



FEDORA WORKSTATION · GNOME · DNF5 · BTRFS

Fedora El Kitabı

İleri kullanıcılar için

Sistemi içinden yönet. DNF5 geçmişi, systemd, SELinux, firewallld, Btrfs anlık görüntüleri, NVIDIA sürücüler, konteynerler ve açılmayan bir sistemi kurtarma. Yeni başlayan bölümünü bitirdiysen buradan devam et.

BÖLÜMLER

1. DNF5 derinlemesine
2. Depolar: RPM Fusion, COPR, üçüncü taraf
3. RPM sorguları
4. Sürüm yükseltme ve firmware
5. systemd servisleri
6. Günlükler: journalctl
7. Ağ: nmcli, ip, ss
8. Güvenlik duvarı: firewallld
9. SELinux
10. Kullanıcılar ve izinler
11. SSH
12. Btrfs, anlık görüntü ve zram
13. Çekirdek, GRUB ve önyükleme
14. Sürücüler: NVIDIA, codec, donanım hızlandırma
15. Toolbox, Podman, Distrobox
16. Metin işleme ve boru hatları
17. Arşiv, yedek, senkronizasyon
18. Kabuğu kişiselleştirme
19. gsettings ile gizli ayarlar
20. Performans ve güç
21. Sorun giderme ve kurtarma

Sarı eğik yazılar (<paket> gibi) senin dolduracağın yerlerdir. Terminalde yapıştırma Ctrl+Shift+V, kopyalama Ctrl+Shift+C'dir. sudo ile başlayan komutlar sistemi değiştirir; ne yaptığını okumadan çalıştırma.

DNF5 derinlemesine

Fedora 41'den beri `dnf` komutu DNF5'tir. Sözdizimi DNF4'e benzer, ancak bazı alt komutlar ve config-manager değişti. Ayarlar `/etc/dnf/dnf.conf` ya da `/etc/dnf/libdnf5.conf.d/` altında durur.

```
$ dnf history list
```

Tüm kurulum/kaldırma işlemlerini numaralarıyla listeler.

```
$ dnf history info <id>
```

Bir işlemde neyin değiştiğini gösterir.

```
$ sudo dnf history undo <id>
```

O işlemi geri alır (kurulanı kaldırır, kaldırılanı kurar).

```
$ sudo dnf history rollback <id>
```

Sistemi o işlemten sonraki duruma geri sarar.

```
$ dnf provides '*/<dosya>'
```

Bir dosyayı ya da komutu hangi paketin sağladığını bulur. Örnek: `dnf provides '*/libssl.so.3'`.

```
$ dnf repoquery -l <paket>
```

Paketin kuracağı dosyaları kurmadan listeler.

```
$ dnf repoquery --whatrequires <paket> --installed
```

Kurulu paketlerden hangilerinin buna bağımlı olduğunu gösterir.

```
$ dnf repoquery --userinstalled
```

Bağımlılık olarak değil, elle kurduğunuz paketleri listeler.

```
$ dnf group list
```

Paket gruplarını listeler.

```
$ sudo dnf install @development-tools
```

Grup kurar (gcc, make, git vb.). C/C++ için `@c-development`.

```
$ sudo dnf reinstall <paket>
```

Bozulan paketi yeniden kurar.

```
$ sudo dnf downgrade <paket>
```

Paketi depodaki bir önceki sürüme düşürür.

```
$ dnf download <paket>
```

Paketi kurmadan .rpm olarak indirir. `--resolve` bağımlılıkları da indirir.

```
$ sudo dnf distro-sync
```

Kurulu paketleri depodaki sürümlerle eşitler; karışık durumları düzeltir.

```
$ sudo dnf clean all && sudo dnf makecache
```

Önbelleği temizleyip meta verileri yeniden indirir.

```
$ sudo dnf config-manager setopt  
max_parallel_downloads=10
```

Paralel indirme sayısını artırır.

```
$ sudo dnf install 'dnf5-  
command(versionlock)'
```

versionlock eklentisini kurar.

```
$ sudo dnf versionlock add <paket>
```

Paketi mevcut sürümüne sabitler; güncellenmez. `versionlock list / delete`.

```
$ dnf needs-restarting -r
```

Güncellemelerden sonra yeniden başlatma gerekip gerekmediğini söyler.

```
$ dnf upgrade --security
```

Yalnızca güvenlik güncellemelerini kurar. `dnf advisory list` duyuruları listeler.

İPUCU Eski çekirdeklerden kaç tane saklanacağını `installonly_limit` belirler (varsayılan 3).

