ARCH LINUX · ROLLING · PACMAN · AUR

Arch Linux El Kitabı

İleri kullanıcılar için

Sistemi içinden yönet. pacman ve AUR'un derinlikleri, kayan sürüm bakımı, systemd, ağ, ufw, Btrfs anlık görüntüleri, çekirdek ve önyükleyici, NVIDIA sürücüleri ve arch-chroot ile kurtarma.

BÖLÜMLER

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.

pacman derinlemesine

pacman ayarları `/etc/pacman.conf`, işlem geçmişi `/var/log/pacman.log` içindedir.

```
$ pacman -Qo <dosya-yolu>
```
Bir dosyanın hangi pakete ait olduğunu söyler.

```
$ pacman -Ql <paket>
```
Kurulu paketin dosyaları.

```
$ sudo pacman -Fy && pacman -F <dosya-adı>
```
Kurulu olmayan paketlerde dosya arar (ör. bir komutu hangi paket sağlıyor).

```
$ pacman -Qm
```
Resmi depolarda olmayan paketler (AUR ve elle kurulanlar).

```
$ pacman -Qkk <paket>
```
Paketin dosyalarını denetler; değişmiş ya da eksik dosyaları gösterir.

```
$ pactree <paket>
```
Bağımlılık ağacı. Tersine (kim buna bağımlı): `pactree -r <paket>` (pacman-contrib).

```
$ sudo pacman -S --needed <paket>
```
Zaten güncel olanı yeniden kurmaz.

```
$ sudo pacman -D --asexplicit <paket>
```
Bağımlılık olarak gelen paketi "elle kurulmuş" işaretler; yetim temizliğinde silinmez.

```
$ paccache -r
```
Önbellekte her paketin son 3 sürümü dışındakileri siler. `paccache -ruk0` kaldırılmış paketlerinkini temizler.

```
$ pacman -Qqe > paketler.txt
```
Elle kurulan paketlerin listesini yedekler. Yeni sistemde: `sudo pacman -S --needed - < paketler.txt`.

İPUCU Bir paketin güncellenmesini engellemek için `/etc/pacman.conf` içindeki `IgnorePkg =` satırına ekle. Uzun süre tutmak kısmi güncelleme riski yaratır.

```
$ grep -iE 'installed|upgraded|removed'
/var/log/pacman.log | tail -30
```
Son paket işlemleri.

```
$ expac -H M '%m\t%n' | sort -h | tail -20
```

En çok yer kaplayan 20 paket (expac paketi).

AUR derinlemesine

AUR yardımcısı olmadan da paket kurabilirsin; yay'ın arkada yaptığı budur.

```
$ git clone https://aur.archlinux.org/<paket>.git && cd <paket>
```

Tarifi indirir.

```
$ less PKGBUILD
```

Tarifi oku: kaynak adresleri, derleme ve kurulum adımları.

```
$ makepkg -si
```

Bağımlılıkları kurar, derler ve paketi kurar.

```
$ git pull && makepkg -si
```

Elle kurulan AUR paketini günceller.

```
$ yay -G <paket>
```

Tarifi derlemeden indirir.

```
$ yay -Qua
```

Güncellemesi bekleyen AUR paketleri.

```
$ yay -Ps
```

Sistem istatistiği: paket sayıları, en büyük paketler, önbellek boyutu.

```
$ yay -Sc
```

yay'ın derleme önbelleğini temizler (`~/.cache/yay`).

İPUCU Derleme ayarları `/etc/makepkg.conf` içindedir. Uzun derlemeleri hızlandırmak için `MAKEFLAGS="-j$(nproc)"` satırını ekle.

Kayan sürümle yaşamak: bakım

Arch'ı sorunsuz tutmanın sırrı düzenli ve bilinçli güncellemedir: sık güncelle, haberleri oku, .pacnew dosyalarını birleştir.

```
$ checkupdates
```

Sistemi değiştirmeden bekleyen güncellemeleri listeler (pacman-contrib).

```
$ yay -Pw
```

Okunmamış Arch haberlerini gösterir; güncellemeden önce bak.

```
$ sudo pacdiff
```

`.pacnew` / `.pacsave` dosyalarını bulur, farklarını gösterip birleştirmeni sağlar.

```
$ sudo pacman -U
/var/cache/pacman/pkg/<paket>-
<sürüm>.pkg.tar.zst
```

Sorunlu bir güncellemeden sonra paketi önbellekteki eski sürümüne düşürür.

