

手动安装 archlinux

- [前期准备](#)
- [设置字体大小](#)
- [更新包管理](#)
- [分区管理](#)
 - [分区规划](#)
 - [系统盘](#)
 - [数据盘](#)
 - [对系统盘进行分区](#)
 - [对数据盘进行分区](#)
 - [挂载文件系统](#)
 - [一些维护命令](#)
- [安装操作系统](#)
- [生成分区表](#)
- [设置时区、编码](#)
- [设置主机名](#)
- [网络管理](#)
- [添加用户](#)
- [安装引导](#)
- [验证](#)
- [完成安装](#)

前期准备

- 一个刷写好archlinux live的U盘
- 一台机器

设置字体大小

```
setfont ter-132n # 正常 较大
or
setfont ter-122b # 粗体 较小
```

网络设置

Note

有线连接一般dhcp自动获取ip并连接

无线连接需使用iwctl手动联网

```
iwctl
# 以下为iwctl命令行
device list # 列出网卡列表
station <nic_name> scan # 扫描无线网络
station <nic_name> get-networks # 列出已扫描的无线网络
station <nic_name> connect <wlan_name> # 连接无线网
exit
# 以下为sh环境
ping www.baidu.com # 检测网络连通性
or
ip a # 查看网络连接信息
```

更新包管理

包管理镜像源列表位置在: `/etc/pacman.d/mirrorlist`

```
cp /etc/pacman.d/mirrorlist /etc/pacman.d/mirrorlist.bak # 备份镜像源列表
reflector --country China --age 24 --sort rate --protocol https --save
/etc/pacman.d/mirrorlist
# more /etc/pacman.d/mirrorlist # 检查镜像源列表
pacman -Syu --noconfirm
```

- `--country`: 设置镜像源地区
- `--age`: 最近{num}h
- `--sort`: 排序: 以传输速度排序
- `--protocol`: 镜像源所用传输协议 https为非明文传输 http为明文传输
- `--save`: 生成的镜像源列表路径

分区管理

```
lsblk -o NAME,UUID,MOUNTPOINT          # 查看分区表
lsblk -f /dev/sda{num}
gdisk /dev/sda                          # 进入gdisk模式为/dev/sda磁盘分区
gdisk /dev/sdb
```

Note

fdisk、gdisk 交互式分区管理工具，fdisk、gdisk 分别管理 mbr、gpt分区

parted 命令式分区管理工具

对应关系：gpt -> UEFI mbr -> BIOS

创建分区过程：

gdisk /dev/sda

o → y # 创建分区表

n → 1 → (默认起始) → +512M → EF00 # EFI分区

n → 2 → (默认起始) → +17G → 8E00 # LVM分区

n → 3 → (默认起始) → +1G → 8200 # Swap分区

n → 4 → (默认起始) → (默认结束) → 8300 # Btrfs快照分区

w → y

fdisk/gdisk常用参数：

Options:

- d delete a partition
- i show detailed information on a partition
- l list known partition types
- n add a new partition
- p print the partition table
- w write table to disk and exit
- ? print this menu

常用分区对应十六进制码：

8200 Linux swap

8300 Linux filesystem

8302 Linux /home

ef00 EFI

8e00 LVM

检查分区信息：

gdisk -l /dev/sda

```
root@archiso ~ # gdisk /dev/sda
GPT fdisk (gdisk) version 1.0.10

Partition table scan:
  MBR: not present
  BSD: not present
  APM: not present
  GPT: not present

Creating new GPT entries in memory.

Command (? for help): n
Partition number (1-128, default 1): 1
First sector (34-41943006, default = 2048) or {+-}size{KMGT}:
Last sector (2048-41943006, default = 41940991) or {+-}size{KMGT}: +512M
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300): ef00
Changed type of partition to 'EFI system partition'

Command (? for help): n
Partition number (2-128, default 2): 2
First sector (34-41943006, default = 1050624) or {+-}size{KMGT}:
Last sector (1050624-41943006, default = 41940991) or {+-}size{KMGT}: +18G
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300):
Changed type of partition to 'Linux filesystem'

Command (? for help): n
Partition number (3-128, default 3):
First sector (34-41943006, default = 38799360) or {+-}size{KMGT}:
Last sector (38799360-41943006, default = 41940991) or {+-}size{KMGT}:
Current type is 8300 (Linux filesystem)
Hex code or GUID (L to show codes, Enter = 8300): 8200
Changed type of partition to 'Linux swap'

Command (? for help): p
Disk /dev/sda: 41943040 sectors, 20.0 GiB
Model: VMware Virtual S
Sector size (logical/physical): 512/512 bytes
Disk identifier (GUID): 8CDA1C22-5DDB-404B-B16B-D93E73F61DE5
Partition table holds up to 128 entries
Main partition table begins at sector 2 and ends at sector 33
First usable sector is 34, last usable sector is 41943006
Partitions will be aligned on 2048-sector boundaries
Total free space is 4029 sectors (2.0 MiB)

Number  Start (sector)    End (sector)  Size      Code  Name
   1            2048             1050623   512.0 MiB   EF00   EFI system partition
   2          1050624          38799359   18.0 GiB    8300   Linux filesystem
   3          38799360          41940991    1.5 GiB    8200   Linux swap

Command (? for help): _
```