Depolar: RPM Fusion, COPR, üçüncü taraf

Depo tanımları `/etc/yum.repos.d/*.repo` dosyalarındadır. DNF5'te config-manager sözdizimi `setopt / addrepo` biçimindedir.

```
$ dnf repolist
```

Etkin depoları listeler. `dnf repolist --all` kapalıları da gösterir.

```
$ sudo dnf config-manager setopt <repo-  
kimliği>.enabled=0
```

Depoyu kapatır (1 ile açar).

```
$ sudo dnf config-manager addrepo --from-  
repofile=<URL>
```

İnternetteki bir .repo dosyasını ekler.

```
$ sudo dnf copr enable <kullanıcı>/<proje>
```

COPR (topluluk derleme servisi) deposunu açar. Güvenilirliği sana kalmış.

```
$ sudo dnf copr list
```

Etkin COPR depolarını gösterir. `copr remove` kaldırır.

```
$ sudo dnf config-manager setopt google-chrome.enabled=1
```

Üçüncü taraf depolar açıksa Chrome deposunu etkinleştirir. Ardından `sudo dnf install google-chrome-stable`.

```
$ sudo dnf config-manager setopt fedora-cisco-openh264.enabled=1
```

Firefox'ta WebRTC için OpenH264. Ardından `sudo dnf install mozilla-openh264`.

```
$ sudo rpm --import <anahtar-URL>
```

Bir deponun GPG imza anahtarını içe aktarır.

```
$ sudo dnf install rpmfusion-free-appstream-data rpmfusion-nonfree-appstream-data
```

RPM Fusion paketlerinin Yazılım uygulamasında görünmesini sağlar.

DİKKAT Bir sürüm yükseltmesi öncesinde COPR ve üçüncü taraf depolar yeni sürüme henüz hazır değilse yükseltme çakışma verebilir. Gerekirse geçici olarak kapat.

RPM sorguları

`rpm` doğrudan kurulu paket veritabanını sorgular. Kurulum için DNF kullan; rpm sorgu ve doğrulama için idealdir.

```
$ rpm -qa | sort
```

Kurulu tüm paketler.

```
$ rpm -qa --last | head
```

En son kurulan paketler, tarihleriyle.

```
$ rpm -qf <dosya-yolu>
```

Bir dosyanın hangi pakete ait olduğunu söyler. Örnek: `rpm -qf /usr/bin/ls`.

```
$ rpm -ql <paket>
```

Kurulu paketin dosyaları.

```
$ rpm -qc <paket>
```

Paketin yapılandırma dosyaları.

```
$ rpm -qi <paket>
```

Paket bilgisi: sürüm, kurulum tarihi, lisans.

```
$ rpm -q --changelog <paket> | head -30
```

Paketin değişiklik günlüğü.

```
$ rpm -V <paket>
```

Paketin dosyalarını doğrular; değişmiş ya da eksik dosyaları gösterir.

```
$ rpm -E %fedora
```

Fedora sürüm numarasını verir; betiklerde kullanışlıdır.

```
$ sudo rpm --rebuilddb
```

Bozulmuş RPM veritabanını yeniden oluşturur.

Sürüm yükseltme ve firmware

Fedora yaklaşık altı ayda bir yeni sürüm çıkarır ve her sürüm yaklaşık 13 ay desteklenir. Yükseltme Yazılım uygulamasından ya da terminalden yapılır.

```
$ sudo dnf upgrade --refresh
```

Önce mevcut sürümü tam güncelle.

```
$ sudo dnf system-upgrade download --  
releasever=<sürüm>
```

Yeni sürümün paketlerini indirir. Örnek: `--releasever=45`.

```
$ sudo dnf offline reboot
```

Yeniden başlatıp yükseltmeyi çevrimdışı modda uygular.

```
$ sudo dnf offline log
```

Önceki çevrimdışı yükseltme günlüklerini gösterir.

```
$ sudo dnf autoremove
```

Yükseltmeden sonra artık gereksiz paketleri temizler.

```
$ sudo dnf install rpmconf && sudo rpmconf -a
```

Yükseltmede oluşan .rpmnew / .rpmsave yapılandırma dosyalarını birleştirmene yardım eder.

```
$ fwupdmgtr get-devices
```

Firmware güncellemesi destekleyen aygıtlar.

```
$ fwupdmgr refresh && fwupdmgr update
```

Üretici firmware güncellemelerini (LVFS) kurar.

iPUCU Bir sürüm atlayarak (ör. 43'ten 45'e) yükseltmek desteklenir, ancak en fazla iki sürüm atlanması önerilir.

systemd servisleri

Servisleri (arka plan hizmetleri), zamanlayıcıları ve açılış sürecini systemd yönetir.

```
$ systemctl status <servis>
```

Durumu, son günlük satırlarını ve PID'i gösterir.

```
$ sudo systemctl start|stop|restart <servis>
```

Servisi başlatır / durdurur / yeniden başlatır.

```
$ sudo systemctl enable --now <servis>
```

Açılıшта başlamasını sağlar ve hemen başlatır.

```
$ sudo systemctl disable --now <servis>
```

