

麒麟信安服务器操作系统

V6 SP1

软件用户手册

文档编号：KYJS-KS-Server-6-SP1-SUM-V1.1



变更记录

版本	修订时间	修订人	修订类型	修订章节	修订内容
V1.0	2026.04.27	徐铖	A	ALL	生成全文
V1.1	2026.5.21	余永康	M	封面、页眉、4.2.2-4.2.4、5、5.1、6、7、8.3.2、8.4	封面、页眉更换新logo; 4.2.2-4.2.4、5、5.1、6、7、8.3.2、8.4章节更换图片

注 1：修订类型分为 A-ADDED，M-MODIFIED，D-DELETED

注 2：对该文件内容增加、删除或修改均需填写此记录，详细记载变更信息，以保证其可追溯性

目 录

1 范围	1
1.1 标识	1
1.2 系统概述	1
1.3 文档概述	1
2 综述	3
2.1 软件应用	3
2.2 硬件环境	4
2.3 系统框架概述	4
2.4 意外事故及运行的备用状态和方式	5
2.5 帮助和问题报告	5
3 系统入门	7
3.1 能力	7
3.2 约定	7
3.3 软件的首次使用	8
3.3.1 熟悉设备	8
3.3.2 访问控制	8
3.3.3 安装和设置	8
3.4 启动	9

3.5 停止	10
3.6 系统升级	10
4 安装	12
4.1 准备工作	12
4.2 安装过程	12
4.2.1 安装启动模式	12
4.2.2 语言选择	14
4.2.3 安装配置	15
4.2.4 安装完成	29
4.2.5 系统登录	30
4.2.6 版本信息	34
5 最小安装	35
5.1 安装完成	35
5.2 多终端 (tty) 切换	38
5.3 系统激活	38
5.4 查看系统信息	39
5.4.1 查看版本信息	39
5.4.2 查看 CPU 信息	39
5.4.3 查看内存信息	39
5.4.4 查看磁盘信息	39
5.4.5 设置网络	40

6 服务器	43
6.1 docker 服务	44
6.2 Cockpit 服务	45
6.3 一次性任务调度服务	45
6.4 存储管理库	46
6.5 授权框架	47
6.6 RPC 端口映射	47
7 虚拟化主机	49
7.1 虚拟机镜像上传	49
7.2 虚拟机创建	50
7.3 vnc 控制虚拟机	51
7.4 虚拟机管理	51
7.5 虚拟机网络管理	52
7.6 虚拟机其他配置	52
8 图形化服务器	53
8.1 图形化桌面	53
8.1.1 桌面导视图	54
8.1.2 电源管理	57
8.1.3 消息	58
8.2 控制面板	59
8.2.1 关于系统	60

8.2.2 声音	66
8.2.3 电源	67
8.2.4 设备	67
8.2.5 登录设置	68
8.2.6 账户	69
8.2.7 用户组	72
8.2.8 日期与时间	76
8.2.9 网络	78
8.2.10 显示	81
8.2.11 个性化	82
8.2.12 应用程序	85
8.2.13 认证管理	86
8.3 桌面应用	90
8.3.1 压缩与解压	90
8.3.2 Atril 文档查看器	91
8.3.3 MATE 之眼图像查看器	92
8.3.4 计算器	93
8.3.5 PulseAudio 音量控制	94
8.3.6 终端	95
8.3.7 磁盘	96
8.3.8 系统日志查看器	97

8.3.9 密码和密钥	98
8.3.10 文本编辑器	99
8.3.11 键盘布局测试器	100
8.3.12 Firefox 浏览器	100
8.3.13 Fcitx 5 配置	101
8.3.14 截图工具	102
8.3.15 防火墙	104
8.3.16 Dconf 编辑器	104
8.3.17 打印设置	105
8.3.18 软件用户手册、软件管理员手册	106
8.3.19 激活信息	107
8.4 安全中心	109
8.4.1 软件概述	109
8.4.2 软件综述	109
8.4.3 软件功能	110
8.4.4 FAQ	138
8.5 远程控制	138
8.5.1 VNC 配置	138
9 备份还原	142
9.1 图形方式	142
9.1.1 创建	142

9.1.2 恢复	143
9.1.3 删除	143
9.1.4 浏览	144
9.1.5 设定	145
9.2 命令方式	150
9.2.1 创建快照	150
9.2.2 还原快照	151
9.2.3 删除快照	152
10 系统急救模式	153
11 bond3+功能	154
11.1 配置 bond	155
11.2 bond3+功能	155
12 运维工具集	156
12.1 工具概述	156
12.2 软件安装	156
12.3 CPU 负载观测与分析功能	157
12.3.1 作用	157
12.3.2 输入	157
12.3.3 输出	158
12.4 文件缓存观测与分析功能	160
12.4.1 作用	160

12.4.2 输入	160
12.4.3 输出	161
12.4.4 /etc/opsinsight.conf 配置文件	163
12.5 内存观测与分析功能	164
12.5.1 作用	164
12.5.2 输入	164
12.5.3 输出	165
12.5.4 /etc/opsinsight.conf 配置文件	166
12.6 io 观测与分析功能	166
12.6.1 作用	166
12.6.2 输入	166
12.6.3 输出	168
12.6.4 /etc/opsinsight.conf 配置文件	168
13 软件源	170
13.1 网络源配置	170
13.2 镜像源配置	170
13.3 安装软件包	174
14 免责声明	174
附录	175

1 范围

1.1 标识

文档标识号：KYJS-KS-Server-6-SP1-SUM-V1.1

标题：麒麟信安服务器操作系统 V6 SP1 软件用户手册

软件名称：麒麟信安服务器操作系统

软件标识：KS-Server-6-SP1

软件版本号：6 SP1

本文档适用的系统和计算机软件配置项 CSCI：麒麟信安服务器操作系统

1.2 系统概述

以用户和市场需求为导向，基于麒麟信安服务器操作系统 V6 版本构建，定位于国防、政府、电力、金融、能源、工业等信息系统建设，兼容主流 CPU、GPU、整机、存储、数据库、中间件等硬件以及主流应用软件，以安全、稳定、高效为突破点，满足重要行业领域用户高安全、高可靠的规模化部署需求。致力于服务器操作系统在系统安全性、易用性、稳定性及兼容性等方面进行更新迭代，同时继承上游社区 openEuler 的良好系统生态、继承 openEuler 社区优秀特性、继承历史版本增加特性，进一步打造产品护城河、提高产品竞争力。

1.3 文档概述

本文档是麒麟信安服务器操作系统软件用户手册，它包含以下内容：

- 1) 麒麟信安服务器操作系统概述；
- 2) 麒麟信安服务器操作系统的入门；

3) 麒麟信安服务器操作系统使用指南

2 综述

2.1 软件应用

硬件设备和操作系统之上的软件应用是程序、数据结构和相关文档的集合，是指用于实现所需要功能的逻辑方法、过程或控制。麒麟信安服务器操作系统中的软件应用包括麒麟信安服务器操作系统自带的典型应用、麒麟信安服务器操作系统支持的典型应用程序和麒麟信安服务器操作系统的国产软件支持能力。

1) 网络应用

网络应用支持：浏览器、文件下载、FTP 客户端、防火墙、远程管理、远程登录、文件共享等应用。

2) 功能服务器应用

功能服务器应用支持：Web 服务器、FTP 服务器、邮件服务器、代理服务器、域名服务器等应用。

3) 数据库服务器应用

数据库服务器应用支持：商业数据库、开源数据库等应用。

4) 文件应用

文件应用支持：文件管理、文件系统、压缩工具、文本编辑、文件加密等应用。

5) 办公软件应用

办公软件应用支持：商业办公软件、开源办公软件等应用。

6) 多媒体应用

多媒体应用支持：CD 烧录、视频播放器、图形查看、图形处理等应用。

2.2 硬件环境

- 支持 x86_64/AMD64 架构机器，包括海光 3 号、海光 4 号、兆芯 KH-40000、兆芯 KH-50000 及常见的 Intel、AMD 处理器。
- 支持 aarch64 架构机器，包括鲲鹏 920、鲲鹏 920 v200、飞腾 2000+、飞腾 S2500、飞腾 S5000C 等。
- 支持 loongarch64 架构机器，包括龙芯 3C5000、龙芯 3C6000 等。
- 支持 sw_64 机器，包括申威 8A。

2.3 系统框架概述

麒麟信安服务器操作系统 V6 SP1 基础系统从下至上分为硬件层、内核层、系统层和应用层。硬件架构上，麒麟信安服务器操作系统 V6 SP1 同源支持 x86_64、aarch64、loongarch64、sw_64 架构；在内核层，麒麟信安服务器操作系统 V6 SP1 主要包含设备管理、文件系统、网络子系统、进程管理，在安全子系统上进一步开发，支持保密套件底座、内核模块保护、文件保护、可信度量等功能；在系统层包括系统服务管理 systemd、系统运行基础库（如 glibc、openssl）和图形开发库（如 gtk+、qt），在此之上，提供安全策略配置和管理、服务管理、软件包管理、日志管理等基本服务。基础软件提供了时间管理工具、shell 终端、运维工具、压缩工具、远程工具等。

系统架构图如图 2-1 所示：

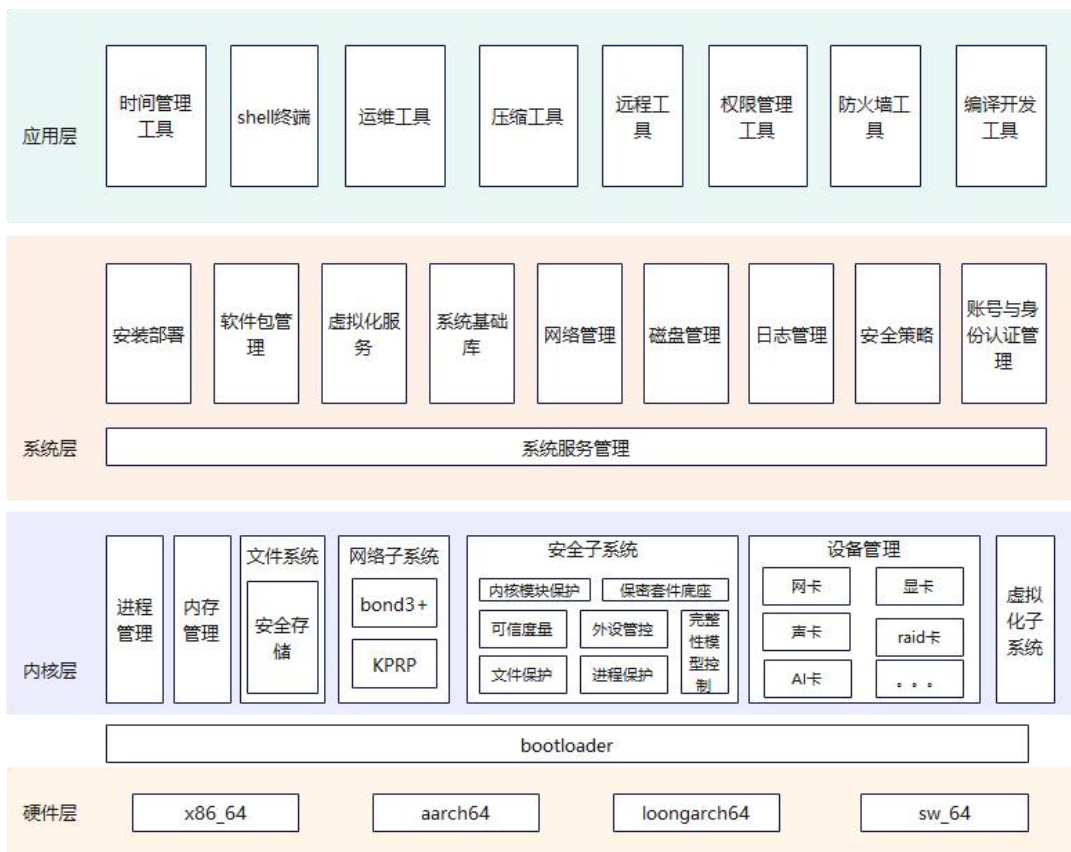


图 2-1 麒麟信安服务器操作系统 V6 SP1 系统架构

2.4 意外事故及运行的备用状态和方式

紧急时刻（如非正常关机）后可使用急救模式进行修复，详细内容参见 10 章节；如需使用备份还原，详细内容参见 9 章节。

2.5 帮助和问题报告

如果您有任何疑问，都可以通过以下方式获得帮助：

- 1) 查阅常见问题解答文档：见附录；
- 2) 求助在线帮助：<https://www.kylinsec.com.cn> ；
- 3) 热线：400-012-6606；
- 4) 扫描下方二维码，反馈相关问题；



3 系统入门

3.1 能力

麒麟信安服务器操作系统围绕用户在部署、运行、管理及跨环境适配等全生命周期需求，构建了一体化的核心能力体系。在基础支撑层面，系统提供多形态安装部署能力，支持 U 盘启动、多架构适配与灵活的配置选项，可满足从单机到集群、物理机到虚拟化的多样化部署场景；同时具备完善的升级与迁移能力，支持跨版本平滑升级、多方式数据迁移，保障业务连续性。接口与数据能力方面，系统外部接口标准化、规范化，可与第三方平台、运维系统及监控体系无缝对接；内部接口通过统一协议保障模块间高效协同，搭配完善的数据存储、备份与一致性校验机制，实现配置、运行、日志及审计数据的安全管理。

在安全与质量保障层面，系统构建了全维度防护体系，既满足用户许可协议与法务合规要求，也支持敏感数据加密、权限管控、操作审计、国密算法及安全中心集成，从系统层、访问层到数据层筑牢安全防线。同时，支持故障自愈与快速恢复，可有效应对硬件故障、网络异常等突发场景，为用户提供稳定、可信、可扩展的操作系统运行支撑。

3.2 约定

1) 单击（点击、左击）

在屏幕选定的区域内快速按下鼠标左键。

2) 双击

在屏幕选定的区域内快速连续按两下鼠标左键。

- 3) “按钮 A” > “按钮 B”

表示操作顺序是先操作按钮 A，再操作按钮 B。

3.3 软件的首次使用

3.3.1 熟悉设备

在熟悉设备阶段，需要了解麒麟信安服务器操作系统安装的目标硬件设备是否满足以下要求：

- 1) 设备具备正常工作的能力；
- 2) 与麒麟信安服务器操作系统所支持的硬件设备兼容；
- 3) 满足麒麟信安服务器操作系统允许的硬件环境要求；

3.3.2 访问控制

麒麟信安服务器操作系统具备访问控制功能，登录系统需要用户口令。在安装时用户自定义 root 密码，也可新增普通用户和设置密码，方法参见 4.2.3.4 章节。系统管理员可设置密码、更改密码和删除用户，具体方法参见 8.2.6 章节。

注意：在对系统进行远程访问控制时，不建议使用 root 用户。

3.3.3 安装和设置

麒麟信安服务器操作系统的安装流程图如下图 3-1 所示：

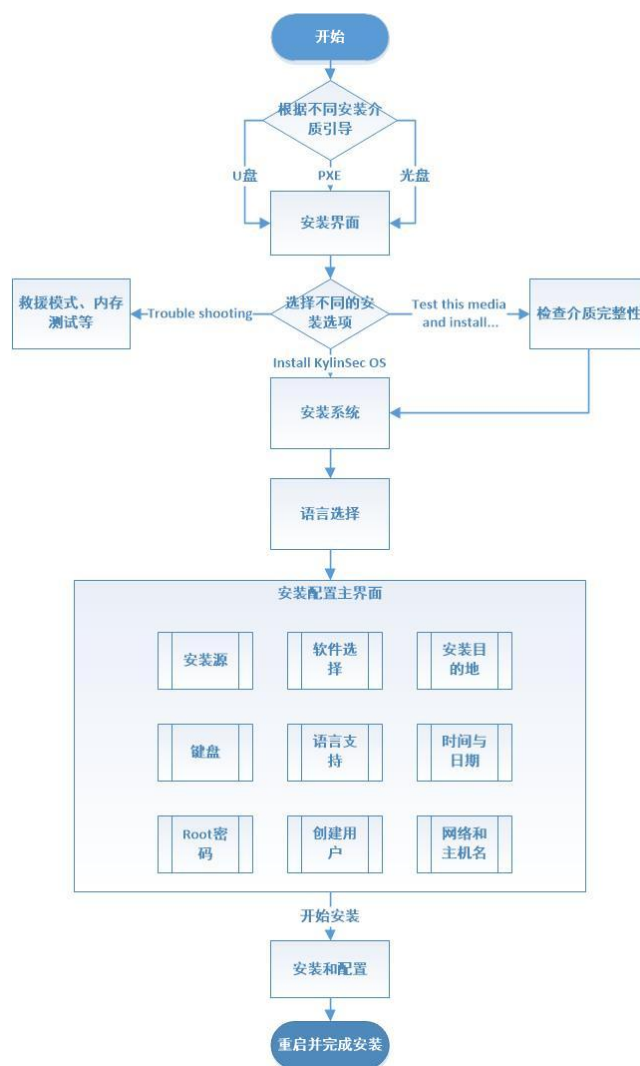


图 3-1 安装过程流程图

详细的安装相关步骤参见本手册 4 章节。

3.4 启动

要启动麒麟信安服务器操作系统, 首先确保计算机硬件状态良好且正确接通了电源。通过按电源按钮给计算机上电, 计算机完成硬件自检后启动引导程序, 引导程序界面如下图 3-2 所示:

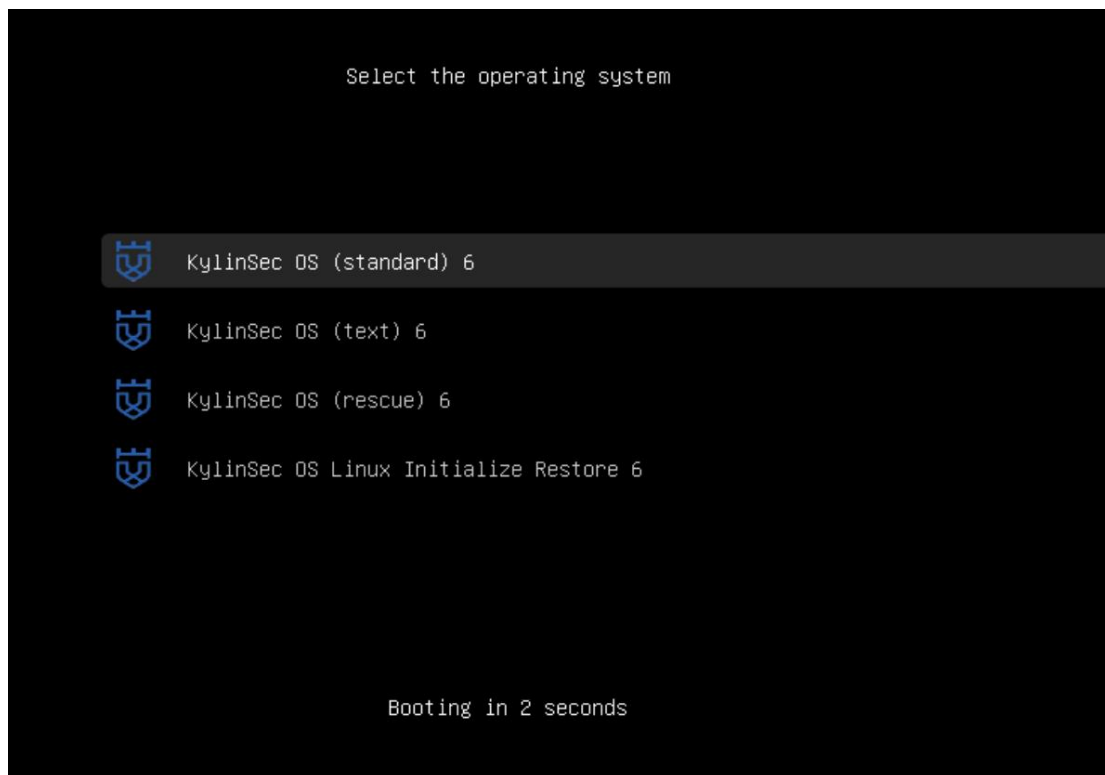


图 3-2 引导程序界面

引导程序经过 5 秒钟的倒计时之后，即从默认的引导选项启动操作系统。如计算机上安装了多个操作系统，可通过键盘上下键切换到其他系统或者引导选项。接下来系统就成功启动，进入到登录界面。

3.5 停止

在系统桌面点击“开始菜单”>“电源选项”>“关机”可停止系统，也可在终端输入 `poweroff` 或 `shutdown` 停止系统。

当硬件故障或者系统掉电也会导致麒麟信安服务器操作系统异常停止工作。

3.6 系统升级

麒麟信安服务器操作系统支持系统升级操作，当存在可更新的软件包时，在系统桌面点击“开始菜单”>“关于系统”>“系统升级”；可以对系统进行升级操作，如图 3-3 所示：

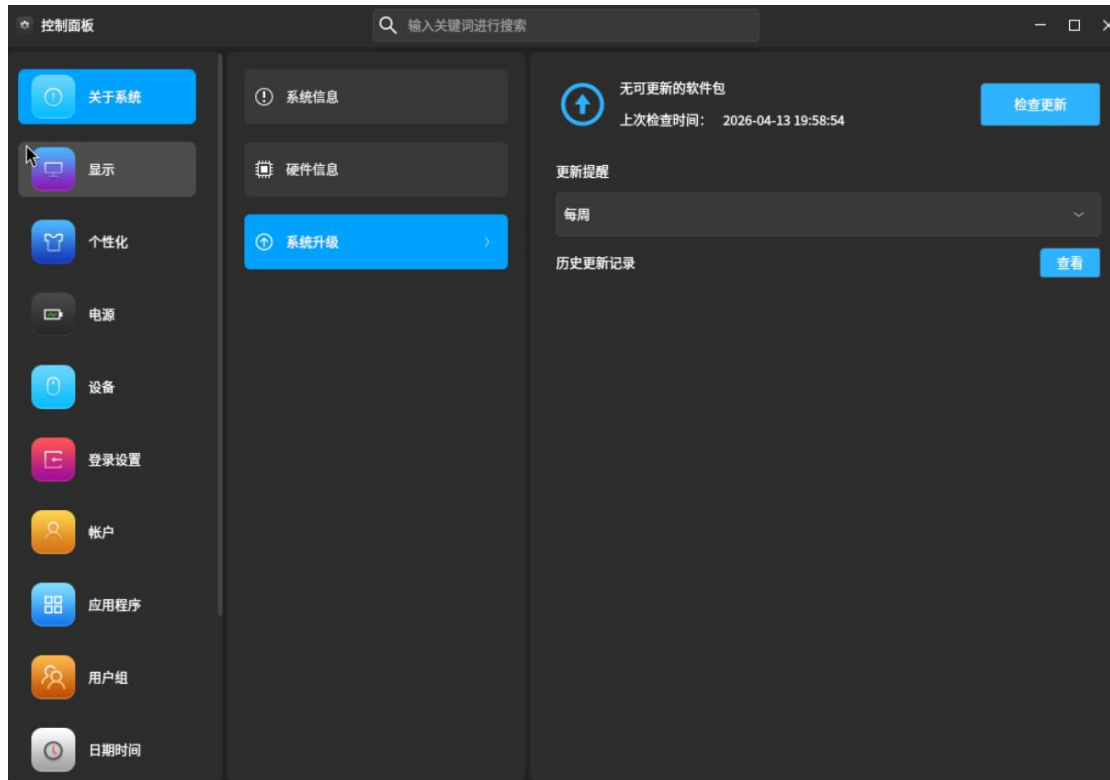


图 3-3 升级界面

除图形化方式外，系统升级也可通过命令行操作完成。在正确配置好升级源后，执行 `dnf update` 命令即可升级系统，在系统升级时会给出升级的具体内容，包括升级的软件或组件名称，如图 3-4 所示：

```
[root@localhost ~]# dnf update firefox
Last metadata expiration check: 0:00:14 ago on 2025年01月15日 星期三 15时54分21秒.
Dependencies resolved.
=====
Package                                Architecture      Version           Repository        Size
=====
Upgrading:
firefox                                x86_64            128.5.0-1.0e2403 update           122 M
Transaction Summary
-----
Upgrade 1 Package
Total download size: 122 M
Is this ok [y/N]: y
Downloading Packages:
firefox-128.5.0-1.0e2403.x86_64.rpm                                61 MB/s | 122 MB  00:01
Total
-----
61 MB/s | 122 MB  00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
执行兼容性检查
=====
软件包      助理      级别      兼容性      兼容      回退
=====
firefox      openEuler-SA-2024-2523      高      1.0      是      是
firefox      openEuler-SA-2024-2342      高      1.0      是      是
firefox      openEuler-SA-2024-2459      低      1.0      是      是
firefox      openEuler-SA-2024-2401      低      1.0      是      是
=====
兼容性检查通过。请参考以下详情文件: /var/lib/dnf/abi-check/compatibility-details-20250115-155451.csv
兼容性检查完成。
```

图 3-4 命令升级

4 安装

首先选择安装介质（光盘、U 盘、PXE）启动，到安装界面选择安装系统，然后选择系统语言，接下来进行安装配置，配置完成后开始安装。

4.1 准备工作

安装麒麟信安服务器操作系统之前，需要准备组件、检查硬件兼容性、备份数据、查看磁盘空间和进行磁盘分区等。推荐配置及其他推荐如表 4-1：

表 4-1 安装前准备表

内容	参数
安装系统的设备	服务器
磁盘空间	建议 60G 以上
CPU	各个架构基准 CPU 即可
内存	建议 2G 以上
介质	光盘、U 盘、网络 PXE
操作系统镜像	ISO 操作系统镜像文件

安装前需要确定好安装的服务器，建议磁盘剩余空间大于 60G。安装介质光盘和 U 盘需要使用刻录工具刻录，或者搭建 PXE 服务器后通过网络安装。

注：具体安装盘刻录方法可咨询相关工作人员。

4.2 安装过程

4.2.1 安装启动模式

安装方式有 Legacy 和 UEFI 两种，其中 x86_64 架构支持 Legacy、UEFI

两种方式，aarch64、loongarch64 和 sw_64 架构机器只支持 UEFI 方式。

开始安装之前需设置计算机启动方式，通过按键进入 BIOS（不同品牌计算机按键不同，可上网查询或在计算机开机后根据界面提示选择），根据安装介质，设置安装启动顺序为 U 盘、光盘或网络优先，可参考图 4-1：

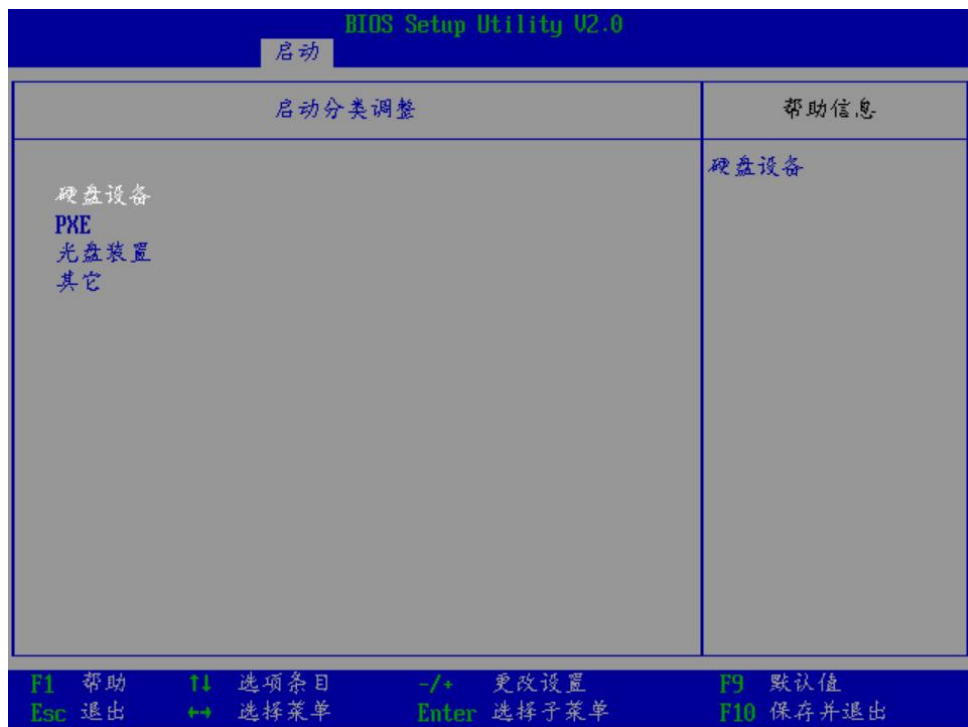


图 4-1 BIOS 安装顺序设置

设置完成后重启计算机，默认会从设置的介质启动。开始进入安装界面，Legacy 和 UEFI 两种启动方式的安装启动界面分别如图 4-2 与图 4-3 所示：

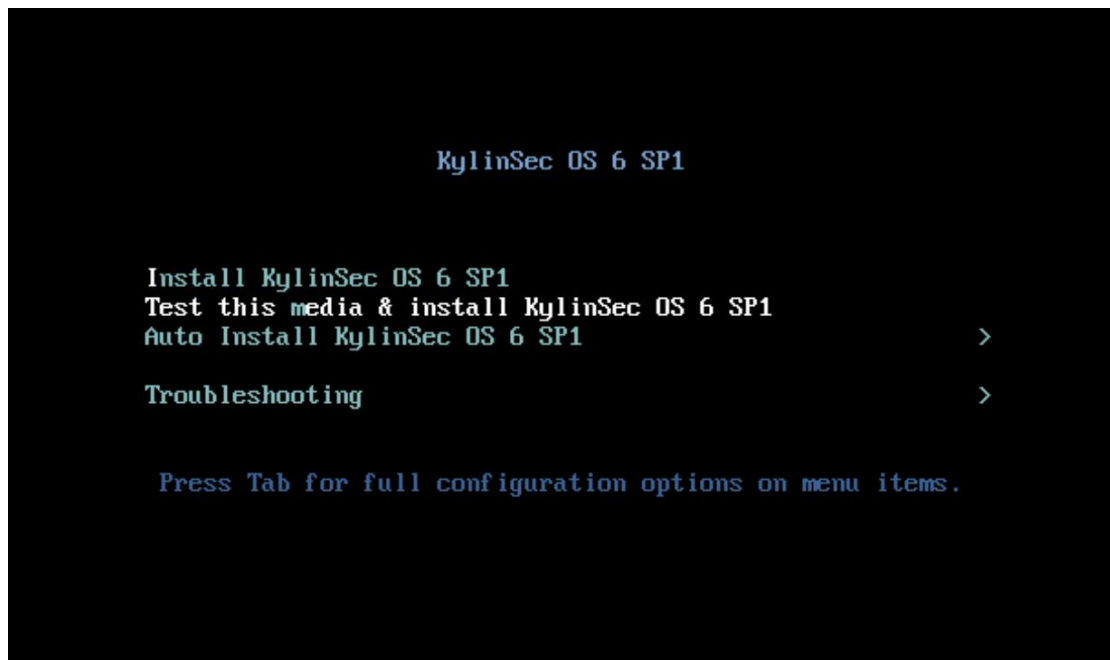


图 4-2 安装启动界面 Legacy

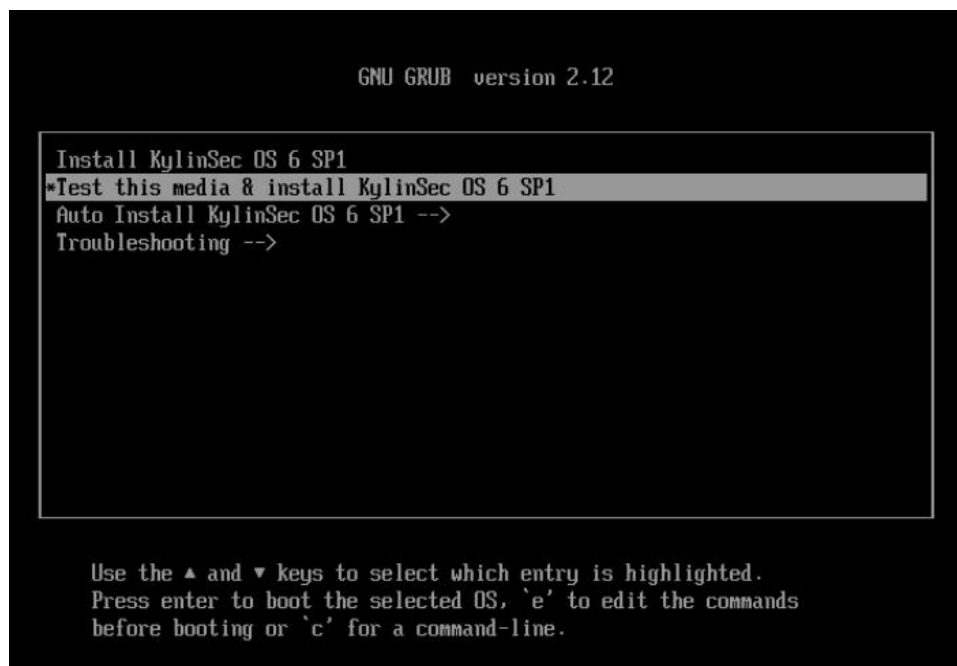


图 4-3 安装启动界面 UEFI

选择 Install KylinSec OS 6 SP1 开始进行安装。

4.2.2 语言选择

开始安装后首先进入语言选择界面，可选择简体中文或英文，选择完成后点

击“继续”按钮进入下一步，如图 4-4 所示：

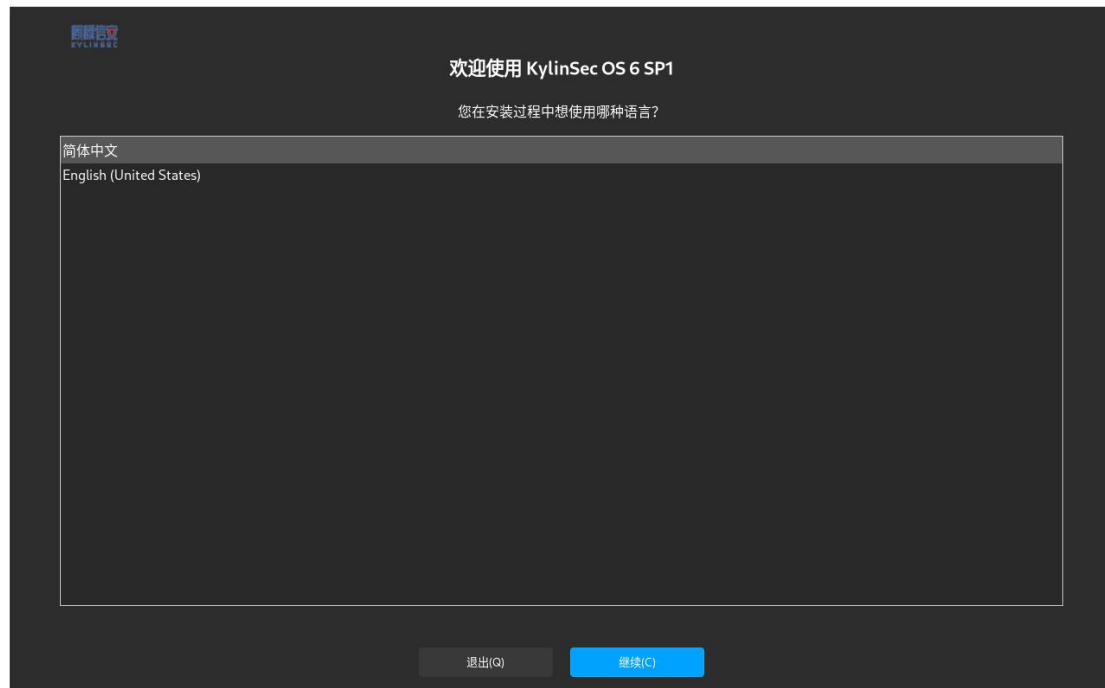


图 4-4 语言选择

4.2.3 安装配置

接下来到系统安装的配置界面，包括本地化、软件、系统和用户设置四个部分。带黄色警告图标的选项是必须设置选项，如安装目的地，没有黄色警告图标选项时方可进入下一步。如图 4-5 所示：



图 4-5 安装配置界面

4.2.3.1 本地化设置

本地化设置包括键盘、语言支持、时间和日期设置，一般情况下建议使用默认设置，也可以点击手动修改。

键盘布局默认是汉语，左侧选择框可添加和移动语言的布局，如图 4-6：

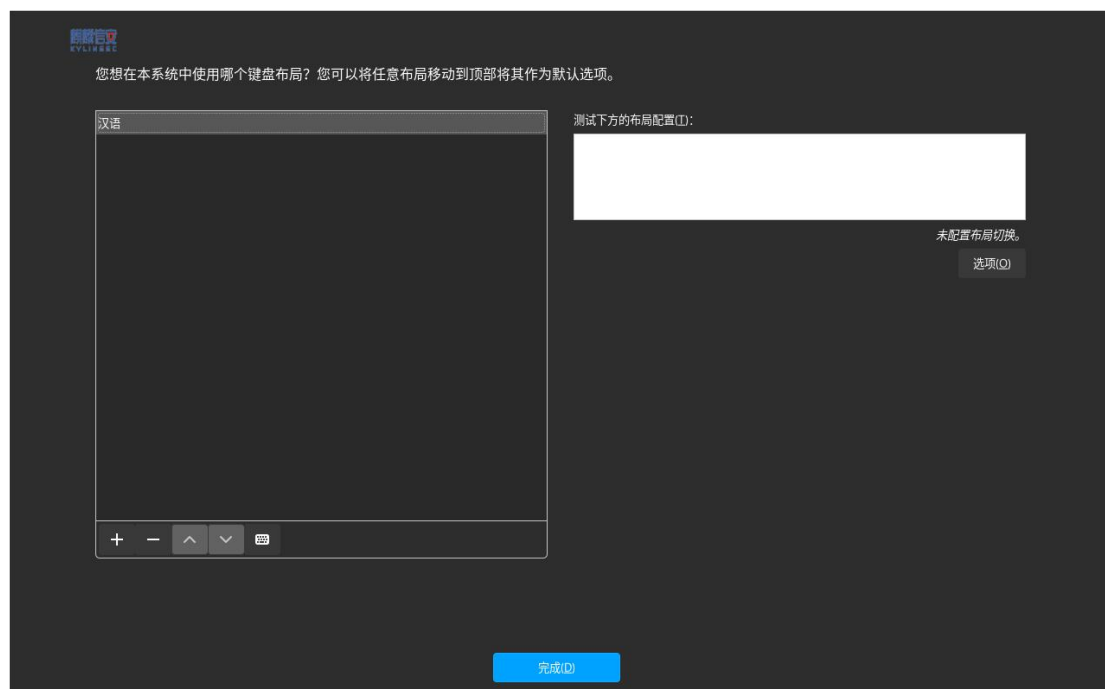


图 4-6 键盘布局设置

默认的语言支持类型为简体中文，可以修改成英文，如图 4-7：

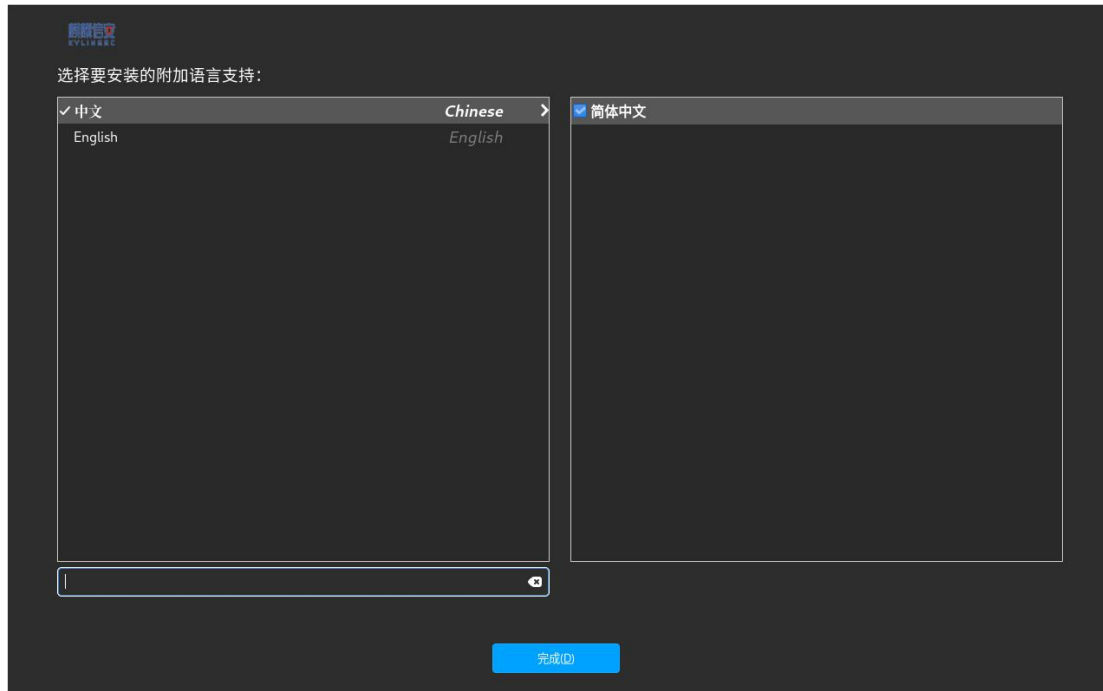


图 4-7 语言支持设置

系统提供了对各国时区的支持。对于大多数国内用户来说，保持默认的亚洲/上海即可，也可以手动修改时区，如图 4-8：

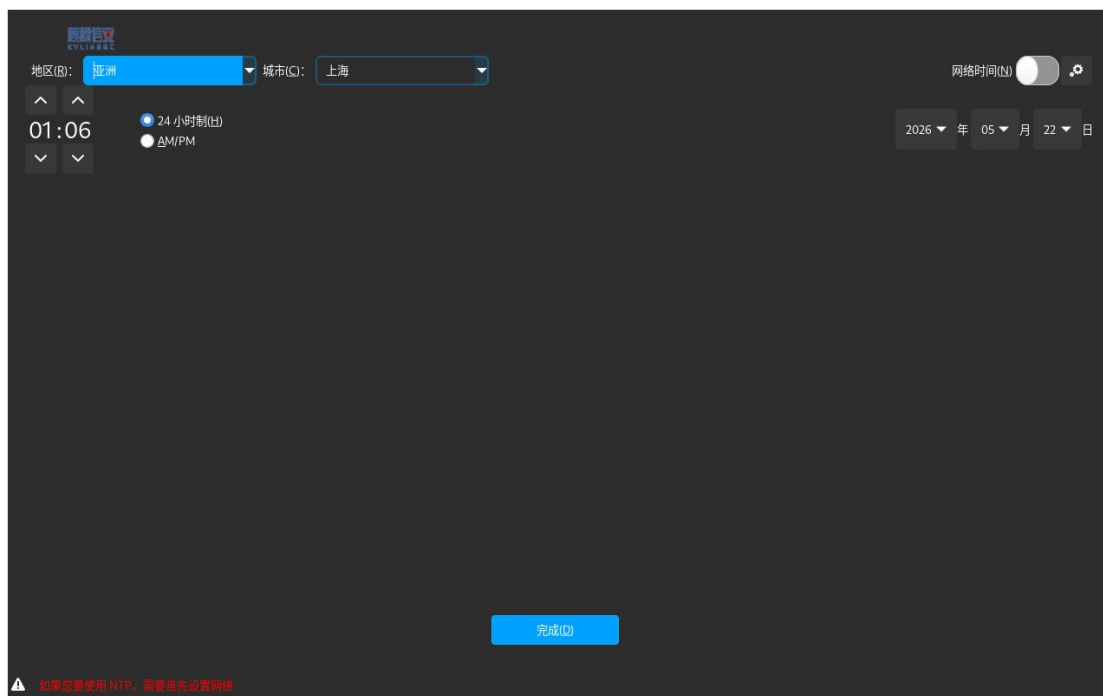


图 4-8 时区设置

注：“网络时间”开关，需在已连接网络的情况下才可以打开，默认为关闭状态。

4.2.3.2 软件设置

软件设置包括安装源和软件选择，安装源默认会自动检测到，无需修改；软件选择可以设置系统安装形态，包括了“图形化服务器”、“最小安装”、“服务器”和“虚拟化主机”形态。