分区规划

使用GPT分区表

系统盘 (/dev/sda)

分区	大小	类型	文件系统	用途
/dev/sda1	512M	EFI系统分区	FAT32	/boot/efi
/dev/sda2	17G	Linux LVM	-	LVM物理卷（系统VG）

分区	大小	类型	文件系统	用途
/dev/sda3	1G	Linux swap	swap	交换空间（512M~1G）
/dev/sda4	剩余空间		btrfs	系统快照存储

数据盘 (/dev/sdb)

分区	大小	类型	文件系统	用途
/dev/sdb1	全部	Linux LVM	-	LVM物理卷（数据VG）

对系统盘进行分区

```
# ===== 系统盘：sda =====
mkfs.fat -F32 /dev/sda1 # 格式化EFI分区

pvcreate /dev/sda2 # 创建物理卷
vgcreate sysvg /dev/sda2 # 创建卷组（命名为sysvg）

# 创建逻辑卷
lvcreate -L 16G -n root sysvg # 根分区
lvcreate -L 100%FREE -n var sysvg # 日志分区

# 格式化逻辑分区
mkfs.btrfs -f -L "ROOT" /dev/sysvg/root # 格式化根分区
mkfs.btrfs -f -L "VAR" /dev/sysvg/var # 格式化可变数据分区

mount /dev/sysvg/root /mnt # 临时挂载分区
btrfs subvolume create /mnt/@ # 创建btrfs子卷结构
btrfs subvolume create /mnt/@var
btrfs subvolume set-default 256 /mnt # 设置默认子卷，256通常是第一个创建的子卷的ID
or
btrfs subvolume list /mnt
btrfs subvolume set-default <子卷ID> /mnt
umount /mnt # 卸载临时挂载

mkswap -f /dev/sda3 # 格式化swap分区
swapon /dev/sda3 # 启用交换分区
swapon --show # 查看交换分区大小
or
free -h

vgcreate snapvg /dev/sda4 # 创建卷组
lvcreate -l 100%FREE -n all_snapshots snapvg # 创建逻辑卷
mkfs.btrfs -f -L "SNAPSHOTS" /dev/sda4 # 格式化逻辑分区
mount /dev/snapvg/all_snapshots /mnt # 挂载临时分区
btrfs subvolume create /mnt/@all_snapshots # 创建子卷
umount /mnt # 卸载临时挂载
```

对数据盘进行分区

```
# ===== 数据盘: sdb =====
pvcreate /dev/sdb1 # 创建物理卷
vgcreate datavg /dev/sdb1 # 创建卷组 (命名为datavg)
lvcreate -L 13G -n home datavg # 用户数据 /home
lvcreate -L 5G -n etc datavg # 配置备份 /etc、/var
lvcreate -l 100%FREE -n otherdata datavg # 其他数据

# 格式化并配置/home
mkfs.btrfs -f -L "HOME" /dev/datavg/home
mount /dev/datavg/home /mnt
btrfs subvolume create /mnt/@home
umount /mnt

# 格式化并配置/etc备份
mkfs.btrfs -f -L "ETC" /dev/datavg/etc
mount /dev/datavg/etc /mnt
btrfs subvolume create /mnt/@configs
btrfs subvolume create /mnt/@ssh_keys
umount /mnt

# 格式化数据存储
mkfs.btrfs -f -L "OTHERDATE" /dev/datavg/otherdata
```

Tip

对于卷等操作:

```
lvdisplay /dev/<卷组>/<逻辑卷> # 查看逻辑卷
lvremove /dev/<卷组>/<逻辑卷> # 删除逻辑卷
lvextend -l +100%FREE /dev/<卷组>/<逻辑卷> # 扩容逻辑卷
vgremove <卷组> # 删除卷组 (前提删除卷组内所有逻辑卷)
btrfs subvolume list show # 列出btrfs所有子卷
btrfs subvolume delete /mnt/<子卷名> # 删除btrfs子卷 (挂载状态)
```