Açılıшта başlamasını engeller ve durdurur.

```
$ sudo systemctl mask <servis>
```

Servisin hiçbir şekilde başlatılmasını engeller.
`unmask` geri açar.

```
$ systemctl --failed
```

Başarısız olan birimleri listeler.

```
$ systemctl list-units --type=service --state=running
```

Çalışan servisler.

```
$ systemctl list-timers
```

Zamanlanmış görevler (cron'un systemd karşılığı).

```
$ systemctl cat <servis>
```

Birim dosyasının içeriğini ve ek parçaları gösterir.

```
$ sudo systemctl edit <servis>
```

Paket dosyasına dokunmadan override (drop-in) oluşturur.

```
$ sudo systemctl daemon-reload
```

Birim dosyalarını değiştirdikten sonra systemd'nin yeniden okumasını sağlar.

```
$ systemctl --user enable --now <servis>
```

Kullanıcı servisleri; tanımlar

`~/.config/systemd/user/` altında durur.

```
$ loginctl enable-linger $USER
```

Kullanıcı servislerinin oturum açılmadan da çalışmasını sağlar.

```
$ systemd-analyze blame
```

Açılışta hangi servisin ne kadar sürdüğünü sıralar.

```
$ systemd-analyze critical-chain
```

Açılıştaki kritik yolu gösterir.

Günlükler: journalctl

Fedora'da sistem günlükleri systemd journal'da tutulur. `/var/log/messages` yoktur.

```
$ journalctl -b
```

Bu açılıştan beri tüm günlük.

```
$ journalctl -b -1 -p err
```

Bir önceki açılıştaki hatalar. Çökme sonrası ilk bakılacak yer.

```
$ journalctl -u <servis> -f
```

Bir servisin günlüğünü canlı izler.

```
$ journalctl -p warning --since "1 hour ago"
```

Son bir saatteki uyarı ve üstü mesajlar.

```
$ journalctl --since today --grep "<kelime>"
```

Bugünkü günlükte metin arar.

```
$ journalctl -k
```

Yalnızca çekirdek mesajları (dmesg karşılığı).

```
$ journalctl --list-boots
```

Kayıtlı açılışları listeler.

```
$ journalctl --disk-usage
```

Günlüklerin kapladığı alan.

```
$ sudo journalctl --vacuum-time=2weeks
```

İki haftadan eski günlükleri siler. `--vacuum-size=500M` de kullanılabilir.

```
$ coredumpctl list
```

Çöken programların kayıtları. `coredumpctl info <PID>` ayrıntı verir.

```
$ sudo dmesg -w
```

Çekirdek halkasını canlı izler (USB takıp çıkarınca ne olduğunu görmek için).

Ağ: nmcli, ip, ss

Ağı NetworkManager yönetir; komut satırı aracı `nmcli`, metin arayüzü `nmtui`.

```
$ nmcli device status
```

Ağ aygıtları ve durumları.

```
$ nmcli device wifi list
```

Görünen Wi-Fi ağları, sinyal gücüyle.

```
$ nmcli device wifi connect "<SSID>" password  
"<parola>"
```

Wi-Fi ağına bağlanır.

```
$ nmcli connection show
```

Kayıtlı bağlantılar.

```
$ nmcli connection up|down "<ad>"
```

Bağlantıyı açar / kapatır.

```
$ sudo nmcli connection modify "<ad>"  
ipv4.method manual ipv4.addresses  
192.168.1.50/24 ipv4.gateway 192.168.1.1  
ipv4.dns "1.1.1.1 9.9.9.9"
```

Statik IP tanımlar. Sonra `nmcli connection up "<ad>"`.

```
$ nmtui
```

Menülü metin arayüzü; grafik ortam yokken işe yarar.

```
$ ip -br a
```

Arayüzler ve IP adresleri, kısa biçimde.

```
$ ip route
```

Yönlendirme tablosu ve varsayılan ağ geçidi.

```
$ resolvectl status
```

DNS sunucuları (systemd-resolved).

```
$ ss -tulpn
```

Dinleyen TCP/UDP portları ve hangi program olduğu (programı görmek için sudo).

```
$ ping -c 4 <adres>
```

Erişilebilirlik testi.

```
$ tracepath <adres>
```

Paketlerin izlediği yol.

```
$ curl -I https://<site>
```

Bir sitenin HTTP başlıklarını getirir.

```
$ dig <alan-adı>
```

DNS sorgusu (bind-utils paketi).