图形化服务器：搭配麒麟信安自研桌面 KiranUI，强化了对即插即用硬件设备的支持和图形化网络管理器；实现了多种用户友好的易用性设计，提供了可视化操作环境,适合作为日常办公、开发调试及图形交互的主机环境。

最小安装：针对服务器使用场景的最小安装合集，仅包含系统运行所必需的内核、基础命令行工具、网络管理、包管理及简单编译环境。

服务器：在最小安装形态的基础上增加了容器、服务管理、定时任务、统一

存储管理接口、端口映射服务、授权框架功能等组件为用户提供了系统运维管理与服务支撑能力。

虚拟化主机：在最小安装形态的基础上，提供虚拟化环境，支持从镜像快速创建虚拟机，配有网络与存储管理及命令行接口，可在主机上运行与管理多个独立虚拟机。

每个安装形态下有许多附加软件，默认都未勾选，用户可以自主选择是否进行勾选安装，如图 4-9：

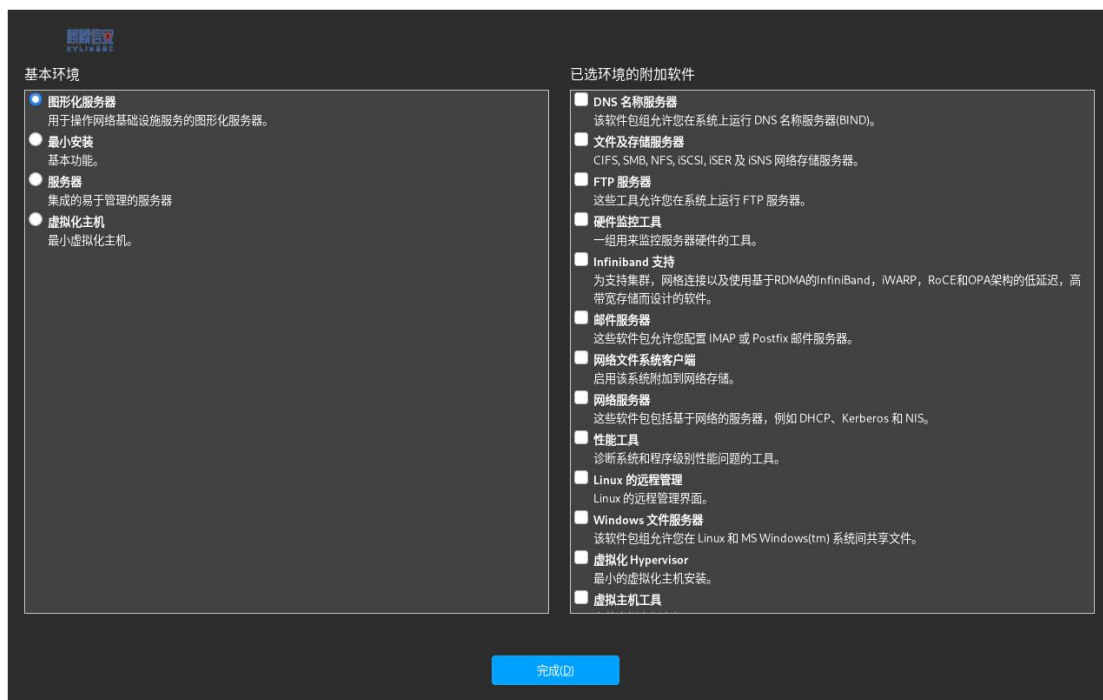


图 4-9 软件选择

4.2.3.3 系统设置

系统设置包括安装目的地、网络和主机名设置。安装目的地是用户必须手动设置的选项，需要勾选安装系统的磁盘，存储配置分为自动和自定义。

自动：自动创建分区方式，如图 4-10 自动分区界面 自动分区界面。若磁盘剩余空间充足，使用 Legacy 方式安装则在剩余可用磁盘空间自动新建固定大

小的分区/, /boot, /swap; 若使用 UEFI 方式安装则自动新建固定大小的分区 /, /boot, /swap, /boot/efi。系统会根据磁盘空间情况, 自动判断是否生成 /home 分区来存储数据。若磁盘剩余空间不足会给出提示, 用户可以格式化磁盘之前的分区来释放空间。

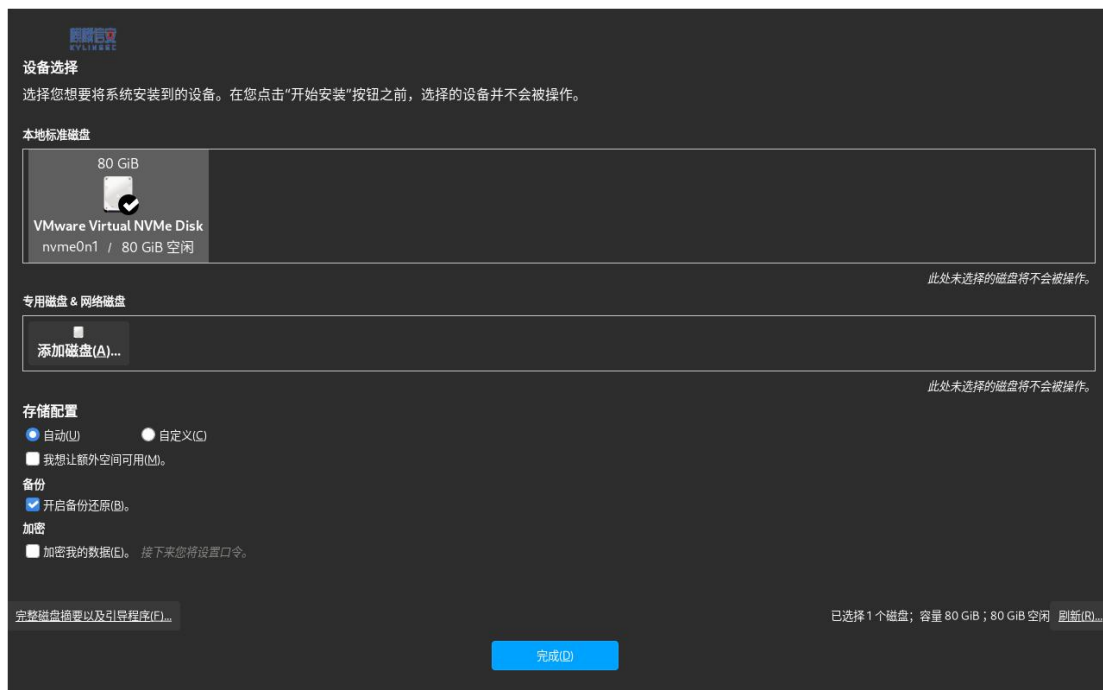


图 4-10 自动分区界面

注:如需创建初始备份快照和备份还原单独使用的 backup 分区,需勾选“开启备份还原”选项,设置 /backup 分区最少需要 10 个 G, 具体备份还原功能使用请查看 9 章节。如需加密数据,则勾选“加密我的数据”。

加密我的数据:该功能会加密机器磁盘,需设置磁盘密码,从加密磁盘启动时,在进入磁盘前需输入正确密码才能正常使用。

勾选“加密我的数据”,点击完成,弹出输入“磁盘加密口令”界面,输入密码后点击“保存口令”,如图 4-11 所示:

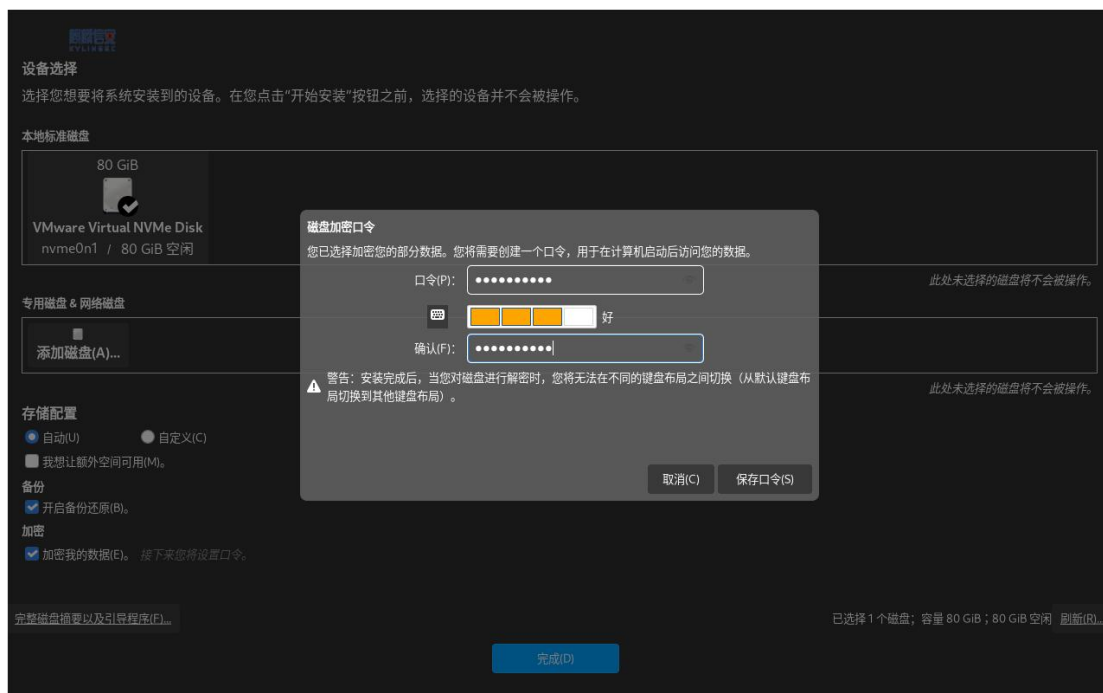


图 4-11 磁盘加密 1

自定义：手动创建分区方式，勾选“自定义”>点击“完成”按钮进入到手动创建分区界面，如图 4-12。

注：如需使用备份还原功能，需勾选“开启备份还原”选项。

如需加密，则勾选“自动地加密默认创建的挂载点”选项，再自定义进行自定义分区操作。

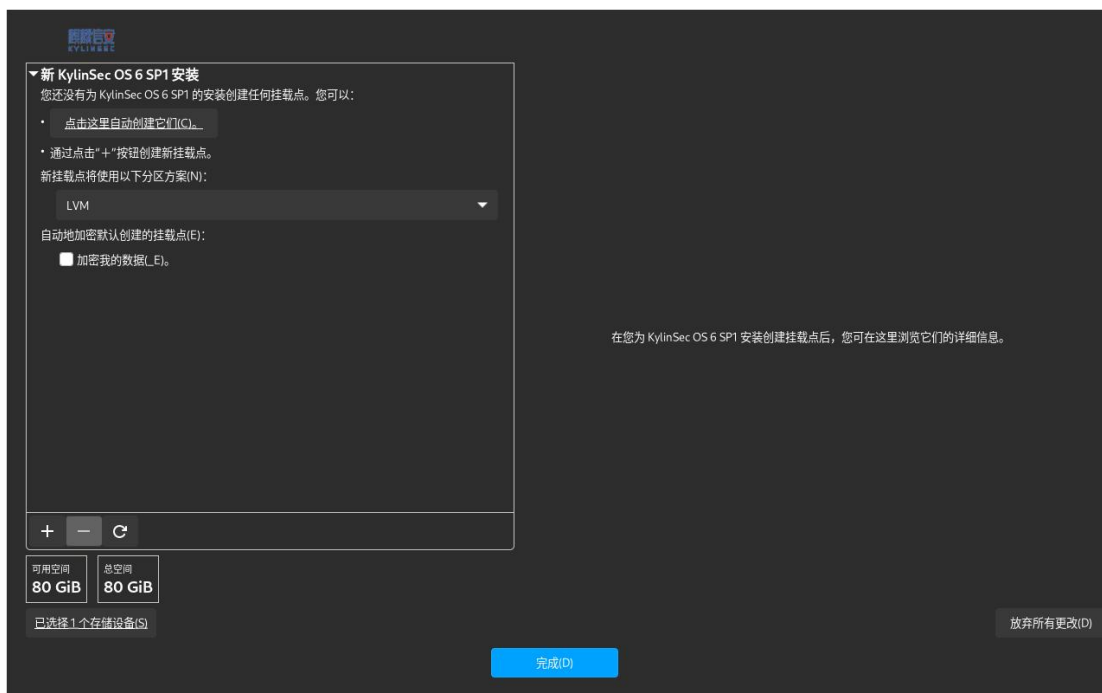


图 4-12 自定义分区

可以通过两种方法手动创建分区，一是点击“点击这里自动创建它们”按钮，此时会创建出分区：/, /boot, /swap，用户选中某个分区输入期望容量可以修改分区的大小，设置完成后点击“完成”按钮。如图 4-13：

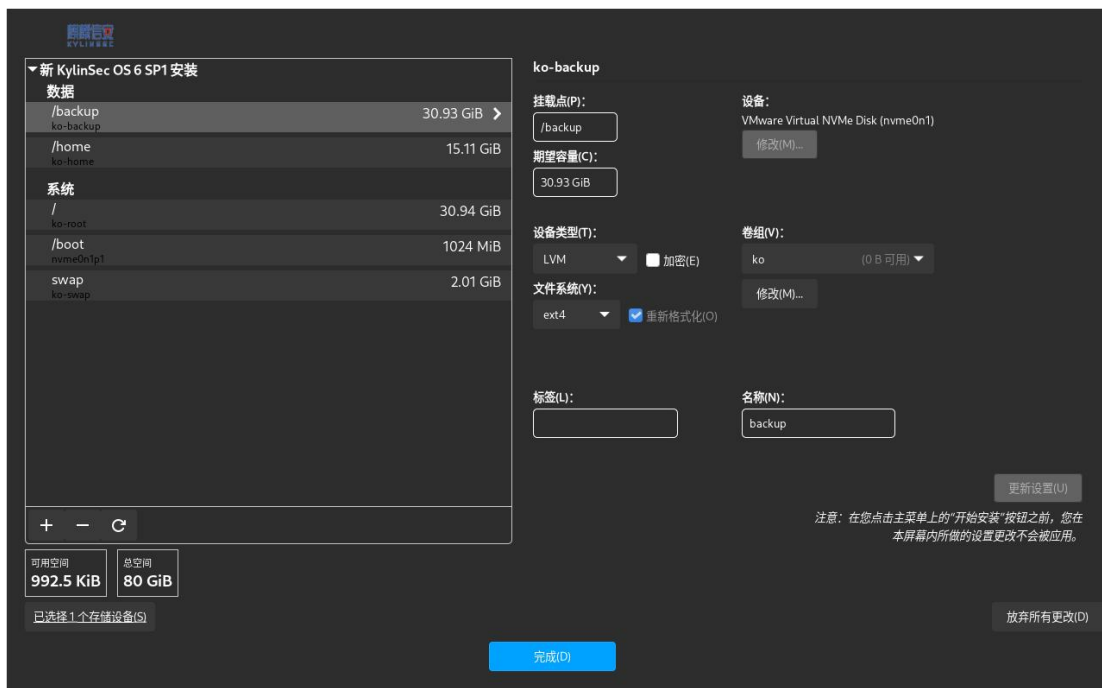


图 4-13 点击自动创建分区

二是点击“+”按钮弹出如下所示弹框，依次添加分区/,/boot,/swap,在磁盘空间有限时/分区建议容量大于 60G，/boot 分区建议容量 2G，/swap 分区建议 2G，可根据磁盘实际情况进行调整，也可添加/home 分区用于数据存储。

如图 4-14：

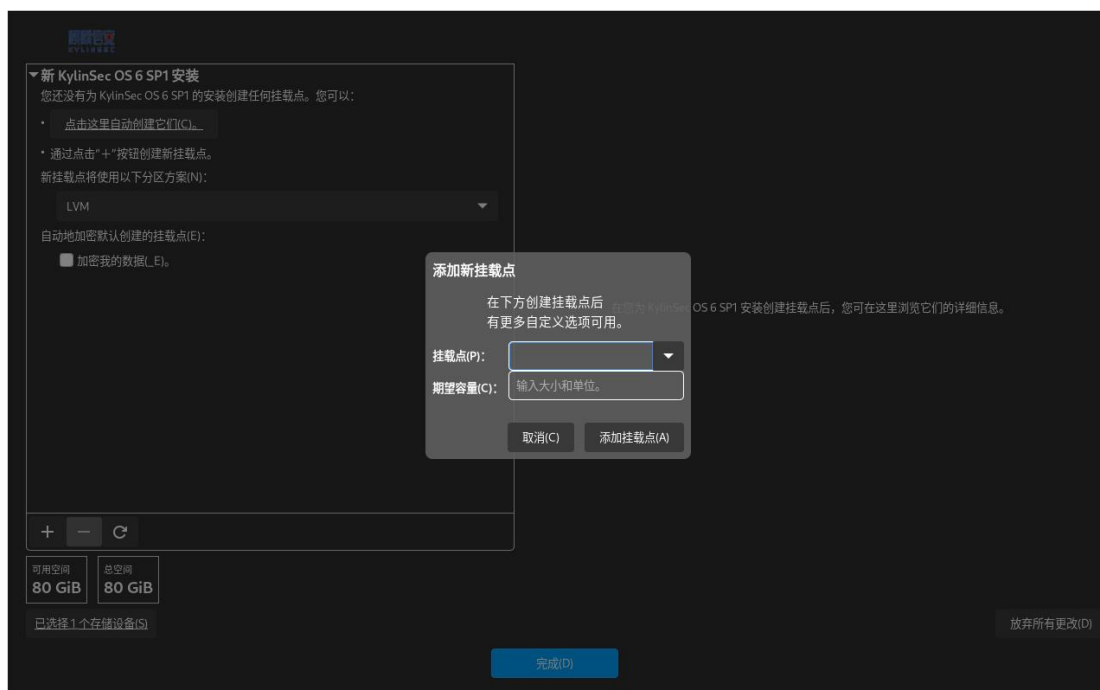


图 4-14 手动添加分区

添加完成后点击“完成”按钮，弹出如所示的更改摘要提示框，点击“接受更改”即可，如下图 4-15：

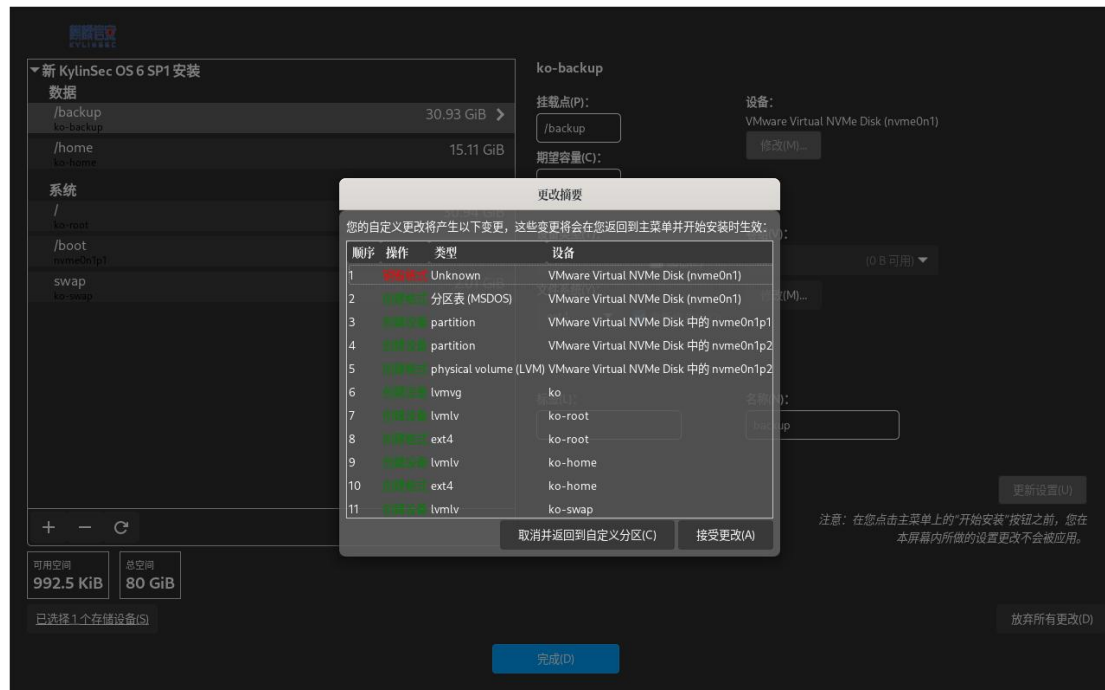


图 4-15 分区更改摘要提示

如果勾选了“自动地加密默认创建的挂载点”选项，则在弹出更改提示框前，弹出“磁盘加密口令”界面，如图 4-16：

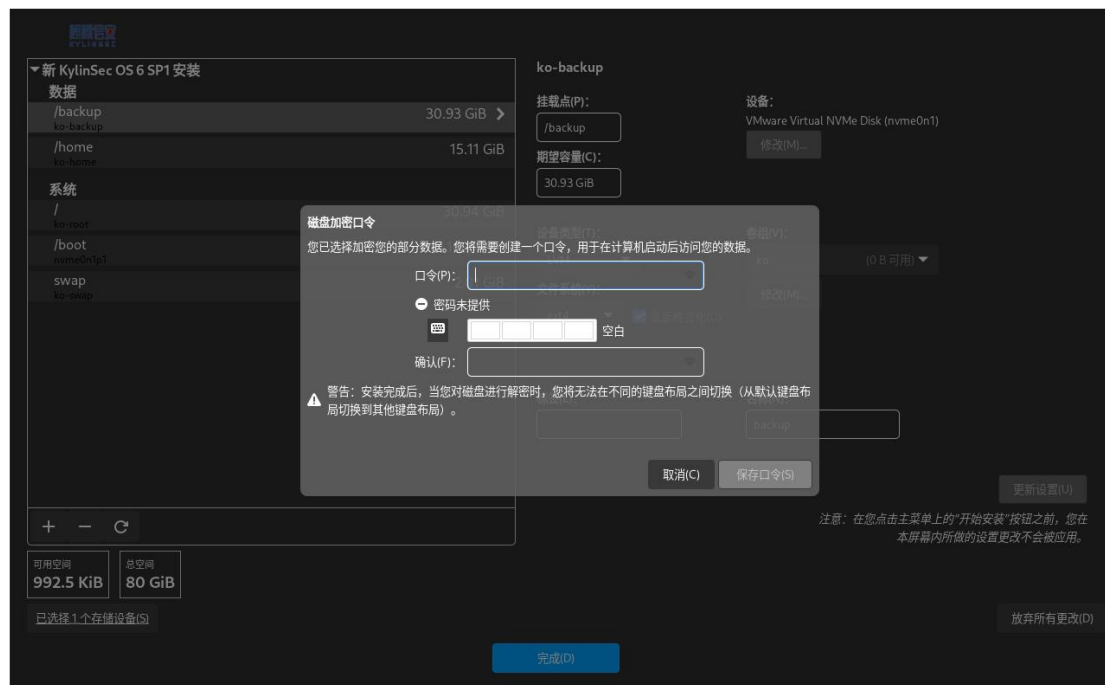


图 4-16 磁盘加密 2

网络和主机名设置:可以在界面左下角的主机名输入框中手动输入后点击

“应用”按钮进行修改，如图 4-17 所示：

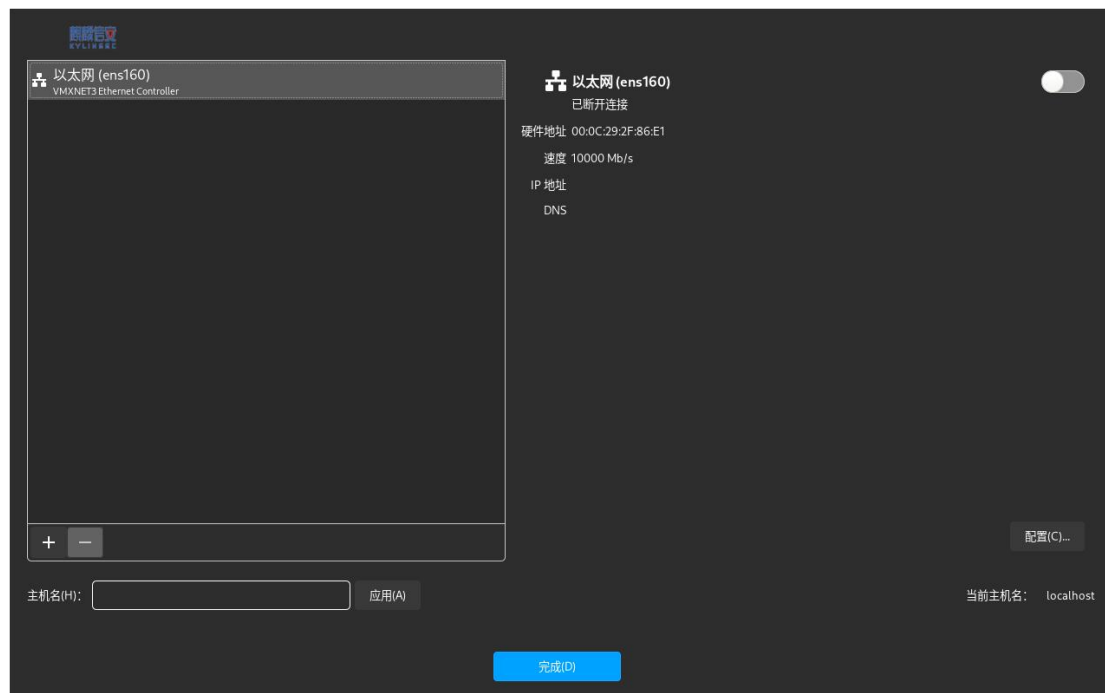


图 4-17 网络和主机名

选中以太网点击界面中的“配置”按钮，会弹出对话框，可手动配置 IP 地址、子网掩码、网关和 DNS 等。设置完成点击“保存”>“完成”按钮即可保存回到安装配置界面。如图 4-18：

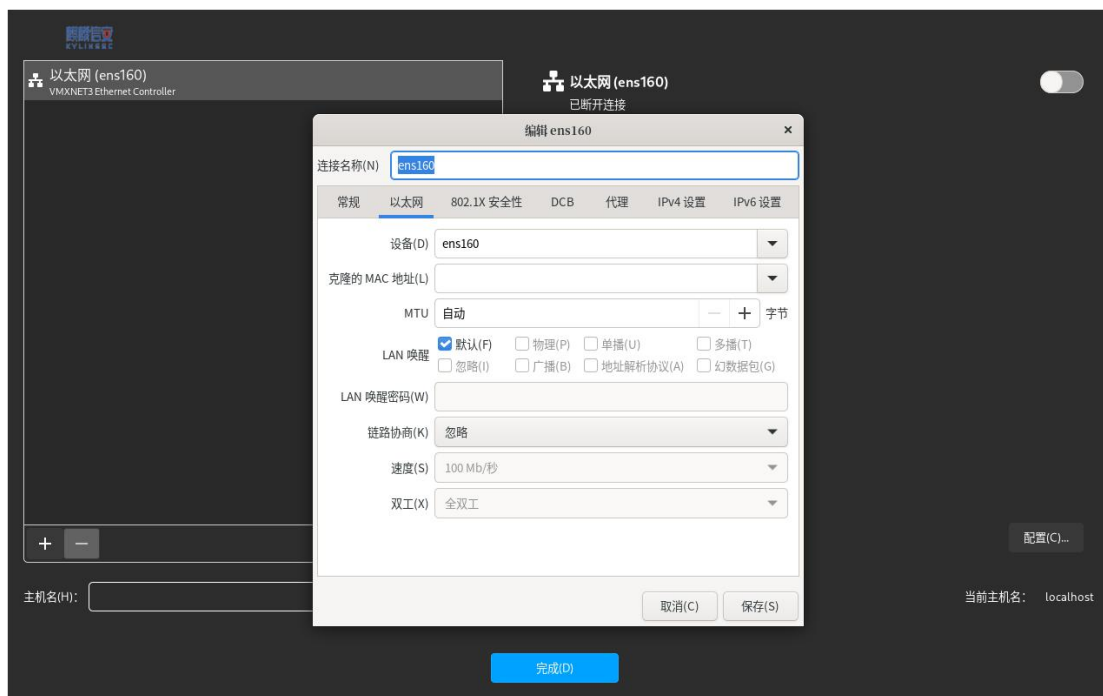


图 4-18 手动配置 IPv4

4.2.3.4 用户设置

用户设置包括了 root 账户、三权分立和创建用户。root 是系统的超级管理员用户，可以根据需要启用 root 账户或禁用 root 账户。禁用 root 用户后，无法使用 root 用户登录并使用系统。需创建新用户去使用系统，创建用户方法如图 4-19 所示。

点击“root 账户”进入 root 密码设置界面

禁用 root 账户：选择“禁用 root 账户”后点击完成。

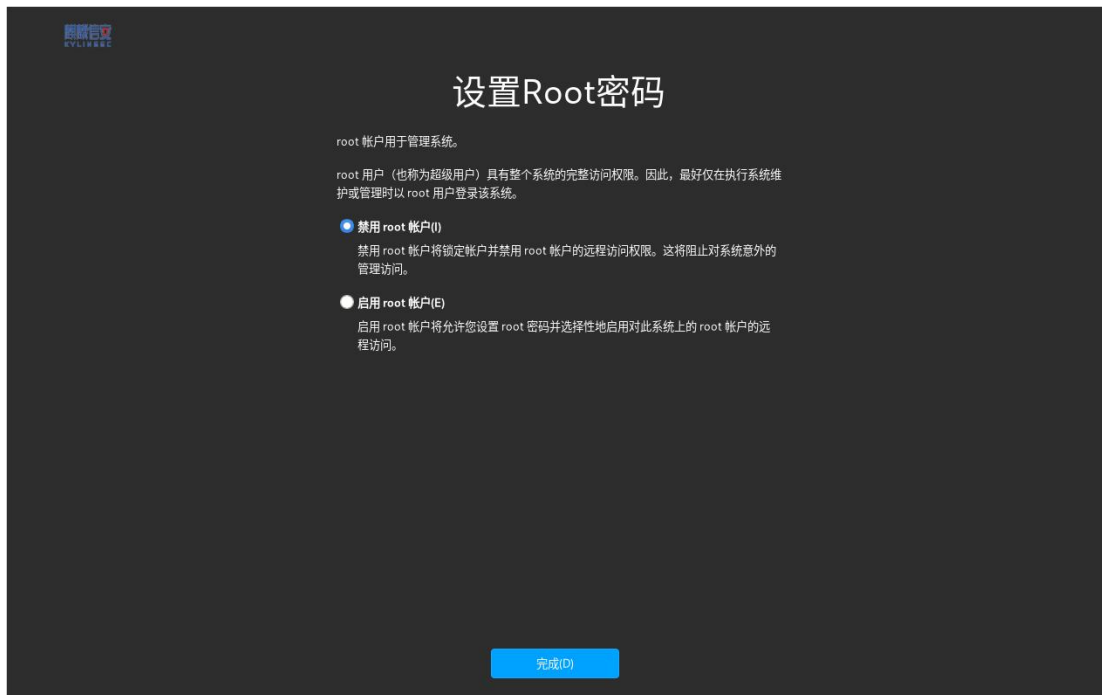


图 4-19 设置 root 密码

启用 root 账户：选中“启用 root 账户”后，输入密码后再次确认密码，两次密码一致，点击“完成”方可设置成功。如图 4-20：

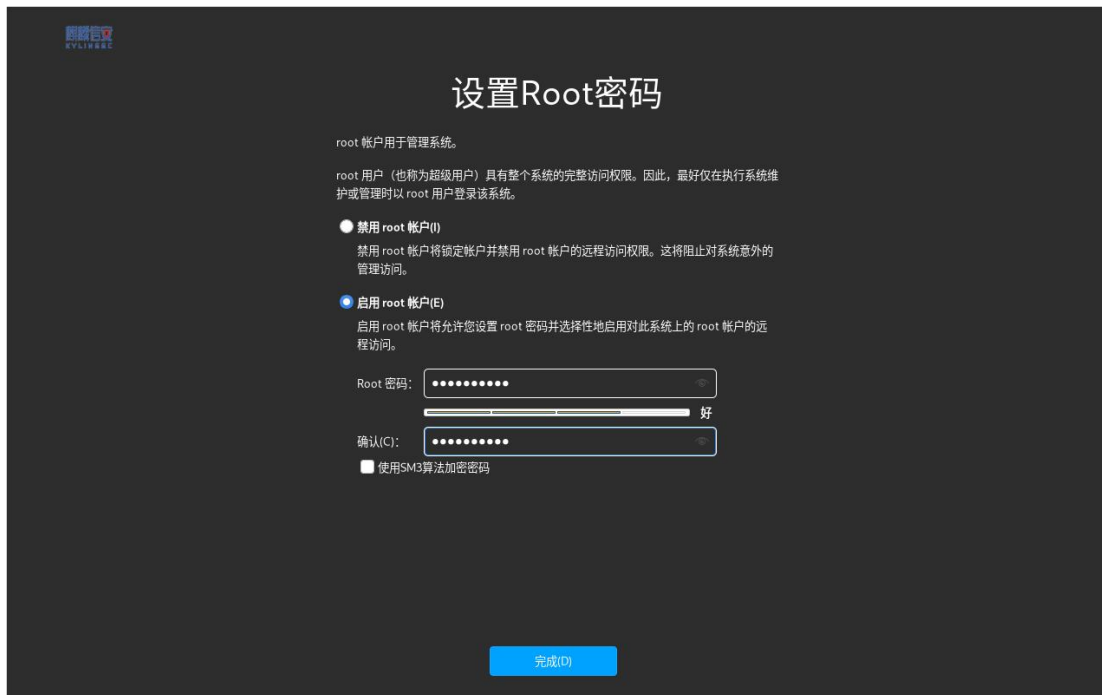


图 4-20 设置 root 密码

系统默认使用 SHA256 算法加密密码，可根据需要是否更改为 SM3 算法，如需使用 SM3 算法，则勾选“使用 SM3 算法加密密码”选项，再点击“完成”。

注意：root 用户应记好自己的密码，并且养成定期更改的好习惯。

系统也提供了自主创建普通用户的功能，点击“创建用户”可以创建一个日常工作使用的账户，输入全名、用户名、密码和确认密码，两次密码一致方可进入下一步，设置完成后点击“完成”按钮即可。如图 4-21：