检查挂载点和子卷的关系:

```
findmnt -t btrfs -o TARGET,SOURCE,FSTYPE,OPTIONS
```

挂载文件系统

```
# ===== 挂载文件系统 =====
vgchange -ay # 激活所有卷组
lvscan # 检查逻辑卷状态
vgs && lvs && pvs # 检查LVM状态

mount UUID=a795255d-8ee7-4c15-9dd6-831364ff77ee /mnt -o subvol=@ # 首先挂载根分区
mkdir -p /mnt/{boot/efi,var,etc,home,snapshots,otherdata} # 创建挂载点目录结构
# 挂载分区
mount /dev/sda1 /mnt/boot/efi
mount UUID=14190dae-1e89-4636-8d1a-87e9a46e4b0d /mnt/var -o subvol=@var
```

```

mount UUID=27333bb2-7961-4406-881a-316d2dfe1718 /mnt/snapshots -o
subvol=@all_snapshots
mount UUID=c5418a7f-2f2b-492c-a8c3-464d60b6881b /mnt/etc -o subvol=@configs
mount UUID=9788b05b-6d48-4158-b527-3e1ce1f61b62 /mnt/home -o subvol=@home
mount UUID=02c8fb43-6a24-474d-8ecf-2997ff7b9d98 /mnt/otherdata

systemctl daemon-reload

# 检查所有挂载点
mount | grep btrfs

# btrfs子卷检查
btrfs subvolume list /mnt/<子卷名>
# eg: btrfs subvolume list /mnt/home

```

一些维护命令

```

# ===== 维护命令 =====
sudo btrfs subvolume list / # 检查Btrfs子卷
sudo btrfs filesystem usage / # 查看空间使用
sudo btrfs filesystem defrag -r /mnt/data # 碎片整理

sudo rsync -av --delete /etc/ /mnt/etc/ # 定期备份/etc
sudo btrfs subvolume snapshot / /.snapshots/${date +%Y%m%d} # 创建系统快照
sudo btrfs subvolume snapshot /home /mnt/snapshots/home_${date +%Y%m%d} # 创建数据快照

sudo chattr +c / # 禁用COW对系统目录
sudo chattr +C /var # 禁用COW对日志目录
sudo chattr +C /var/lib/mysql # 禁用COW对数据库
sudo chattr +C /var/lib/docker # 禁用COW对容器

# 检查文件系统
sudo btrfs scrub start /mnt/data

# 平衡文件系统
sudo btrfs balance start -dusage=50 /mnt/data

# 碎片整理
sudo btrfs filesystem defrag -r /mnt/data

# 启用压缩
btrfs property set /mnt compression zstd
btrfs property set /mnt/home compression zstd:3

# btrfs透明压缩
sudo btrfs filesystem defrag -czstd /mnt/data

# 系统盘用高压缩比（zstd:3），数据盘用平衡模式（zstd:1）
sudo btrfs property set /mnt/data compression zstd:1

# 创建系统快照
sudo btrfs subvolume snapshot -r / /.snapshots/${date +%Y%m%d}

```

```

# 自动清理旧快照
sudo find /.snapshots -maxdepth 1 -mtime +7 -exec btrfs subvolume delete {} \;

# 系统崩溃时:
# * 从Live USB启动
# * 挂载系统盘并回滚快照:
mount /dev/sda4 /mnt/snaps
btrfs subvolume snapshot /mnt/snaps/@system_snaps/20230801 /mnt/@restore
btrfs subvolume set-default 256 /mnt/@restore

# 数据误删恢复:
btrfs send /mnt/snaps/@user_home_backup | btrfs receive /home/new_restore

# 对根分区启用zstd压缩
sudo btrfs property set / compression zstd:3

# 对已有文件压缩
btrfs filesystem defragment -r -czstd /
systemctl enable --now btrfs-maintenance.timer

# 查看压缩率
sudo compsize -x /

# btrfs自动维护
cat > /etc/systemd/system/btrfs-maintenance.service <<EOF
[Unit]
Description=Btrfs Monthly Maintenance

[Service]
Type=oneshot
ExecStart=/usr/bin/btrfs balance start -dusage=50 /
ExecStart=/usr/bin/btrfs scrub start /
EOF