Güvenlik duvarı: firewalld

Fedora Workstation'da varsayılan bölge `FedoraWorkstation` 'dır; SSH ve 1025–65535 arası portlar gelen bağlantılara açıktır. Kurallar `--permanent` olmadan yeniden başlatınca kaybolur.

```
$ sudo firewall-cmd --state
```

firewalld çalışıyor mu.

```
$ sudo firewall-cmd --get-active-zones
```

Etkin bölgeler ve bağlı arayüzler.

```
$ sudo firewall-cmd --list-all
```

Varsayılan bölgedeki tüm kurallar.

```
$ sudo firewall-cmd --permanent --add-service=http
```

Bir servisi kalıcı olarak açar.

```
$ sudo firewall-cmd --permanent --add-port=8080/tcp
```

Bir portu kalıcı olarak açar.

```
$ sudo firewall-cmd --permanent --remove-port=8080/tcp
```

Portu kapatır.

```
$ sudo firewall-cmd --reload
```

Kalıcı kuralları uygular.

```
$ sudo firewall-cmd --set-default-zone=public
```

Daha kısıtlayıcı bir bölgeye geçer (ör. halka açık ağlarda).

```
$ sudo firewall-cmd --get-services
```

Hazır servis tanımları.

```
$ sudo dnf install firewall-config
```

Grafik yönetim aracı.

SELinux

SELinux bir program ele geçirilse bile erişebileceği yerleri sınırlar. Bir şey "Permission denied" verip dosya izinleri doğruysa neden çoğunlukla SELinux'tur. Çözüm kapatmak değil, doğru etiketi vermektir.

```
$ getenforce
```

Mod: Enforcing / Permissive / Disabled.

```
$ sestatus
```

Ayrıntılı durum.

```
$ sudo ausearch -m AVC -ts recent
```

Son engellemeleri (AVC reddi) gösterir.

```
$ sudo sealert -a /var/log/audit/audit.log
```

Engellemeleri açıklar ve çözüm komutu önerir (setroubleshoot-server paketi).

```
$ ls -Z <yol>
```

Dosyanın SELinux bağlamını (etiketini) gösterir. Süreçler için `ps -eZ`.

```
$ sudo restorecon -Rv <yol>
```

Etiketleri varsayılan kurallara göre düzeltir. mv ile taşınan dosyalarda sık gerekir.

```
$ sudo semanage fcontext -a -t httpd_sys_content_t "/srv/web(/.*)?"
```

Bir klasör için kalıcı etiket kuralı ekler; ardından restorecon çalıştır.

```
$ sudo semanage port -a -t http_port_t -p tcp 8081
```

Bir servisin standart dışı portta dinlemesine izin verir.

```
$ getsebool -a | grep <anahtar>
```

SELinux açma/kapama seçenekleri (boolean).

```
$ sudo setsebool -P httpd_can_network_connect on
```

Bir boolean'ı kalıcı olarak açar.

```
$ sudo setenforce 0
```

Geçici olarak Permissive moda alır (yalnızca teşhis için; yeniden başlatınca döner).

```
$ sudo touch /.autorelabel && sudo reboot
```

Açılışta tüm dosya sistemini yeniden etiketler.

DİKKAT SELinux'u `/etc/selinux/config` üzerinden kalıcı olarak kapatmak yerine sorunu sealert ile çöz. Podman'da bağlanan klasörler için `-v /yol:/yol:Z` kullan.

Kullanıcılar ve izinler

Fedora'da sudo yetkisi `wheel` grubuna üyelikle verilir.

```
$ id
```

Kullanıcı kimliğin ve grupların.

```
$ sudo useradd -m -G wheel <kullanıcı>
```

Ev klasörlü, sudo yetkili yeni kullanıcı.

Ardından `sudo passwd <kullanıcı>`.

```
$ sudo usermod -aG <grup> <kullanıcı>
```

Kullanıcıyı gruba ekler (`-a` olmadan diğer gruplardan çıkarır). Oturumu yeniden açınca etkinleşir.

```
$ sudo userdel -r <kullanıcı>
```

Kullanıcıyı ev klasörüyle siler.

```
$ passwd
```

Kendi parolanı değiştirir.

```
$ chmod +x <dosya>
```

Dosyayı çalıştırılabilir yapar.

```
$ chmod 644 <dosya>
```

Sahibi okur/yazar, diğerleri okur. 755: klasörler ve betikler için tipik.

```
$ chmod -R u+rwX,go-rwx <klasör>
```

Klasörü yalnızca sahibine açar.

```
$ sudo chown -R <kullanıcı>:<grup> <yol>
```

Sahibini değiştirir.

```
$ setfacl -m u:<kullanıcı>:rw <dosya>
```

Belirli bir kullanıcıya ek izin verir (ACL). `getfacl` görüntüler.

```
$ sudo -i
```

Root kabuğu açar. İşin bitince exit.

```
$ sudo visudo -f /etc/sudoers.d/<ad>
```

Sudo kurallarını sözdizimi denetimiyle düzenler.