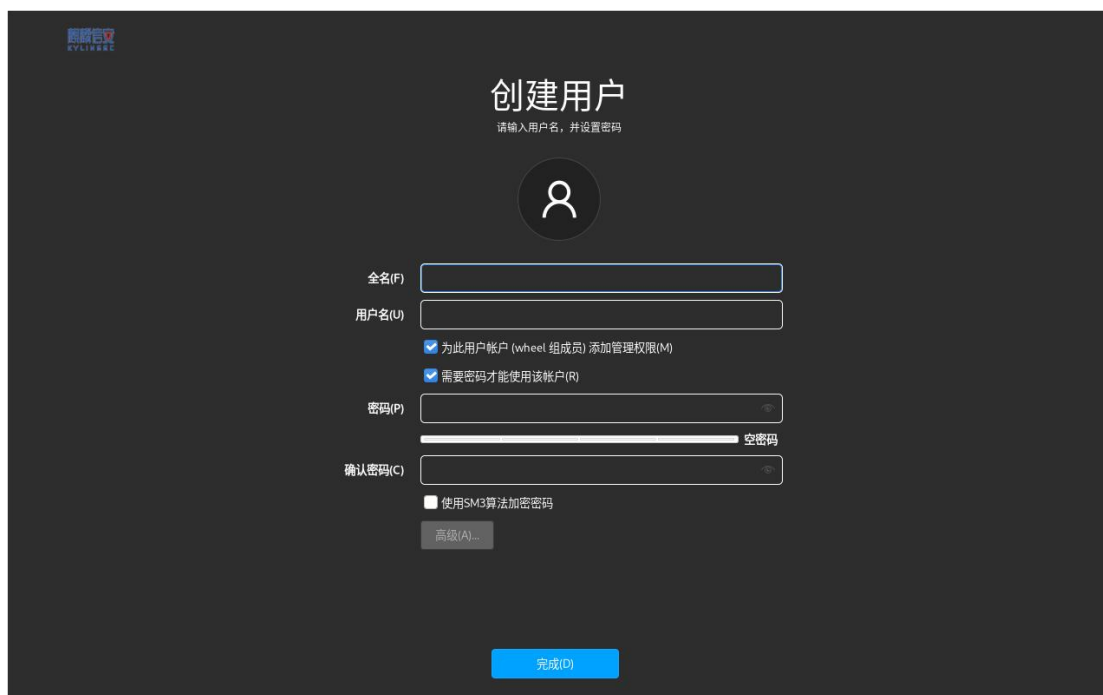


图 4-21 创建普通用户

至此所有的安装配置已经完成，回到安装配置界面，所有的黄色警告图标消失，点击“开始按钮”按钮即可开始安装系统，点击“退出”按钮将退出安装，如图 4-22：

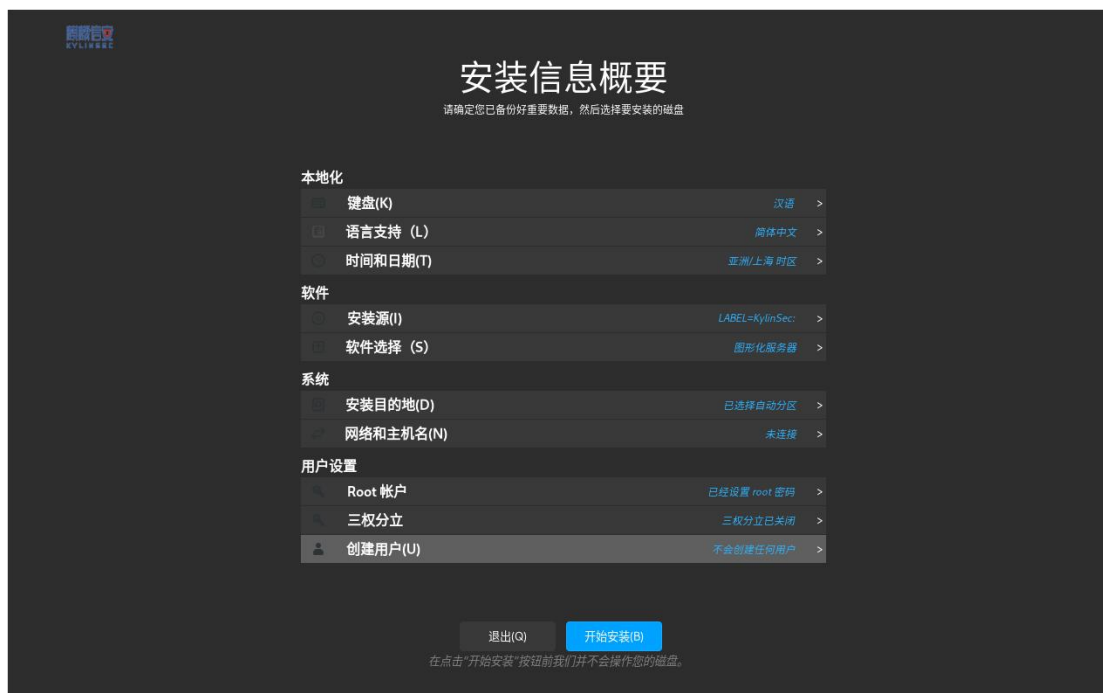


图 4-22 安装配置完成

安装过程如图 4-23 所示：



图 4-23 安装过程

4.2.4 安装完成

安装完成界面如图 4-24，点击“重启系统”后系统重启。



图 4-24 安装完成界面

初次启动系统需要接受“用户许可证”，如图 4-25 所示，点击“许可信息”>勾选“我同意许可协议”>点击“完成”>点击“结束配置”，即可进入登录界面。

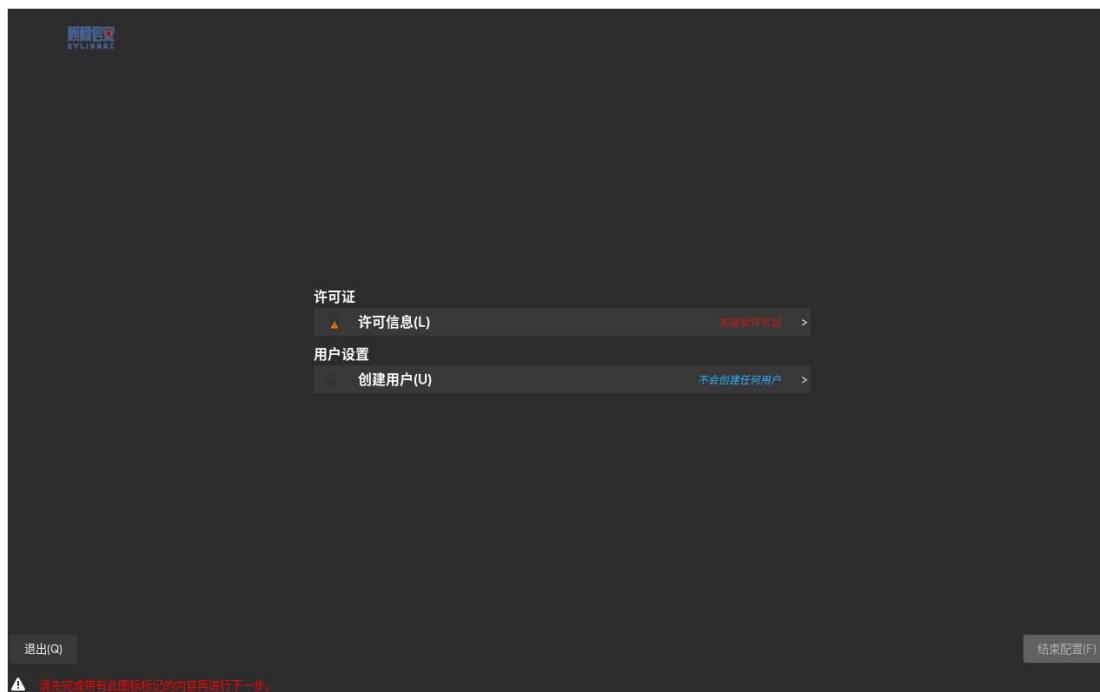


图 4-25 用户许可证

4.2.5 系统登录

4.2.5.1 图形化界面

系统登录图形化界面显示了用户名输入框、时间日期，电源按钮。界面支持自适应调整，支持屏幕放缩，支持多屏显示。系统支持手动输入用户名和密码方式登录系统，也支持选择用户后输入密码登录（当系统存在普通用户，登录界面左下角会显示普通用户名列表），登录界面如图 4-26 所示：

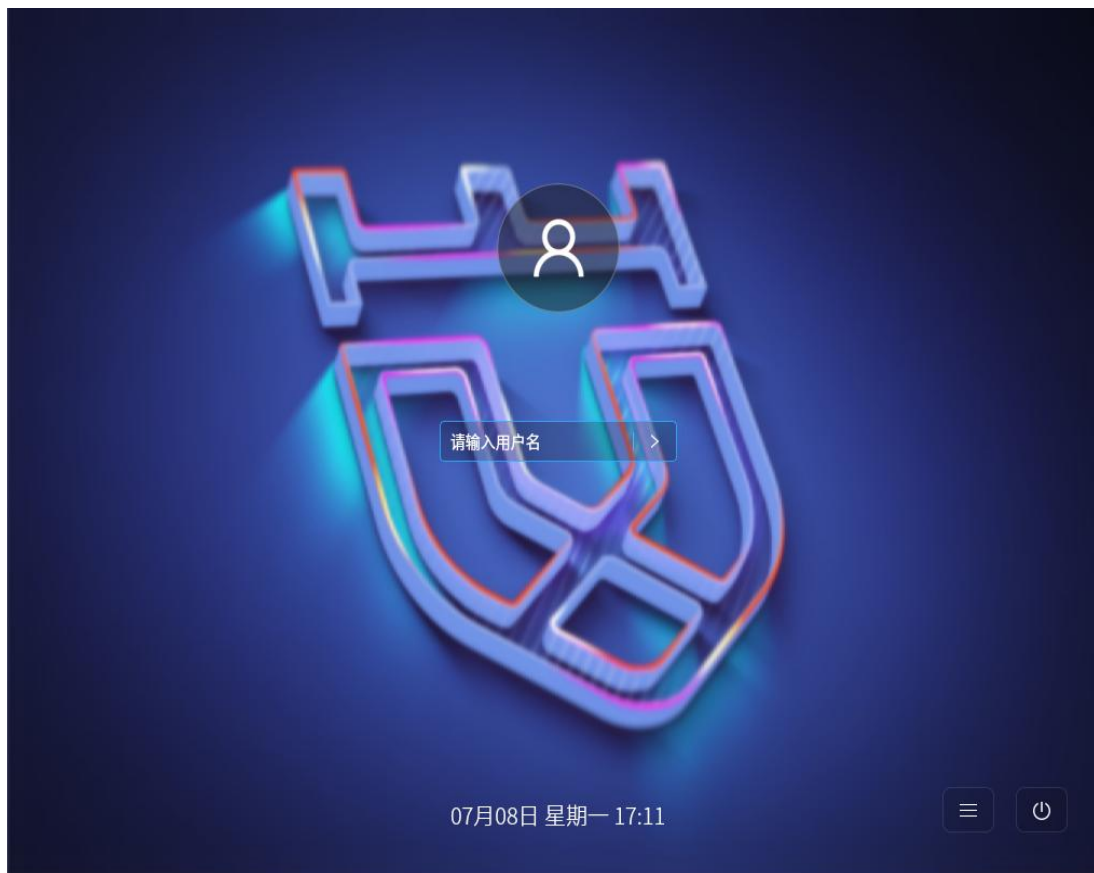
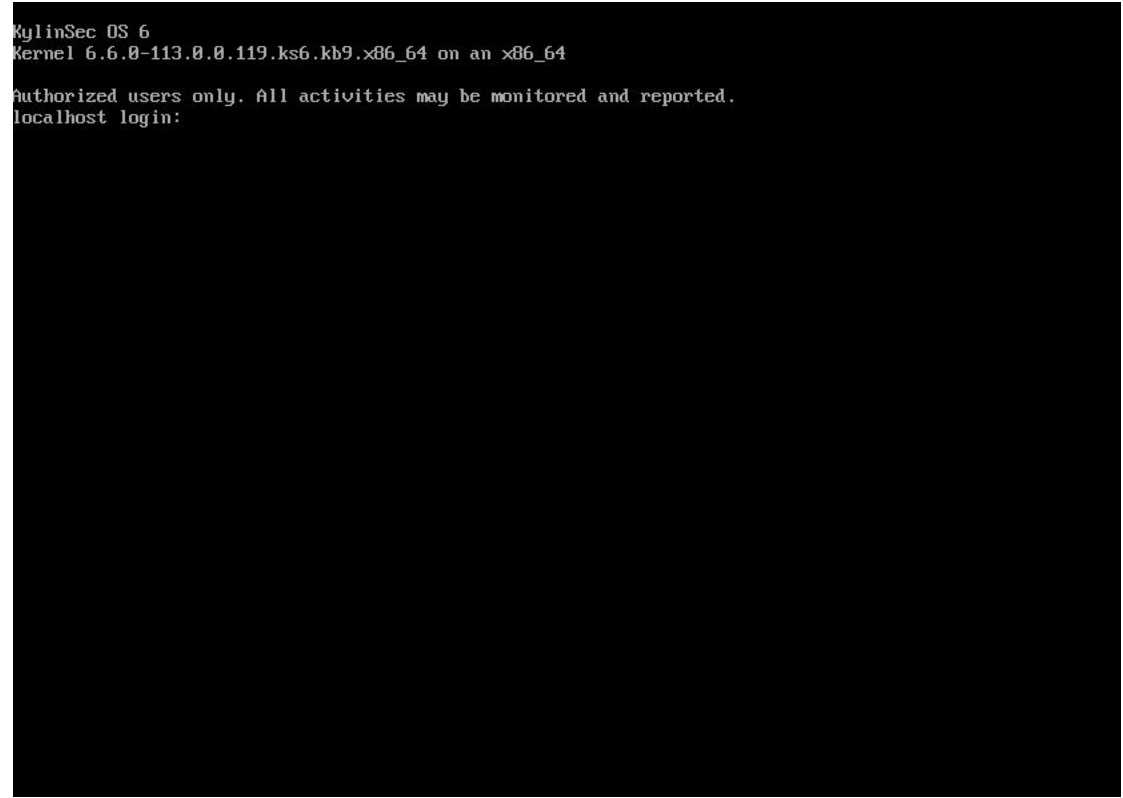


图 4-26 图形化登录

4.2.5.2 最小安装界面

最小安装模式下，系统直接进入文本登录界面，如图 4-27 所示。界面仅显示系统版本、内核信息及登录提示符，不加载图形界面。用户需手动输入用户名和密码登陆系统，界面简洁，无额外交互元素。

A terminal window with a black background and white text. The text displays the operating system version and kernel information, followed by a security notice and a login prompt.

```
KylinSec OS 6  
Kernel 6.6.0-113.0.0.119.ks6.kb9.x86_64 on an x86_64  
  
Authorized users only. All activities may be monitored and reported.  
localhost login:
```

图 4-27 最小安装登录

4.2.5.3 服务器界面

服务器模式下，系统启动后进入文本登录界面，如图 4-28 所示。界面仅显示系统版本、内核信息、及登录提示符，不加载图形界面。用户需手动输入用户名和密码登陆系统，界面简洁，无额外交互元素。

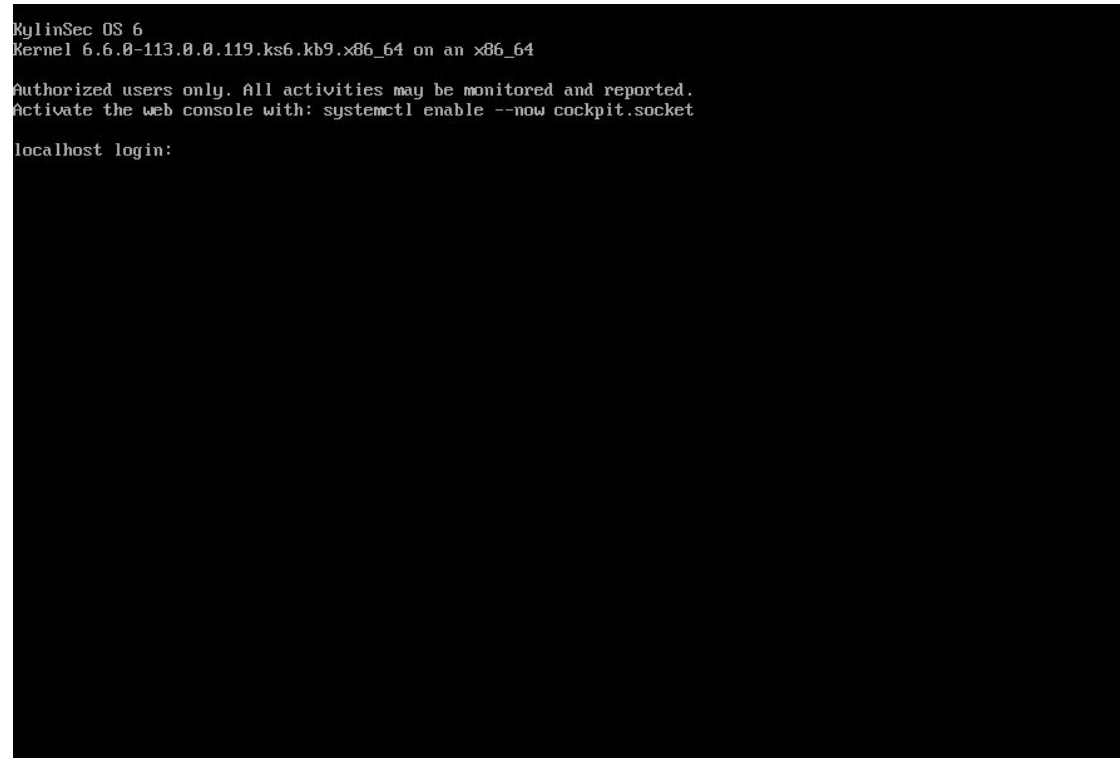
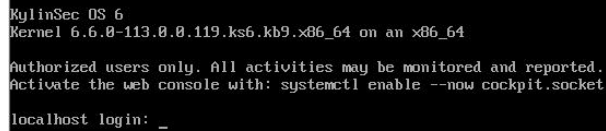


图 4-28 服务器登录

4.2.5.4 虚拟化主机界面

虚拟化主机模式下，系统进入文本登录界面，如图 4-29 所示。界面显示系统版本、内核信息及登录提示符，提示可通过命令启用 Web 控制台。用户需手动输入用户名和密码登录系统，界面与服务器模式一致，满足虚拟化主机的命令行管理需求。



```
KylinSec OS 6
Kernel 6.6.0-113.0.0.119.ks6.kb9.x86_64 on an x86_64

Authorized users only. All activities may be monitored and reported.
Activate the web console with: systemctl enable --now cockpit.socket

localhost login: _
```

图 4-29 虚拟化主机界面

4.2.6 版本信息

麒麟信安服务器操作系统中，可以通过命令行查看版本信息 `ks-showinfo`，

如图 4-30 所示：



```
[root@localhost ~]# ks-showinfo
[dist]
name = KylinSec-Server
milestone = 6-SP1-beta-260330-2117
arch = x86_64
dist_id = KylinSec-Server-6-SP1-beta-260330-2117-x86_64
installclass = server-environment
touser =
```

图 4-30 版本信息

5 最小安装

最小安装形态仅包含系统运行所必需的内核、基础命令行工具、网络管理、包管理及简单编译环境。安装后用户可通过 `yum/dnf install <package>` 按需安装软件包。

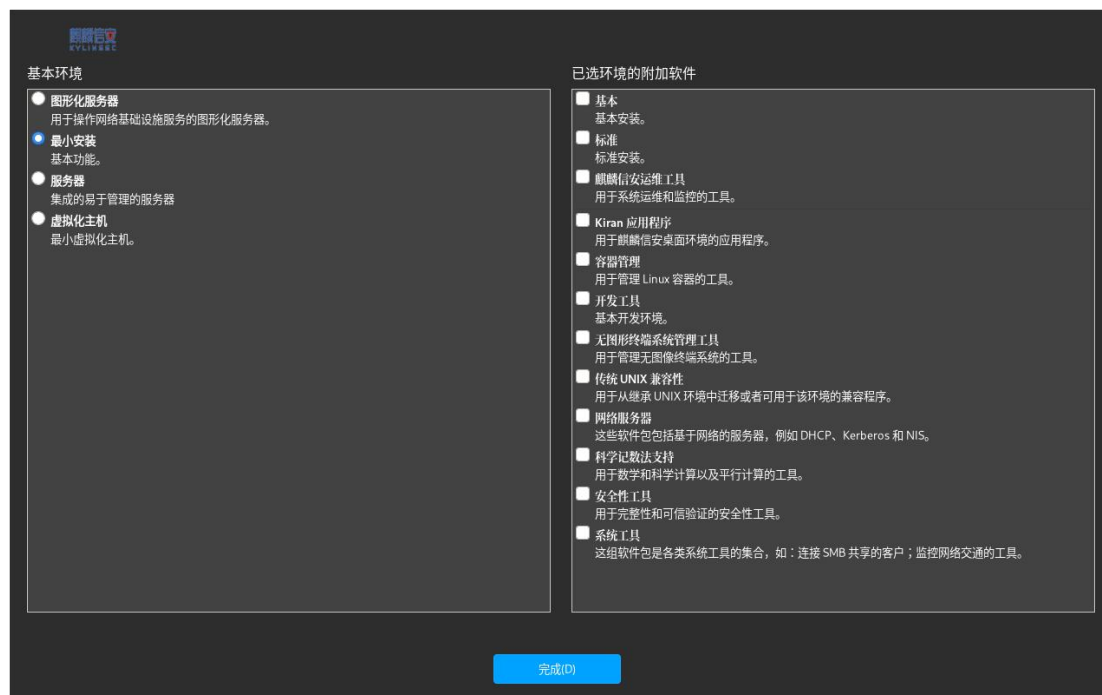


图 5-1 最小化安装界面

5.1 安装完成

安装完成界面如图 5-2，点击“重启系统”后系统重启。



图 5-2 安装完成界面

系统启动后, 首先会进入许可界面, 需接受“用户许可证”后, 方可正常进入系统, 如图 5-3 所示:

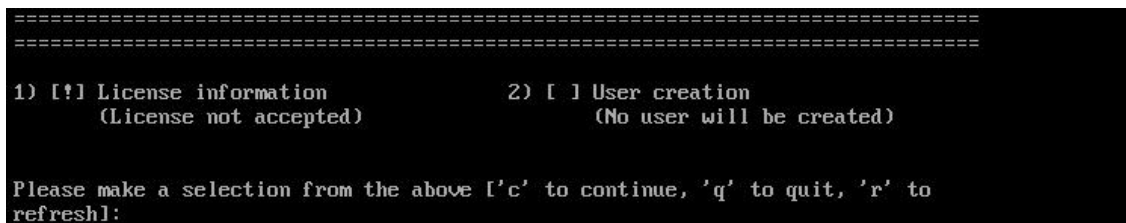


图 5-3 许可界面

接受“用户许可证”: 如图 5-4 所示, 在提示界面输入“1”>输入“Enter”翻页>输入“2”同意许可证内容>输入“c”继续, 将回到图 5-3 所示界面, 输入“c”即可进入登录界面。

```
Please make a selection from the above ['c' to continue, 'q' to quit, 'r' to
refresh]:
1
=====
License information

1) Read the License Agreement
2) [ ] I accept the license agreement.

Please make a selection from the above ['c' to continue, 'q' to quit, 'r' to
refresh]:
```

图 5-4 用户许可证界面

“创建一个新用户”：输入“2”进入创建用户，如图 5-5 所示，输入“3”>输入用户名>输入“Enter”>输入“5”>输入密码（无明文显示）>输入“Enter”>再次输入密码>输入“Enter”>输入“c”继续。

```
=====
User creation

1) [x] Create user
2) Full name
3) User name
4) [x] Use password
5) Password
6) [x] Administrator
7) Groups
   wheel

Please make a selection from the above ['c' to continue, 'q' to quit, 'r' to
refresh]:
```

图 5-5 创建新用户界面

确认界面如图 5-6，此时“许可证”与“用户创建”均已完成勾选，输入“c”即可进入登录界面。

```
=====
1) [x] License information          2) [x] User creation
   (License accepted)              (Administrator admin will be
                                   created)

Please make a selection from the above ['c' to continue, 'q' to quit, 'r' to
refresh]:
```

图 5-6 确认界面

5.2 多终端（tty）切换

系统默认提供多个虚拟终端（tty），支持用户同时打开多个独立的命令行窗口，可分别执行不同的系统操作，互不干扰。系统默认启用了 6 个可登录的虚拟终端，可通过组合键 “Ctrl” + “Alt” + “F1”（可替换为 “F1” - “F6”）快速切换。

5.3 系统激活

通过终端命令：`ksl-register -h` 查看激活相关的命令和功能，如图 5-7 所示，执行该命令将输出系统激活工具的全部参数列表及功能详解。

```
[root@localhost ~]# ksl-register -h
Usage:
  ksl-register [OPTION...]

Help Options:
  -h, --help                Show help options

Application Options:
  -r, --register             Activate the machine. If the activation
                             code is specified, then activate machine using activation code, and if the server host is specified, then activate machine online. Otherwise, activate using U
                             SBKey device.
  -l, --license-code        Specify the activation code, the argument
                             is deprecated, please use -a.
  -a, --activation-code      Specify the activation code.
  -d, --server-host         Specify the activation server host.
  --service-uid             Specify the service uid for activation
                             server.
  -f, --activation-file      Specify the activation file
  -s, --show                Show activation info.
  -v, --version             Output version information and exit.
  -g, --get-registration-status status [expired-status] [machinecode] [customercode] [authcode] [expiredate] [kylinsec-telephone]
                             Get the specified variable.
  -o, --activation-object-name Name of activation object. If not specified,
                             KylinSecOS is used.
  -j, --json                Use JSON output format.
  --wait-service            Wait for ksl-daemon service start.
  --reset-install-time      Allow resetting installation time once
                             .
  --launch-dbus             Start D-Bus daemon, typically used for
                             emergency situations, in conjunction with displaying activation information.
```

图 5-7 ksl-register -h 命令输出

通过终端命令：`ksl-register -s` 获取系统机器码，如图 5-8 所示。

```
[root@localhost ~]# ksl-register -s
Activation status:  Not activated.
Trial expiration:  2826-10-13.
Machine code:      0732-7451-082C-22DF.
Installation time:  2826-04-15 01:50:14.
```

图 5-8 系统机器码

激活系统有 3 种方式：激活码、在线激活、Usbkey。

（1）提供机器码给支持人员，获取激活码。

通过终端命令激活系统：`ksl-register -r -a 激活码 -s`，如图 5-9 所示。

```
[root@localhost ~]# ksl-register -r -a C6TRV-7H9CW-FKJQZ-611XI-VH1C1 -s
激活状态:      已激活.
激活模式:      订阅激活.
激活有效期:    2026-05-14.
售后服务期限:  2026-05-14.
激活类型:      激活码.
激活码:        C6TRV-7H9CW-FKJQZ-611XI-VH1C1.
机器码:        07C7-813E-88DE-DAD8.
激活时间:      2026-04-14 16:50:40.
安装时间:      2026-04-07 13:29:02.
```

图 5-9 命令激活系统

(2) 支持人员搭建在线激活服务器，保证待激活系统与在线激活服务器能 ping 通，系统时间同步。

通过终端命令激活系统：`ksl-register -r -d “在线激活服务器 IP” --service-uid “服务号” -s`

(3) 支持人员提供 UsbKey，保证 UsbKey 有激活点数。

通过终端命令激活系统：`ksl-register -r -s`

5.4 查看系统信息

5.4.1 查看版本信息

命令如下：

```
[root@localhost ~]# ks-showinfo
```

5.4.2 查看 CPU 信息

命令如下：

```
[root@localhost ~]# lscpu
```

5.4.3 查看内存信息

命令如下：

```
[root@localhost ~]# free
```

5.4.4 查看磁盘信息

命令如下：

```
[root@localhost ~]# fdisk -l
```

5.4.5 设置网络

5.4.5.1 nmcli 命令

nmcli 是 NetworkManager 的一个命令行工具，使用 nmcli 命令配置的网络配置可以立即生效且系统重启后配置也不会丢失。

nmcli 命令的基本格式格式如下：

```
[root@localhost ~]# nmcli help
用法: nmcli [选项] 对象 { 命令 | help }

选项
-a, --ask                询问缺少的参数
-c, --colors auto|yes|no 是否在输出中使用颜色
-e, --escape yes|no      转义值中的列分隔符
-f, --fields <字段,...>|all|common 指定要输出的字段
-g, --get-values <字段,...>|all|common -m tabular -t -f 的快捷方式
-h, --help              打印此帮助
-m, --mode tabular|multiline 输出模式
-o, --overview          概览模式
-p, --pretty            美化输出
-s, --show-secrets      允许显示密码
-t, --terse            简介输出
-v, --version          显示程序版本
-w, --wait <秒数>      设定操作完成的等待超时

对象
g[eneral]              NetworkManager 的常规状态和操作
n[etworking]          整体网络控制
r[adio]               NetworkManager 无线电开关
c[onnection]          NetworkManager 的连接
d[evice]              NetworkManager 管理的设备
a[gent]               NetworkManager 机密 (secret) 或 polkit 代理
m[onitor]             监视 NetworkManager 更改
```

显示 NetworkManager 状态：

```
[root@localhost ~]# nmcli general status
STATE      CONNECTIVITY  WIFI-HW  WIFI    WWAN-HW  WWAN
已连接 (仅本地) 受限          missing  已启用  missing  已启用
```

显示所有连接：

```
[root@localhost ~]# nmcli connection show
```

NAME	UUID	TYPE	DEVICE
------	------	------	--------

只显示当前活动连接，添加-a, --active:

```
[root@localhost ~]# nmcli connection show -a
```

NAME	UUID	TYPE	DEVICE
------	------	------	--------

显示由 NetworkManager 识别到设备及其状态:

```
[root@localhost ~]# nmcli device status
```

DEVICE	TYPE	STATE	CONNECTION
lo	loopback	连接 (外部)	lo

使用 nmcli 工具启动和停止网络接口，在 root 权限下执行如下命令:

```
[root@localhost ~]# nmcli connection up id enp2s0
连接已成功激活 (D-Bus 活动路径: /org/freedesktop/NetworkManager/ActiveConnection/4)
[root@localhost ~]# nmcli device disconnect enp2s0
成功断开设备 "enp2s0"。
```

5.4.5.2 设备管理

连接到设备：使用如下命令，NetworkManager 将连接到对应网络设备，尝试找到合适的连接配置，并激活配置。如果不存在相应的配置连接，NetworkManager 将创建并激活具有默认设置的新配置文件。

```
[root@localhost ~]# nmcli device connect "$IFNAME"
```

断开设备连接：使用如下命令，NetworkManager 将断开设备连接，并防止设备自动激活。

```
[root@localhost ~]# nmcli device disconnect "$IFNAME"
```

5.4.5.3 设备网络连接

列出目前可用的网络连接:

```
[root@localhost ~]# nmcli con show
```

NAME	UUID	TYPE	DEVICE
lo	193a2e6f-cdfa-4573-9452-c4cd5944f319	loopback	lo
virbr0	d6b5525e-f527-45bb-be9b-bd6f39e311e3	bridge	virbr0
enp2s0	99d404ba-7ce8-4e28-baab-a16e8b1e56de	ethernet	--

添加一个网络连接会生成相应的配置文件，并与相应的设备关联。检查可用的设备，方法如下：nmcli device status

DEVICE	TYPE	STATE	CONNECTION
lo	loopback	连接（外部）	lo
virbr0	bridge	连接（外部）	virbr0
enp2s0	ethernet	不可用	--

5.4.5.4 配置动态 IP 连接

要使用 DHCP 分配网络时，可以使用动态 IP 配置添加网络配置文件，命令格式如下：

```
[root@localhost ~]# nmcli connection add type ethernet con-name connection-name ifname interface-name
```

type ethernet: 网络类型为以太网。

con-name <Connection-Name>: 连接名称为 <Connection-Name>。

ifname <Interface-Name>: 接口名称为 <Interface-Name>。

例如创建名为 net-test 的动态连接配置文件，在 root 权限下使用以下命令：

```
[root@localhost ~]# nmcli connection add type ethernet con-name net-test ifname enp2s0
连接 "net-test" (7e8f28c8-a8e0-4164-87c2-7fd8e684e76c) 已成功添加。
```

激活连接并检查状态：

```
[root@localhost ~]# nmcli con up net-test
连接已成功激活 (D-Bus 活动路径: /org/freedesktop/NetworkManager/ActiveConnection/5)
[root@localhost ~]# nmcli device status
DEVICE  TYPE      STATE      CONNECTION
enp2s0  ethernet  已连接     net-test
lo      loopback  连接（外部） lo
virbr0  bridge    连接（外部） virbr0
```

5.4.5.5 配置静态 IP 连接

添加静态 IPv4 配置的网络连接，可使用以下命令：

```
[root@localhost ~]# nmcli connection add type ethernet con-name connection-name ifname interface-name ip4 your_ipv4 gw4 ipv4_gw
```

ip4 <Your_IPv4>[/<24>]: 将连接的 IPv4 地址设定为<Your_IPv4>。

gw4 <IPv4_gw>: 设置网关为<IPv4_gw>。

例如创建名为 net-static 的静态连接配置文件，在 root 权限下使用以下命令：

```
[root@localhost ~]# nmcli connection add type ethernet con-name static-test ifname enp2s0 ip4 10.90.21.111 gw4 10.90.21.1  
连接 "static-test" (205a2bea-743c-4ad5-874c-e8cb1a12e01d) 已成功添加。
```

激活连接并检查状态：

```
[root@localhost ~]# nmcli con up static-test ifname enp2s0  
连接已成功激活 (D-Bus 活动路径: /org/freedesktop/NetworkManager/ActiveConnection/6)  
[root@localhost ~]# nmcli device status  
DEVICE    TYPE      STATE      CONNECTION  
enp2s0    ethernet  已连接      static-test  
lo        loopback  连接 (外部)  lo  
virbr0    bridge    连接 (外部)  virbr0
```

5.4.5.6 配置主机名

查看当前的主机名，使用如下命令：

```
[root@localhost ~]# hostnamectl status
```

在 root 权限下，设定系统中的所有主机名，使用如下命令：

```
[root@localhost ~]# hostnamectl set-hostname name_you_wanted_to
```

6 服务器

服务器形态是专门为持续、稳定、高效地提供网络服务而设计的专用操作系统。它与日常使用的桌面系统不同，核心目标不是“交互体验”，而是“可靠服务”。

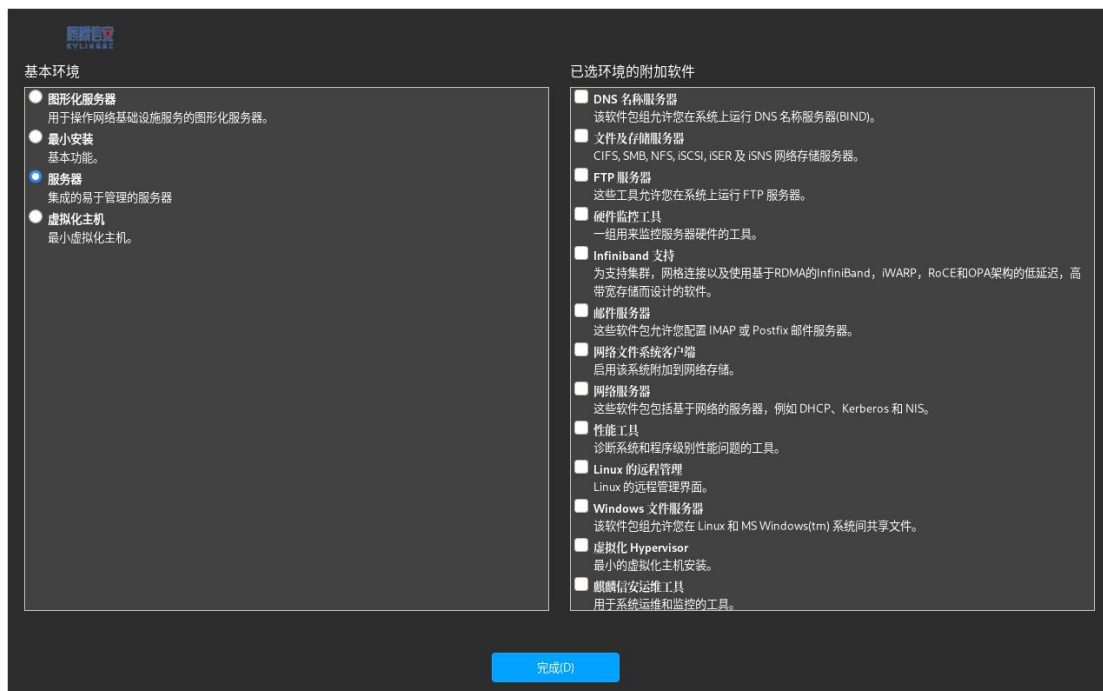


图 6-1 服务器安装界面

服务器形态在最小安装形态基础上增加了容器（docker）、服务管理（cockpit）、定时任务（atd）、统一存储管理接口（libstoragemgmt）、RPC 端口映射服务（rpcbind）、授权框架功能（polkit）等，为用户提供了系统的运维管理与服务支撑能力，满足生产环境下容器化部署、远程运维、存储资源管理等需求。

6.1 docker 服务

docker：用户可通过 docker images 查看本地镜像，使用 docker run -it --name <容器名> <镜像名> /bin/bash 启动容器，通过 docker ps -a 列出所有容器，使用 docker stop <容器 ID> 停止运行中的容器。容器服务运行状态如图 6-2：

```
[root@localhost ~]# systemctl status docker
○ docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; preset: disabled)
   Active: inactive (dead)
   TriggeredBy: ○ docker.socket
   Docs: https://docs.docker.com
[root@localhost ~]# systemctl start docker
[root@localhost ~]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; preset: disabled)
   Active: active (running) since Mon 2026-04-13 06:42:57 CST; 1s ago
   TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
   Main PID: 11569 (dockerd)
   Tasks: 10
   Memory: 112.6M ()
   CGroup: /system.slice/docker.service
           └─11569 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Apr 13 06:42:55 localhost.localdomain systemd[1]: Starting Docker Application Container Engine...
Apr 13 06:42:56 localhost.localdomain dockerd[11569]: time="2026-04-13T06:42:56.452610717+08:00" level=info msg="Starting up"
Apr 13 06:42:56 localhost.localdomain dockerd[11569]: time="2026-04-13T06:42:56.490399472+08:00" level=info msg="Loading containers: start."
Apr 13 06:42:56 localhost.localdomain dockerd[11569]: time="2026-04-13T06:42:56.602001112+08:00" level=info msg="Firewall: interface docker0 already part of docker bridge"
Apr 13 06:42:56 localhost.localdomain dockerd[11569]: time="2026-04-13T06:42:56.930466930+08:00" level=info msg="Loading containers: done."
Apr 13 06:42:57 localhost.localdomain dockerd[11569]: time="2026-04-13T06:42:57.007611670+08:00" level=info msg="Docker daemon" commit=fca782d graphdriver=overhead
Apr 13 06:42:57 localhost.localdomain dockerd[11569]: time="2026-04-13T06:42:57.007790718+08:00" level=info msg="Daemon has completed initialization"
Apr 13 06:42:57 localhost.localdomain dockerd[11569]: time="2026-04-13T06:42:57.124844159+08:00" level=info msg="API listen on /run/docker.sock"
Apr 13 06:42:57 localhost.localdomain systemd[1]: Started Docker Application Container Engine.
lines 1-28/28 (END)
```

图 6-2 容器服务

6.2 Cockpit 服务

Cockpit: 启动 Cockpit Web 管理平台可使用 `systemctl enable --now cockpit.socket`, 然后通过浏览器访问 `https://服务器 IP:9090` 图形化监控系统资源、管理容器、查看日志。日常服务管理采用 `systemctl start|stop|restart|status <服务名>` 控制各类服务, 例如 `systemctl restart httpd` 重启 Web 服务。Cockpit 管理界面如图 6-3:

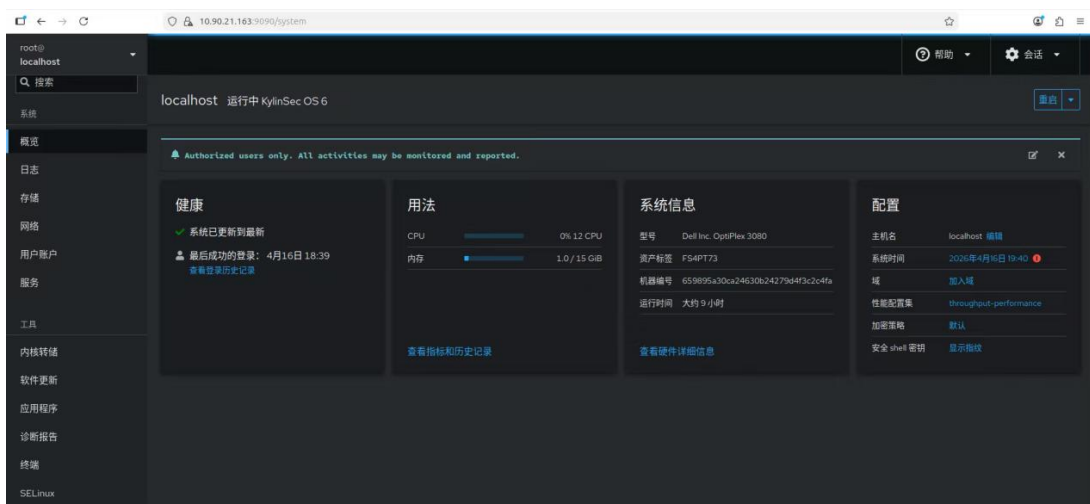


图 6-3 Cockpit 服务管理界面

6.3 一次性任务调度服务

atd : 为用户设置的“一次性闹钟”的系统服务, 可以安排在某个特定时间

点自动执行某个任务，如自动关机、发送提醒或运行脚本，且只执行一次，如图

6-4:

```
[root@localhost ~]# systemctl start atd.service
[root@localhost ~]# systemctl status atd.service
● atd.service - Deferred execution scheduler
   Loaded: loaded (/usr/lib/systemd/system/atd.service; enabled; preset: enabled)
   Active: active (running) since Thu 2026-04-16 10:47:53 CST; 15min ago
     Docs: man:atd(8)
    Main PID: 11355 (atd)
      Tasks: 1 (limit: 8714)
     Memory: 232.0K ( )
    CGroup: /system.slice/atd.service
            └─11355 /usr/sbin/atd -f

Apr 16 10:47:53 localhost.localdomain systemd[1]: Started Deferred execution scheduler.
Apr 16 10:47:53 localhost.localdomain (atd)[11355]: atd.service: Referenced but unset environment variable evaluates to an empty string: OPTS
Apr 16 10:49:00 localhost.localdomain atd[11558]: Starting job 1 (a0000101c3bd09) for user 'root' (0)
Apr 16 10:49:00 localhost.localdomain atd[11558]: pam_unix(atd:session): session opened for user root(uid=0) by root(uid=0)
Apr 16 10:49:00 localhost.localdomain atd[11558]: pam_unix(atd:session): session closed for user root
[root@localhost ~]# systemctl stop atd.service
[root@localhost ~]# systemctl status atd.service
○ atd.service - Deferred execution scheduler
   Loaded: loaded (/usr/lib/systemd/system/atd.service; enabled; preset: enabled)
   Active: inactive (dead) since Thu 2026-04-16 11:03:28 CST; 1s ago
     Docs: man:atd(8)
    Duration: 15min 34.788s
    Process: 11355 ExecStart=/usr/sbin/atd -f $OPTS (code=exited, status=0/SUCCESS)
    Main PID: 11355 (code=exited, status=0/SUCCESS)

Apr 16 10:47:53 localhost.localdomain systemd[1]: Started Deferred execution scheduler.
Apr 16 10:47:53 localhost.localdomain (atd)[11355]: atd.service: Referenced but unset environment variable evaluates to an empty string: OPTS
Apr 16 10:49:00 localhost.localdomain atd[11558]: Starting job 1 (a0000101c3bd09) for user 'root' (0)
Apr 16 10:49:00 localhost.localdomain atd[11558]: pam_unix(atd:session): session opened for user root(uid=0) by root(uid=0)
Apr 16 10:49:00 localhost.localdomain atd[11558]: pam_unix(atd:session): session closed for user root
Apr 16 11:03:28 localhost.localdomain systemd[1]: Stopping Deferred execution scheduler...
Apr 16 11:03:28 localhost.localdomain systemd[1]: atd.service: Deactivated successfully.
Apr 16 11:03:28 localhost.localdomain systemd[1]: Stopped Deferred execution scheduler.
```