> İPUCU Önbellekte yoksa Arch Linux Archive'dan (archive.archlinux.org/packages) eski sürümü indirip aynı komutla kurabilirsin. AUR'daki `downgrade` aracı bunu menüyle yapar.

```
$ sudo pacman -Sy archlinux-keyring && sudo
pacman -Su
```

İmza hatalarında anahtarlığı önce günceller (bu sıranın tek istisnası).

```
$ sudoedit /etc/xdg/reflector/reflector.conf
```

reflector.timer'ın kullanacağı ülke ve sıralama ayarları.

```
$ sudo timeshift --create --comments
"güncelleme öncesi"
```

Büyük güncellemeden önce anlık görüntü. AUR'daki `timeshift-autosnap` her pacman işleminden önce kendiliğinden alır.

```
$ systemctl --failed && journalctl -b -p err
```

Güncellemeden sonra hızlı sağlık kontrolü.

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>
```

Açılışta başlamasını sağlar ve hemen başlatır.

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

Açılışta 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-unit-files --state=enabled`	Açılışta başlayan birimler.
`$ 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

Sistem günlükleri systemd journal'da tutulur. Arch'ta günlük kalıcıdır; ayrı bir syslog servisi 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 (systemd-coredump kuruluysa). `coredumpctl info <PID>` ayrıntı verir.
`$ sudo dmesg -w`	Çekirdek halkasını canlı izler (USB takıp çıkarınca ne olduğunu görmek için).

Ağ: nmcli, ip, ss

Masaüstü kurulumlarında 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.
`$ sudo hostnamectl set-hostname <yeni-ad>`	Bilgisayarın adını değiştirir.
`$ ip -br a`	Arayüzler ve IP adresleri, kısa biçimde.
`$ ip route`	Yönlendirme tablosu ve varsayılan ağ geçidi.
`$ resolvectl status`	DNS sunucuları (systemd-resolved kullanılıyorsa). Aksi halde `cat /etc/resolv.conf`.
`$ 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 paketi).
`$ iwctl`	iwd'nin etkileşimli aracı (kurulum ISO'sunda Wi-Fi için): `device list`, `station wlan0 scan`, `station wlan0 get-networks`, `station wlan0 connect "<SSID>"`.

`$ sudo systemctl enable --now NetworkManager`	NetworkManager servisini açar; masaüstü ağ menüsü bunu kullanır.

Güvenlik duvarı: ufw

Arch'ta varsayılan olarak güvenlik duvarı yoktur. ufw (Uncomplicated Firewall), nftables üzerinde basit bir arayüzdür.

`$ sudo pacman -S ufw && sudo systemctl enable --now ufw`	ufw'yi kurar ve servisini etkinleştirir.
`$ sudo ufw status verbose`	Durum ve kurallar.
`$ sudo ufw default deny incoming && sudo ufw default allow outgoing`	Gelen bağlantıları kapat, gidenlere izin ver (masaüstü için makul varsayılan).
`$ sudo ufw enable`	Güvenlik duvarını açar ve açılışta etkin kalmasını sağlar.
`$ sudo ufw allow ssh`	SSH'ye (22) izin verir. `sudo ufw limit ssh` kaba kuvvet denemelerini yavaşlatır.
`$ sudo ufw allow 8080/tcp`	Bir portu açar.
`$ sudo ufw allow from 192.168.1.0/24 to any port 445`	Yalnızca yerel ağdan belirli bir porta izin verir.
`$ sudo ufw status numbered`	Kuralları numaralı listeler.
`$ sudo ufw delete <numara>`	Numarasıyla kural siler.
`$ sudo ufw app list`	Paketlerin tanımladığı hazır uygulama profilleri.
`$ sudo pacman -S gufw`	Grafik arayüz.

DİKKAT Docker kendi iptables/nftables kurallarını yazdığı için yayımladığı portlar ufw kurallarını atlar. Docker kullanıyorsan portları `127.0.0.1:8080:80` biçiminde yerel adrese bağla.

Kullanıcılar ve izinler

Bu sistemde sudo yetkisi `wheel` grubuna üyelikle verilir.

`$ id`	Kullanıcı kimliğin ve grupların.
`$ sudo useradd -m -G wheel -s /bin/bash <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.