SSH

```
$ sudo systemctl enable --now sshd
```

Bu bilgisayarda SSH sunucusunu açar (Ayarlar > Sistem > Güvenli Kabuk ile aynı).

```
$ ssh-keygen -t ed25519 -C "<etiket>"
```

Anahtar çifti oluşturur (`~/.ssh/id_ed25519`).

```
$ ssh-copy-id <kullanici>@<sunucu>
```

Açık anahtarını sunucuya yükler; sonra parolasız girersin.

```
$ ssh <kullanici>@<sunucu> -p <port>
```

Bağlanır.

```
$ ssh -L 8080:localhost:80  
<kullanici>@<sunucu>
```

Yerel 8080 portunu sunucunun 80 portuna tünel yapar.

```
$ scp <dosya> <kullanici>@<sunucu>:~/
```

Dosyayı sunucuya kopyalar. Klasör için `-r` .

iPUCU `~/.ssh/config` ile kısa adlar tanımla:

Host sunucum · HostName 192.168.1.10 · User kullanıcı · IdentityFile

`~/.ssh/id_ed25519`

Sonra yalnızca `ssh sunucum` .

iPUCU Parolayla girişi kapatmak için `/etc/ssh/sshd_config.d/10-yerel.conf` dosyasına `PasswordAuthentication no` yaz ve `sudo systemctl restart sshd` .

Btrfs, anlık görüntü ve zram

Fedora varsayılan olarak Btrfs kullanır: `root` alt birimi `/` 'a, `home` alt birimi `/home` 'a bağlanır ve zstd sıkıştırması açıktır. Takas alanı disk yerine RAM'de sıkıştırılmış zram'dır.

```
$ sudo btrfs filesystem usage /
```

Btrfs'e göre gerçek doluluk (df sıkıştırmayı yansıtmaz).

```
$ sudo btrfs subvolume list /
```

Alt birimleri listeler.

```
$ sudo btrfs subvolume snapshot -r /  
/.snapshots/<ad>
```

Kökün salt okunur anlık görüntüsü (önce `/.snapshots` klasörünü oluştur).

```
$ sudo btrfs scrub start /
```

Veri bütünlüğünü sağlama toplamlarıyla denetler. `scrub status /` ile izle.

```
$ sudo compsize /
```

Sıkıştırma oranını gösterir (compsize paketi).

```
$ sudo dnf install snapper btrfs-assistant
```

Otomatik anlık görüntüler için Snapper ve grafik yöneticisi.

```
$ sudo snapper -c root create-config /
```

Kök için Snapper yapılandırması oluşturur.

```
$ sudo snapper -c root create --description "  
<açıklama>"
```

Elle anlık görüntü alır. `snapper -c root list` listeler.

```
$ sudo snapper -c root undochange <önceki>..  
<sonraki>
```

İki görüntü arasındaki değişiklikleri geri alır.

```
$ zramctl
```

zram aygıtı, boyutu ve sıkıştırma algoritması.

```
$ swapon --show
```

Etkin takas alanları.

iPUCU Timeshift'in Btrfs modu Ubuntu tarzı `@ / @home` adlandırması ister; Fedora'nın `root / home` düzeninde çalışmaz. Fedora'da Snapper ya da Timeshift'in `rsync` modu kullanılır.

iPUCU zram boyutunu değiştirmek için `/etc/systemd/zram-generator.conf` oluştur; varsayılanlar `/usr/lib/systemd/zram-generator.conf` içinde.

Çekirdek, GRUB ve önyükleme

Fedora GRUB girdilerini BLS (Boot Loader Specification) biçiminde `/boot/loader/entries/` altında tutar. Çekirdek parametreleri için grubby kullanılır.

```
$ rpm -q kernel
```

Kurulu çekirdekler.

```
$ sudo grubby --info=ALL
```

Tüm önyükleme girdileri ve parametreleri.

```
$ sudo grubby --default-kernel
```

Varsayılan çekirdek.

```
$ sudo grubby --set-default /boot/vmlinuz-  
<sürüm>
```

Varsayılan çekirdeği değiştir.

```
$ sudo grubby --update-kernel=ALL --args="  
<parametre>"
```

Tüm çekirdeklere parametre ekler. Kaldırmak için `--remove-args`.

```
$ cat /proc/cmdline
```

Çalışan çekirdeğin parametreleri.

```
$ sudo grub2-editenv - unset menu_auto_hide
```

GRUB menüsünü her açılışta gösterir.

```
$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

`/etc/default/grub` değişikliklerinden sonra yapılandırmayı yeniden üretir.

```
$ sudo dracut -f
```

Çalışan çekirdek için initramfs'i yeniden oluşturur.

```
$ lsmod | grep <modül>
```

Yüklü çekirdek modülleri. `modinfo` ayrıntı, `sudo modprobe` yükler.