图 6-4 一次性任务调度服务

6.4 存储管理库

libstoragemgmt: 它为你提供了一个管理界面，你可以用命令和方式来管理存储设备，如图 6-5 和图 6-6:

```
[root@localhost ~]# systemctl status libstoragemgmt
● libstoragemgmt.service - libstoragemgmt plug-in server daemon
   Loaded: loaded (/usr/lib/systemd/system/libstoragemgmt.service; enabled; preset: enabled)
   Active: active (running) since Thu 2026-04-16 17:04:33 CST; 5h 59min left
    Main PID: 10772 (lsmd)
      Tasks: 1 (limit: 8714)
     Memory: 272.0K ( )
    CGroup: /system.slice/libstoragemgmt.service
            └─10772 /usr/bin/lsmd -d

Apr 16 17:04:33 localhost systemd[1]: Started libstoragemgmt plug-in server daemon.
[root@localhost ~]# systemctl stop libstoragemgmt
[root@localhost ~]# systemctl status libstoragemgmt
○ libstoragemgmt.service - libstoragemgmt plug-in server daemon
   Loaded: loaded (/usr/lib/systemd/system/libstoragemgmt.service; enabled; preset: enabled)
   Active: inactive (dead) since Thu 2026-04-16 11:05:40 CST; 1s ago
    Process: 10772 ExecStart=/usr/bin/lsmd -d (code=exited, status=0/SUCCESS)
    Main PID: 10772 (code=exited, status=0/SUCCESS)

Apr 16 17:04:33 localhost systemd[1]: Started libstoragemgmt plug-in server daemon.
Apr 16 11:05:40 localhost.localdomain systemd[1]: Stopping libstoragemgmt plug-in server daemon...
Apr 16 11:05:40 localhost.localdomain systemd[1]: libstoragemgmt.service: Deactivated successfully.
Apr 16 11:05:40 localhost.localdomain systemd[1]: Stopped libstoragemgmt plug-in server daemon.
[root@localhost ~]# systemctl start libstoragemgmt
[root@localhost ~]# systemctl status libstoragemgmt
● libstoragemgmt.service - libstoragemgmt plug-in server daemon
   Loaded: loaded (/usr/lib/systemd/system/libstoragemgmt.service; enabled; preset: enabled)
   Active: active (running) since Thu 2026-04-16 11:05:58 CST; 1s ago
    Main PID: 11692 (lsmd)
      Tasks: 1 (limit: 8714)
     Memory: 180.0K ( )
    CGroup: /system.slice/libstoragemgmt.service
            └─11692 /usr/bin/lsmd -d

Apr 16 11:05:58 localhost.localdomain systemd[1]: Started libstoragemgmt plug-in server daemon.
[root@localhost ~]#
```

图 6-5 存储管理库

```
root@localhost profile.d# export LSCSI_URL="sim://"
root@localhost profile.d# lsmcli -u sim:// plugin-info
Description: Storage simulator Version: 1.9.8
root@localhost profile.d# lsmcli -u sim:// list --type systems
ID      Name      Status      Info      PD Ver      Mode      Read Cache Percentage
-----
sim-01  LSI simulated storage plug-in  OK          LSI_SIMULATOR_DATA_4.1  HW RAID: 10

root@localhost profile.d# lsmcli -u sim:// list --type pools
ID      Name      Element Type      Total Space      Free Space      Status      Info      System ID
-----
POOL_ID_00001  Pool 1      POOL_VOLUME_FS.SYSTEM_RESERVED.DELTA  2199823255552    1649267441664    OK          1      sim-01
POOL_ID_00002  Pool 2(sub pool of Pool 1)  VOLUME_FS.DELTA  549755013088    549755013088    OK          1      sim-01
POOL_ID_00003  Pool 3      VOLUME_FS.DELTA  549755013088    549755013088    OK          1      sim-01
POOL_ID_00004  lsm_test_aggr  VOLUME_FS.DELTA  2306043009213699952  2306043009213699952  OK          1      sim-01

root@localhost profile.d# lsmcli -u sim:// list --type disks
ID      Name      Type      Size      Status      System ID      SCSI ID      Bb33      Disk Paths      Revolutions Per Minute      Link Type
-----
DISK_ID_00001  2T10 SATA Disk 1  SATA  2199823255552    OK          sim-01      50006f1d4ed1432f  7200  /dev/sda  15000  SATA/SATA
DISK_ID_00002  2T10 SATA Disk 2  SATA  2199823255552    OK          sim-01      50afed120dfc4a02  7200  /dev/sdb  15000  SATA/SATA
DISK_ID_00003  2E10 SAS Disk 3  SAS  1152921504606046976  OK          sim-01      50b3c51afac4b33a  15000  /dev/sdc  15000  SAS
DISK_ID_00004  2E10 SAS Disk 4  SAS  1152921504606046976  OK          sim-01      5004f1f0eac59077  15000  /dev/sdd  15000  SAS
DISK_ID_00005  2E10 SAS Disk 5  SAS  1152921504606046976  OK          sim-01      509c0b449b0b6310  15000  /dev/sde  15000  SAS
DISK_ID_00006  2E10 SAS Disk 6  SAS  1152921504606046976  OK          sim-01      50c60316f7067bdf  15000  /dev/sdf  15000  SAS
DISK_ID_00007  2E10 SAS Disk 7  SAS  1152921504606046976  OK          sim-01      50d297435003af79  15000  /dev/sdg  15000  SAS
DISK_ID_00008  2E10 SAS Disk 8  SAS  1152921504606046976  OK          sim-01      50b1d03231196cf7  15000  /dev/sdh  15000  SAS
DISK_ID_00009  512GiB SSD Disk 9  SSD  549755013088    OK          sim-01      50d539e9009aa7e6  Non-Rotating Medium  /dev/sdi  15000  SATA/SATA
DISK_ID_00010  512GiB SSD Disk 10  SSD  549755013088    OK          sim-01      5046a03f4d01234  Non-Rotating Medium  /dev/sdj  15000  SATA/SATA
DISK_ID_00011  512GiB SSD Disk 11  SSD  549755013088    OK          sim-01      50e5e54f1d0bc25  Non-Rotating Medium  /dev/sdk  15000  SATA/SATA
DISK_ID_00012  512GiB SSD Disk 12  SSD  549755013088    OK          sim-01      504276f0e4c5b7c5  Non-Rotating Medium  /dev/sdl  15000  SATA/SATA
DISK_ID_00013  512GiB SSD Disk 13  SSD  549755013088    OK          sim-01      5032f244f4191540  Non-Rotating Medium  /dev/sdm  15000  SATA/SATA
DISK_ID_00014  2T10 SSD Disk 14  SSD  2199823255552    OK          sim-01      50652a4e31e20345  Non-Rotating Medium  /dev/sdn  15000  SAS
DISK_ID_00015  2T10 SSD Disk 15  SSD  2199823255552    OK          sim-01      50047f6b2a1d1809  Non-Rotating Medium  /dev/sdo  15000  SAS
DISK_ID_00016  2T10 SSD Disk 16  SSD  2199823255552    OK          sim-01      504cc2f5962a5e43  Non-Rotating Medium  /dev/sdp  15000  SAS
DISK_ID_00017  2T10 SSD Disk 17  SSD  2199823255552    OK          sim-01      5073a7965076ce52  Non-Rotating Medium  /dev/sdq  15000  SAS
DISK_ID_00018  2T10 SSD Disk 18  SSD  2199823255552    OK          sim-01      50d34402723ac0d4  Non-Rotating Medium  /dev/sdr  15000  SAS
DISK_ID_00019  2T10 SSD Disk 19  SSD  2199823255552    OK          sim-01      500a3c596760a3cd  Non-Rotating Medium  /dev/sds  15000  SAS
DISK_ID_00020  2T10 SSD Disk 20  SSD  2199823255552    OK          sim-01      5066e102c934ff1  Non-Rotating Medium  /dev/sdt  15000  SAS

root@localhost profile.d#
```

图 6-6 存储管理库状态

6.5 授权框架

polkit: 它在你执行需要管理员权限的操作，比如修改系统设置时，询问你的密码进行授权，在便利和安全之间取得平衡。服务运行状态如图 6-7:

```
root@localhost profile.d# busctl list | grep polkit
:1.3                               1178 polkitd polkitd :1.3 polkit.service -
org.freedesktop.PolicyKit1         1178 polkitd polkitd :1.3 polkit.service -

root@localhost profile.d# ls -la /etc/polkit-1/rules.d/
total 24
drwxr-xr-x. 2 polkitd root 4096 Apr 16 06:34 .
drwxr-xr-x. 5 root root 4096 Apr 16 06:34 ..
-rw-r--r--. 1 root root 974 Jan 4 09:28 49-polkit-pkla-compat.rules
root@localhost profile.d# ls -la /usr/share/polkit-1/rules.d/
total 40
drwxr-xr-x. 2 polkitd root 4096 Apr 16 06:35 .
drwxr-xr-x. 4 root root 4096 Apr 16 06:34 ..
-rw-r--r--. 1 root root 326 Jul 28 2023 50-default.rules
-rw-r--r--. 1 root root 369 Jan 2 12:52 org.freedesktop.bolt.rules
-rw-r--r--. 1 root root 257 Nov 9 2023 org.freedesktop.packagekit.rules
root@localhost profile.d# systemctl start polkit
root@localhost profile.d# systemctl status polkit
● polkit.service - Authorization Manager
   Loaded: loaded (/usr/lib/systemd/system/polkit.service; static)
   Active: active (running) since Thu 2026-04-16 17:04:37 CST; 3h 24min left
     Docs: man:polkit(8)
   Main PID: 1178 (polkitd)
    Tasks: 4 (limit: 8714)
   Memory: 2.7M ( )
   CGroup: /system.slice/polkit.service
           └─1178 /usr/lib/polkit-1/polkitd --no-debug

Apr 16 17:04:36 localhost systemd[1]: Starting Authorization Manager...
Apr 16 17:04:36 localhost polkitd[1178]: Started polkitd version 123
Apr 16 17:04:37 localhost polkitd[1178]: Loading rules from directory /etc/polkit-1/rules.d
Apr 16 17:04:37 localhost polkitd[1178]: Loading rules from directory /usr/share/polkit-1/rules.d
Apr 16 17:04:37 localhost polkitd[1178]: Finished loading, compiling and executing 4 rules
Apr 16 17:04:37 localhost polkitd[1178]: Acquired the name org.freedesktop.PolicyKit1 on the system bus
Apr 16 17:04:37 localhost systemd[1]: Started Authorization Manager.
root@localhost profile.d#
```

图 6-7 授权框架

6.6 RPC 端口映射

rpcbind: 它是网络服务的“电话总机”，帮助客户端程序准确找到服务器

上对应服务的“分机号码”，确保远程访问能正常连接。如图 6-8 和图 6-9：

```
[root@localhost profile.d]# systemctl start rpcbind
[root@localhost profile.d]# systemctl status rpcbind
● rpcbind.service - RPC Bind
   Loaded: loaded (/usr/lib/systemd/system/rpcbind.service; enabled; preset: enabled)
   Active: active (running) since Thu 2026-04-16 17:04:15 CST; 3h 22min left
   TriggeredBy: ● rpcbind.socket
     Docs: man:rpcbind(8)
    Main PID: 1031 (rpcbind)
      Tasks: 1 (limit: 8714)
     Memory: 2.1M ()
    CGroup: /system.slice/rpcbind.service
           └─1031 /usr/bin/rpcbind -r -w -f

Apr 16 17:04:15 localhost systemd[1]: Starting RPC Bind...
Apr 16 17:04:15 localhost systemd[1]: Started RPC Bind.
[root@localhost profile.d]# systemctl stop rpcbind
Stopping 'rpcbind.service', but its triggering units are still active:
rpcbind.socket
[root@localhost profile.d]# systemctl status rpcbind
○ rpcbind.service - RPC Bind
   Loaded: loaded (/usr/lib/systemd/system/rpcbind.service; enabled; preset: enabled)
   Active: inactive (dead) since Thu 2026-04-16 13:42:25 CST; 15s ago
   TriggeredBy: ● rpcbind.socket
     Docs: man:rpcbind(8)
   Process: 1031 ExecStart=/usr/bin/rpcbind $RPCBIND_OPTIONS -w -f (code=exited, status=0/SUCCESS)
   Main PID: 1031 (code=exited, status=0/SUCCESS)

Apr 16 17:04:15 localhost systemd[1]: Starting RPC Bind...
Apr 16 17:04:15 localhost systemd[1]: Started RPC Bind.
Apr 16 13:42:25 localhost.localdomain systemd[1]: Stopping RPC Bind...
Apr 16 13:42:25 localhost.localdomain systemd[1]: rpcbind.service: Deactivated successfully.
Apr 16 13:42:25 localhost.localdomain systemd[1]: Stopped RPC Bind.
[root@localhost profile.d]# systemctl start rpcbind
[root@localhost profile.d]#
```

图 6-8 RPC 端口映射状态

```
[root@localhost profile.d]# rpcinfo -p localhost
program vers proto port service
100000 4 tcp 111 portmapper
100000 3 tcp 111 portmapper
100000 2 tcp 111 portmapper
100000 4 udp 111 portmapper
100000 3 udp 111 portmapper
100000 2 udp 111 portmapper
[root@localhost profile.d]# netstat -tlnp | grep 111
tcp 0 0 0.0.0.0:111 0.0.0.0:* LISTEN 14895/rpcbind
tcp6 0 0 :::111 :::* LISTEN 14895/rpcbind
[root@localhost profile.d]# ss -tlnp | grep 111
LISTEN 0 4096 0.0.0.0:111 0.0.0.0:* users:(("rpcbind",pid=14895,fd=8))
LISTEN 0 4096 [::]:111 [::]:* users:(("rpcbind",pid=14895,fd=11))
[root@localhost profile.d]# rpcinfo -e localhost
rpcinfo: invalid option -- 'e'
Usage: rpcinfo [-m | -s] [host]
       rpcinfo -p [host]
       rpcinfo -T netid host prognum [versnum]
       rpcinfo -l host prognum versnum
       rpcinfo [-n portnum] -u | -t host prognum [versnum]
       rpcinfo -a serv_address -T netid prognum [version]
       rpcinfo -b prognum versnum
       rpcinfo -d [-T netid] prognum versnum
[root@localhost profile.d]# rpcinfo -s localhost
program version(s) netid(s) service owner
100000 2,3,4 local,udp,tcp,udp6,tcp6 portmapper superuser
[root@localhost profile.d]#
```

图 6-9 RPC 端口映射

7 虚拟化主机

虚拟化主机形态是一种专门用于运行和管理虚拟机的操作系统。它充当物理服务器与虚拟化环境之间的“桥梁”，将一台物理计算机的 CPU、内存、硬盘、网络等硬件资源进行抽象、分割与集中管理，分配给一个或多个相互隔离的虚拟机使用。

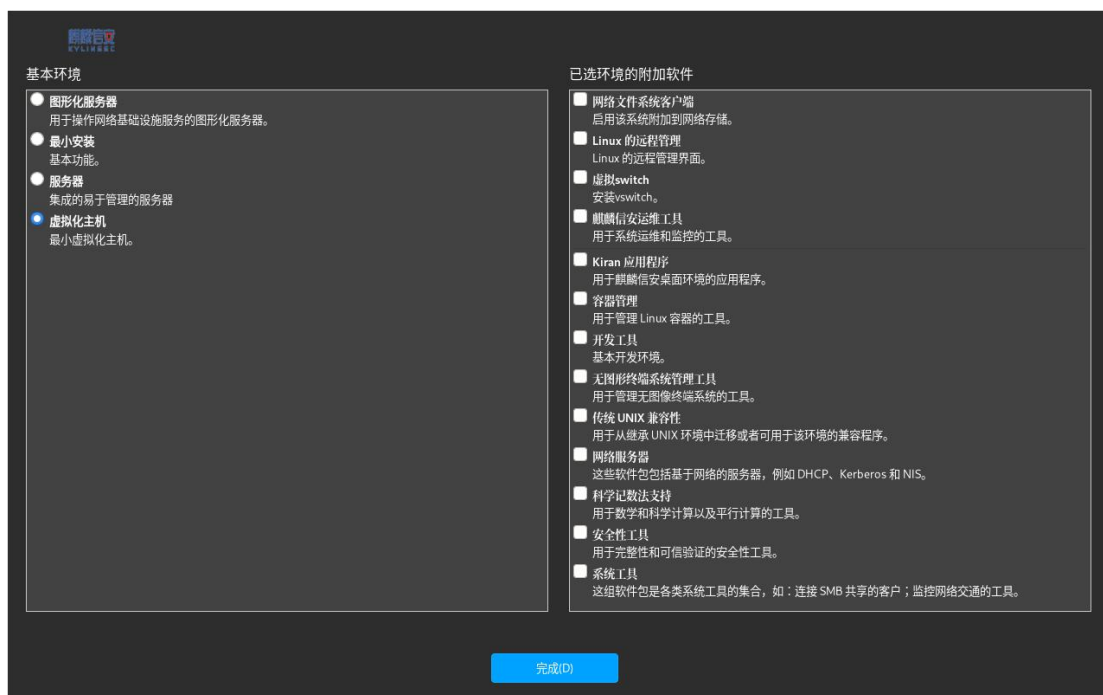


图 7-1 虚拟化主机安装界面

虚拟化主机形态在最小安装形态基础上主要提供了虚拟化环境，支持从 ISO 或磁盘镜像快速创建启动虚拟机，提供网络和存储特性，并配套命令行管理接口。这些功能让系统具备虚拟化的能力，用户可直接在宿主机上运行、管理多个独立虚拟机。

7.1 虚拟机镜像上传

用户可通过 libvirt 服务创建和管理虚拟机。创建虚拟机方法如下，首先开启虚拟化服务，在操作系统中上传一个需要的镜像，确认镜像完整正确，复制

镜像到对应文件夹 `cp CentOS-7.iso /var/lib/libvirt/images/`，如图 7-2：

```
[root@localhost ~]# ls
anaconda-ks.cfg  initial-setup-ks.cfg  KylinSec-Server-6-SP1-beta-260308-2027-x86_64.iso
[root@localhost ~]# cp KylinSec-Server-6-SP1-beta-260308-2027-x86_64.iso /var/lib/libvirt/images/
```

图 7-2 上传镜像

7.2 虚拟机创建

使用命令 `virt-install --name vm1 --memory 2048 --vcpus 2 --disk size=20 --cdrom /var/lib/libvirt/images/KylinSec-Server-xxx.iso --os-variant centos7.0 --graphics vnc,listen=0.0.0.0,port=5901` 来安装虚拟机，如图 7-3。安装好操作系统后需要关闭虚拟机，修改虚拟机 XML 配置文件以设置从硬盘启动，使用 `virsh edit` 命令，在配置 `<os></os>` 里面，找到修改配置 `<boot dev='hd'/>`，如图 7-4，重启虚拟机可以进入操作系统。

```
[root@localhost ~]# virt-install --name test-vm1 --ram 2048 --vcpus 4 --cdrom /var/lib/libvirt/images/KylinSec-Server-6-SP1-beta-260308-2027-x86_64.iso --disk path=/home/test-vm1,size=20 --graphics vnc,listen=0.0.0.0
WARNING: 无法连接到图形控制台，没有安装 virt-viewer，请安装 'virt-viewer' 软件包。
WARNING: 没有检测到用于启动客户机，默认为 --wait -1
开始安装.....
创建磁盘.....
0 B 00:00:00
虚拟机仍在运行，安装可能正在进行中。
请等待安装完成。
```

图 7-3 安装过程

```
<?xml version='1.0' type='text/xml'>
<domain type='kvm'>
  <name>test-vm</name>
  <uuid>a00090c-a097-4b79-b084-0063e4edf22a</uuid>
  <memory unit='KiB'>2097152</memory>
  <currentMemory unit='KiB'>2097152</currentMemory>
  <cpu placement='static'>2</cpu>
  <os>
    <type arch='x86_64' machine='pc-i440fx-8.2'>hvm</type>
    <boot dev='hd'>/>
  </os>
  <features>
    <acpi/>
    <apic/>
  </features>
  <cpu mode='host-passthrough' check='none' migratable='on'>
    <clock offset='utc'>
      <timer name='rtc' tickpolicy='catchup'>/>
      <timer name='pit' tickpolicy='delay'>/>
      <timer name='hpet' present='no'>/>
    </clock>
    <on_poweroff>destroy</on_poweroff>
    <on_reboot>restart</on_reboot>
    <on_crash>destroy</on_crash>
  </cpu>
  <pm>
    <suspend-to-mem enabled='no'>/>
    <suspend-to-disk enabled='no'>/>
  </pm>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='raw'>/>
      <source file='/var/lib/libvirt/images/test-vm.qcow2'>/>
      <target dev='vda' bus='virtio'>/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'>/>
    </disk>
    <disk type='file' device='cdrom'>
      <driver name='qemu' type='raw'>/>
      <source file='/var/lib/libvirt/images/KylinSec-Server-6-SP1-beta-260308-2027-x86_64.iso'>/>
      <target dev='hda' bus='ide'>/>
      <readonly/>
      <address type='drive' controller='0' bus='0' target='0' unit='0'>/>
    </disk>
    <controller type='usb' index='0' model='ich9-ehci1'>/>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'>/>
    </controller>
    <controller type='pci' index='0' model='pci-root'>/>
    <controller type='ide' index='0'>/>
    <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x0'>/>
    </controller>
    <interface type='network'>
      <mac address='52:54:00:98:73:0d'>/>
      <source network='default'>/>
      <model type='virtio'>/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'>/>
    </interface>
  </devices>
</domain>
```

图 7-4 虚拟机配置文件

7.3 vnc 控制虚拟机

输入命令后使用 vnc 服务远程连接虚拟机完成安装，如图 7-5

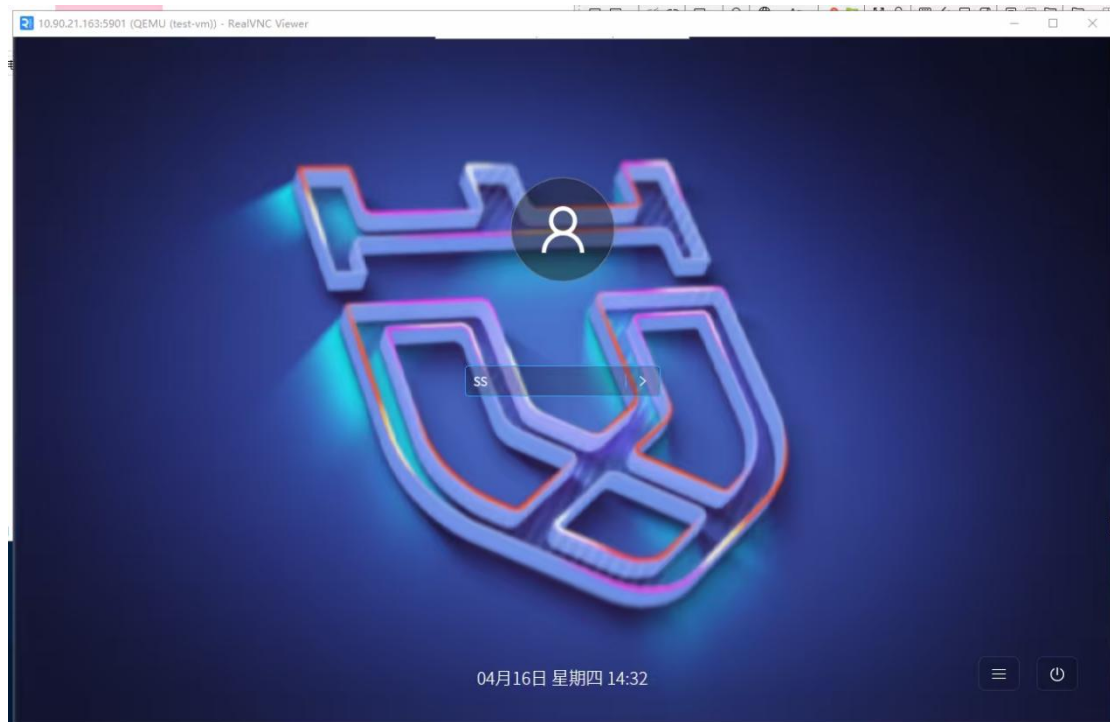


图 7-5 vnc 远程连接

7.4 虚拟机管理

安装完成后使用 `virsh list --all` 查看所有虚拟机状态，`virsh start vm1` 启动指定虚拟机，`virsh shutdown vm1` 关闭虚拟机，`virsh undefine vm1` 删除虚拟机配置，如图 7-6：

```
root@localhost:~  
[root@localhost ~]# virsh list --all  
Id    名称    状态  
-----  
7     test-vm  运行  
  
[root@localhost ~]# virsh shutdown test-vm  
域 'test-vm' 正在关闭  
  
[root@localhost ~]# virsh list --all  
Id    名称    状态  
-----  
-     test-vm  关闭  
  
[root@localhost ~]# virsh start test-vm
```

图 7-6 虚拟机列表

7.5 虚拟机网络管理

虚拟化形态还提供了桥接网络支持，使虚拟机与宿主机处于同一局域网，并集成 virt-top 工具实时监控虚拟机资源使用情况。使用 virsh net-list --all 列出所有虚拟网络，virsh net-start <网络名> 启动一个未激活的虚拟网络，virsh net-autostart <网络名>- 设置虚拟网络随系统自动启动，还可以使用 virsh net-info <网络名>- 查看指定虚拟网络的详细信息。

7.6 虚拟机其他配置

若用户在安装虚拟机后要求使用 vnc，需要修改 vnc 配置，使用 virsh edit test-vm 命令，找到配置<serial></serial>中的<graphics>,修改为如图图 7-7 所示配置，放行端口和放行 ip 可以自定义。修改后需要重启虚拟机，并且在主机放行对应端口，关闭防火墙。使用 virsh snapshot-create-as 创建快照以便回滚，支持 virsh console 通过串口登录虚拟机。

```
<address type='usb' bus='0' port='1' />
</input>
<input type='mouse' bus='ps2' />
<input type='keyboard' bus='ps2' />
<graphics type='vnc' port='5901' autoport='no' listen='0.0.0.0'>
  <listen type='address' address='0.0.0.0' />
</graphics>
```

图 7-7 修改 vnc 配置

8 图形化服务器

图形化服务器形态是在服务器操作系统基础上,集成了图形用户界面的完整版本。它在保留服务器核心功能的同时,提供了可视化操作环境,适合作为日常办公、开发调试及图形交互的主机环境。

8.1 图形化桌面

麒麟信安服务器操作系统的 KiranUI 图形化桌面由湖南麒麟信安科技股份有限公司主导开发。KiranUI 桌面由文件管理器、窗口管理器、任务栏、控制面板,以及一些图形工具等组成。给用户提供一个易于使用、开放、轻巧快速的桌面体验,同时也保留了充分的可定制性。桌面展示如图 8-1

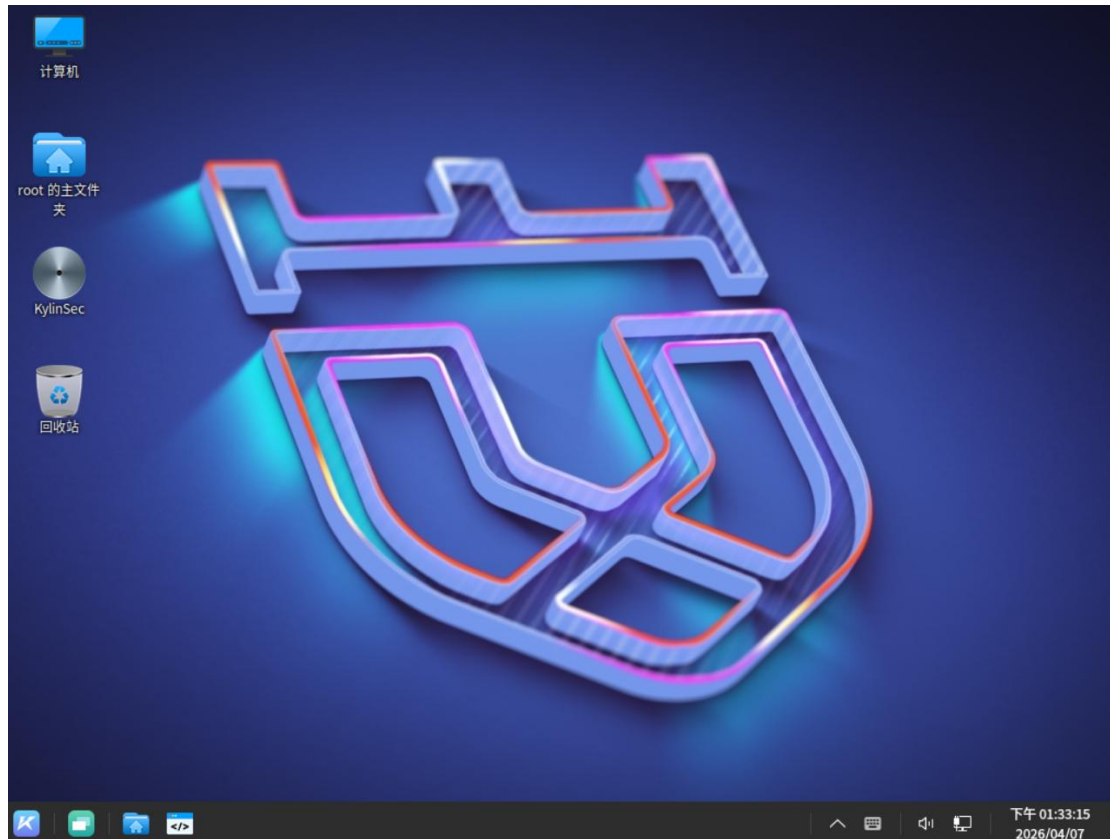


图 8-1 系统桌面

8.1.1 桌面导视图

桌面是用户与计算机交互的区域，桌面上显示基础图标计算机、主目录文件夹和回收站，可以快速双击打开。桌面下方区域是系统面板，包括了任务栏、托盘区域和日期与时间。如图 8-2 所示：

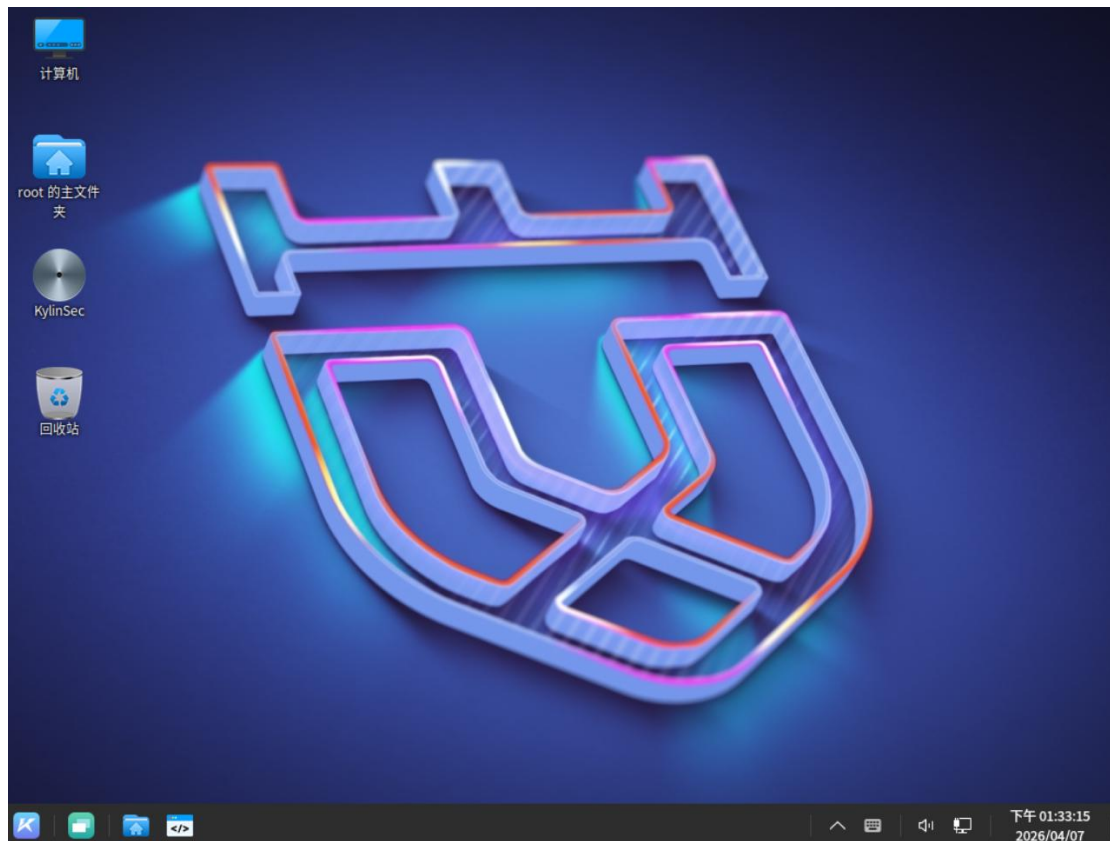


图 8-2 系统桌面导视图

计算机：双击可以显示从本计算机访问的所有本地和远程磁盘和文件夹。

主文件夹：双击可以显示用户家目录下的内容。

回收站：存放已删除的文件。

在桌面上，点击鼠标右键进行系统常规功能操作如图标管理、创建文件夹、创建文档、桌面背景、主题、等快捷方式。具体如下表 8-1：

表 8-1 桌面右键菜单表

名称	描述
创建文件夹	创建一个新的文件夹
创建启动器	创建一个新的启动器
创建文档	创建一个空的纯文本文档

在终端中打开	在当前桌面目录下直接打开终端应用
刷新	可以刷新整个桌面
按名称整理桌面	可以通过名称来排序桌面文件
按修改时间组织桌面	可以通过你修改的时间来排序桌面文件
按大小组织桌面	可以通过文件大小来排序桌面文件
按类型组织桌面	按类型来排序桌面文件
保持网格排列对齐	勾选了保持对齐，桌面图标会按照网格对齐排列
锁定图标位置	勾选了锁定图标位置，会将桌面图标位置进行锁定
图标大小	设置桌面图标为大图标、中等图标、小图标
粘贴	粘贴复制的内容
个性化设置	打开个性化设置工具，可设置主题、字体和壁纸

系统面板位于桌面下方区域，包括了任务栏、托盘区域和日期与时间。任务栏用于查看系统启动应用，默认放置开始菜单、文件管理器、工作区，在任务栏可以进行应用程序打开、关闭、放大、最小化等操作。托盘区域可以设置输入法、调节音量、设置网络，托盘区域右边显示日期和时间。如下图 8-3 所示：

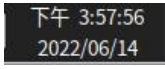


图 8-3 系统面板导视图

系统面板区域图标和说明如下表 8-2 所示：

表 8-2 系统面板说明表

图标	名称	描述
----	----	----

	开始菜单	单击会弹出系统级联的菜单，显示系统应用
	工作区	单击显示系统工作区，可新增和切换工作区
	文件浏览器	文件夹、文件管理工具
	终端	单击将启动终端，可输入指令操作
	输入法	设置输入法，切换语言
	声音	调节声音大小
	网络设置	显示当前网络状态，可设置网络连接
	时间和日期	显示当前时间和日期，点击可打开设计和日期设置工具
	显示桌面	位于桌面最右下角，单击可将所有应用最小化，快速回到桌面

8.1.2 电源管理

系统提供了电源管理功能，可以锁定屏幕、切换用户、注销、重启和关机。

点击“开始菜单”>“电源选项”进入电源选项设置，如图 8-4 所示：

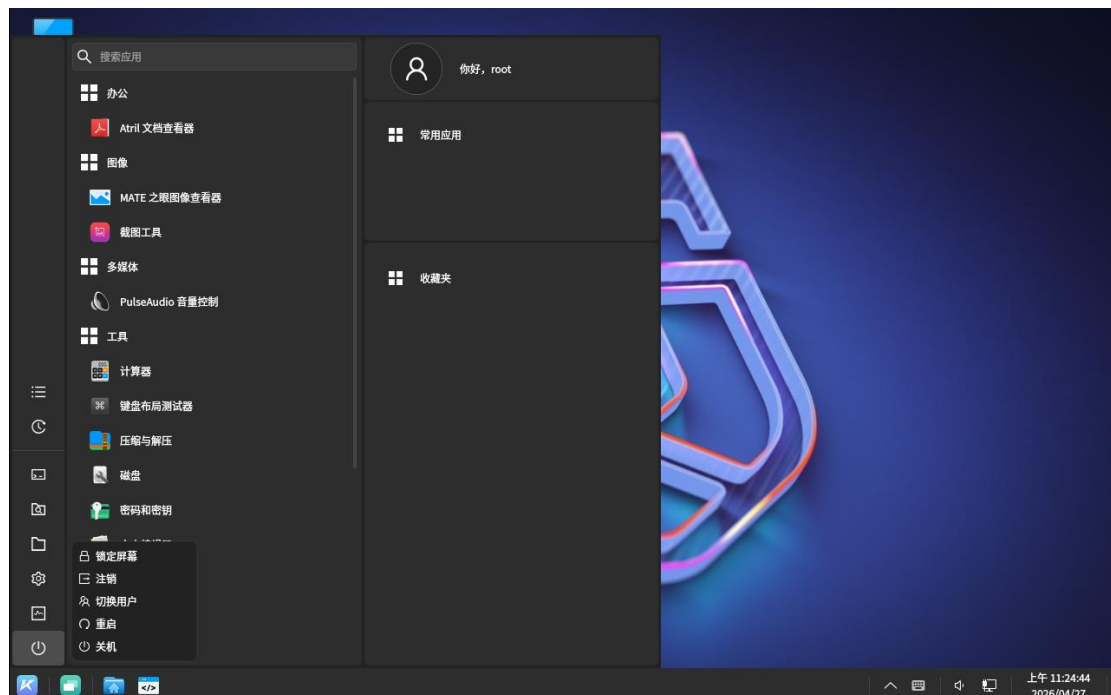


图 8-4 电源按钮

电源管理中各个选项的说明如下表 8-3 所示：

表 8-3 电源管理说明表

名称	描述
锁定屏幕	系统进入锁屏状态，显示锁屏界面，输入密码可登录
切换用户	切换登录其他用户，之前的用户信息依然保留
注销	注销当前用户退出到登录界面，用户信息不会保留
重启	计算机重启
关机	计算机关机