```
$ sudo EDITOR=nano visudo
```

`%wheel ALL=(ALL:ALL) ALL` satırının başındaki # işaretini kaldır; yoksa wheel grubu sudo kullanamaz.

SSH

```
$ sudo pacman -S openssh && sudo systemctl enable --now sshd
```

SSH sunucusunu kurar ve başlatır.

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

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

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

Açık anahtarını sunucuya yükler; sonra parolasız girersin.

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

Bağlanır.

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

Yerel 8080 portunu sunucunun 80 portuna tünel yapar.

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

Dosyayı sunucuya kopyalar. Klasör için `-r`.

İPUCU `~/.ssh/config` ile kısa adlar tanımla:
`Host sunucum · HostName 192.168.1.10 · User kullanici · IdentityFile ~/.ssh/id_ed25519`
Sonra yalnızca `ssh sunucum`.

İPUCU Parolayla girişi kapatmak için `/etc/ssh/sshd_config.d/10-yerel.conf` dosyasına `PasswordAuthentication no` yaz ve SSH servisini yeniden başlat.

Btrfs, anlık görüntüler ve zram

archinstall Btrfs seçildiğinde `@`, `@home`, `@log`, `@pkg` ve `@.snapshots` alt birimlerini oluşturur. Bu düzen hem Timeshift hem Snapper ile kullanılabilir.

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

Alt birimleri listeler.

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

Btrfs'e göre gerçek doluluk.

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

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

```
$ sudo pacman -S snapper snap-pac
```

Snapper ve her pacman işleminden önce/sonra otomatik anlık görüntü alan snap-pac.

```
$ sudo snapper -c root list
```

Snapper anlık görüntüleri.

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

İki görüntü arasındaki değişiklikleri geri alır.

```
$ sudo pacman -S grub-btrfs && sudo systemctl enable --now grub-btrfsd
```

Anlık görüntüleri GRUB menüsüne ekler; bozuk sistemi eski görüntüden açarsın. Timeshift ile kullanmak için servisi `--timeshift-auto` seçeneğiyle düzenle.

```
$ sudo pacman -S zram-generator
```

RAM'de sıkıştırılmış takas. `/etc/systemd/zram-generator.conf` dosyasına `[zram0]` ve `zram-size = ram / 2` yaz, yeniden başlat.

```
$ zramctl && swapon --show
```

zram ve takas durumu.

DİKKAT Snapper'ın `create-config` komutu kendi `.snapshots` alt birimini oluşturmak ister ve archinstall'ın `@.snapshots` alt birimiyle çakışır. Kurulum sırasını Arch Wiki'nin Snapper sayfasından takip et.

Çekirdek, initramfs ve önyükleyici

Arch'ta birden fazla çekirdek birlikte kurulabilir. Yedek olarak `linux-lts` kurmak, yeni çekirdek sorun çıkarırsa açılış menüsünden geri dönmeni sağlar.

```
$ sudo pacman -S linux-lts linux-lts-headers
```

Uzun süreli destek çekirdeğini yedek olarak kurar. Diğerleri: `linux-zen`, `linux-hardened`.

```
$ uname -r && pacman -Q linux
```
Çalışan ve kurulu çekirdek sürümü; farklıysa yeniden başlatman gerekir.

```
$ sudoedit /etc/mkinitcpio.conf
```
initramfs ayarları (HOOKS, MODULES).

```
$ sudo mkinitcpio -P
```
Tüm çekirdekler için initramfs'i yeniden üretir.

