsh
- Zsh для Fish
- Fish для Fish и Zsh
- Tilix
- Vim
- Neovim
- Выбираем как настроить Kali

Меня часто называют докладчиком, чаще работает среда (до настолько Night Core), но иногда докладчик. Но на докладах, то на докладах, то среда на докладе. В докладе я рассказываю, как быстро повысить комфорт в любом Linux или WSL. Для меня это неразрывные вещи для себя и докладчиков. В этой статье попробую и тем, кто не докладчик, рассказать, как быстро повысить комфорт в любом Linux или WSL.

Рассмотрим три варианта оболочек интерактивных оболочек ПО, для удобства назовём их дотфайлы. Дотфайлы — это файлы конфигурации для оболочек, которые начинаются с точки, и по их расширению по-хорошему и можно и нужно ориентироваться — точка в начале имени, расширение .rc.

Что касается того как мы будем ставить дотфайлы — дотфайлы, которые нужны нескольким пользователям, то мы их будем ставить в каталог /opt/ и каждый пользователь у себя в домашнем каталоге будет подключать свой дотфайл из каталога /opt/.

Хорошо знать, что есть всегда, универсально подходящие дотфайлы для любых версий интерпретатора ОС. Чтобы такие универсальные дотфайлы для любых версий, при обновлении ОС, не приходилось переписывать, мы будем использовать минимальный набор, который гарантированно будет работать, при любой версии ОС.

Полезные для интерактивных оболочек утилиты (бинарники) — это уже или предустановлены по умолчанию, или ставятся из репозиториев ОС. Мы не будем использовать бинарники, которые нужно ставить из сторонних репозиториев, потому что у каждого дистрибутива свой менеджер пакетов, и у каждого дистрибутива свой набор репозиториев, и у каждого дистрибутива свой набор пакетов. Мы будем использовать только то, что есть в репозиториях ОС, то, что есть в репозиториях ОС, то, что есть в репозиториях ОС.

Итак, с чего начать, для начала, рассмотрим дотфайлы для оболочек Zsh, Fish и Tilix. Чтобы сделать более комфортную для себя среду, нам потребуется ещё и Vim, и Neovim, и Tilix. Для Vim и Neovim мы будем использовать один дотфайл, для Tilix — один дотфайл.

Начнём с дотфайлов — конфигурационных файлов для приложений на Linux.

Первый — .zshrc — конфигурационный файл для приложений на Linux.

```
~/.zshrc — Notepad++                                    [ansi, dos, unix]
Файл  Правка  Поиск  Вид  Кодировки  Синтаксисы  Настройки  Инструменты  Макросы  Выполнить  Плагины  Окна  ?
DOMAIN_NAME="ru"
export LANG="ru_RU.UTF-8"
export LC_ALL="ru_RU.UTF-8"
export LC_CTYPE="ru_RU.UTF-8"
export LC_NUMERIC="ru_RU.UTF-8"
export LC_TIME="ru_RU.UTF-8"
export LC_COLLATE="ru_RU.UTF-8"
export LC_MONETARY="ru_RU.UTF-8"
export LC_MESSAGES="ru_RU.UTF-8"
export LC_PAPER="ru_RU.UTF-8"
export LC_NAME="ru_RU.UTF-8"
export LC_ADDRESS="ru_RU.UTF-8"
export LC_TELEPHONE="ru_RU.UTF-8"
export LC_MEASUREMENT="ru_RU.UTF-8"
export LC_IDENTIFICATION="ru_RU.UTF-8"
export TZ="Europe/Moscow"
export EDITOR="/usr/bin/vim"
export VISUAL="/usr/bin/vim"
export PAGER="/usr/bin/less"
export LESSCHARSET="utf-8"
export LESSCHARDEF="8bcfpuvzdeilnrsbtwxyz"
export LESS="-Ri -M -j.5 -x4 -R -S -X -z-1 -P?f%f:(stdin). ?lline %lj:%pB\%:byte %bB ?bytes:%T ?eEOF:%P ?pb\%:%Ldlines"
export MANPAGER="/usr/bin/less -s -M +Gg"
export MANOPT="-P -r -l -Dn -Dl -Dp -Ds -Dt -Dv -Dx -Dy -Dc -Dd -Df -Dg -Dh -Dm -Dr -Dw -Dk -Db -Dq -Du -Dn -Dl -Dp -Ds -Dt -Dv -Dx -Dy -Dc -Dd -Df -Dg -Dh -Dm -Dr -Dw -Dk -Db -Dq -Du"
export GREP_OPTIONS="--color=auto --binary-files=text --exclude-dir=.git --exclude-dir=.svn --exclude-dir=.hg --exclude-dir=.bzr --exclude-dir=.cvs --exclude-dir=.idea --exclude-dir=.vscode --exclude-dir=.cache --exclude-dir=.tmp --exclude-dir=.temp --exclude-dir=.log --exclude-dir=.logs --exclude-dir=.backup --exclude-dir=.bak --exclude-dir=.old --exclude-dir=.new --exclude-dir=.dist --exclude-dir=.build --exclude-dir=.dev --exclude-dir=.test --exclude-dir=.tests --exclude-dir=.spec --exclude-dir=.specs --exclude-dir=.doc --exclude-dir=.docs --exclude-dir=.man --exclude-dir=.manpages --exclude-dir=.info --exclude-dir=.infopages --exclude-dir=.pkg --exclude-dir=.packages --exclude-dir=.lib --exclude-dir=.libs --exclude-dir=.bin --exclude-dir=.binaries --exclude-dir=.obj --exclude-dir=.objs
```