8.1.3 消息

麒麟信安服务器操作系统提供了消息提示功能，默认会在桌面右上角弹出消息提示框。如：系统网络连接成功时会弹出信息提示框，如图 8-5 所示：

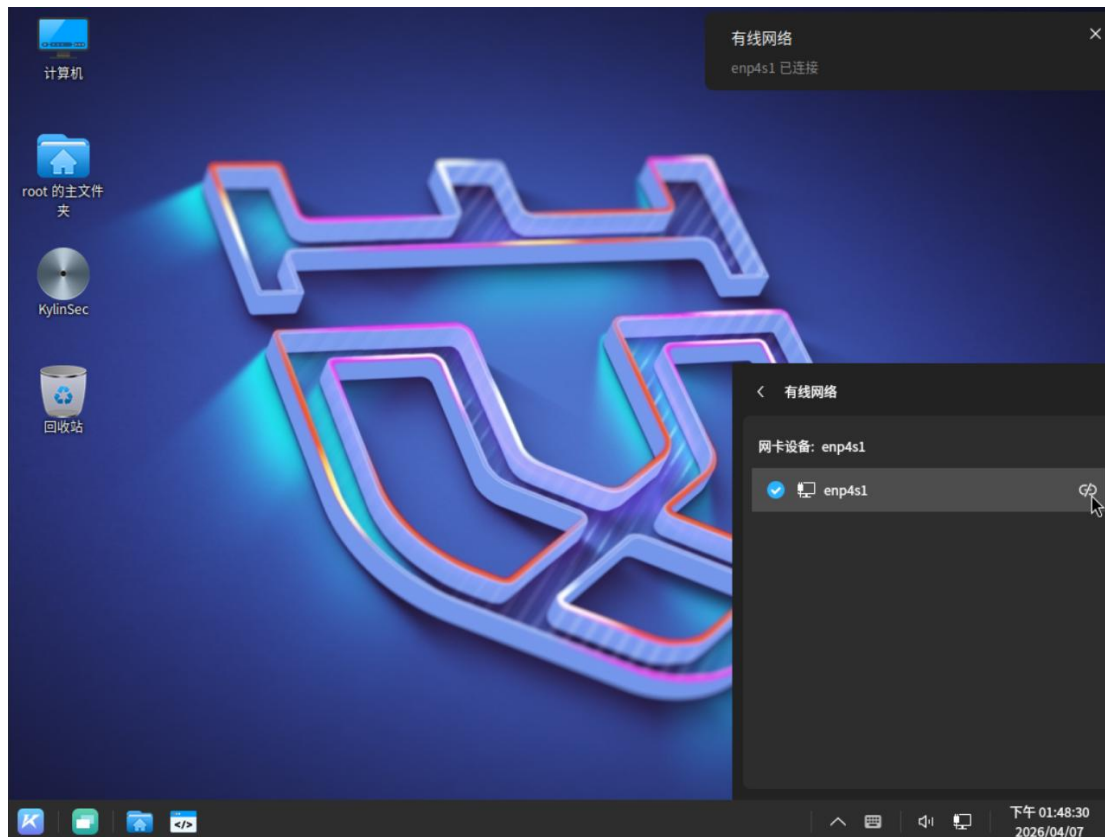


图 8-5 系统弹出提示弹窗

8.2 控制面板

控制面板是麒麟信安服务器操作系统中提供的一个高度集成的图形化配置环境，几乎包含所有的配置和管理工具，包括桌面定制、系统配置管理工具以及网络服务配置工具等。它主要包括以下功能：

- 1) 执行系统配置和管理任务；
- 2) 运行网络服务配置；
- 3) 定制具有个人特色的桌面环境。

点击“开始菜单”>“控制面板”图标，打开控制面板对话框，如下图 8-6 所示：

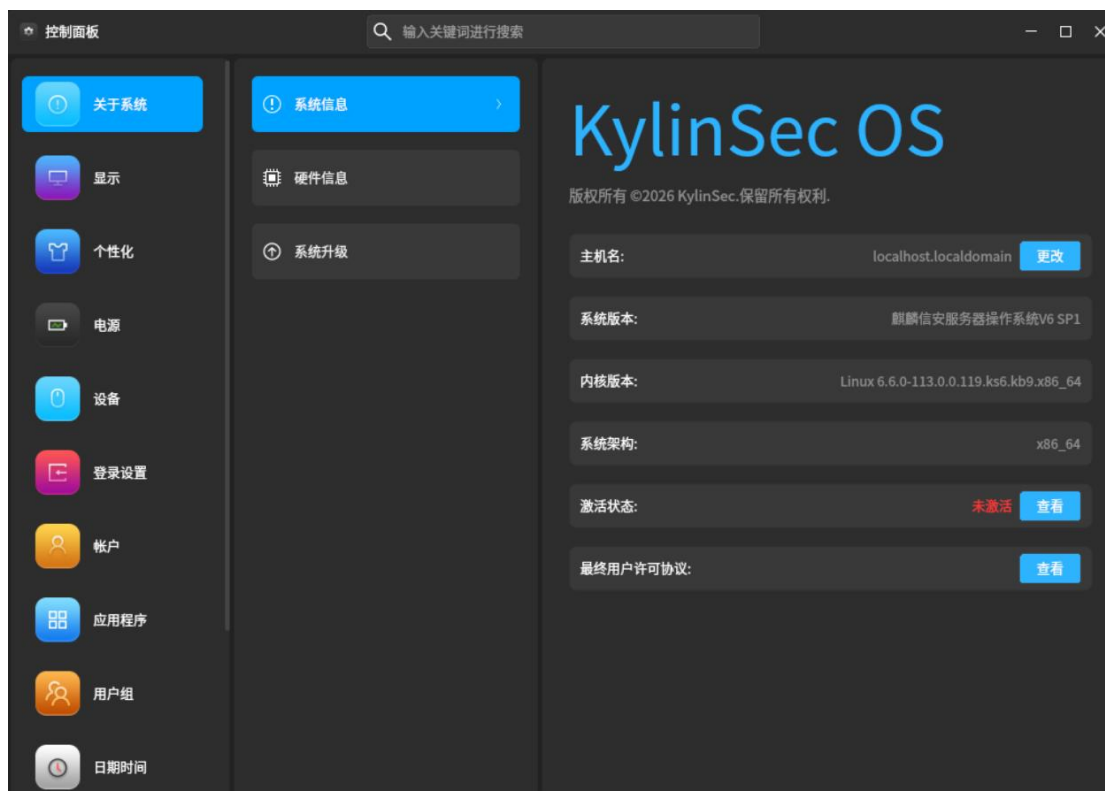


图 8-6 控制面板界面

8.2.1 关于系统

8.2.1.1 系统信息

系统信息工具中包括了系统信息和硬件信息内容,其中系统信息显示了当前系统的主机名、系统版本、内核版本、系统架构、激活状态、最终用户许可协议信息。用户可点击查看和导出许可协议和版本协议。硬件信息显示了计算机 CPU、内存、硬盘、显卡和网卡信息。

点击“控制面板”的“系统信息”打开工具,如下图 8-7 所示:

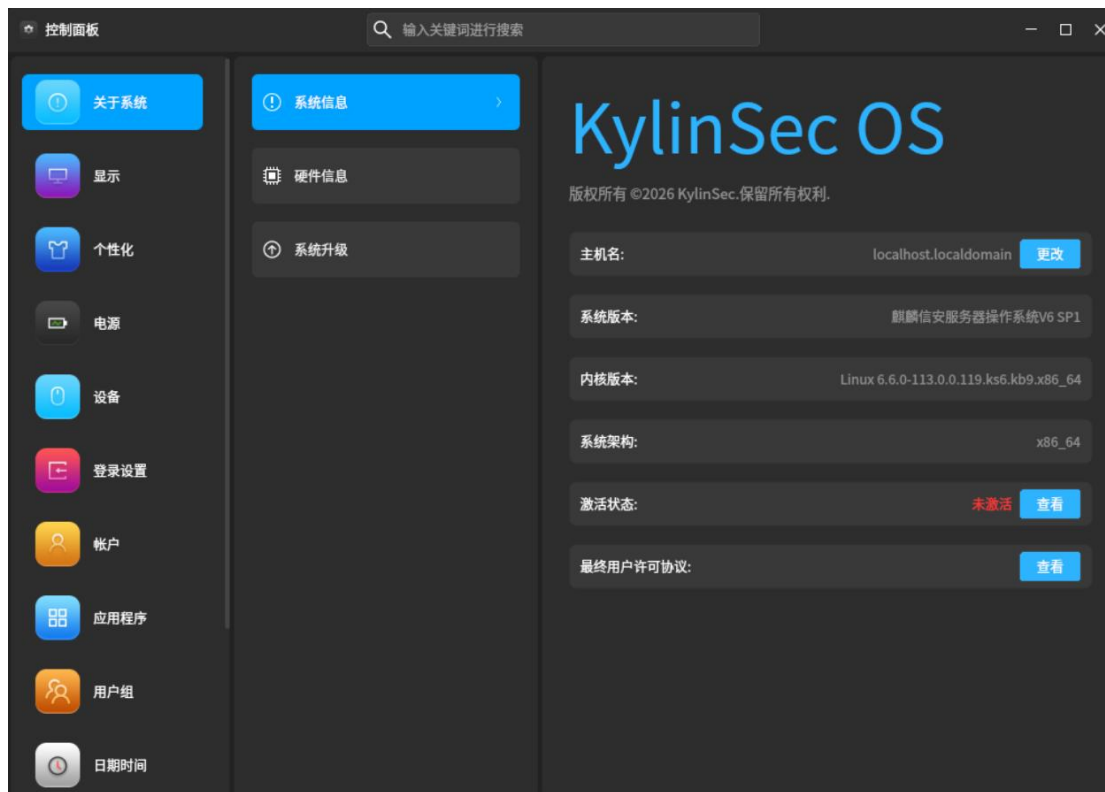


图 8-7 系统信息

点击“更改”按钮后输入主机名可更改主机名。

点击“查看”按钮可分别查看最终用户许可协议，可以导出最终用户许可协议。

8.2.1.1.1 系统激活

激活系统有四种方式：激活码激活方式、插入 UsbKey 设备激活、在线激活和文件激活。

点击“系统信息”>激活状态点击“查看”>“激活”，打开激活工具，如下图 8-8 所示：

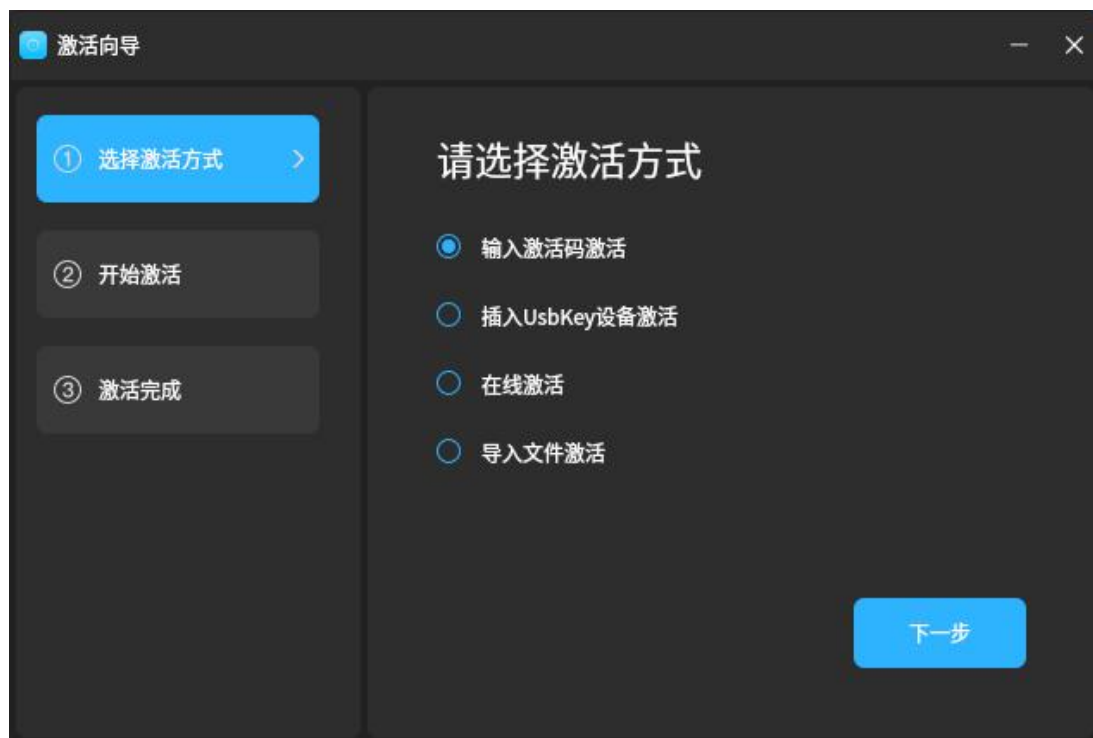


图 8-8 系统激活

输入激活码激活：提供机器码向麒麟信安科技股份有限公司获取授权码，选中“输入激活码激活”>“下一步”，输入激活码后，点击“激活”，即可激活成功。如图 8-9 和图 8-10 所示：

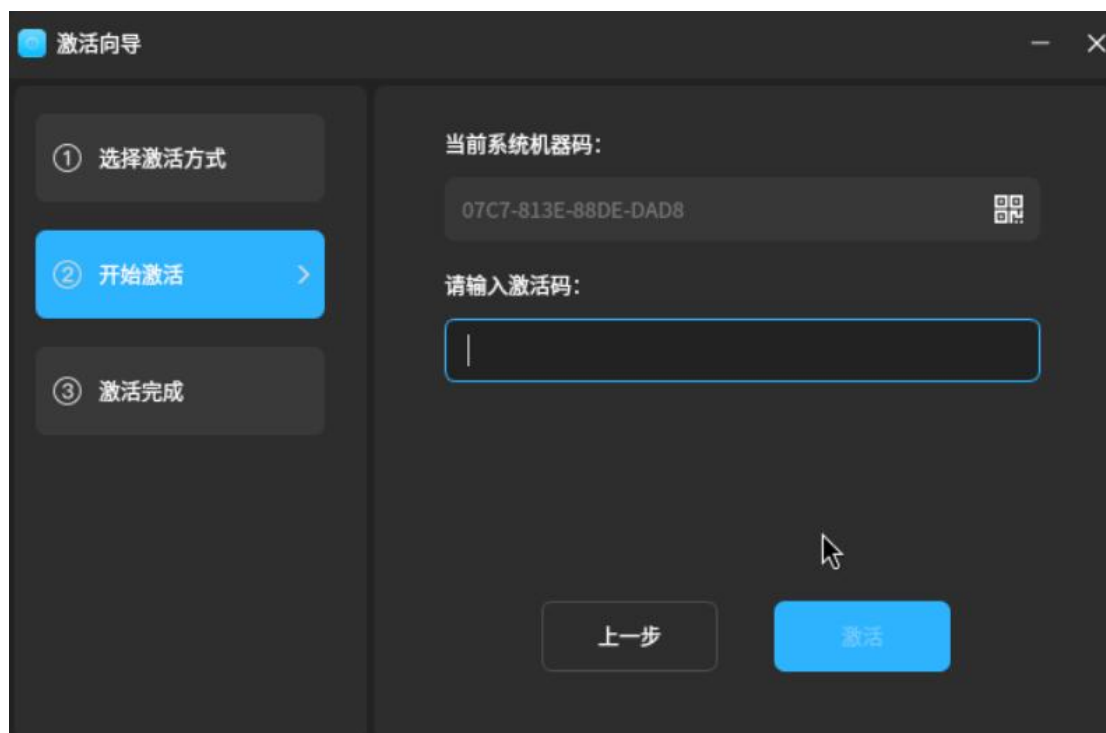


图 8-9 输入激活码激活

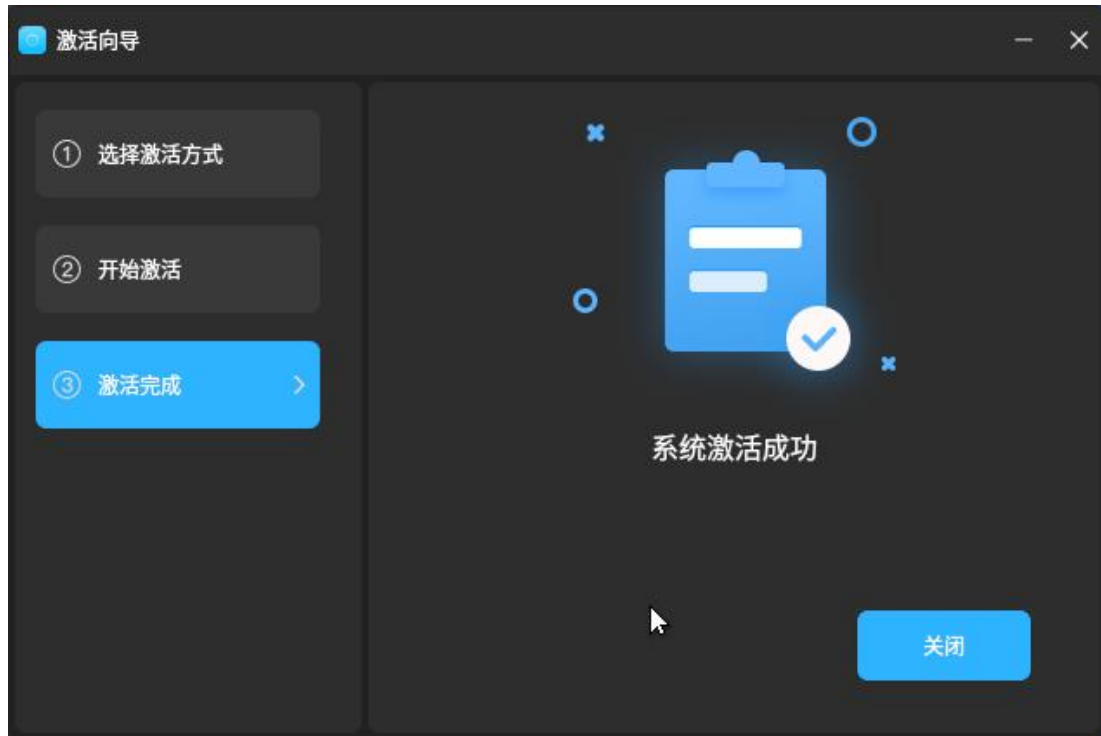


图 8-10 激活成功

激活后的系统信息界面如下图 8-11 所示：

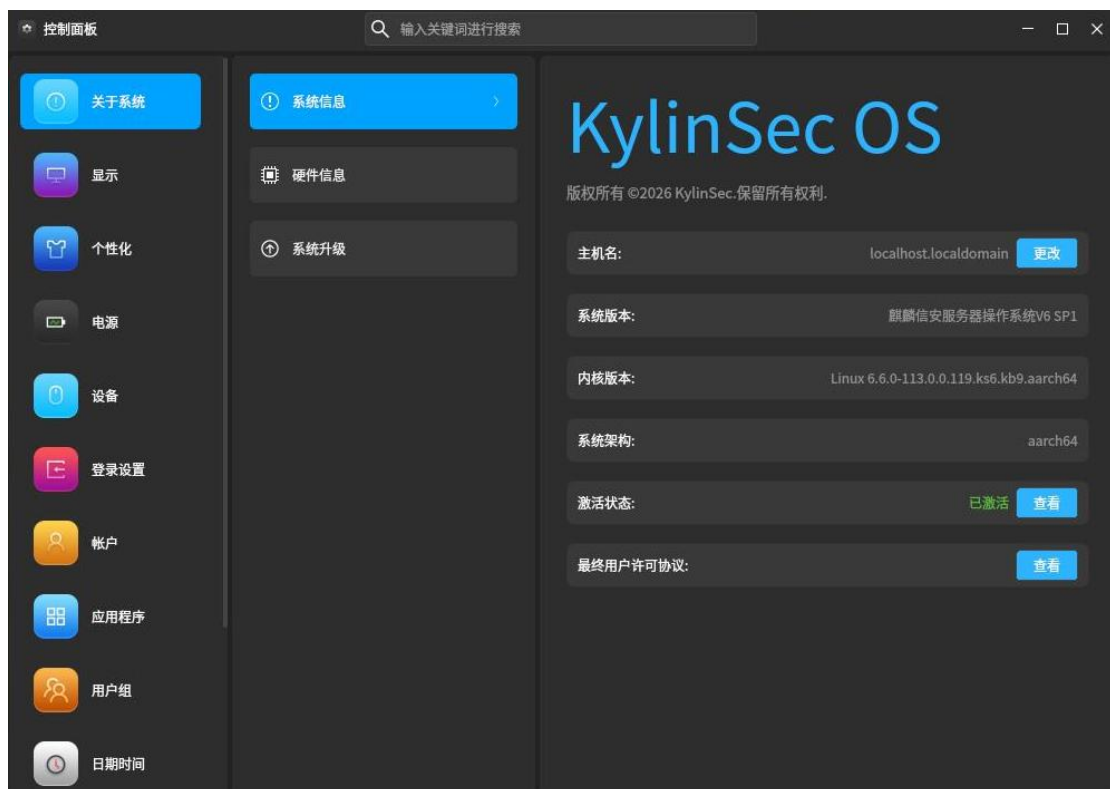


图 8-11 激活后的系统信息界面

插入 UsbKey 设备激活：在待激活计算机上插入 UsbKey 设备，选中“插入 UsbKey 设备激活”>“下一步”>“激活”即可激活成功，如图 8-12：

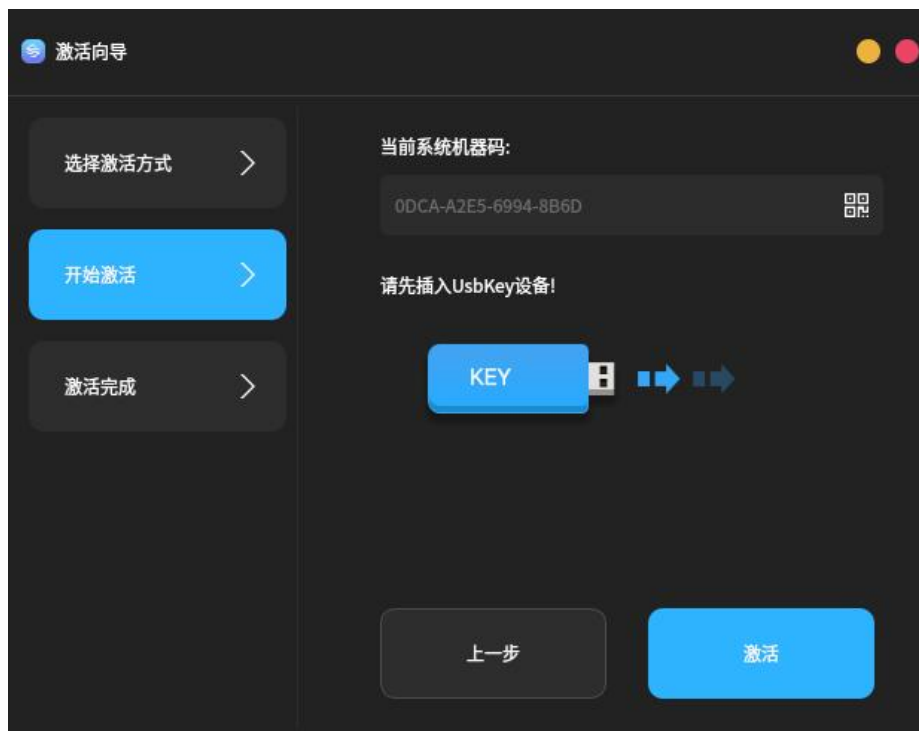


图 8-12 插入 UsbKey 激活

在线激活：系统提供了在线激活方式，首先在服务器上搭建在线激活服务器，配置好 IP。选中“在线激活”>“下一步”>输入服务器 IP 和服务号>“激活”即可，如图 8-13 所示：

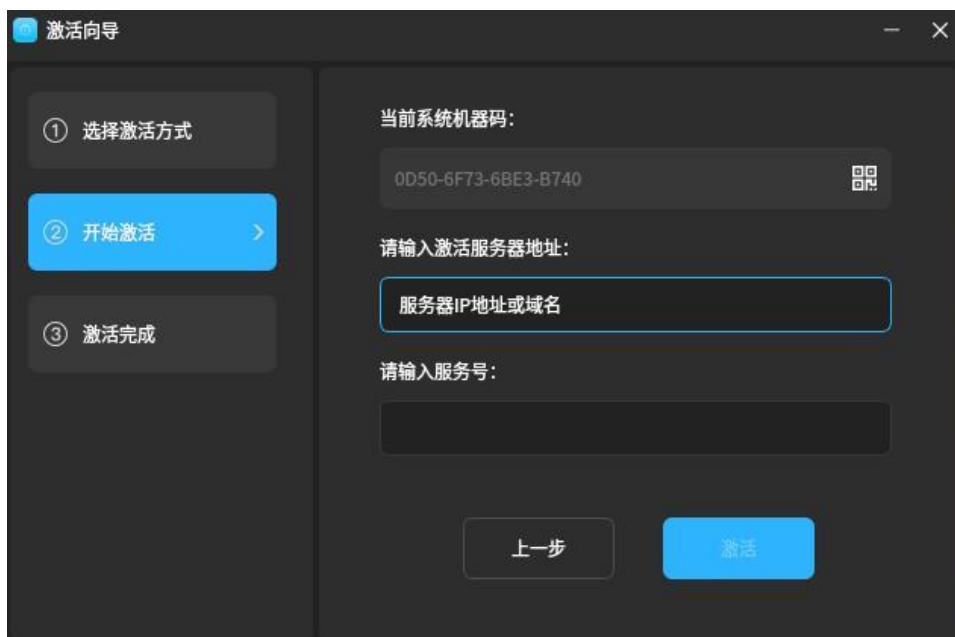


图 8-13 在线激活

导入文件激活：系统提供了导入文件激活方式，如图 8-14 所示：

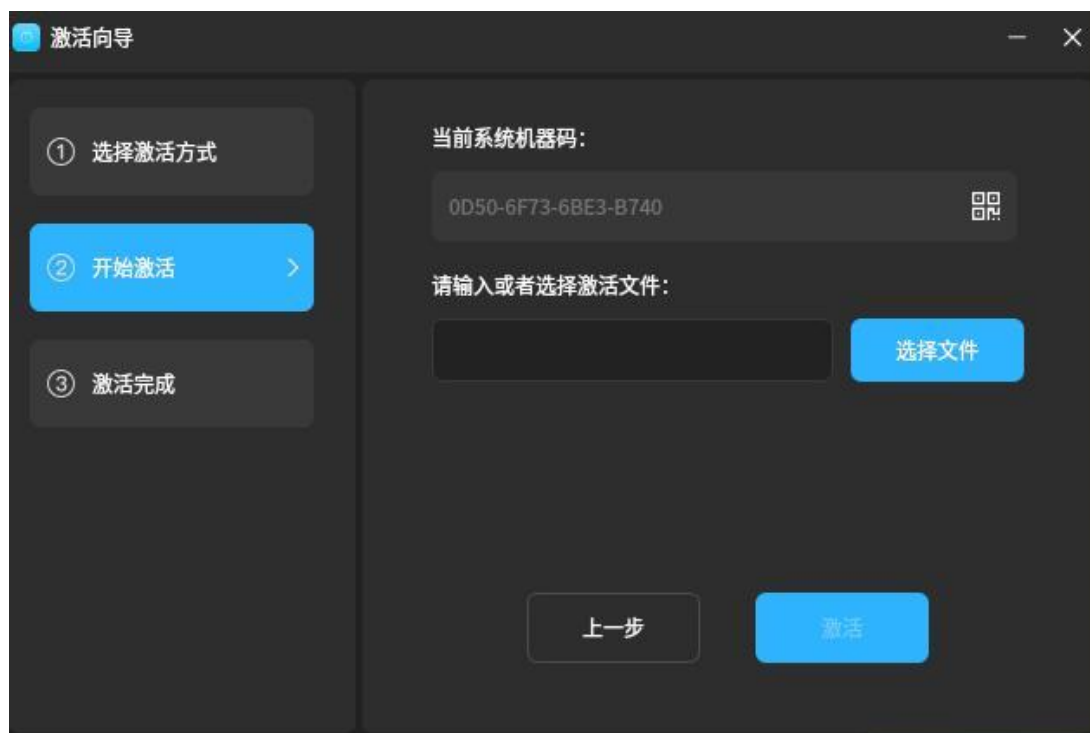


图 8-14 导入文件激活

8.2.1.1.2 硬件信息

硬件信息可查看到主机的 cpu、内存等相关硬件版本信息。如图 8-15：

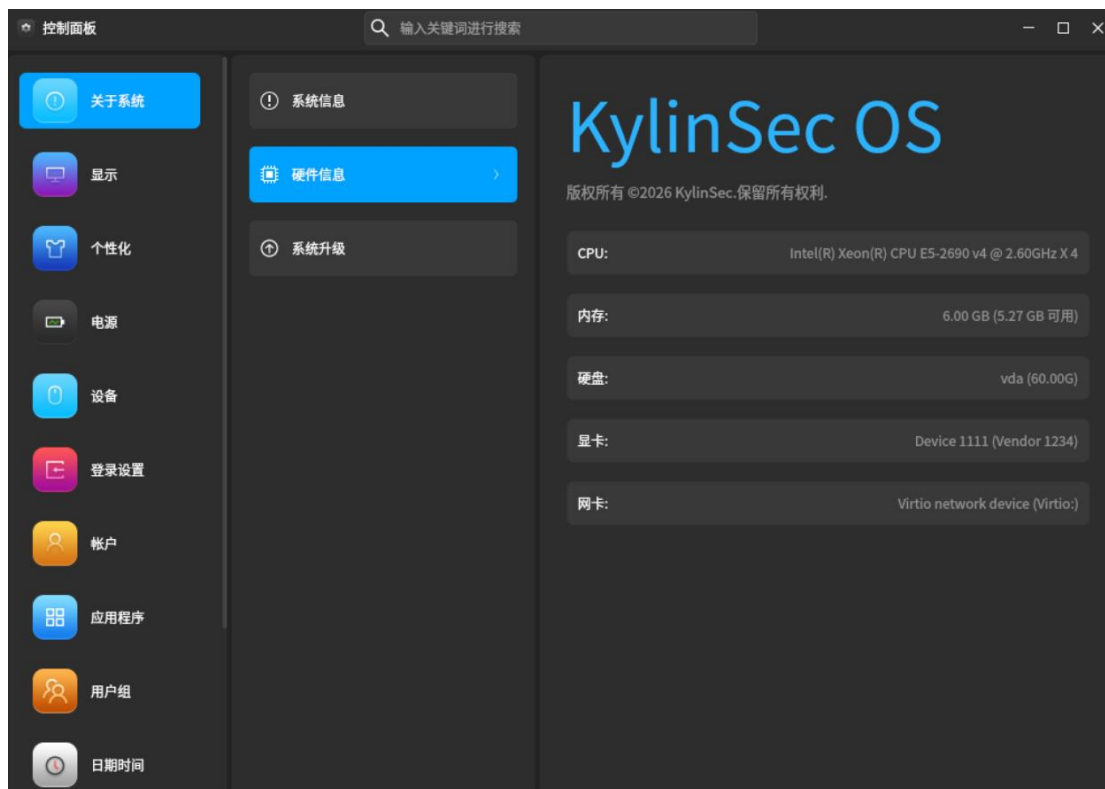


图 8-15 硬件信息

8.2.2 声音

麒麟信安服务器操作系统中，可以通过选择控制面板中的“声音”来控制对系统音量大小，测试扬声器及设置音频输入输出。

点击“控制面板” > “声音”，如图 8-16 所示：

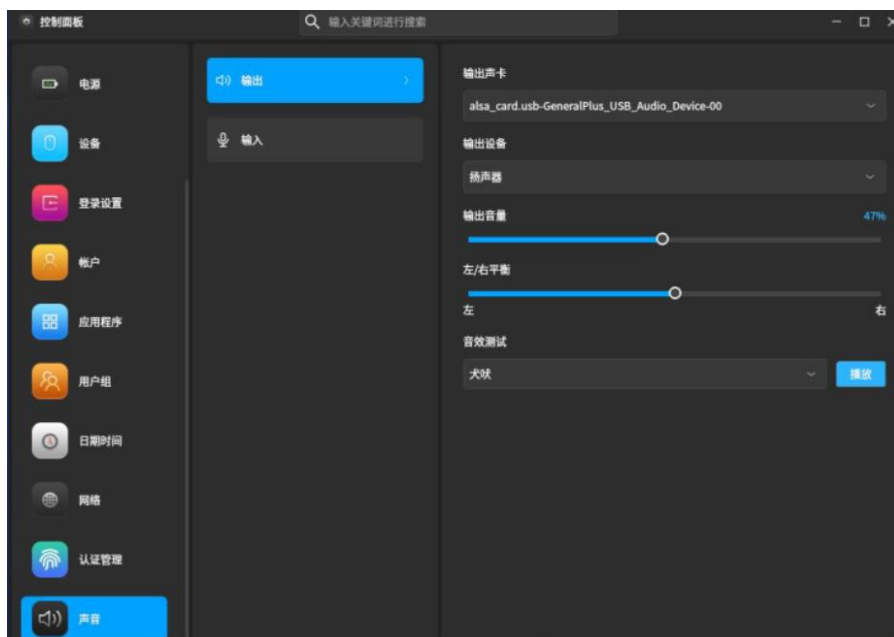


图 8-16 声音

8.2.3 电源

系统提供了电源的通用设置和电源设置功能，可设置计算机空闲时间、空闲是否锁定屏幕等。

点击“控制面板”>“电源”打开工具，如图 8-17 所示：

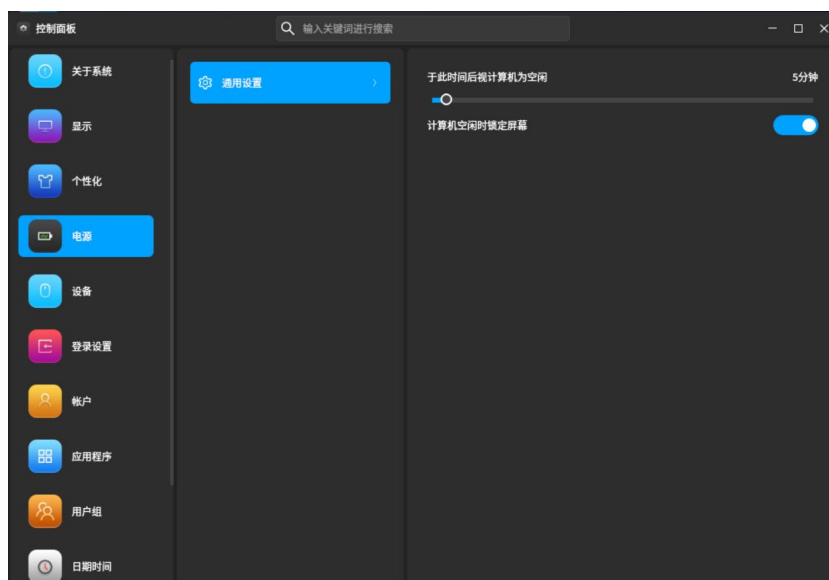


图 8-17 电源

8.2.4 设备

设备主要包含了键盘通用选项、鼠标设置、键盘布局和触摸板设置四个功能，如图 8-18：设备可以设置通用选项、键盘布局和鼠标设置，可自定义键盘按键延时、速度，设置鼠标手持模式、移动速度、自然滚动和模拟中键功能，可手动添加不同的布局，触摸板设置支持调节指针速度、滚动方式和自然滚动方向，适配右手操作习惯。

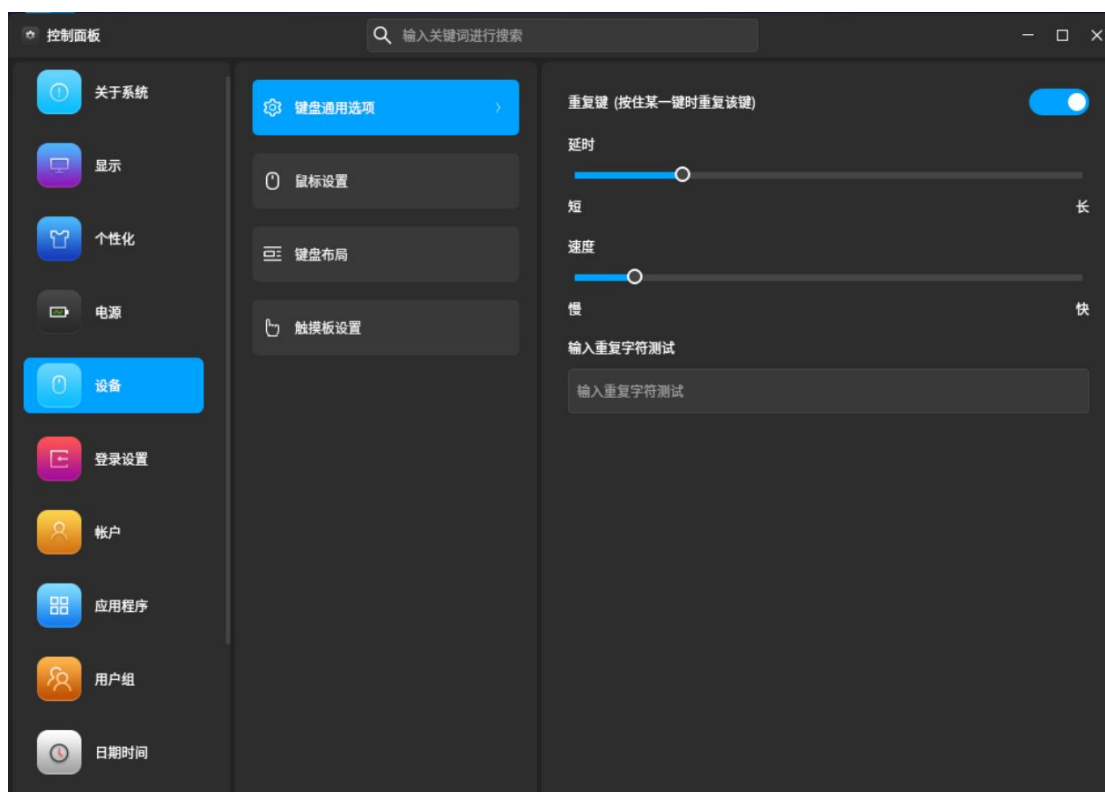


图 8-18 设备

8.2.5 登录设置

登录设置包含了登录选项与自动登录设置功能，可自定义登录界面壁纸、缩放比例、是否显示用户列表、选择某个用户开机自动登录等，如图 8-19：

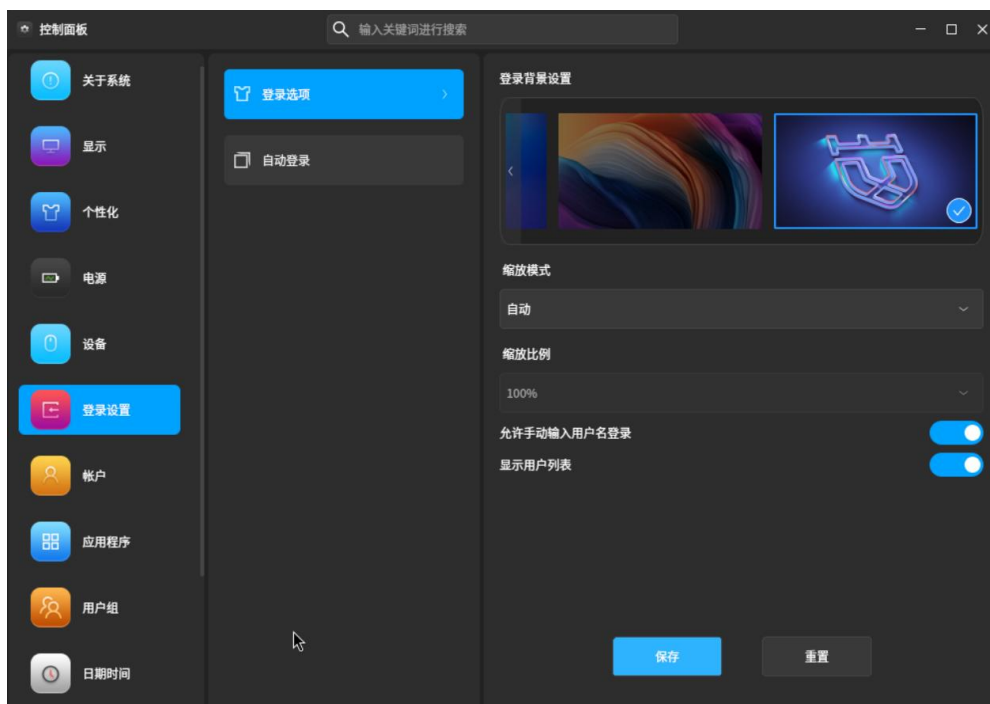


图 8-19 登录设置

8.2.6 账户

账户是对用户进行管理的一个简单易用工具,可以通过这个工具对用户进行配置和管理;点击“控制面板”>“账户管理工具”可打开工具,如图 8-20 所示:

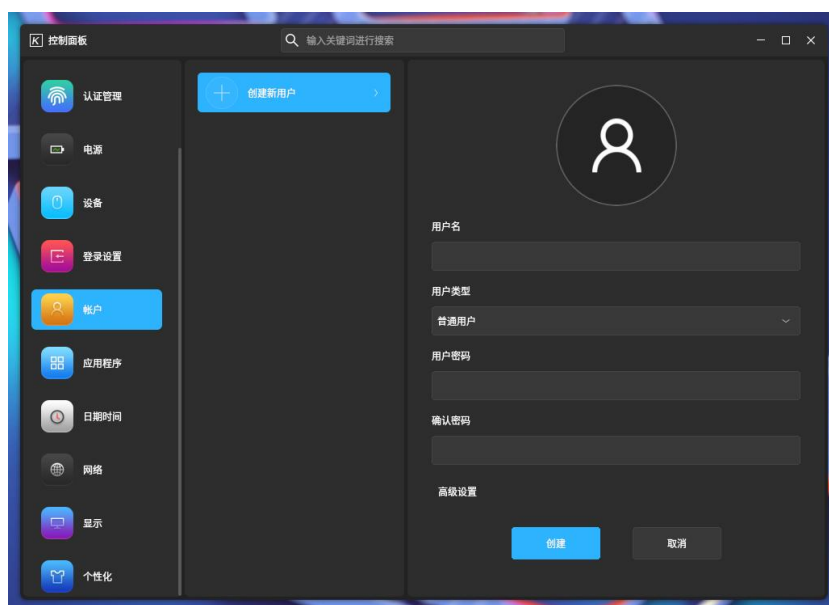


图 8-20 账户

创建用户时的高级设置选项可以设置用户的登录 shell、指定用户 ID 和指定用户目录，如图 8-21 所示：

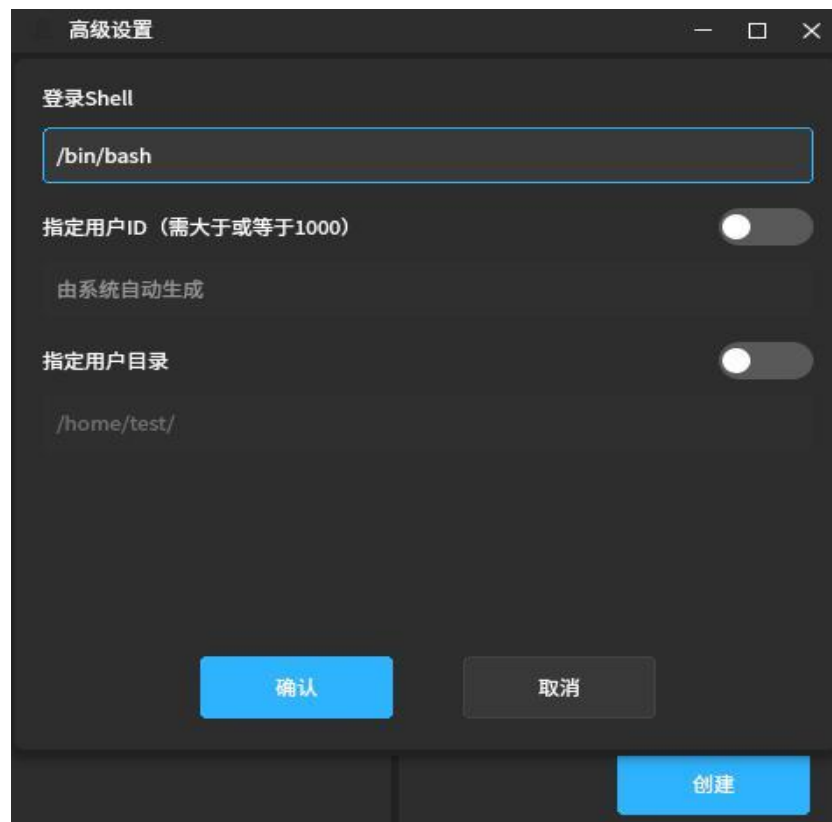


图 8-21 高级设置

8.2.6.1 编辑用户

系统提供了修改用户头像和密码的功能，点击用户头像区域可选择系统提供的头像或自主上传图片作为头像，选择后点击“确认”保存，如下图 8-22 所示：

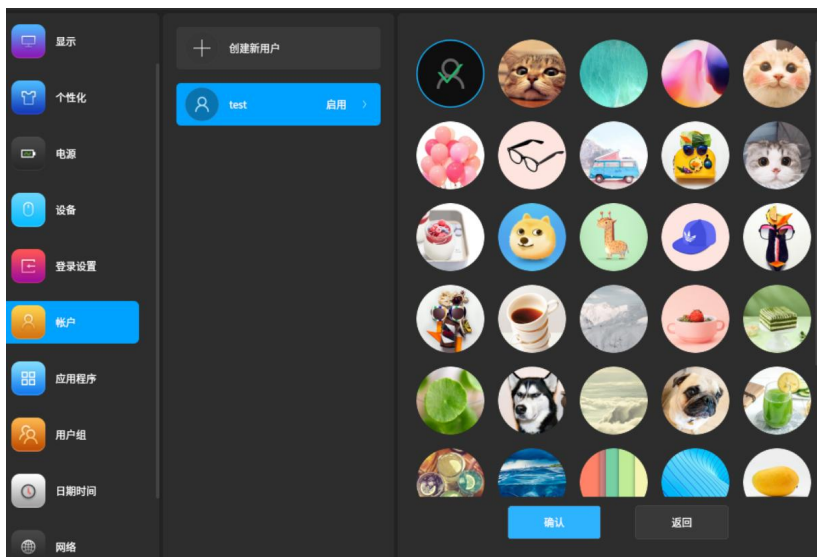


图 8-22 头像修改

选择某个用户后点击“修改密码”按钮可修改用户密码，输入新密码和确认密码后点击“保存”即可修改成功，如下图 8-23 所示：

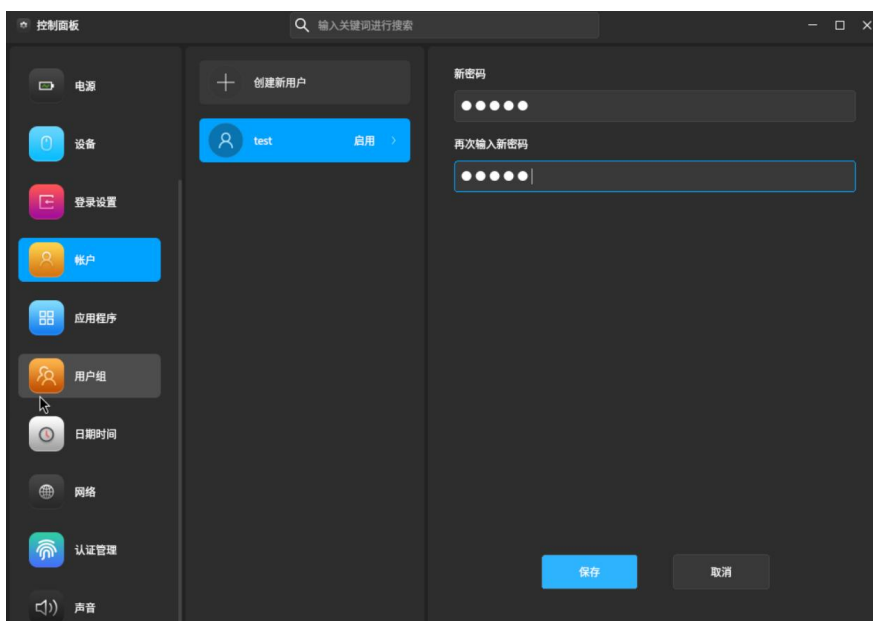


图 8-23 修改密码

8.2.6.2 删除用户

首先选中左侧信息栏里的欲删除的用户，然后在右侧工具栏上点击“删除”按钮，如图 8-24 所示：

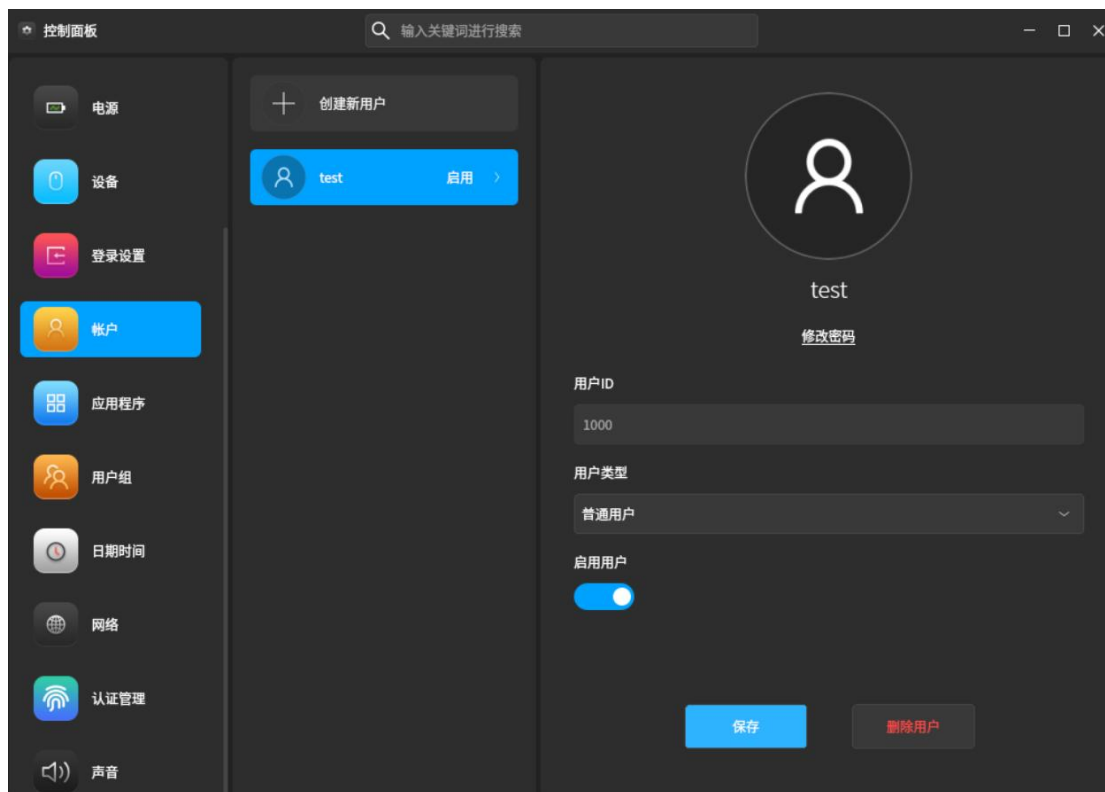


图 8-24 删除用户

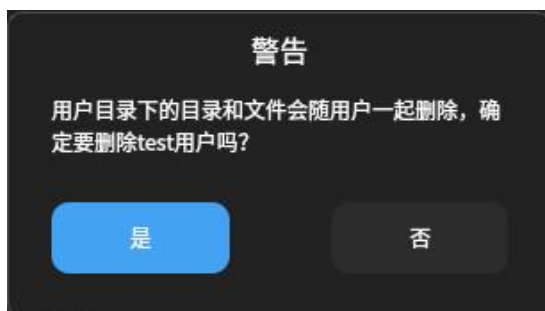


图 8-25 删除用户提示

在弹出图 8-25 所示的对话框中单击“否”撤消删除，单击“是”确认删除。

8.2.7 用户组

用户组是对组进行管理的一个简单易用工具,可以通过这个工具对组群进行配置和管理，点击“控制面板”>“用户组”可打开工具，如图 8-26 所示：

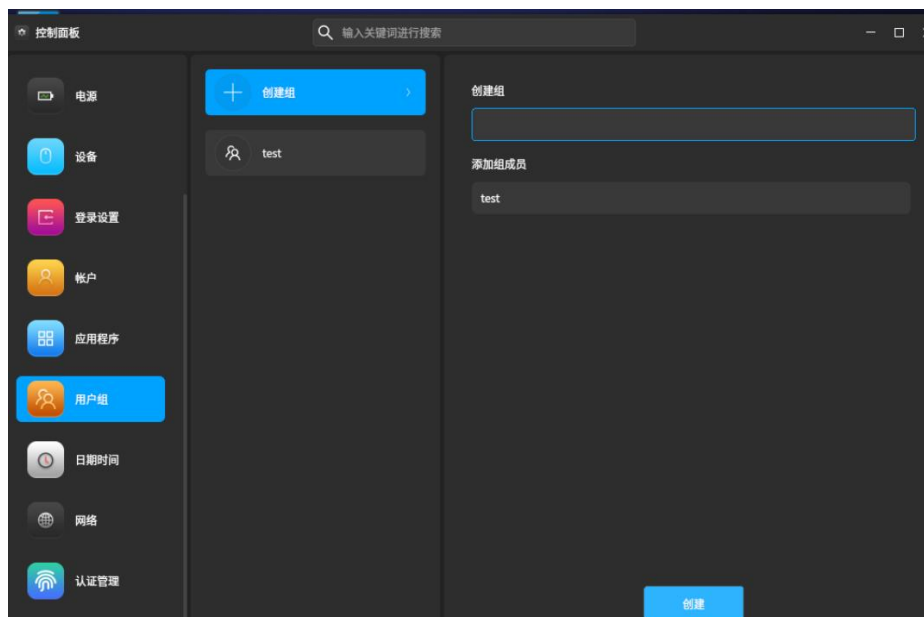


图 8-26 用户组

8.2.7.1 编辑用户组

系统提供了创建用户组和对用户组修改的功能,输入用户组名称点击创建即可成功创建用户组,如图 8-27 所示;点击用户组名称区域可进行名称修改,点击左下角添加成员,可以添加成员进所选的用户组,如图 8-28 所示:

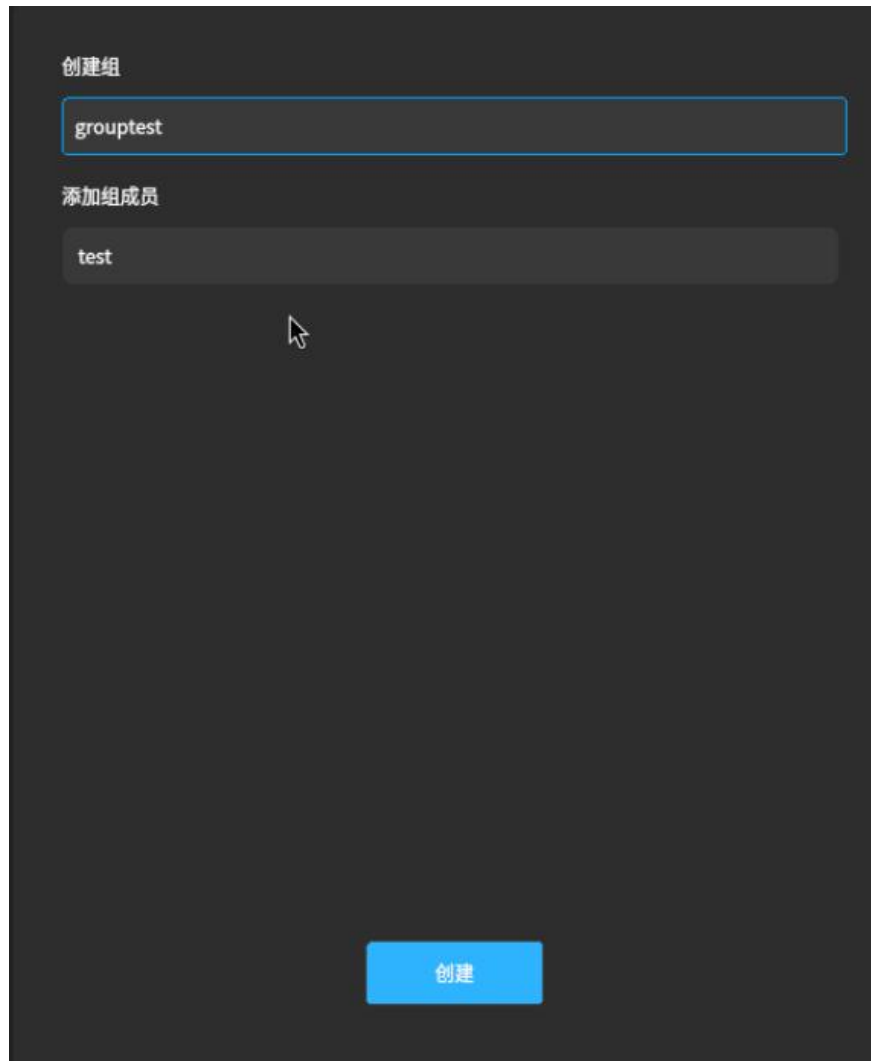


图 8-27 创建用户组

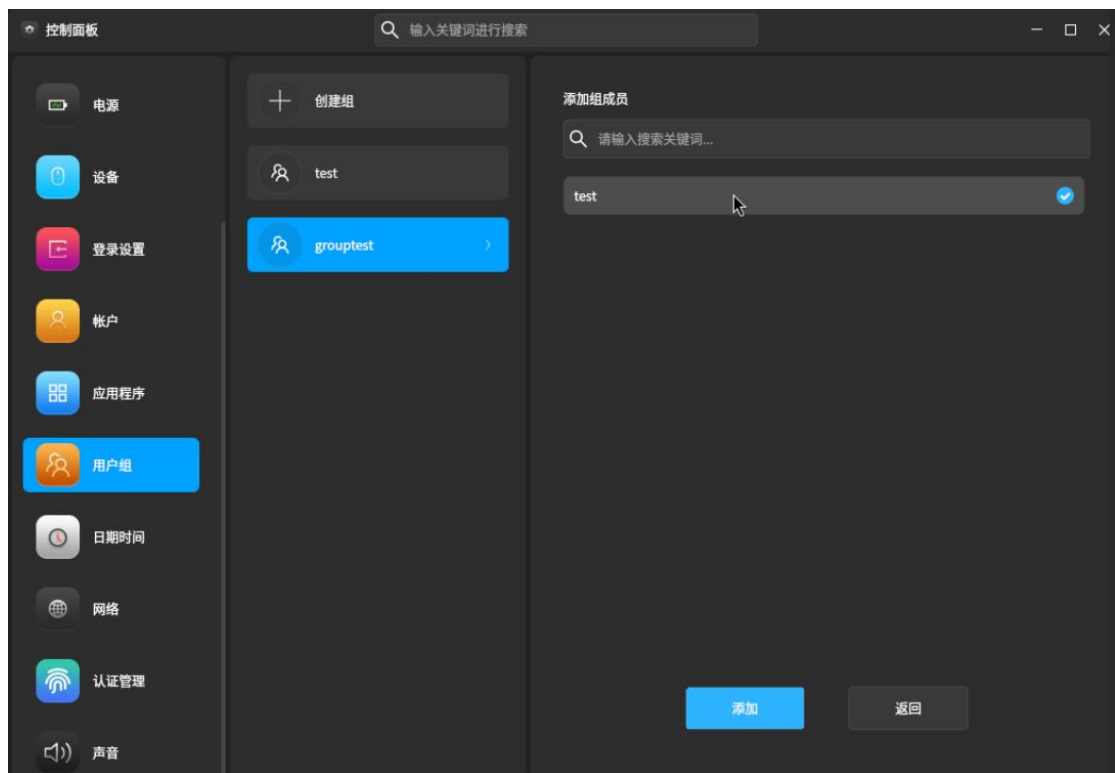


图 8-28 添加用户组成员

8.2.7.2 删除用户组

选中左侧信息栏里的欲删除的用户组，然后在右侧工具栏上点击“删除”按钮，如图 8-29 所示：

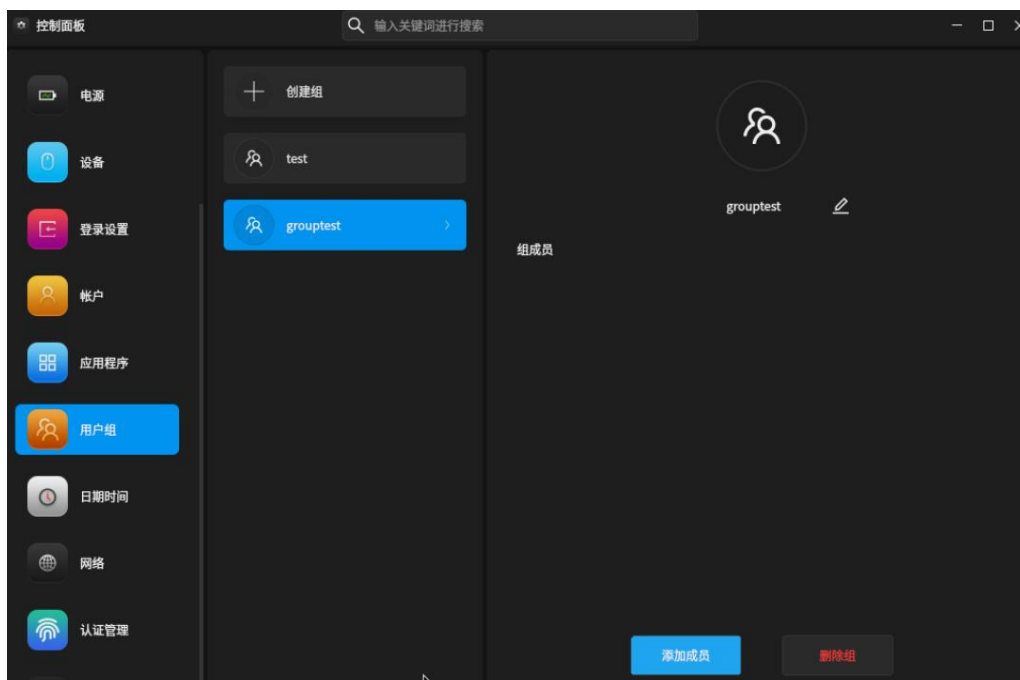


图 8-29 删除用户组

8.2.8 日期与时间

在麒麟信安服务器操作系统中，可以通过选择控制面板中的“日期时间”工具设置时区、时间、日期时间格式。

点击“控制面板”>“日期时间”打开工具，如图 8-30 所示：

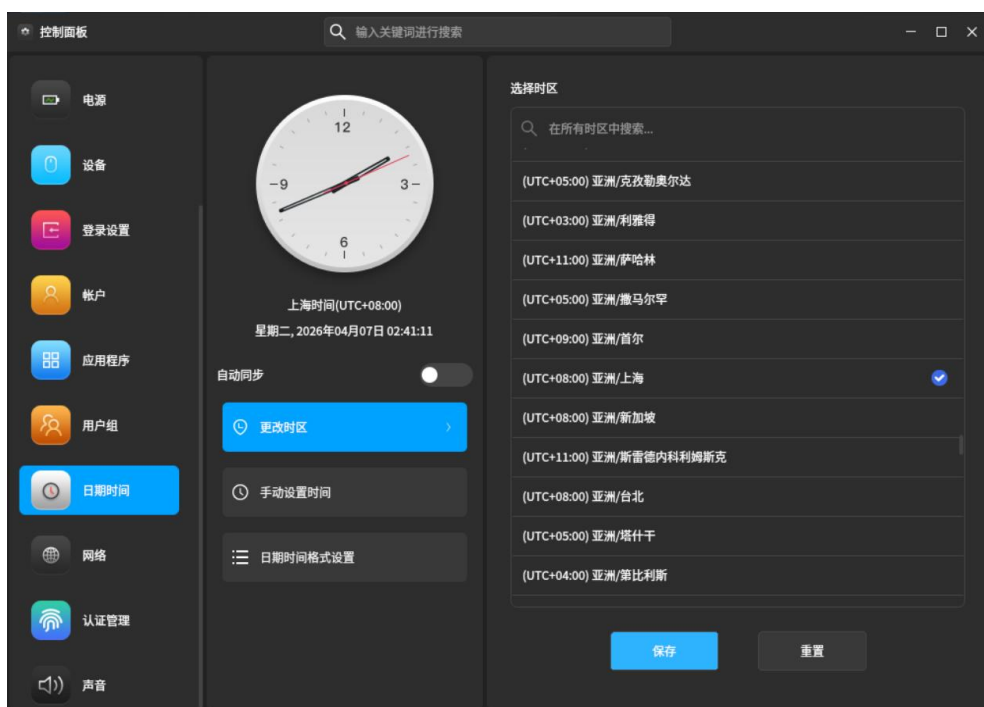


图 8-30 时间和日期管理工具

8.2.8.1 设置时区

系统提供了自动同步时间和手动设置时区的功能，当能够连接外网时打开“自动同步”开关会自动更新日期和时间，也可以更改时区为其他时区，选择时区后点击“保存”，如图 8-31：

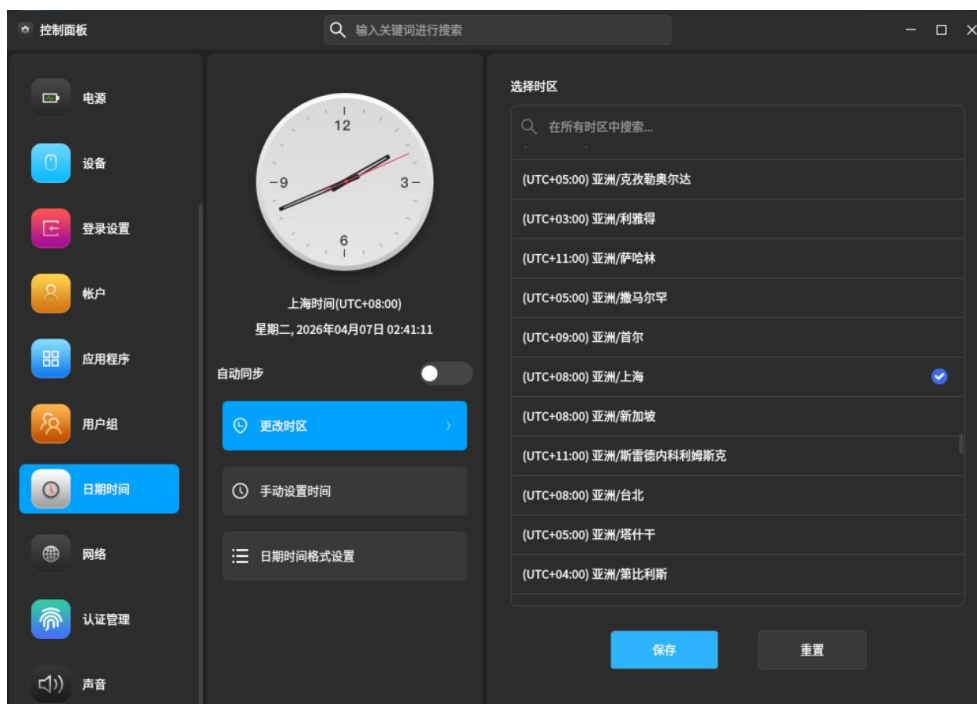


图 8-31 更改时区

8.2.8.2 手动设置时间

关闭“自动同步”开关，可以手动更改日期和时间，如图 8-32 所示：

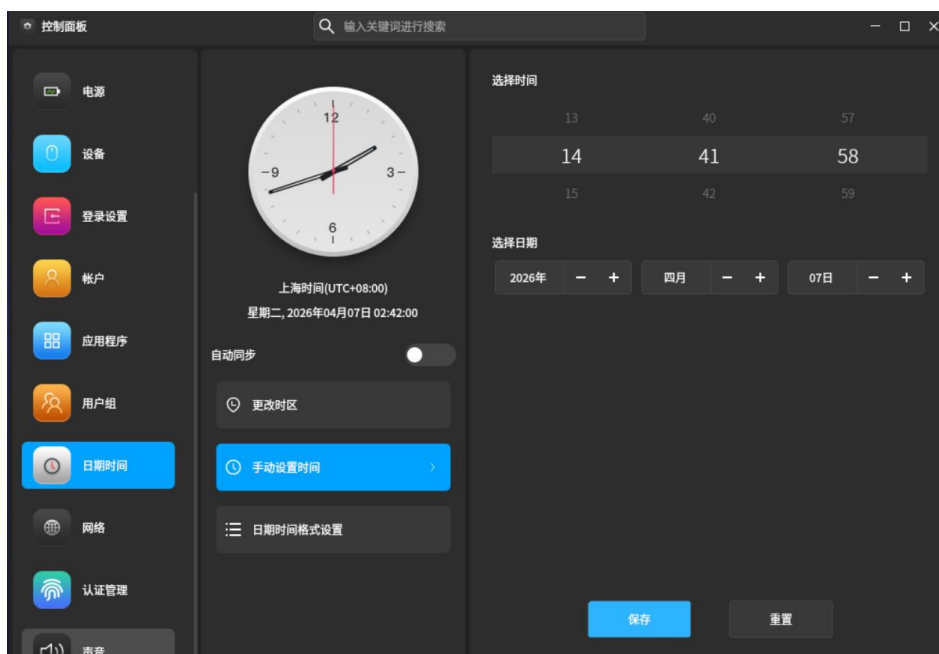


图 8-32 日期和时间设置

8.2.8.3 日期时间格式设置

系统提供了日期时间格式设置功能，可以自定义长日期显示格式、短日期显示格式、时间格式、时间显示秒，如图 8-33 所示：

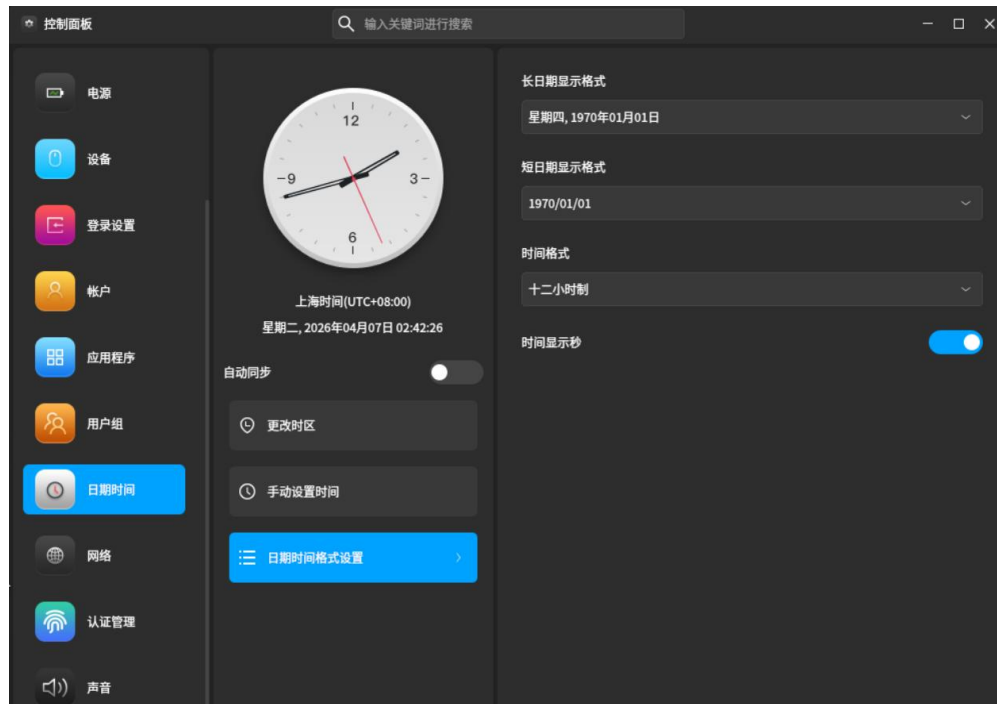


图 8-33 日期时间格式设置

8.2.9 网络

麒麟信安服务器操作系统提供了可以设定、配置、VPN 和管理各种网络类型的桌面网络设置工具，可以对移动宽带设备、IPv6 提供改进的支持。

点击“控制面板” > “网络” 打开工具，如图 8-34：

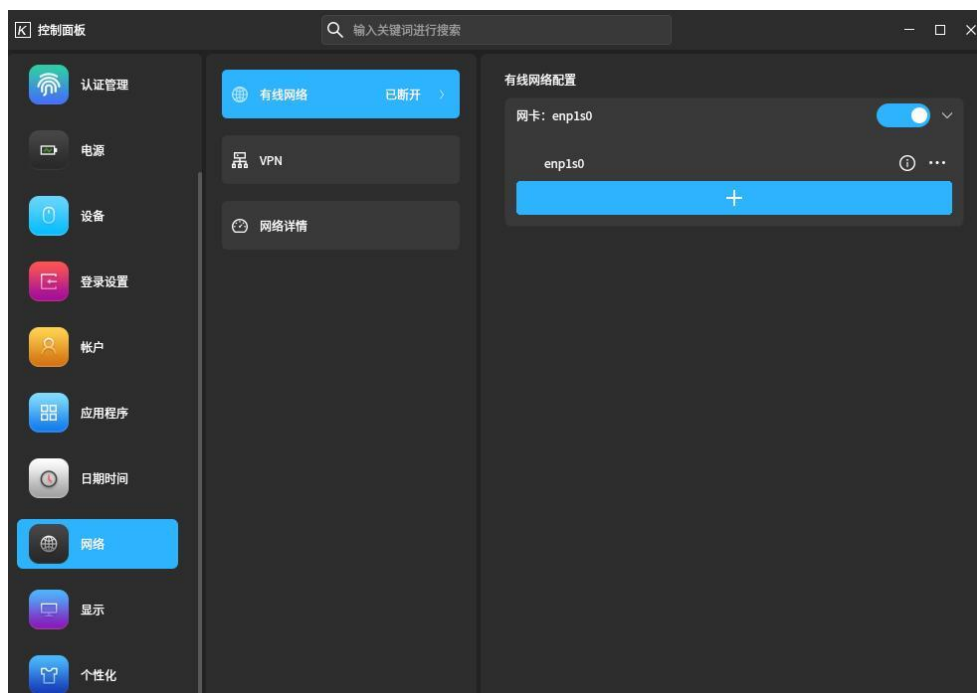


图 8-34 网络

选择当前可用的“有线网络”，点击“”按钮，弹出如图 8-35 所示对话框：

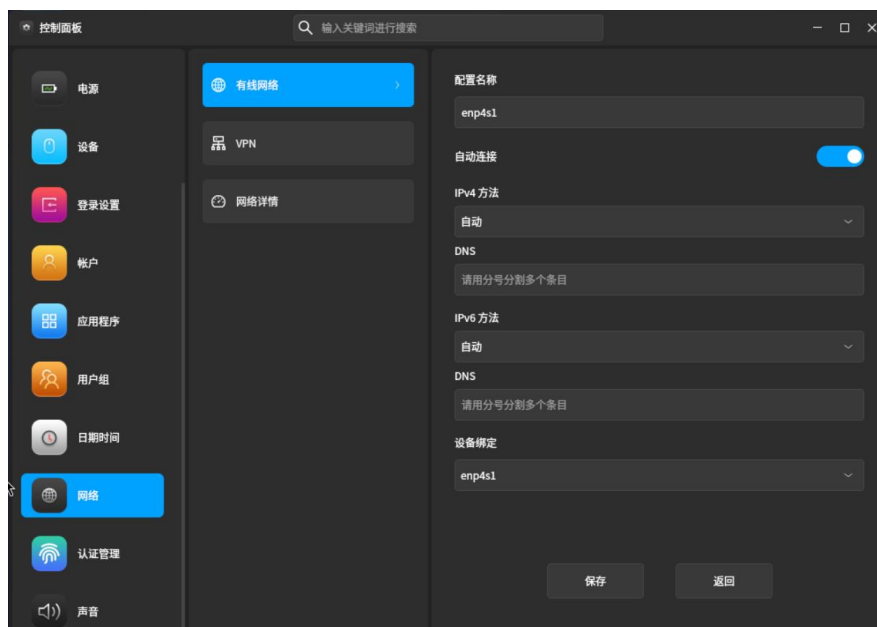


图 8-35 编辑网卡

“IPv4 方法”是用户常用到的设置，可以选择 DHCP 或手动的方式配置网络。

DHCP 方式系统会自动给用户分配 IP 地址，手动方式需要输入 IP、子网掩码和

网关，也可以根据实际情况输入 DNS，配置完成后点击“保存”，自动生效。

如图 8-36 所示：

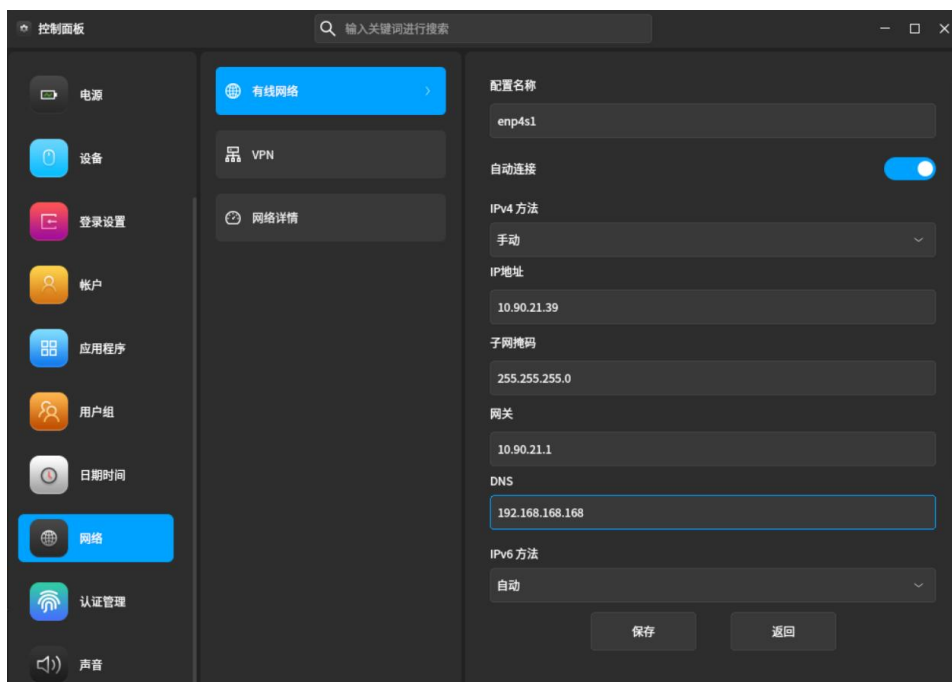


图 8-36 IPv4 设置

VPN，根据需要点击“+”添加，可选择 VPN 类型、输入 VPN 服务的用户、网关、密码等，配置完成后点击“保存”，自动生效，如图 8-37：

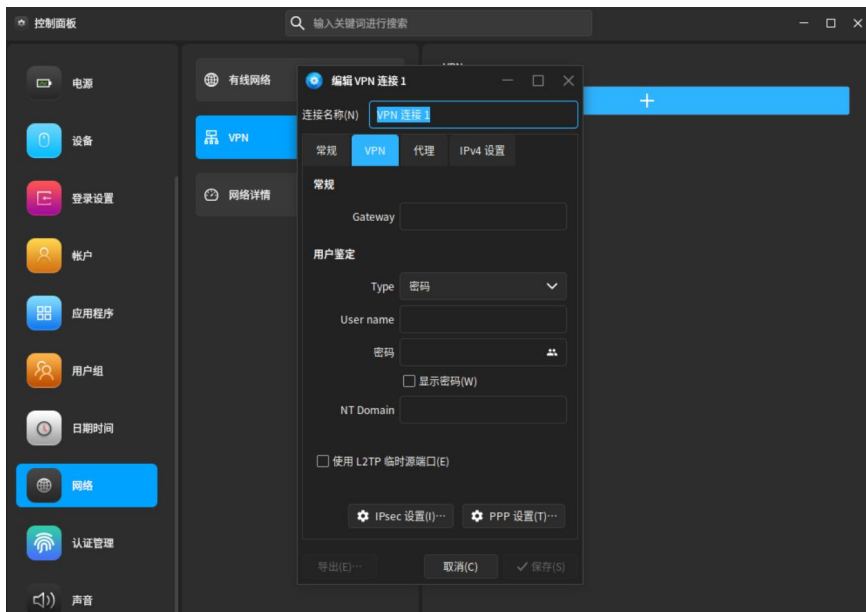


图 8-37 VPN 设置

网络详情可查看到当前网口、mac 地址、IP 等相关信息，点击“网络详情”即可查看，如图 8-38 所示：

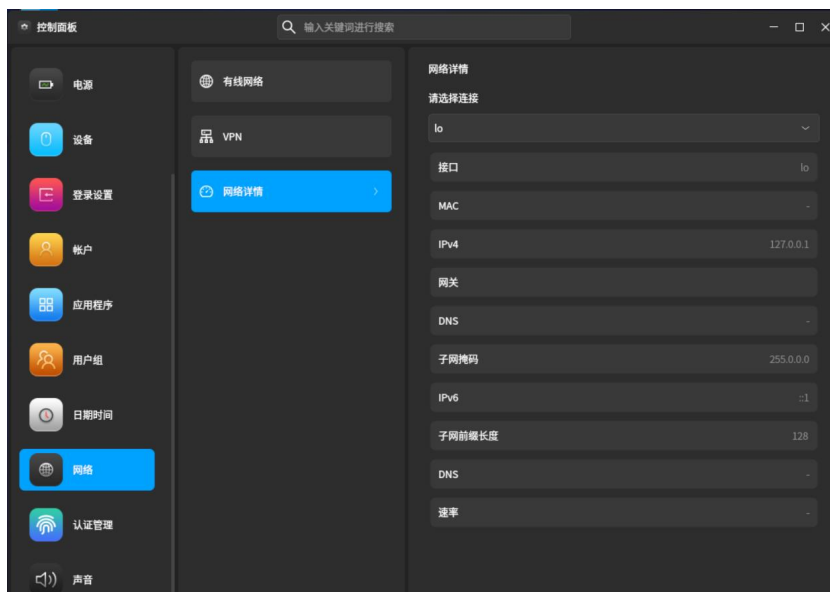


图 8-38 网络详情

8.2.10 显示

麒麟信安服务器操作系统提供了显示属性定制工具。通过点击“控制面板”>“显示”，进入显示设置界面，如图 8-39：

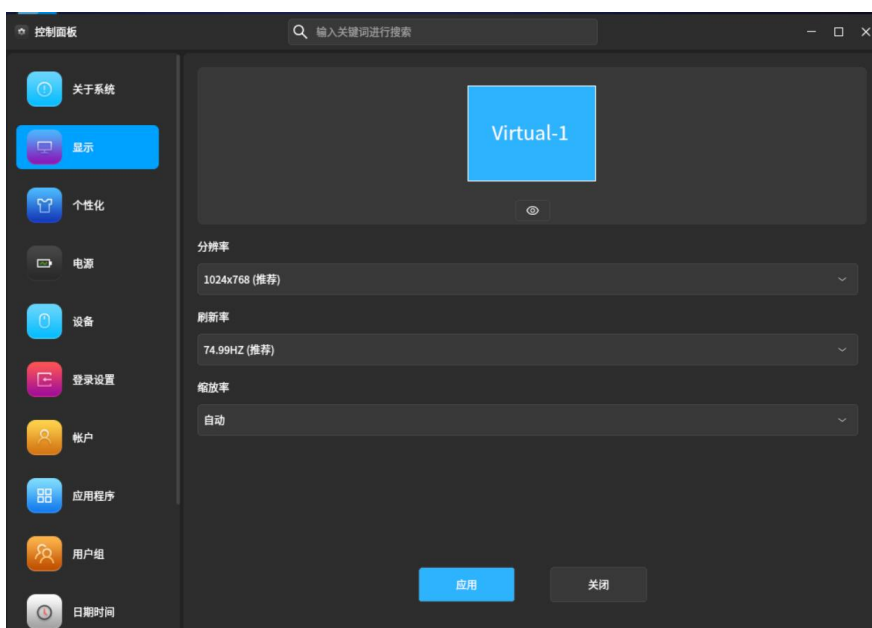


图 8-39 显示设置

可以根据需要设置分辨率、刷新率、缩放率，设置完成后点击“应用”。

8.2.11 个性化

麒麟信安服务器操作系统提供的个性化工具。实现了对系统的主题、壁纸、字体、快捷键和屏保五个方面的统一配置和管理。

选择“控制面板”>“个性化”，进入个性化设置界面，如下图 8-40：

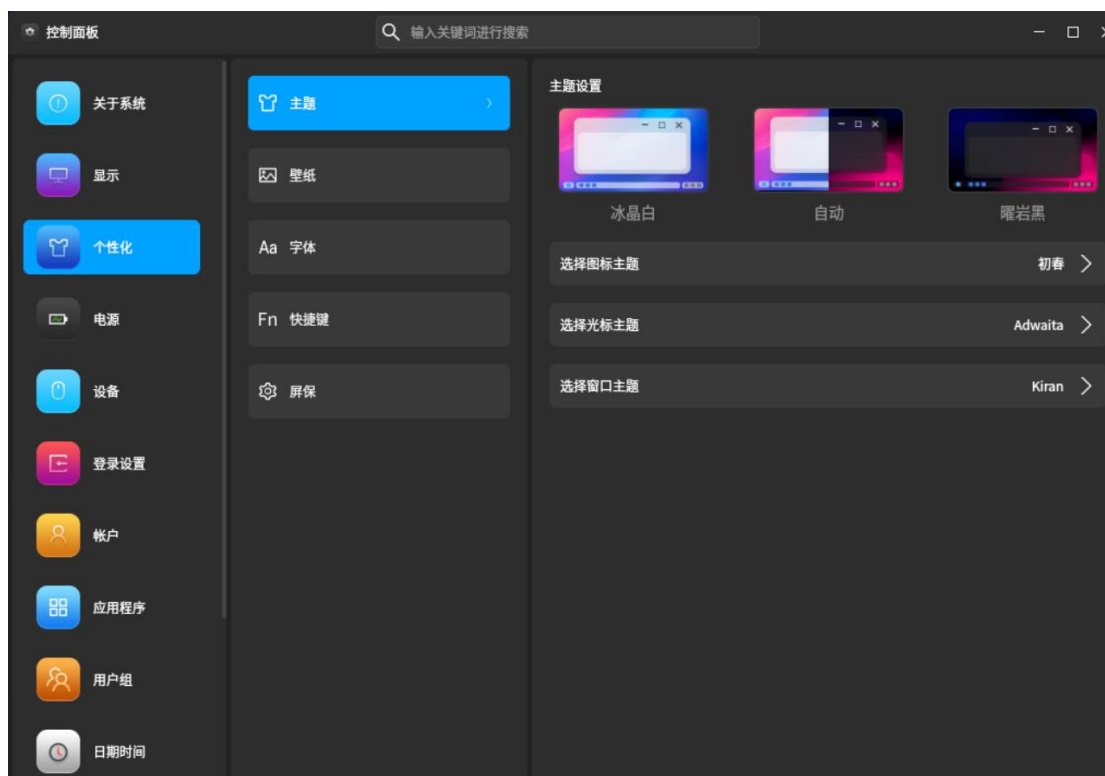


图 8-40 个性化设置

8.2.11.1 主题设置

主题可以对系统的对话框风格，菜单风格，系统面板风格，图标风格进行统一设置。系统中默认提供了深色和浅色主题，点击主题浏览对话框中的主题，即可设置系统主题。系统也提供了图标和光标主题设置，点击设置区域可以切换主题，如图 8-41：