```
$ bootctl status
```
systemd-boot kullanılıyorsa durum ve girdiler.

İPUCU systemd-boot'ta çekirdek parametreleri `/boot/loader/entries/*.conf` dosyalarındaki `options` satırındadır; bekleme süresi `/boot/loader/loader.conf` içindeki `timeout`.

```
$ sudoedit /etc/default/grub
```
GRUB kullanılıyorsa ayarlar; parametreler `GRUB_CMDLINE_LINUX_DEFAULT` satırına.

```
$ sudo grub-mkconfig -o /boot/grub/grub.cfg
```
GRUB yapılandırmasını yeniden üretir (yeni çekirdek ya da mikro kod sonrası).

```
$ cat /proc/cmdline
```
Çalışan çekirdeğin parametreleri.

```
$ lsmod | grep <modül>
```
Yüklü modüller. `modinfo` ayrıntı, `sudo modprobe` yükler.

Sürücüler: NVIDIA, AMD, Intel

Grafik sürücüleri resmi depolardadır. 32 bit oyun desteği için multilib açık olmalı.

```
$ sudo pacman -S nvidia-open nvidia-utils
lib32-nvidia-utils
```
NVIDIA Turing (GTX 16xx / RTX 20xx) ve sonrası için sürücü, `linux` çekirdeğiyle. LTS için `nvidia-open-lts`, diğer çekirdekler için `nvidia-open-dkms`.

İPUCU NVIDIA kurduktan sonra `/etc/mkinitcpio.conf` içindeki HOOKS satırından `kms` 'i çıkar (nouveau'nun yüklenmesini engeller) ve `sudo mkinitcpio -P` çalıştır. Daha eski kartlar için AUR'daki eski sürücü dallarına Arch Wiki'nin NVIDIA sayfasından bak.

```
$ nvidia-smi
```
GPU durumu ve kullanımı.

`$ sudo pacman -S mesa vulkan-radeon lib32-mesa lib32-vulkan-radeon`	AMD: OpenGL, Vulkan ve donanım video çözme.
`$ sudo pacman -S mesa vulkan-intel lib32-vulkan-intel intel-media-driver`	Intel: OpenGL, Vulkan ve donanım video çözme.
`$ sudo pacman -S libva-utils && vainfo`	VA-API donanım hızlandırma desteğini doğrular.
`$ lspci -k | grep -A 3 -E "VGA|3D"`	Ekran kartı ve kullanılan çekirdek sürücüsü.
`$ sudo pacman -S steam`	Steam (multilib gerekir; kurulumda Vulkan sürücüsünü sorar, kartına uygun olanı seç).

Podman, Docker, Distrobox

Ana sistemi kirletmeden başka dağıtımların araçlarını kullanmak için Distrobox, servisler için Podman ya da Docker kullanılır. Podman, Docker komutlarıyla büyük ölçüde uyumludur ve root gerektirmez.

`$ sudo pacman -S podman distrobox`	Podman ve Distrobox'ı kurar.
`$ distrobox create -i registry.fedoraproject.org/fedora:latest -n <ad>`	Başka bir dağıtımın konteynerini oluşturur (ör. `-i archlinux`, `-i ubuntu:24.04`).
`$ distrobox enter <ad>`	İçine girer; ev klasörün paylaşılır. İçeride `distrobox-export --app <uygulama>` uygulamayı ana menüye ekler.
`$ podman run -it --rm debian bash`	Geçici bir konteyner 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
docker.io/library/nginx
```

Klasör bağlayarak servis çalıştırır.

```
$ sudo pacman -S docker docker-compose &&
sudo systemctl enable --now docker
```

Docker'ı kurar ve servisini başlatır.

```
$ sudo usermod -aG docker $USER
```

Docker'ı sudo'suz kullanmak için (oturumu yeniden aç). Bu grup root'a eşdeğer yetki verir.

İPUCU Podman konteynerini systemd servisi yapmak için Quadlet kullan: `~/.config/containers/systemd/` altına `.container` dosyası yaz, `systemctl --user daemon-reload` ve başlat.