```

我的分区情况如下:

```

root@archiso ~ # lsblk -f

```

NAME	FSTYPE	FSVER	LABEL	UUID
loop0	squashfs	4.0		
sda		0	100%	/run/archiso/airootfs
└─sda1	vfat	FAT32		3F1D-C638
		14.5G	0%	/mnt/boot/efi
└─sda2	LVM2_member	LVM2 001		J3Ep2y-
0qEr-x7Mj-kwdH-vxtz-ftYb-kxQXkf				
└─sysvg-root	btrfs		ROOT	a795255d-
8ee7-4c15-9dd6-831364ff77ee		14.5G	0%	/mnt
└─sysvg-var	btrfs		VAR	14190dae-
1e89-4636-8d1a-87e9a46e4b0d		904.6M	1%	/mnt/var
└─sda3	swap	1		355c95c7-
dfc8-45de-9446-03598c7a677c			[SWAP]	


```

└─sda4                                LVM2_member LVM2 001                                2914C6-
DBtN-DSP3-cyTH-cDm3-AISB-KtWtDo
└─snapvg-all_snapshots btrfs                                SNAPSHOTS 27333bb2-
7961-4406-881a-316d2dfe1718        1.3G        0% /mnt/snapshots
sdb

└─sdb1                                LVM2_member LVM2 001                                LVkkOb-
cEnr-ejsa-HJ6E-mDns-o2RE-2pGudm
├─datavg-home                btrfs                                HOME      9788b05b-
6d48-4158-b527-3e1ce1f61b62        12.5G        0% /mnt/home
├─datavg-etc                 btrfs                                ETC        c5418a7f-
2f2b-492c-a8c3-464d60b6881b        4.5G        0% /mnt/etc
└─datavg-otherdata          btrfs                                OTHERDATE 02c8fb43-
6a24-474d-8ecf-2997ff7b9d98        1.8G        0% /mnt/otherdata
sr0                            iso9660          Joliet Extension ARCH_202506 2025-06-
01-09-10-39-00                  0 100% /run/archiso/bootmnt
root@archiso ~ # mount | grep btrfs
/dev/mapper/sysvg-root on /mnt type btrfs
(rw,relatime,space_cache=v2,subvolid=256,subvol=/@)
/dev/mapper/sysvg-var on /mnt/var type btrfs
(rw,relatime,space_cache=v2,subvolid=256,subvol=/@var)
/dev/mapper/snapvg-all_snapshots on /mnt/snapshots type btrfs
(rw,relatime,space_cache=v2,subvolid=256,subvol=/@all_snapshots)
/dev/mapper/datavg-etc on /mnt/etc type btrfs
(rw,relatime,space_cache=v2,subvolid=256,subvol=/@configs)
/dev/mapper/datavg-home on /mnt/home type btrfs
(rw,relatime,space_cache=v2,subvolid=256,subvol=/@home)
/dev/mapper/datavg-otherdata on /mnt/otherdata type btrfs
(rw,relatime,space_cache=v2,subvolid=5,subvol=/)

```

安装操作系统

```

pacstrap /mnt linux linux-firmware linux-headers base base-devel vim bash-
completion
arch-chroot /mnt
pacman -Syu --noconfirm
pacman -S --noconfirm btrfs-progs lvm2 sudo nano dhcpcd openssh snapper
btrfsmaintenance

```

- `btrfs-progs`: Btrfs 文件系统工具
- `lvm2`: LVM 管理工具
- `nano`: 文本编辑器
- `sudo`: 权限管理
- `dhcpcd`: 网络连接

Note

`arch-chroot /mnt` 将当前的根目录切换到 `/mnt`，即进入挂载的目标文件系统。

`pacstrap` 这是一个 Arch Linux 安装工具，用于将指定的软件包安装到目标挂载点（这里是 `/mnt`）。`pacstrap` 是 `pacman` 的一个封装工具，专门用于安装初始的系统软件包。