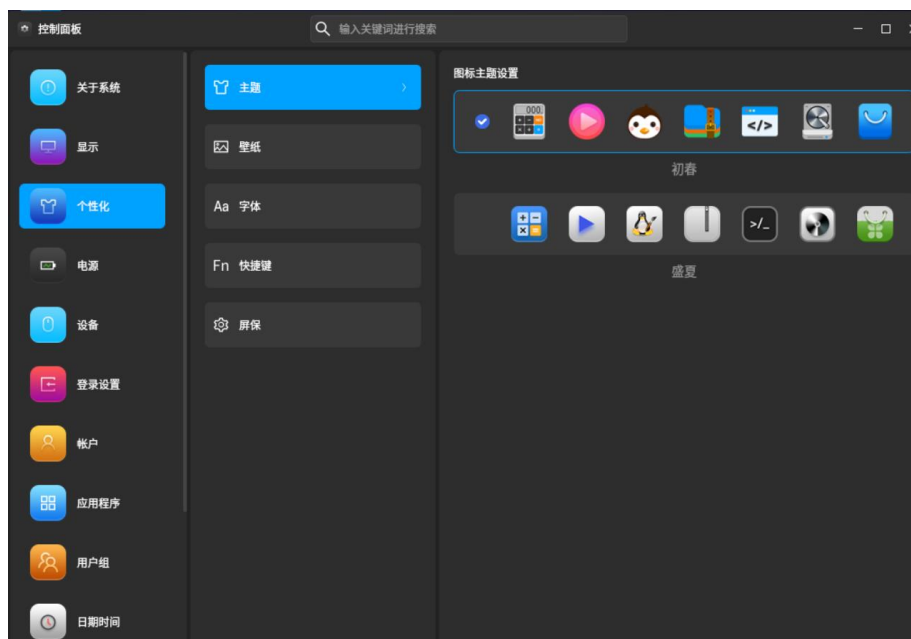


图 8-41 主题设置

8.2.11.2 壁纸设置

系统提供了选择桌面壁纸和锁屏壁纸设置功能，可以使用系统自带的壁纸，也可以手动添加图片设置。如图 8-42：

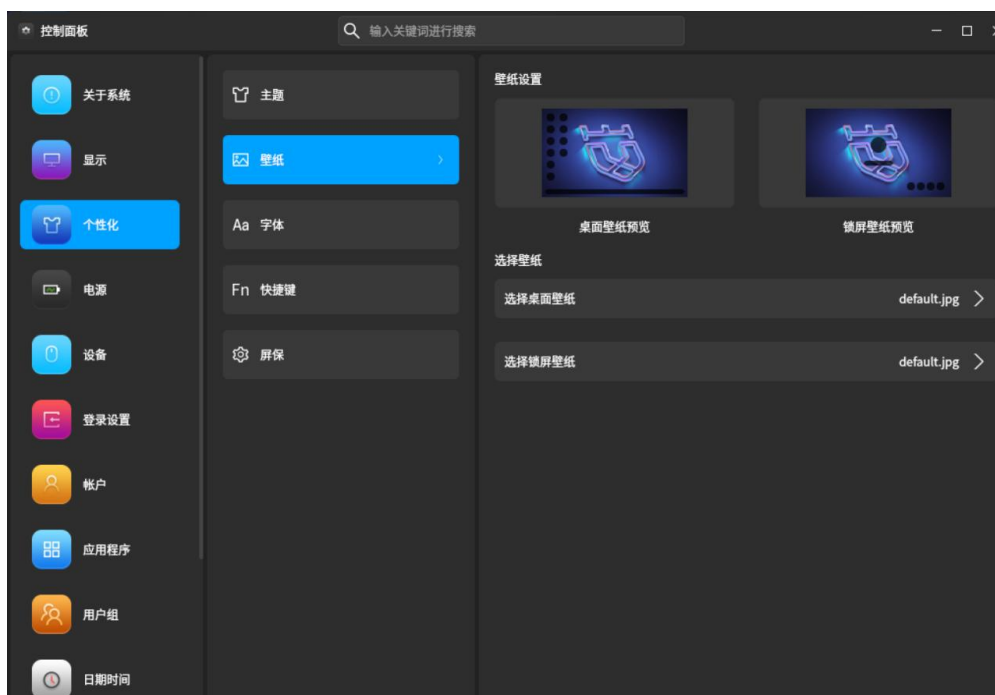


图 8-42 壁纸设置

8.2.11.3 字体

用户可以通过字体设置来设置系统图形界面的各种类型的字体,可设置系统字体、等宽字体和字号。如图 8-43:

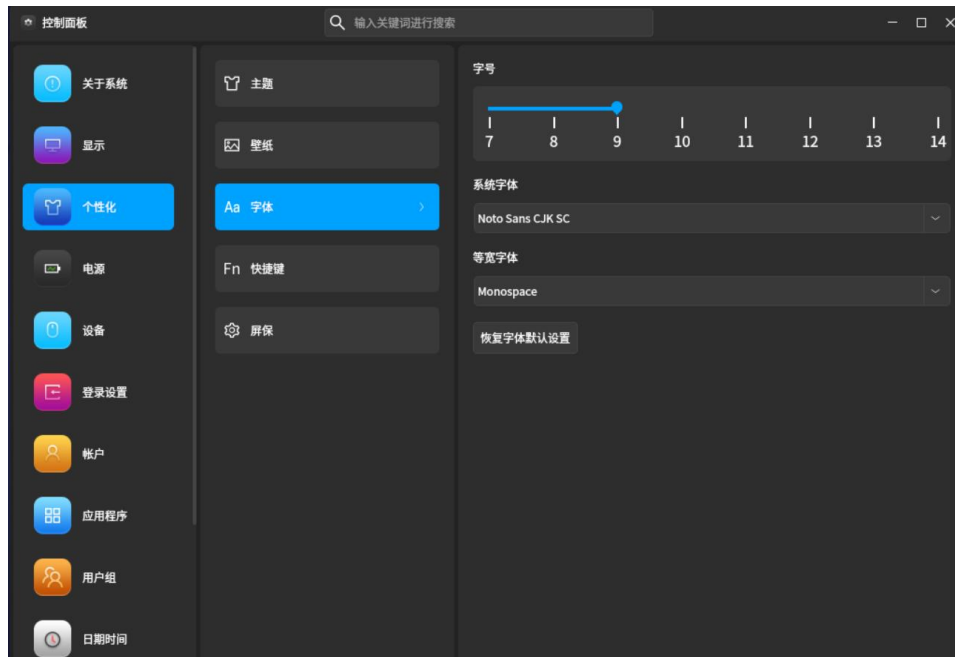


图 8-43 字体设置

8.2.11.4 快捷键

“快捷键”可以对系统快捷键进行设置,如图 8-44:

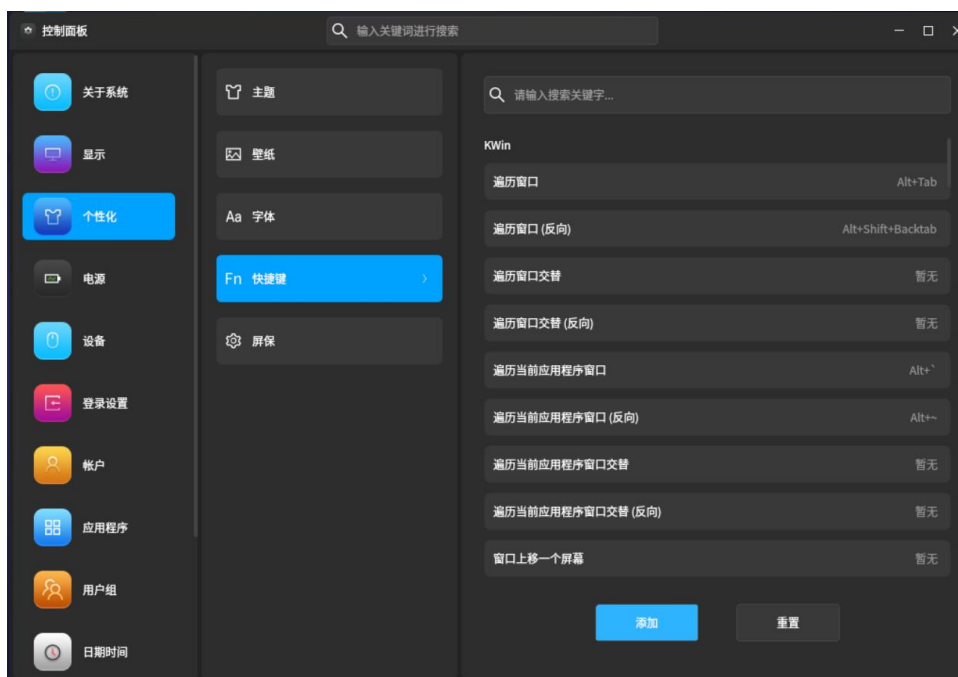


图 8-44 快捷键设置

8.2.11.5 屏保

“屏保”可以设置系统空闲时屏幕保护，如图 8-45：

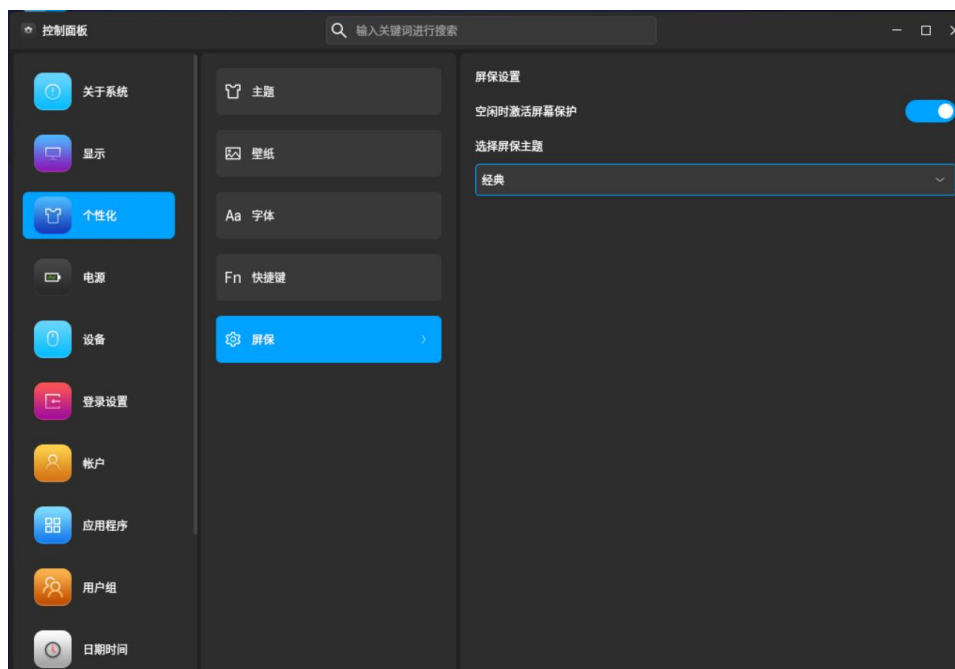


图 8-45 屏保设置

8.2.12 应用程序

选择“开始菜单”>“应用程序”，其子菜单有：“默认程序”和“开机自启”；可以进行各类程序默认设置及开机启动；如图 8-46：

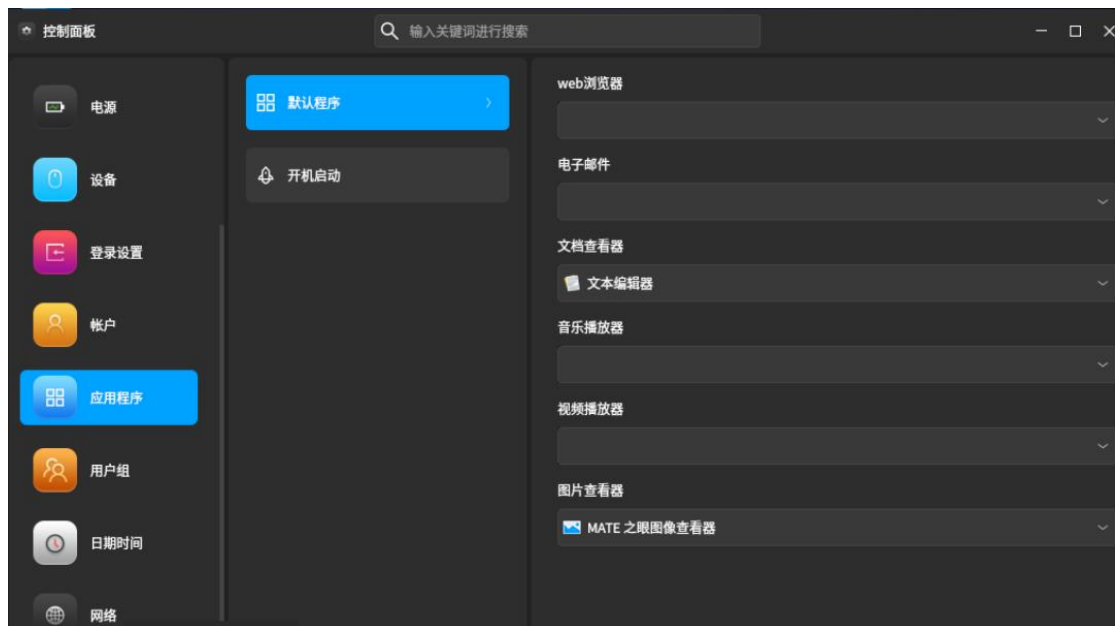


图 8-46 应用程序

8.2.13 认证管理

麒麟信安服务器操作系统中，用户可以在控制面板的认证管理页面对指纹、指静脉、虹膜、人脸等一些信息进行配置，如图 8-47 所示：

指纹：接入指纹设备，安装驱动包，点击“指纹列表下方“+”号，开始录入指纹。

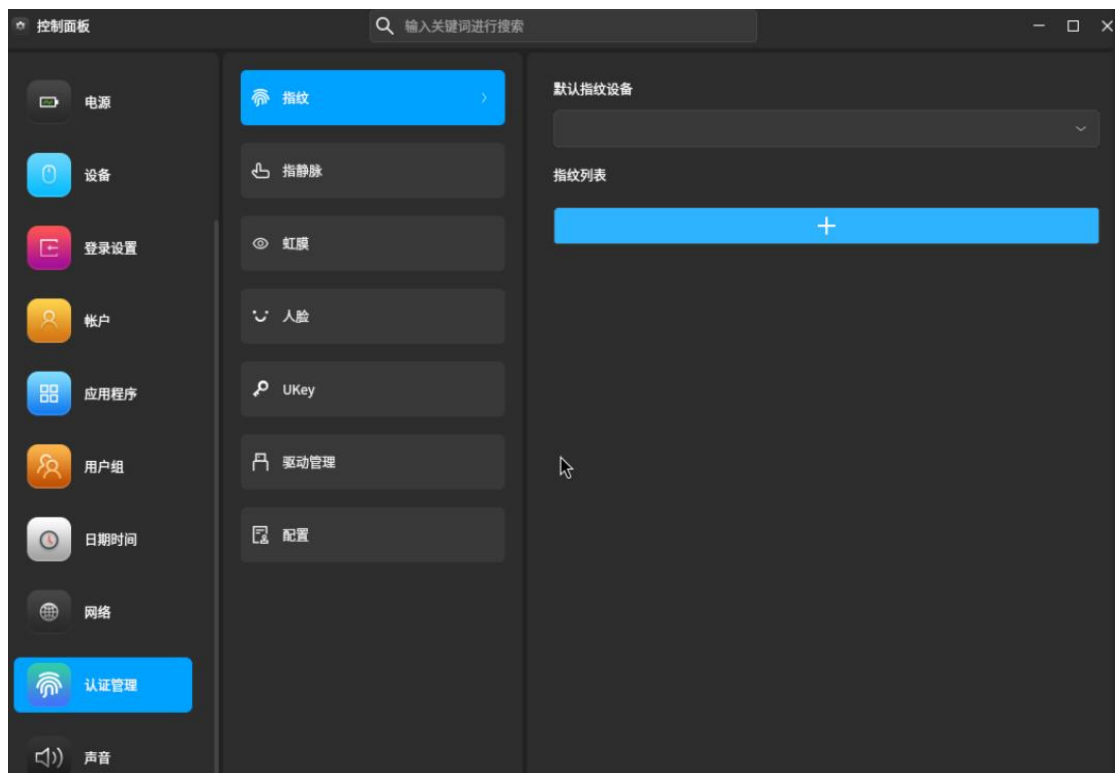


图 8-47 认证管理

系统弹出“校验当前用户密码”提示框，输入当前登录用户的密码进行身份认证，如图 8-48 所示：

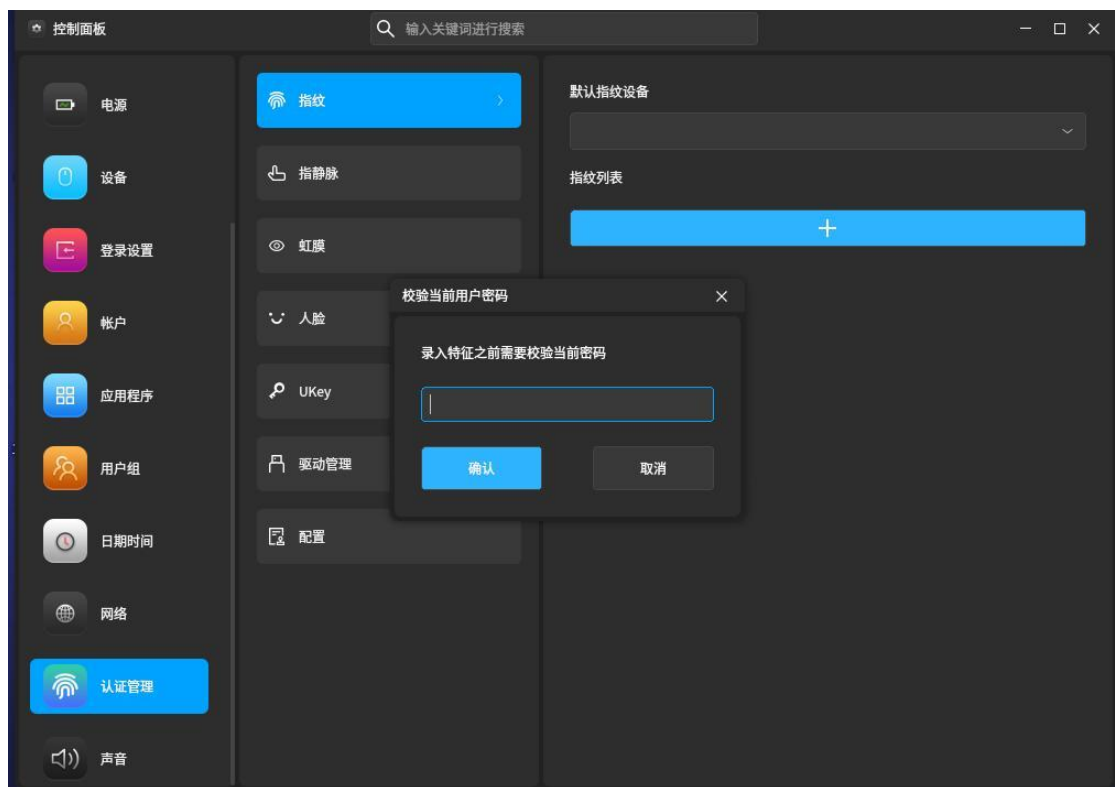


图 8-48 指纹认证密码校验

完成密码校验后，系统将进入指纹采集界面，如图 8-49 所示。此时请按照界面提示，将手指按压在指纹识别设备上，完成指纹特征的录入操作。

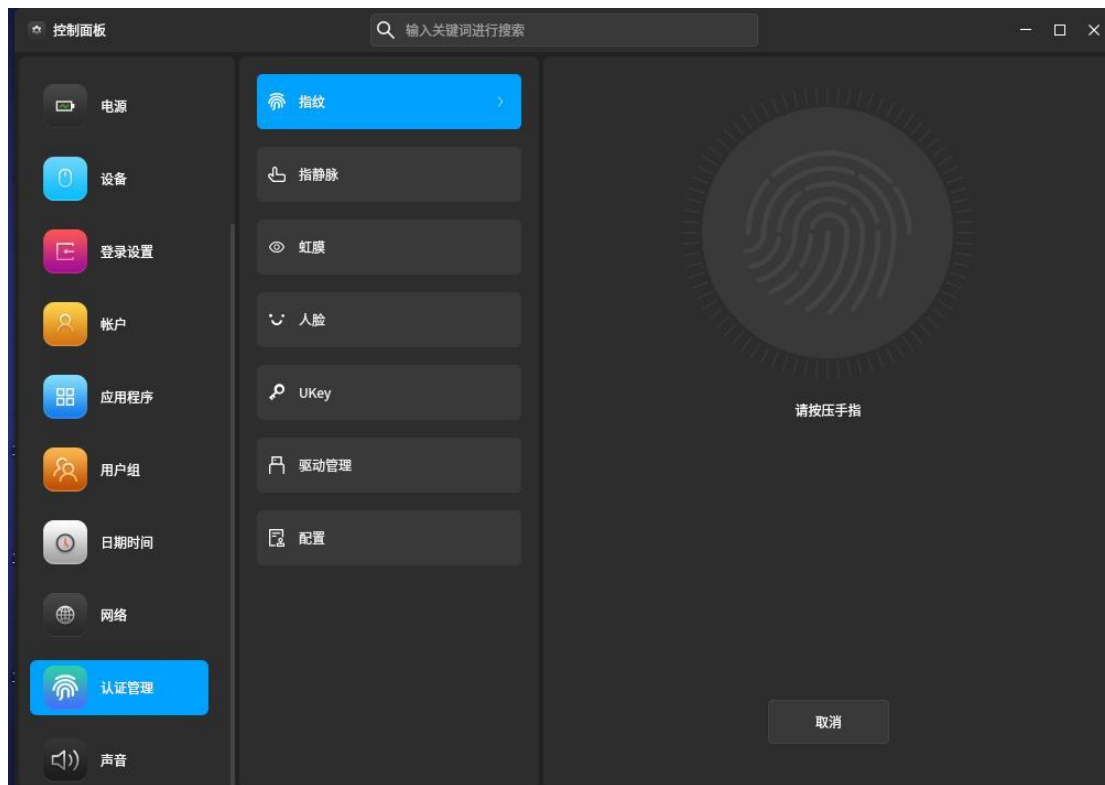


图 8-49 指纹录入界面

指纹录入过程中，按照界面指引多次按压手指，直至系统采集完成并弹出成功提示框，如图 8-50 所示。提示框显示已成功录入，点击“是”即可完成指纹认证的配置。

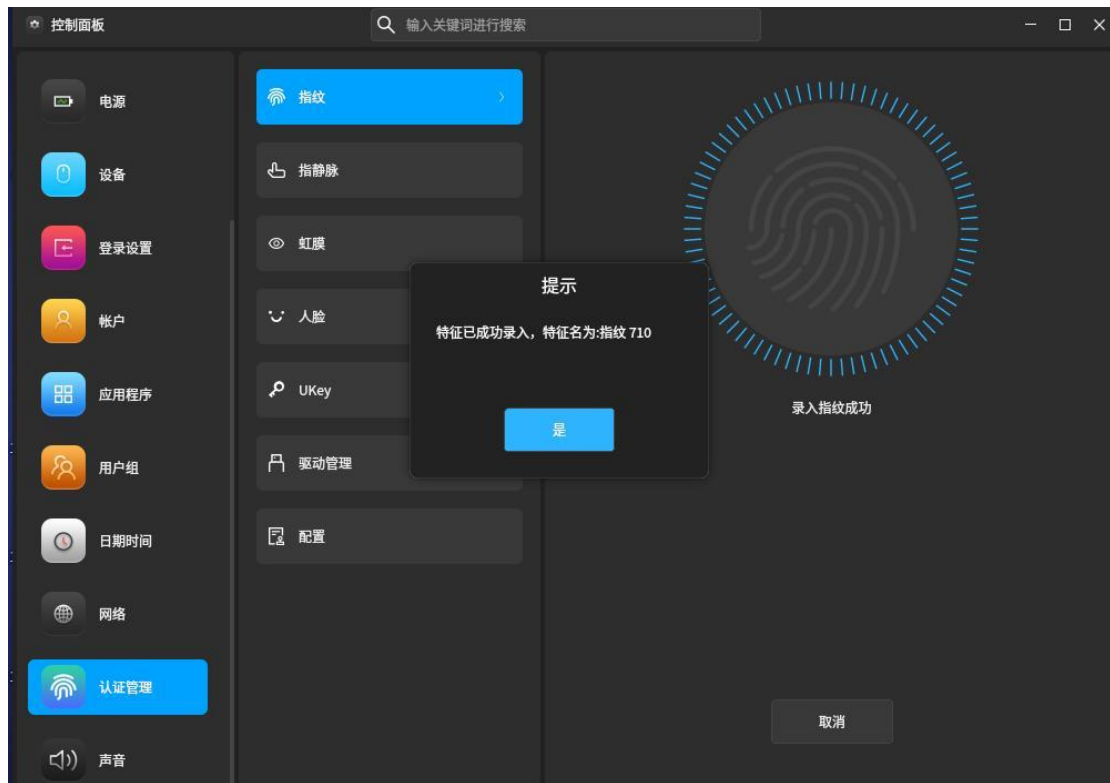


图 8-50 录入成功界面

录入完成后，指纹特征会自动添加到指纹列表中，如图 8-51 所示。在该界面可查看已录入的指纹名称、当前默认指纹设备，并支持添加新指纹、编辑或删除已有指纹。

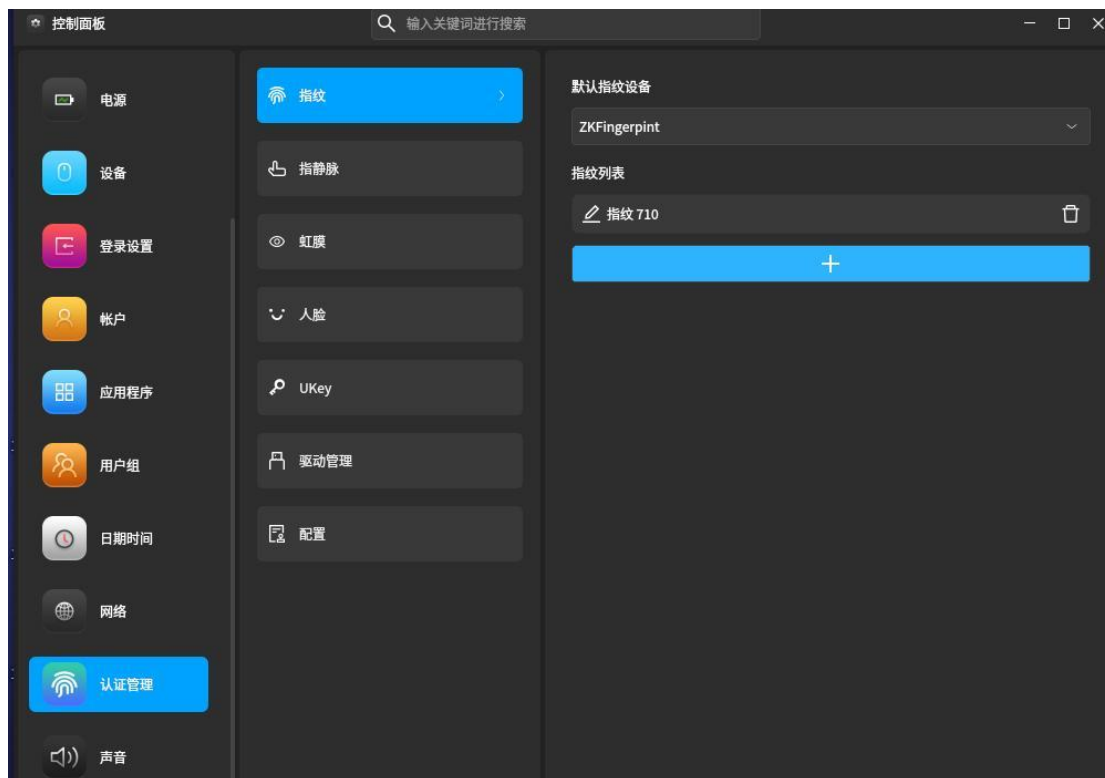


图 8-51 指纹管理列表

8.3 桌面应用

系统提供了多种常用应用工具，点击“开始菜单”按钮可查看到系统上的所有应用，可以将应用程序收藏到收藏夹中，方便打开应用程序。

8.3.1 压缩与解压

归档管理器是一款界面友好、使用方便的压缩与解压缩软件，支持 7z、jar、tar、tar.bz2、tar.gz、tar.lz、tar.lzm、tar.lzo、tar.Z、zip 等多种压缩包格式。

点击“开始菜单”>“压缩与解压”可以启动归档管理器，如图 8-52 所示：

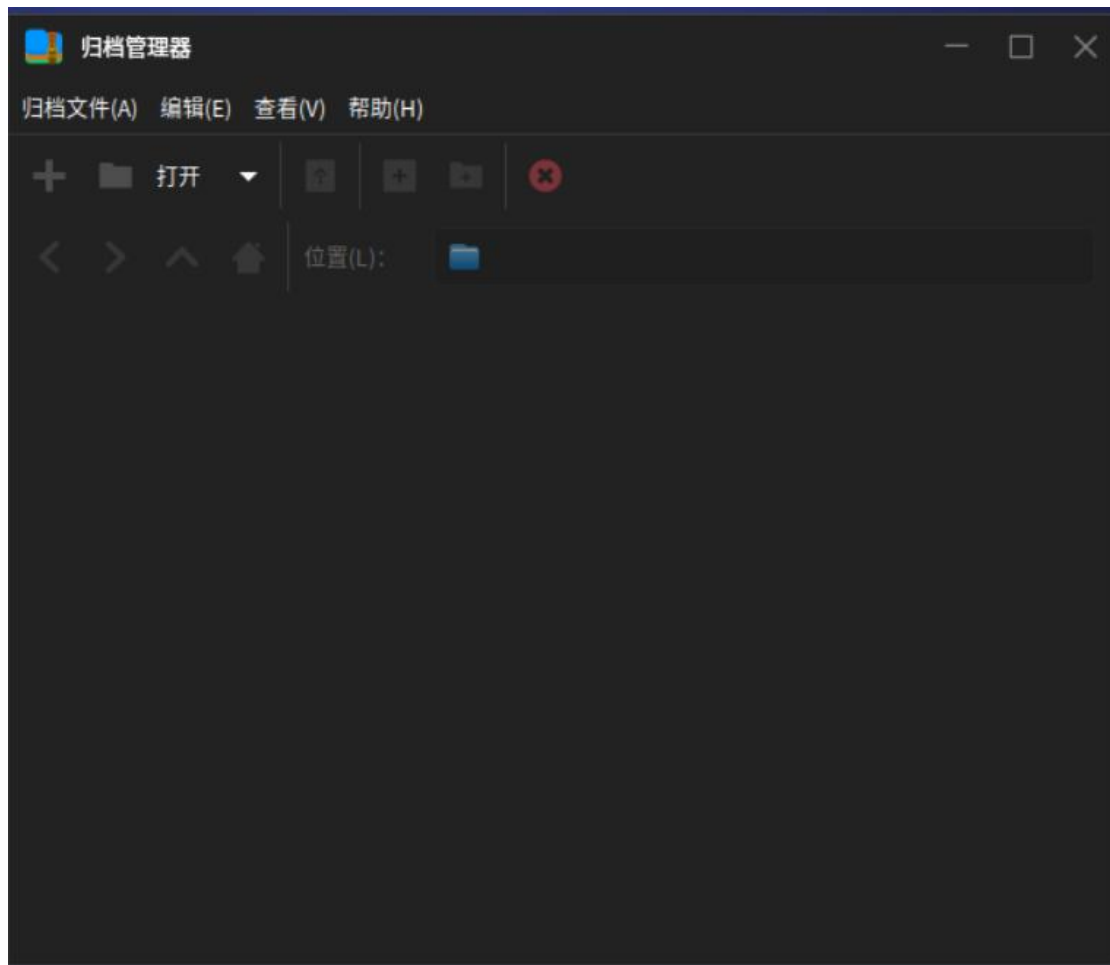


图 8-52 归档管理器工具

创建归档文件：点击“+”按钮新建归档文件，输入名称，选择路径后点击“创建”。

打开归档文件：点击“打开”后选择文件可打开。

解压缩：选中归档文件点击“解压缩”>选择解压目录>“解压缩”。

8.3.2 Atril 文档查看器

Atril 文档查看器是一款简单易用的桌面文档查看程序。

点击“开始菜单”>点击“Atril 文档查看器”>点击“文件”选中需要查看的文件，如图 8-53 所示：

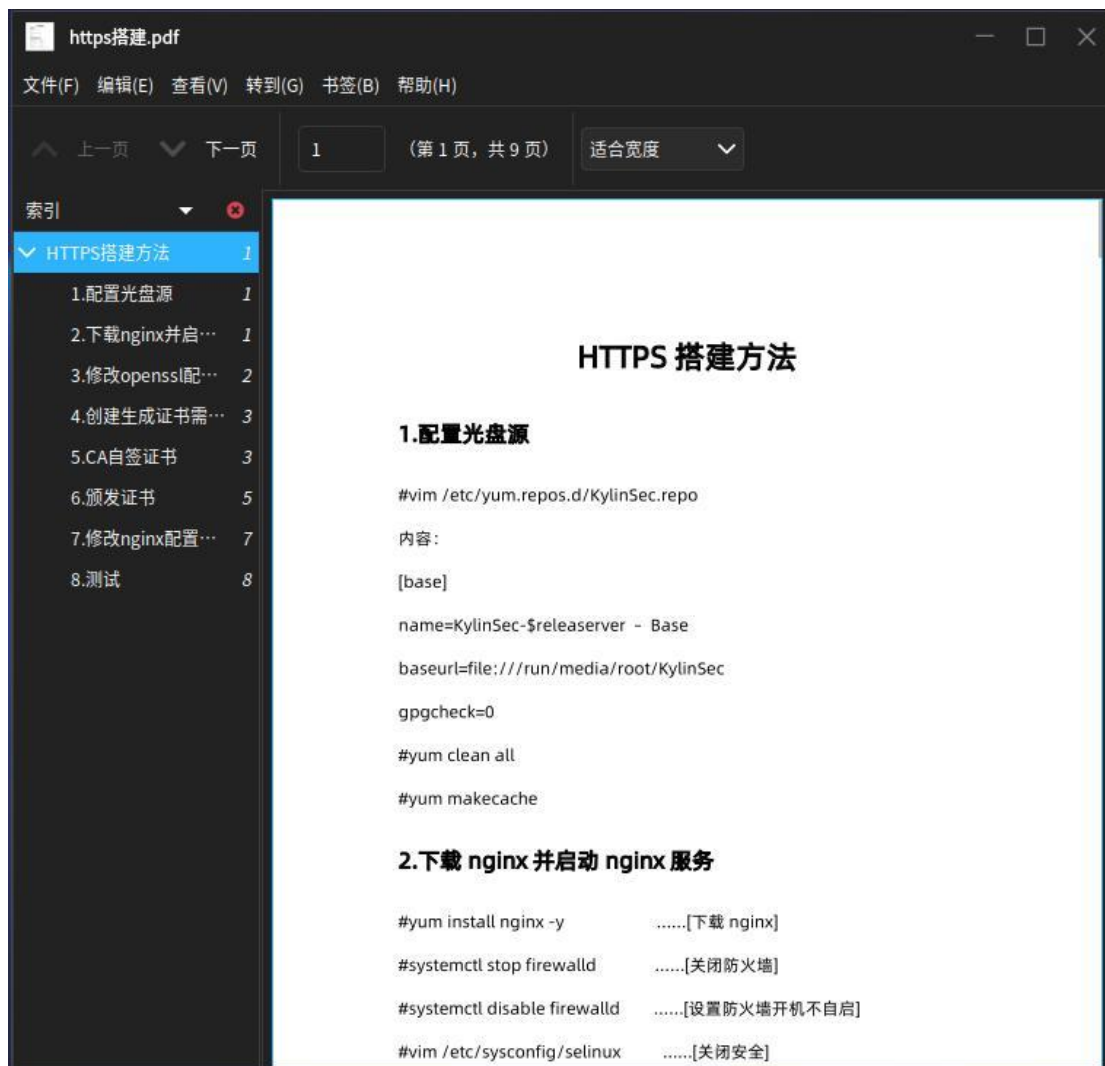


图 8-53 文档查看器

8.3.3 MATE 之眼图像查看器

图像查看器是一款支持多种格式图片（如 BMP、ICO、GIF、JPEG、PNG、SVG、TAG、TIFF、WBMP、XBM、XPM）打开、支持图片放大和缩小的看图工具。