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 duyarsız, 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

Bash ayarları ev klasöründeki `~/.bashrc` dosyasındadır. Takma adları ve değişkenleri dosyanın sonuna ekle.

```
$ nano ~/.bashrc
```

Kabuk ayar dosyasını açar.

```
alias guncelle='sudo pacman -Syu'
```

Tek kelimeyle sistem güncellemesi. Dosyaya yazıp kaydet.

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

Kullanışlı ayrıntılı liste.

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

Değişiklikleri yeni terminal açmadan yükler.

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

Kendi betiklerinin klasörünü PATH'e ekler.

```
$ echo $SHELL
```

Kullandığın kabuk.

```
$ sudo pacman -S zsh && 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.

İPUCU 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 GNOME ayarları

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

`$ 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.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 varsayılana 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 pacman -S dconf-editor`	Tüm anahtarlara açıklamalarıyla göz atabileceğin grafik araç.

Performans ve güç

Güç profilleri için `sudo pacman -S power-profiles-daemon` kurup `sudo systemctl enable --now power-profiles-daemon` ile etkinleştir; GNOME ve KDE bunu kullanır.

`$ powerprofilesctl list`	Güç profilleri ve etkin olan (power-profiles-daemon).
`$ powerprofilesctl set performance`	Profil değiştirir: power-saver / balanced / performance.
`$ sudo pacman -S lm_sensors`	İşlemci ve disk sıcaklıkları; kurduktan sonra `sensors`.
`$ sudo pacman -S btop nvtop`	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. Haftalık otomatik yapmak için `sudo systemctl enable --now fstrim.timer`.
`$ upower -i $(upower -e | grep BAT)`	Pil sağlığı, kapasite ve şarj bilgisi.
`$ free -h && swapon --show`	RAM ve takas alanları.

Sorun giderme ve kurtarma

Sistem açılmadığında ya da bir güncelleme bir şeyi bozduğunda sırasıyla denenecekler. Arch'ta en güçlü araç kurulum ISO'su ve `arch-chroot`'tur.

İPUCU **1. Yedek çekirdekle ya da anlık görüntüyle aç.** linux-lts kuruluysa açılış menüsünden onu seç; grub-btrfs kuruluysa eski bir anlık görüntüden açabilirsin.

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

`$ sudo systemctl restart display-manager`	Grafik oturum yöneticisini (GDM, SDDM) yeniden başlatır; açık oturum kapanır.

İPUCU **3. ISO'dan arch-chroot.** Arch kurulum USB'siyle aç ve aşağıdakileri sırayla çalıştır (ISO'da zaten root olursun). Disk adlarını `lsblk` ile kendi sistemine göre değiştir.

`# mount -o subvol=@ /dev/<nvme0n1p2> /mnt`	Btrfs kök alt birimini bağlar (ext4 ise `-o subvol=@` olmadan).
`# mount /dev/<nvme0n1p1> /mnt/boot`	EFI bölümünü bağlar (archinstall onu /boot'a bağlar).
`# arch-chroot /mnt`	Kurulu sisteme geçer; /dev, /proc, /sys bağlarını kendisi yapar.
`# pacman -Syu && mkinitcpio -P`	chroot içinde yarım kalan güncellemeyi tamamlar ve initramfs'i yeniden üretir.
`# grub-mkconfig -o /boot/grub/grub.cfg`	GRUB kullanıyorsan yapılandırmayı yeniler. systemd-boot için `bootctl install`.
`# pacman -Qkk 2>/dev/null | grep -v ' 0 altered'`	Dosyaları bozulmuş paketleri bulur; bulunanları `pacman -S <paket>` ile yeniden kur.

Komutlar güncel Arch Linux (pacman 7, archinstall ile kurulmuş GNOME ya da KDE Plasma masaüstü) için yazıldı. Kayan sürüm olduğu için paket adları zamanla değişebilir; şüphede Arch Wiki esastır. sudo ile başlayan komutlar sistemi değiştirir, ne yaptığını okumadan çalıştırma.