区别: `arch-chroot /mnt` 可以执行更为复杂的配置, 而 `pacstrap` 则仅向目标系统中安装软件包

⚠ Warning

全新安装时应首先使用 `pacstrap` 安装linux内核和工具包, 否则 `chroot` 直接进入会报错, 未安装系统

生成分区表

live环境:

```
genfstab -U /mnt
genfstab -U /mnt >> /mnt/etc/fstab
```

- `-U`: UEFI

设置时区、编码

```
arch-chroot /mnt # 进入新系统
ln -sf /usr/share/zoneinfo/Asia/Shanghai /etc/localtime # 设置时区为shanghai
hwclock --systohc # 同步硬件时间

or

timedatectl set-timezone Asia/Shanghai # 设置时区
timedatectl set-ntp true # 同步系统时间
timedatectl status # 验证

cp /etc/locale.gen /etc/locale.gen.bak
# 取消注释指定编码
sed -i 's/^#(en_US.UTF-8 UTF-8)/\1/' /etc/locale.gen
sed -i 's/^#(zh_CN.UTF-8 UTF-8)/\1/' /etc/locale.gen

sed -i -E 's/^#(en_US|zh_CN\..*)/\1/' /etc/locale.gen # 批量取消注释
grep -E '^(en_US|zh_CN)' /etc/locale.gen # 验证修改结果
locale-gen # 应用修改
echo "LANG=en_US.UTF-8" > /etc/locale.conf # 设置系统默认语言
```

以下操作非特殊说明环境, 均为 `chroot` 进入新系统的环境

设置主机名

```
echo "archlinux" > /etc/hostname
# 配置hosts文件
cat >> /etc/hosts <<EOF
127.0.0.1 localhost
::1 localhost
127.0.1.1 archlinux.localdomain archlinux
EOF
```

网络管理

```
# 无线网络使用networkmanager
pacman -S --noconfirm networkmanager
systemctl enable NetworkManager # 设置开机自启, 注意大小写

# 有线网络使用dhcp
systemctl enable dhcpcd.service
```

该处操作为**安装好系统重启后**的操作:

```
ip a          # 查看网卡名称
or
nmcli con show # 查看网卡名称
nmcli con mod ipv4.method manual \
            ipv4.address <ip_addr>/<netmask> \
            ipv4.gateway <gateway> \
            ipv4.dns "114.114.114.114 8.8.8.8"
```

- `ipv4.method`: 网络连接方式
- `ipv4.address`: 设置ip地址
- `ipv4.gateway`: 设置网关
- `ipv4.dns`: 设置域名解析服务器, 引号内可以设置多个用空格隔开

或者可以使用 `nmtui` 交互式操作管理网络

```
ip -o -4 route show default | awk '{print $5}' | head -n1  获取第一个网卡的名称
```

添加用户

首先设置root用户密码:

```
[root@archiso /]# passwd
[root@archiso /]# useradd -m -G wheel -s /bin/bash <username>
[root@archiso /]# passwd <username>
[root@archiso /]# sed -i '/^# %wheel ALL=(ALL:ALL) ALL/s/^# //' /etc/sudoers
[root@archiso /]# su - <username>
[<username>@archiso ~]$ sudo -v
```

💡 Tip

如果忘记root密码, 可以再次进入live环境, 使用 `arch-chroot` 进入系统执行 `passwd` 命令

以上步骤为: 修改root密码 -> 添加普通用户 -> 修改普通用户密码 -> 将普通用户添加到sudoers -> 切换到普通用户 -> 验证sudo权限

安装引导

```
pacman -S --noconfirm grub efibootmgr efivar adm-ucode/intel-ucode # intel芯片安
装intel-ucode, amd芯片安装amd-ucode
grubinstall /dev/sda # 安装grub
cp /etc/default/grub /etc/default/grub.bak # 重要操作前记得备份
sudo sed -i 's/GRUB_CMDLINE_LINUX_DEFAULT="loglevel=3
quiet"/GRUB_CMDLINE_LINUX_DEFAULT="loglevel=3"/g' /etc/default/grub # 启动时生成日
志
sed -i 's/GRUB_TIMEOUT=5/GRUB_TIMEOUT=2/' /etc/default/grub #
可选: 缩短grub等待时间
sudo sed -i 's/GRUB_GFXMODE=auto/GRUB_GFXMODE=1920x1080/g' /etc/default/grub #
可选: 修改grub分辨率为1920x1080
grub-mkconfig -o /boot/grub/grub.cfg # 生成grub配置
```

验证

```
mount | grep btrfs
btrfs filesystem show
```

完成安装

退出 `chroot` 卸载全部分区后重启, 在关机后拔掉u盘即可

```
root@archiso ~ # umount -R /mnt
root@archiso ~ # reboot
```

杂项设置

文件 `~/.bashrc`

```
cat << 'EOF' | tee -a ~/.bashrc
[ ! -e ~/.dircolors ] && eval "$(dircolors -p > ~/.dircolors)
[ -e /bin/dircolors ] && eval "$(dircolors -b ~/.dircolors)
EOF
source ~/.bashrc
```

快照工具

```
sudo pacman -S --noconfirm snapper
snapper -c root create-config /
sudo systemctl enable snapper-timeline.timer snapper-cleanup.timer
```