```
$ mokutil --sb-state
```

Secure Boot açık mı.

Sürücüler: NVIDIA, codec, donanım hızlandırma

Bu adımlar RPM Fusion (free + nonfree) eklenmiş olmasını gerektirir.

```
$ sudo dnf install akmod-nvidia xorg-x11-drv-  
nvidia-cuda
```

NVIDIA sürücüsü. Kurulumdan sonra modülün derlenmesi için 5 dakika kadar bekle, sonra yeniden başlat.

```
$ modinfo -F version nvidia
```

Modül derlendi mi; sürüm dönüyorsa hazır.

```
$ nvidia-smi
```

GPU durumu ve kullanımı.

```
$ sudo kmodgenca -a
```

Secure Boot için modül imzalama anahtarı üretir.

```
$ sudo mokutil --import  
/etc/pki/akmods/certs/public_key.der
```

Anahtarı MOK'a kaydeder. Bir parola belirle; yeniden başlatınca mavi MOK ekranında "Enroll MOK" seç ve aynı parolayı gir.

```
$ sudo dnf swap mesa-va-drivers mesa-va-  
drivers-freeworld
```

AMD'de H.264/HEVC donanım video çözme.

```
$ sudo dnf install intel-media-driver
```

Intel (Broadwell ve sonrası) donanım video çözme.

```
$ vainfo
```

VA-API desteğini doğrular (libva-utils paketi).

```
$ lspci -k | grep -A 3 -E "VGA|3D"
```

Ekran kartı ve kullanılan çekirdek sürücüsü.

```
$ sudo dnf install akmod-wl
```

Broadcom Wi-Fi kartları için sürücü.

DİKKAT NVIDIA sürücüsünü nvidia.com'daki .run dosyasıyla kurma; çekirdek güncellemelerinde bozulur. akmod her çekirdek için modülü kendisi yeniden derler.

Toolbx, Podman, Distrobox

Ana sistemi kirletmeden geliştirme araçları kurmak için Toolbx, başka dağıtımların araçları için Distrobox, servisler için Podman kullanılır. Podman, Docker komutlarıyla büyük ölçüde uyumludur ve root gerektirmez.

```
$ toolbox create
```

Sürümüne uygun bir Fedora geliştirme konteyneri oluşturur.

```
$ toolbox enter
```

Konteynere girer; ev klasörün paylaşılır, içinde sudo dnf serbestçe kullanılabilir.

```
$ toolbox list
```

Konteynerleri ve imajları listeler. `toolbox rm -f <ad>` siler.

```
$ sudo dnf install distrobox
```

Distrobox'ı kurar.

```
$ distrobox create -i ubuntu:24.04 -n <ad>
```

Ubuntu konteyneri oluşturur; Arch için `-i archlinux`.

```
$ distrobox enter <ad>
```

İçine girer. İçeride `distrobox-export --app <uygulama>` uygulamayı ana menüye ekler.

```
$ podman run -it --rm fedora bash
```

Geçici bir Fedora konteyneri başlatır.

```
$ podman ps -a
```

Tüm konteynerler.

```
$ podman images
```

Yerel imajlar. `podman system prune -a` kullanılmayanları siler.

```
$ podman run -d -p 8080:80 -v  
~/site:/usr/share/nginx/html:Z  
docker.io/library/nginx
```

Klasör bağlayarak servis çalıştırır; `:Z` SELinux etiketi içindir.

İPUCU Konteyneri systemd servisi yapmak için Quadlet kullan:

`~/config/containers/systemd/` altına `.container` dosyası yaz, `systemctl --user daemon-reload` ve başlat.

```
$ sudo dnf install podman-compose
```

`docker-compose` dosyalarını Podman ile çalıştırır.

Metin işleme ve boru hatları

`|` bir komutun çıktısını diğerine verir. `>` dosyaya yazar (üstüne), `>>` ekler, `2>&1` hataları da aynı yere yönlendirir. `&&` önceki başarılıysa, `||` başarısızsa çalışır.

```
$ grep -rni "<metin>" <klasör>
```

Klasörde büyük/küçük harf duyarlı, satır numaralı yinelemeli arama.

```
$ grep -v "^#" <dosya> | grep -v "^$"
```

Yorum ve boş satırları atar; yapılandırma dosyalarını okumak için.

```
$ sed -i 's/<eski>/<yeni>/g' <dosya>
```

Dosyada bul-değiştir. Önce `-i` olmadan deneyip çıktıya bak.

```
$ awk '{print $1, $3}' <dosya>
```

Satırların 1. ve 3. sütununu yazar.

```
$ sort <dosya> | uniq -c | sort -rn | head
```

En sık tekrar eden satırlar.

```
$ wc -l <dosya>
```

Satır sayısı.

```
$ cut -d: -f1 /etc/passwd
```

İki nokta ile ayrılmış 1. alan (kullanıcı adları).

```
$ find . -name "*.log" -mtime +7 -delete
```

7 günden eski .log dosyalarını siler.

```
$ find . -type f -name "*.jpg" -print0 |  
xargs -0 -I{} cp {} ~/Resimler/
```

Bulunan dosyaları tek tek kopyalar (boşluklu adlarda güvenli).

```
$ <komut> | tee <dosya>
```

Çıktıyı hem ekrana hem dosyaya yazar.

```
$ echo "<satır>" | sudo tee -a /etc/<dosya>
```

Root dosyasına ekleme yapar (`sudo echo >>` çalışmaz).

```
$ diff -u <eski> <yeni>
```

İki dosyanın farkı.

```
$ watch -n 2 <komut>
```

Komutu 2 saniyede bir tekrarlar.

```
$ curl -s <URL> | jq '.'
```

JSON çıktısını biçimlendirir (jq paketi).

Arşiv, yedek, senkronizasyon

```
$ tar -czvf <arşiv>.tar.gz <klasör>
```

Klasörü gzip ile arşivler.

```
$ tar --zstd -cvf <arşiv>.tar.zst <klasör>
```

zstd ile arşivler; gzip'ten hızlı ve daha iyi sıkıştırır.

```
$ tar -xvf <arşiv>
```

Her tür tar arşivini açar. Hedef klasör için `-C <klasör>`.

```
$ tar -tvf <arşiv>
```

Açmadan içeriği listeler.

```
$ zip -r <arşiv>.zip <klasör>
```

Zip oluşturur. Açmak için `unzip <arşiv>.zip`.

```
$ 7z x <arşiv>.7z
```

7z / rar gibi biçimleri açar (7zip paketi).

```
$ rsync -avh --progress <kaynak>/ <hedef>/
```

Klasörü eşitler; yalnızca değişenleri kopyalar. Sondaki / "içeriği" anlamına gelir.

```
$ rsync -avh --delete --dry-run <kaynak>/  
<hedef>/
```

Hedefte kaynakta olmayanları silerek aynala; önce --dry-run ile ne olacağını gör.

```
$ rsync -avz -e ssh <kaynak>/  
<kullanıcı>@<sunucu>:<yol>/
```

SSH üzerinden uzak yedek.

```
$ rclone config
```

Google Drive, OneDrive, S3 gibi bulut hedefleri tanımlar; `rclone sync` ile eşitler.

Kabuğu kişiselleştirme

Fedora'nın varsayılan `~/.bashrc` dosyası `~/.bashrc.d/` klasöründeki tüm dosyaları otomatik okur. Kendi eklemelerini oraya ayrı dosyalar olarak koymak düzenli tutar.

```
$ mkdir -p ~/.bashrc.d && nano  
~/.bashrc.d/takma-adlar.sh
```

Takma adlar için ayrı dosya.

```
alias guncelle='sudo dnf upgrade --refresh &&  
flatpak update -y'
```

Tek kelimeyle sistem + Flatpak güncellemesi. Dosyaya yazıp kaydet.

```
alias ll='ls -lah --group-directories-first'
```

Kullanışlı ayrıntılı liste.

```
$ source ~/.bashrc
```

Değişiklikleri yeni terminal açmadan yükler.

```
export PATH="$HOME/bin:$PATH"
```

PATH'e klasör ekler. `~/.local/bin` Fedora'da zaten PATH'tedir.

```
$ echo $SHELL
```

Kullandığın kabuk.

```
$ sudo dnf install zsh util-linux-user &&  
chsh -s /usr/bin/zsh
```

Zsh kurar ve varsayılan yapar (oturumu yeniden aç). Fish için aynı yöntem.

```
$ type <komut>
```

Bir adın takma ad mı, fonksiyon mu, dosya mı olduğunu gösterir.

iPUCU Betik yazarken ilk satıra `#!/usr/bin/env bash` , ikinci satıra `set -euo pipefail` koy; hatada durur, tanımsız değişkeni yakalar. Sonra `chmod +x betik.sh` .

gsettings ile gizli ayarlar

GNOME ayarlarının hepsi dconf veritabanında durur. Ayarlar uygulamasında olmayan seçeneklere gsettings ile ulaşırın.

```
$ gsettings set org.gnome.desktop.interface  
color-scheme 'prefer-dark'
```

Koyu tema (açık için 'default').

```
$ gsettings set org.gnome.desktop.interface  
clock-show-seconds true
```

Saatte saniyeleri gösterir.

```
$ gsettings set org.gnome.desktop.interface  
clock-show-weekday true
```

Saatin yanında gün adını gösterir.

```
$ gsettings set org.gnome.mutter center-new-  
windows true
```

Yeni pencereleri ekran ortasında açar.

```
$ gsettings set  
org.gnome.desktop.peripherals.touchpad tap-  
to-click true
```

Dokunarak tıklama.

```
$ gsettings set  
org.gnome.desktop.wm.preferences focus-mode  
'sloppy'
```

Fare üstüne gelen pencere odaklanır.

```
$ gsettings set org.gnome.shell disable-user-  
extensions true
```

Tüm eklentileri tek seferde kapatır (sorun giderme). false ile geri açılır.

```
$ gsettings list-recursively  
org.gnome.desktop.interface
```

Bir şemadaki tüm anahtarlar ve değerleri.

```
$ gsettings reset <şema> <anahtar>
```

Bir ayarı varsayılamaya döndürür.

```
$ dconf dump / > ~/gnome-ayarlar.ini
```

Tüm GNOME ayarlarını dosyaya yedekler.

```
$ dconf load / < ~/gnome-ayarlar.ini
```

Yedeği geri yükler; yeni kurulumda ayarları taşımak için.

```
$ sudo dnf install dconf-editor
```

Tüm anahtarlara açıklamalarıyla gözümlenebileceğin grafik araç.

Performans ve güç

Fedora güç profillerini tuned (tuned-ppd uyumluluk katmanıyla) yönetir; Ayarlar > Güç'teki modlar buna bağlıdır.

```
$ powerprofilesctl list
```

Güç profilleri ve etkin olan.

```
$ powerprofilesctl set performance
```

Profil değiştirir: power-saver / balanced / performance.

```
$ tuned-adm active
```

Etkin tuned profili. `tuned-adm list` tümünü listeler.

```
$ sudo dnf install lm_sensors && sensors
```

İşlemci ve disk sıcaklıkları.

```
$ sudo dnf install btop nvidia-smi
```

Gelişmiş sistem ve GPU izleyici.

```
$ sudo iotop -o
```

Diske en çok yazan süreçler (iotop paketi).

```
$ systemd-analyze
```

Toplam açılış süresi (firmware, yükleyici, çekirdek, kullanıcı alanı).

```
$ sudo fstrim -av
```

SSD'de TRIM çalıştırır (fstrim.timer haftalık olarak zaten yapar).

```
$ cat /sys/class/power_supply/BAT0/capacity
```

Pil yüzdesi; `upower -i $(upower -e | grep BAT)` ayrıntılı pil sağlığı verir.

Sorun giderme ve kurtarma

Sistem açılmadığında ya da bir güncelleme bir şeyi bozduğunda sırasıyla denenecekler.

iPUCU 1. Eski çekirdekle aç. GRUB menüsünde bir önceki girdiyi seç. Çalışıyorsa sorun çekirdektedir; `sudo dnf history undo` ya da bir sonraki güncellemeyi bekle.

iPUCU 2. Metin konsoluna geç. Ctrl+Alt+F3 (F2–F6). Giriş yapıp `journalctl -b -p err` ile hatalara bak.

```
$ sudo systemctl restart gdm
```

Grafik oturum yöneticisini yeniden başlatır.

```
$ sudo dnf history undo <id>
```

Sorunlu güncellemeyi geri alır.

```
$ sudo dnf distro-sync && sudo dnf check
```

Paket tutarsızlıklarını giderir ve bozuk bağımlılıkları raporlar.

iPUCU 3. Tek kullanıcı / kurtarma modu. GRUB'da girdinin üstünde `e` 'ye bas, `linux` satırının sonuna `systemd.unit=rescue.target` ekle, Ctrl+X ile başlat.

iPUCU 4. Canlı USB'den chroot. Fedora canlı USB ile aç, terminalde sırasıyla aşağıdakileri çalıştır. Disk adlarını `lsblk` ile kendi sistemine göre değiştir.

```
$ sudo mount -o subvol=root /dev/<nvme0n1p3> /mnt
```

Btrfs kök alt birimini bağlar.

```
$ sudo mount /dev/<nvme0n1p2> /mnt/boot &&  
sudo mount /dev/<nvme0n1p1> /mnt/boot/efi
```

/boot ve EFI bölümlerini bağlar.

```
$ for d in dev proc sys run; do sudo mount --  
rbind /$d /mnt/$d; done && sudo chroot /mnt
```

Sistem klasörlerini bağlayıp kurulu sisteme geçer; içeride `dnf`, `grub2`, `dracut` kullanılabilir.

```
$ sudo dnf reinstall shim-* grub2-efi-*  
grub2-common
```

EFI önyükleyicisini onarır (chroot içinde, `sudo` olmadan).

Komutlar Fedora Workstation'ın güncel sürümleri (DNF5, GNOME, Ptyxis terminali) için yazıldı. Menü adları Türkçe arayüze göre verildi; İngilizce arayüzde parantez içindeki karşılıklara bak. `sudo` ile başlayan komutlar sistemi değiştirir, ne yaptığını okumadan çalıştırma.