点击“开始菜单”>“图像查看器”打开工具，如图 8-54 所示：

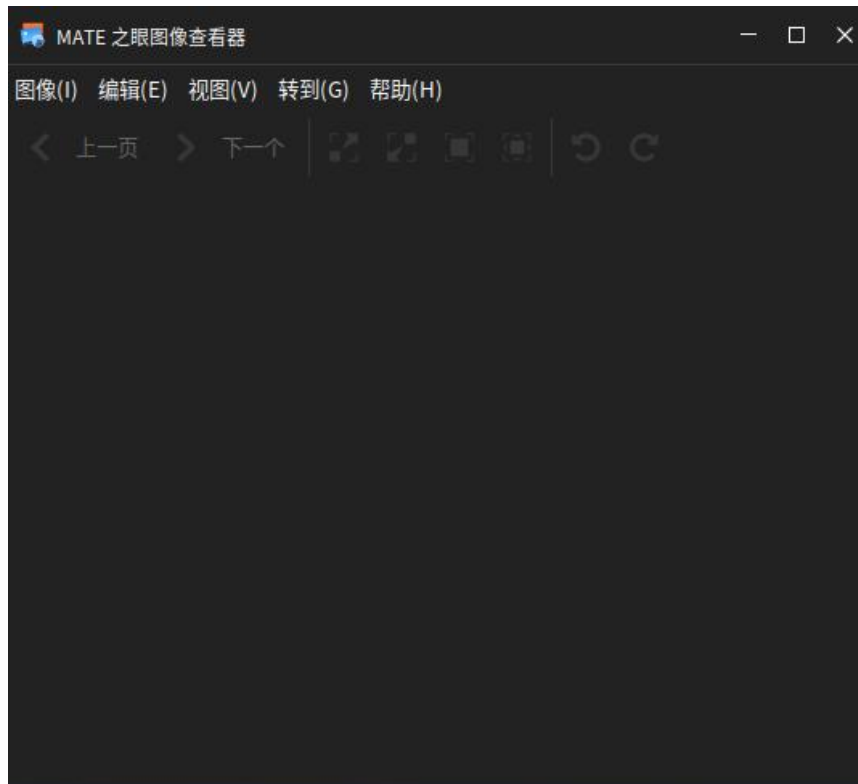


图 8-54 图像查看器

打开图像：点击“图像”>“打开”后弹出文件选择弹窗，选择需要查看的图像，点击“打开”即可。

8.3.4 计算器

计算器一款快捷而简易的计算器，为用户提供加、减、乘、除等基本的数学计算。除了标准模式外，还提供了科学计算和程序员计算功能。

点击“开始菜单”>“计算器”启动工具，也可以在 shell 提示符下键入“kiran-calculator”启动计算器。如图 8-55 所示：

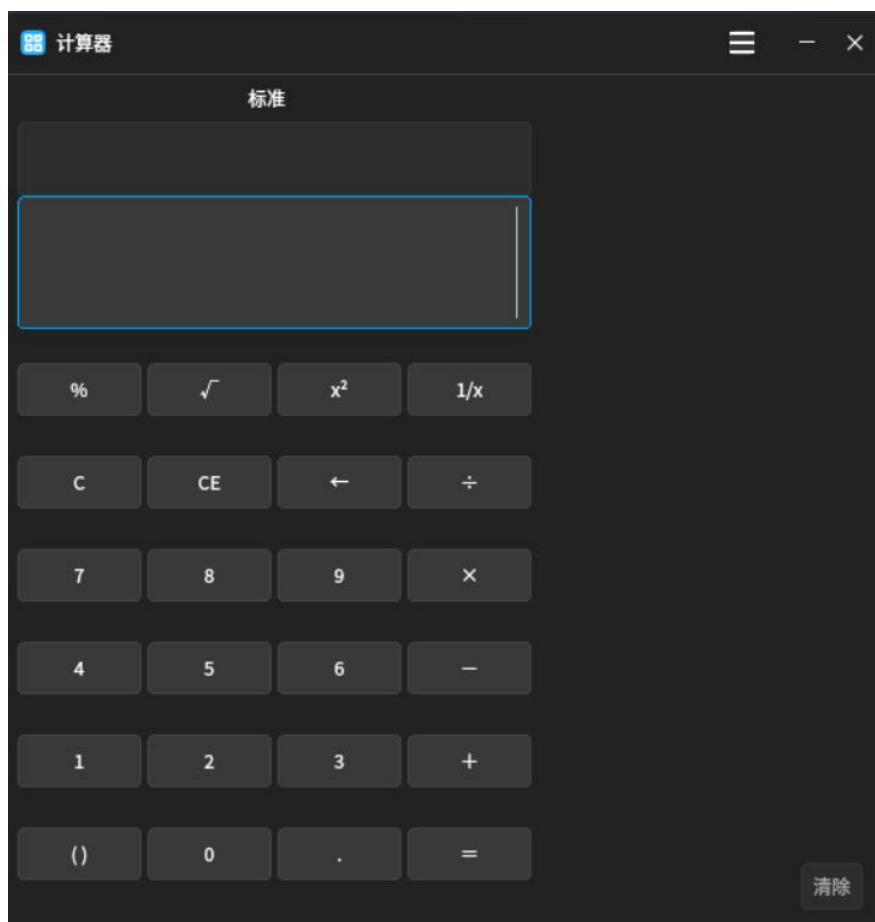


图 8-55 计算器

切换计算模式：点击工具右上角更多图标，点击下拉框中的“标准、科学、程序员”可切换计算模式。

计算：使用鼠标键入或点击计算机中数字和运算符，键入回车或点击“=”可计算出结果。

8.3.5 PulseAudio 音量控制

PulseAudio 音量控制可以设置设备音量大小。

点击“开始菜单”>“PulseAudio 音量控制” 启动工具，如图 8-56 所示：

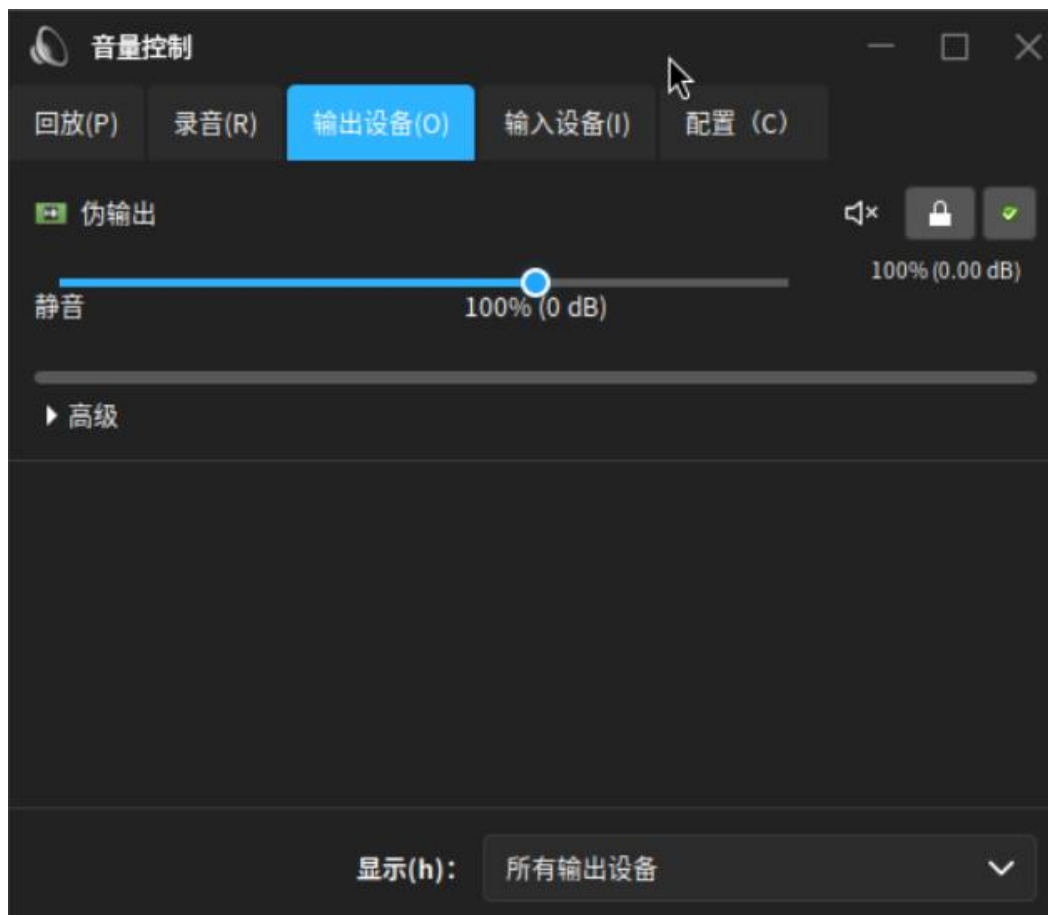
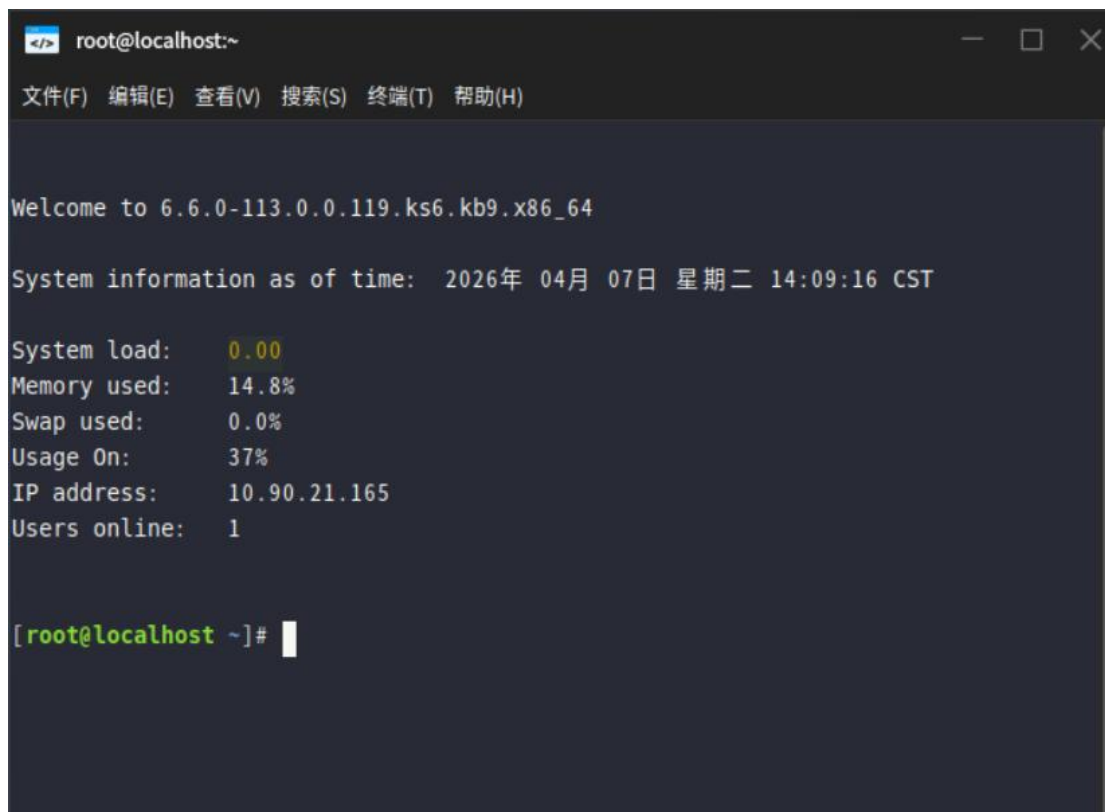


图 8-56 PulseAudio 音量控制

8.3.6 终端

终端是麒麟信安服务器操作系统使用系统命令操作的媒介,通过在终端窗口键入系统命令实现与系统交互。

点击“开始菜单”>“终端”或者系统面板上的图标,可以打开终端。如图 8-57 所示:



```
root@localhost:~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

Welcome to 6.6.0-113.0.0.119.ks6.kb9.x86_64

System information as of time: 2026年 04月 07日 星期二 14:09:16 CST

System load:      0.00
Memory used:      14.8%
Swap used:         0.0%
Usage On:          37%
IP address:        10.90.21.165
Users online:      1

[root@localhost ~]#
```

图 8-57 终端

操作窗口以上图为例，root@localhost~，其中：

root：登录系统的用户名

localhost：计算机名

~：当前打开终端时的路径

退出终端程序，点击窗口右上角的关闭按钮，或在键入 exit，也可按快捷键 <Ctrl + D>。

8.3.7 磁盘

磁盘是一款可查看、修改和配置磁盘与媒体的工具，可以通过该工具创建和恢复磁盘映像，也可对磁盘进行分区和格式化操作。

点击“开始菜单”>“磁盘”打开磁盘工具，如下图 8-58 所示：

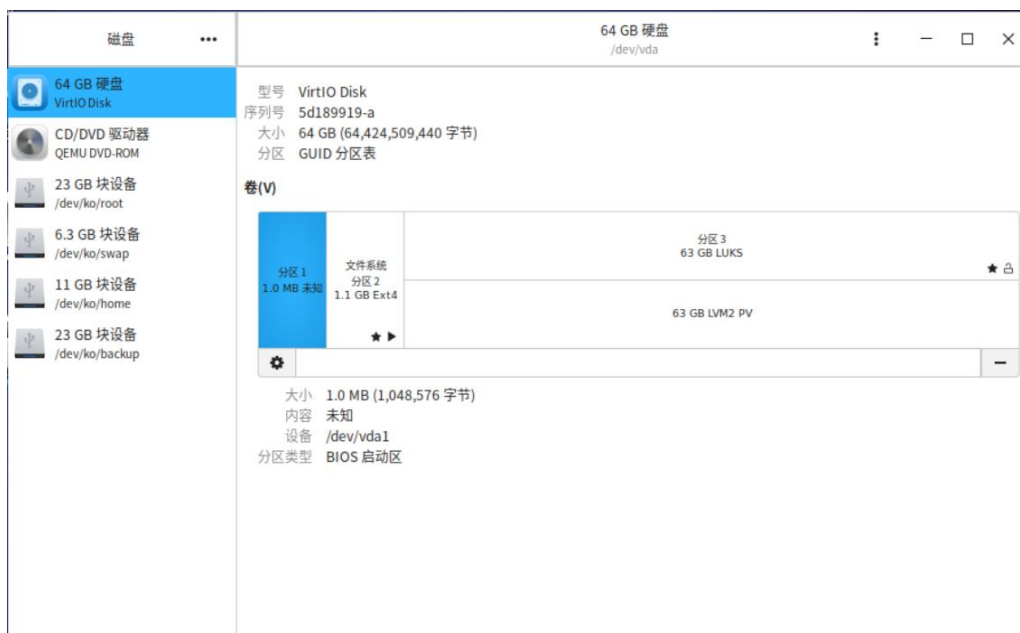


图 8-58 磁盘使用界面

添加分区：选中空闲的磁盘空间，点击“+”按钮弹出新建分区弹窗，输入分区名称、大小等信息后确定即可添加成功。

修改分区类型：选中待修改的分区，点击设置按钮>“编辑分区”，修改分区类型后确定即可修改成功。

8.3.8 系统日志查看器

日志工具显示了系统运行时产生的日志，可以通过键入搜索词来搜索日志，并通过单击查看有关每个事件的详细信息。

点击“开始菜单”>“日志”可打开日志工具，如下图 8-59：

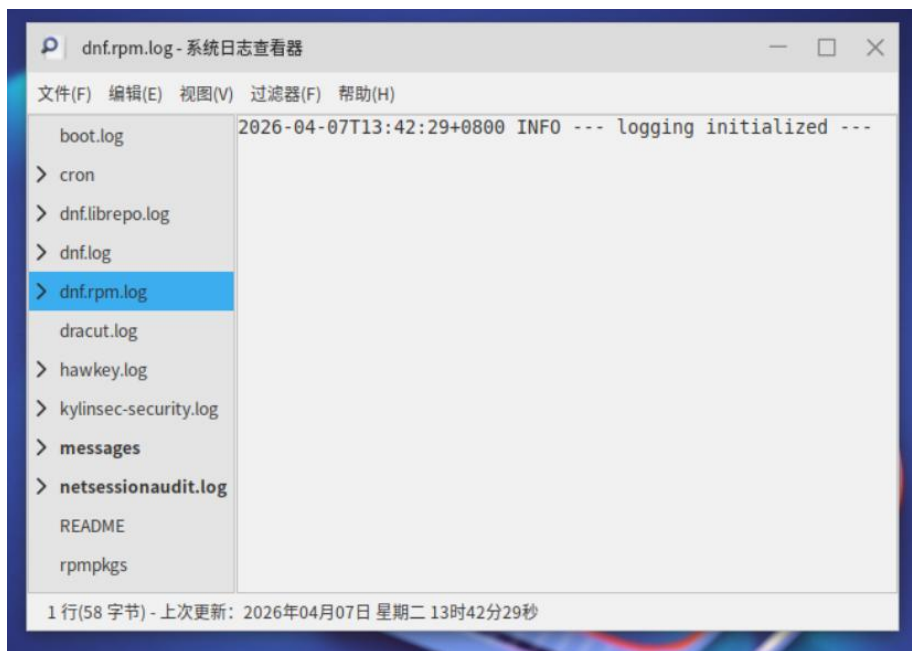


图 8-59 日志

8.3.9 密码和密钥

密码和密钥工具能创建和存储安全令牌，确保用户在计算机上的信息安全，并在用户通过网络发送信息时确保安全。可以复制并保存难以记住的密码。

点击“开始菜单”>“密码和密钥”启动工具，如下图 8-60：

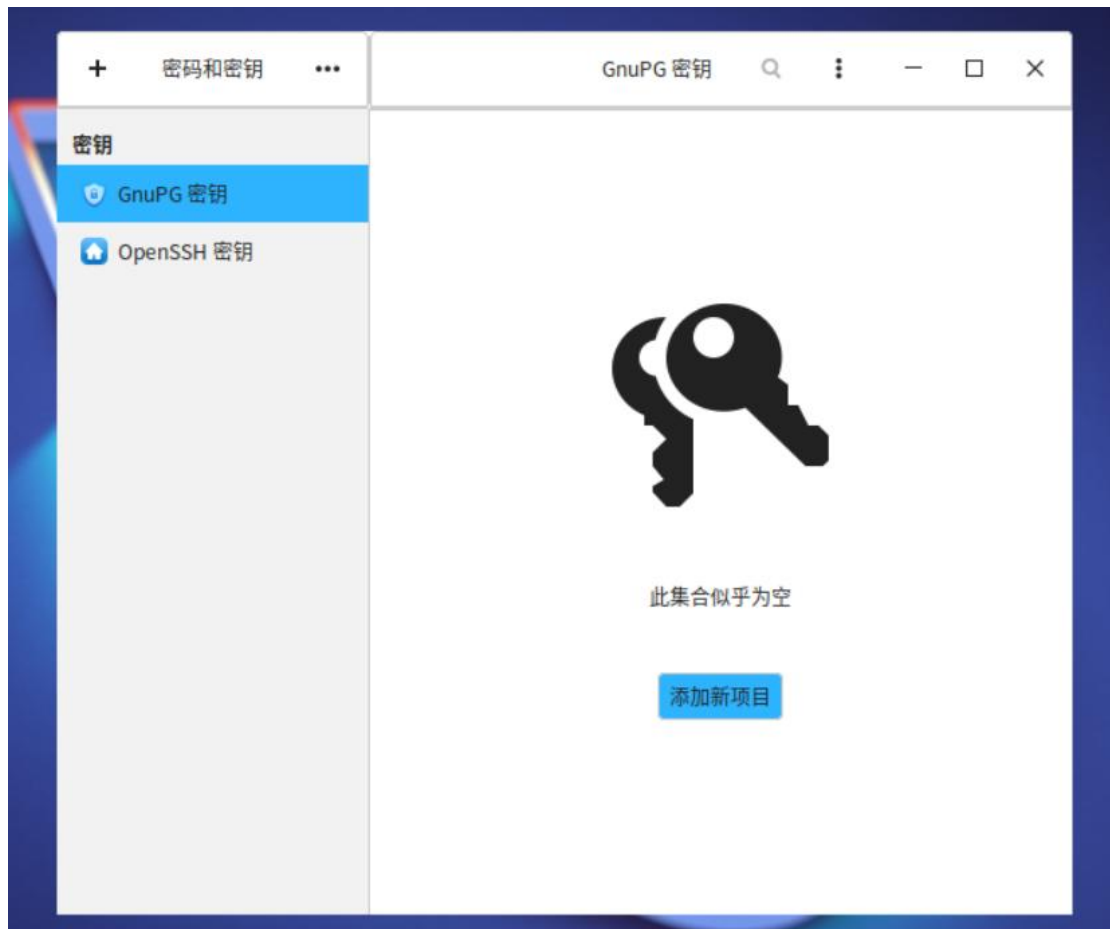


图 8-60 密码和密钥

8.3.10 文本编辑器

文本编辑器是一款快速记录文字的文档编辑工具,用于查看和修改纯文本文件。

点击“开始菜单”>“文本编辑器”启动工具,或者在终端键入 pluma 启动。

工具如图 8-61 所示:

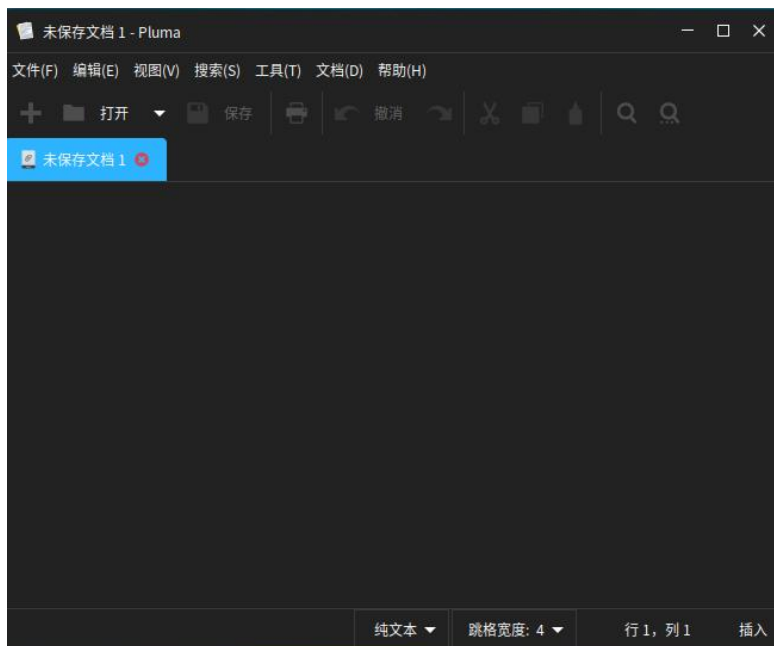


图 8-61 文本编辑器

8.3.11 键盘布局测试器

键盘布局测试器，帮助测试键盘布局。

点击“开始菜单”>“键盘布局测试器”如图 8-62 所示：

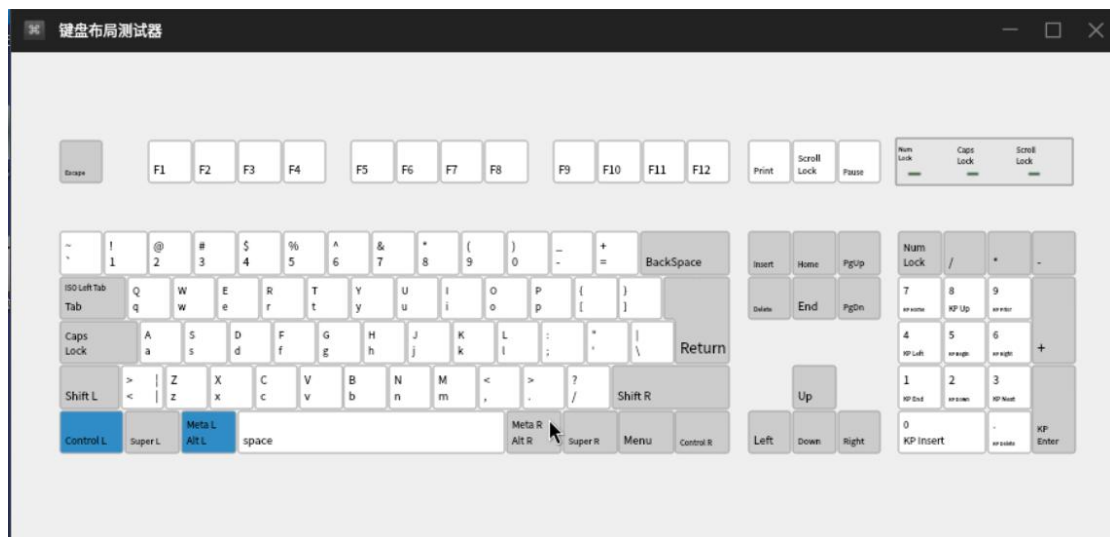


图 8-62 键盘布局测试器

8.3.12 Firefox 浏览器

Firefox 浏览器，是一个自由及开放源代码网页浏览器，使用 Gecko 排版引

擎，支持多种操作系统。它体积小速度快，还有其他一些高级特征，主要特性有：标签式浏览、使用网上冲浪更快、可以禁止弹出式窗口、自定制工具栏、扩展管理、更好的搜索特性、快速而方便的侧栏。

在软件源中通过 `dnf` 命令下载安装 Firefox 后，点击“开始菜单”>“Firefox 浏览器”启动浏览器工具，如图 8-63：

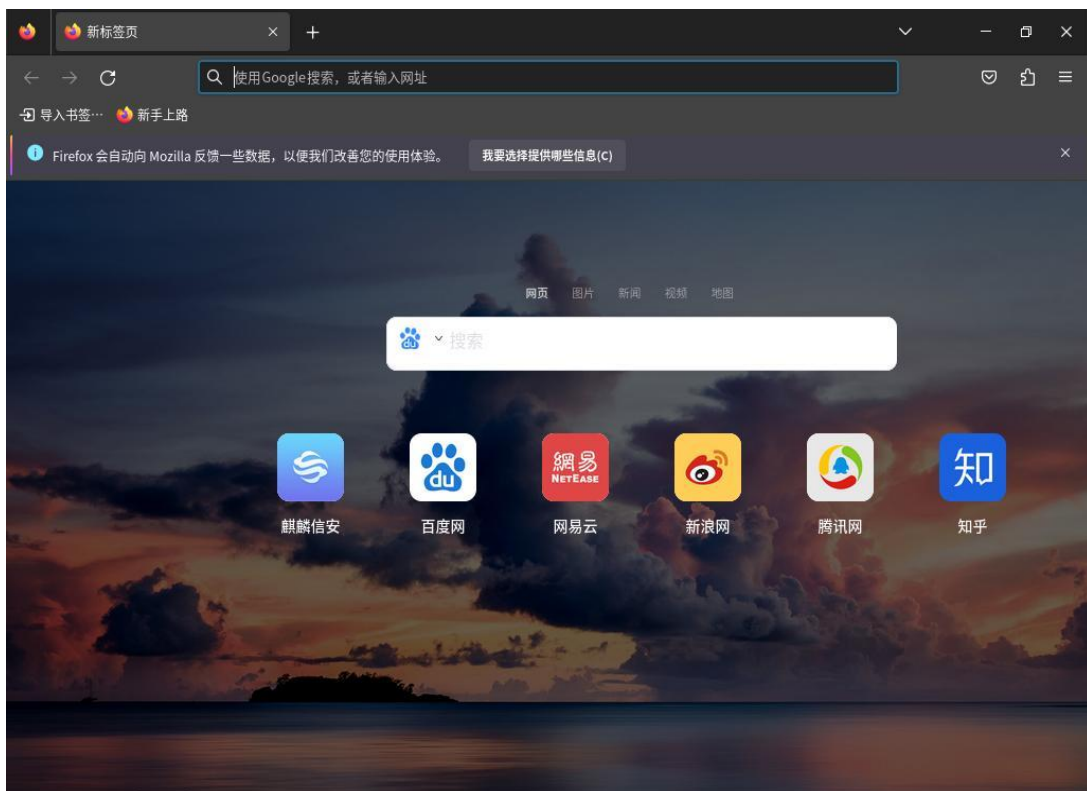


图 8-63 Firefox 浏览器

访问 Web 资源：在地址栏中输入想访问的 Internet 地址或 URL，然后按 Enter，稍等片刻即可转到您想访问的页面。

8.3.13 Fcitx 5 配置

Fcitx 5 配置是可以让用户个性化地使用 Fcitx 输入法而进行的设置和调整，可以根据个人使用习惯进行设置，让它更符合个人使用。

点击桌面底部导航栏 >  打开工具，如下图 8-64：

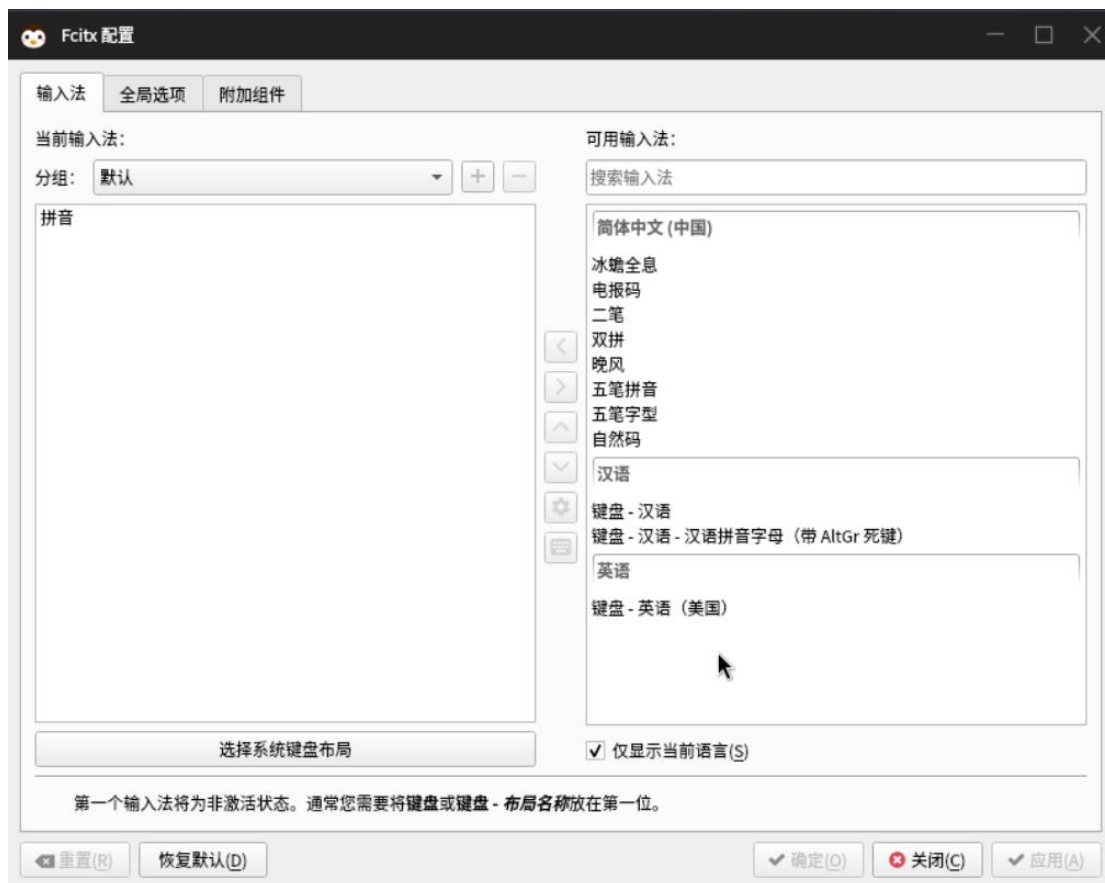


图 8-64 Fcitx 5 配置

8.3.14 截图工具

截图工具是麒麟信安服务器操作系统自带的一款小巧灵活的屏幕捕捉软件，操作界面简洁、使用极为方便。该软件启动时会在托盘处添加截图工具图标，点击该图标后，直接弹出屏幕捕捉界面，可自行选择截图范围。可通过右击该图标打开“打开启动器”，可选择需要抓取的范围是整个桌面，或者方形区域，可设置截图延迟时间。

点击“开始菜单”>“截图工具”，可以启动截图工具。点击托盘区域的截图图标开始手动选择区域截图，按住鼠标左键拖曳出矩形区域，松开鼠标即可完成截图，如下图 8-65 所示：

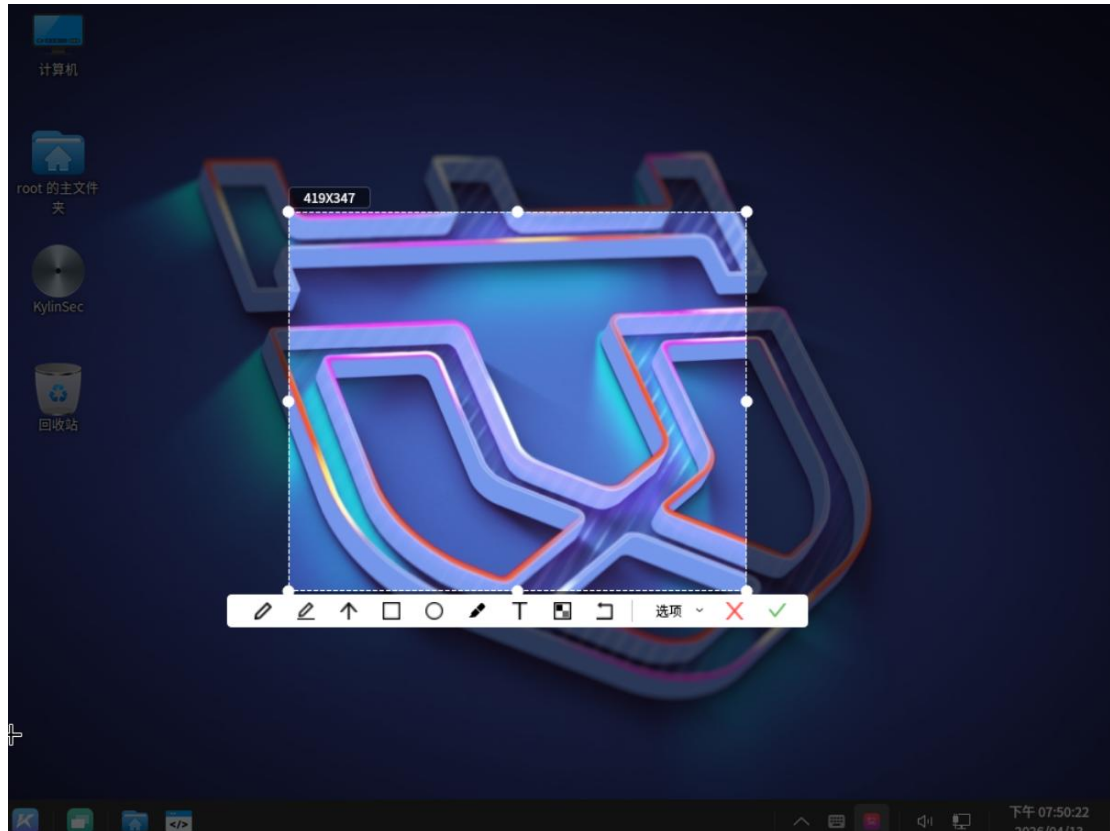


图 8-65 截图界面

工具可在截图上进行编辑，如绘画、添加矩形框、马赛克、文字等，点击“选项”可选择存储位置和保存格式，编辑完成后点击“√”即可保存，点击“×”则取消截图。

鼠标右键托盘区域的截图图标，点击“打开启动器”弹出如下图 8-66 所示弹窗：



图 8-66 启动器界面

在启动器可设置截图区域是方形区域或全屏，还可设置截图延迟时间，输入整数秒数后，点击“获取新屏幕截图”将会按照设置截图。

8.3.15 防火墙

防火墙是位于计算机和它所连接网络之间的安全防线，依据预设规则决定允许或阻止特定数据通过。

点击“开始菜单”>“防火墙”，打开防火墙配置页面，可以对服务、端口、协议等选项进行配置，如图 8-67 所示：

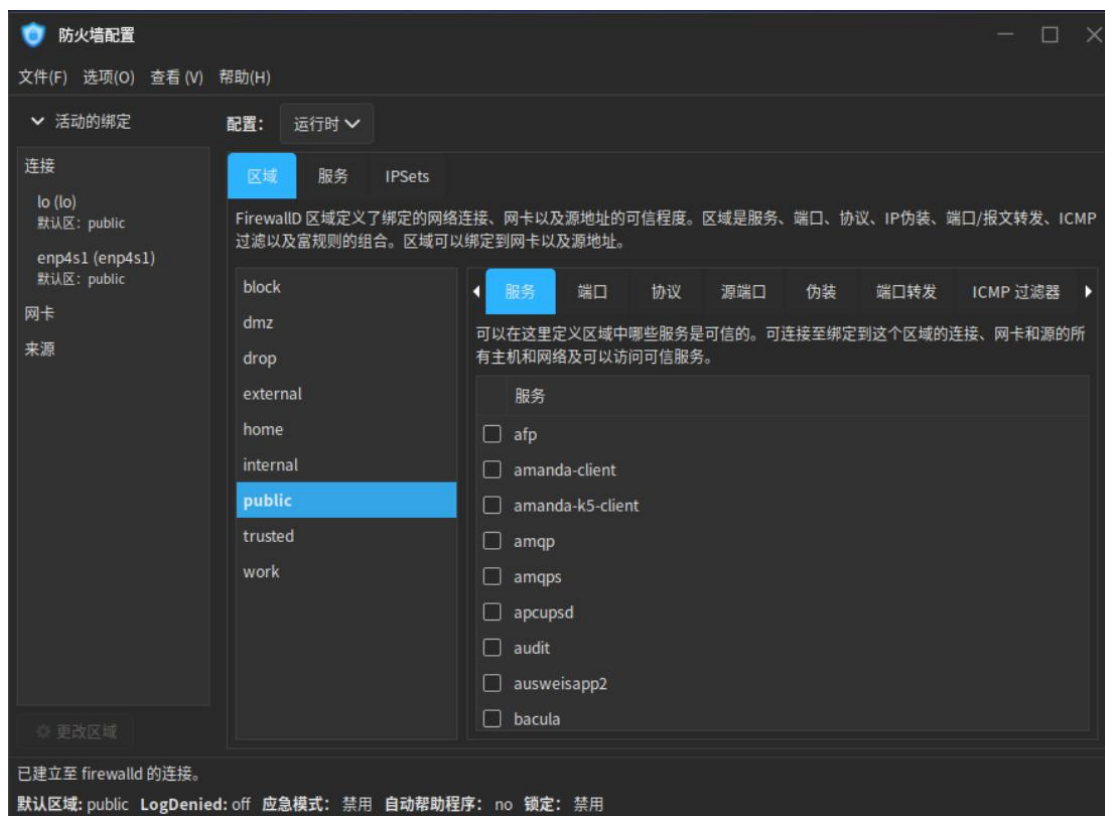


图 8-67 防火墙

8.3.16 Dconf 编辑器

Dconf 编辑器是应用程序内部设置的图形化查看器和编辑器。

点击“开始菜单”>“Dconf 编辑器”，可以启动 Dconf 系统配置编辑器，如图 8-68 所示：

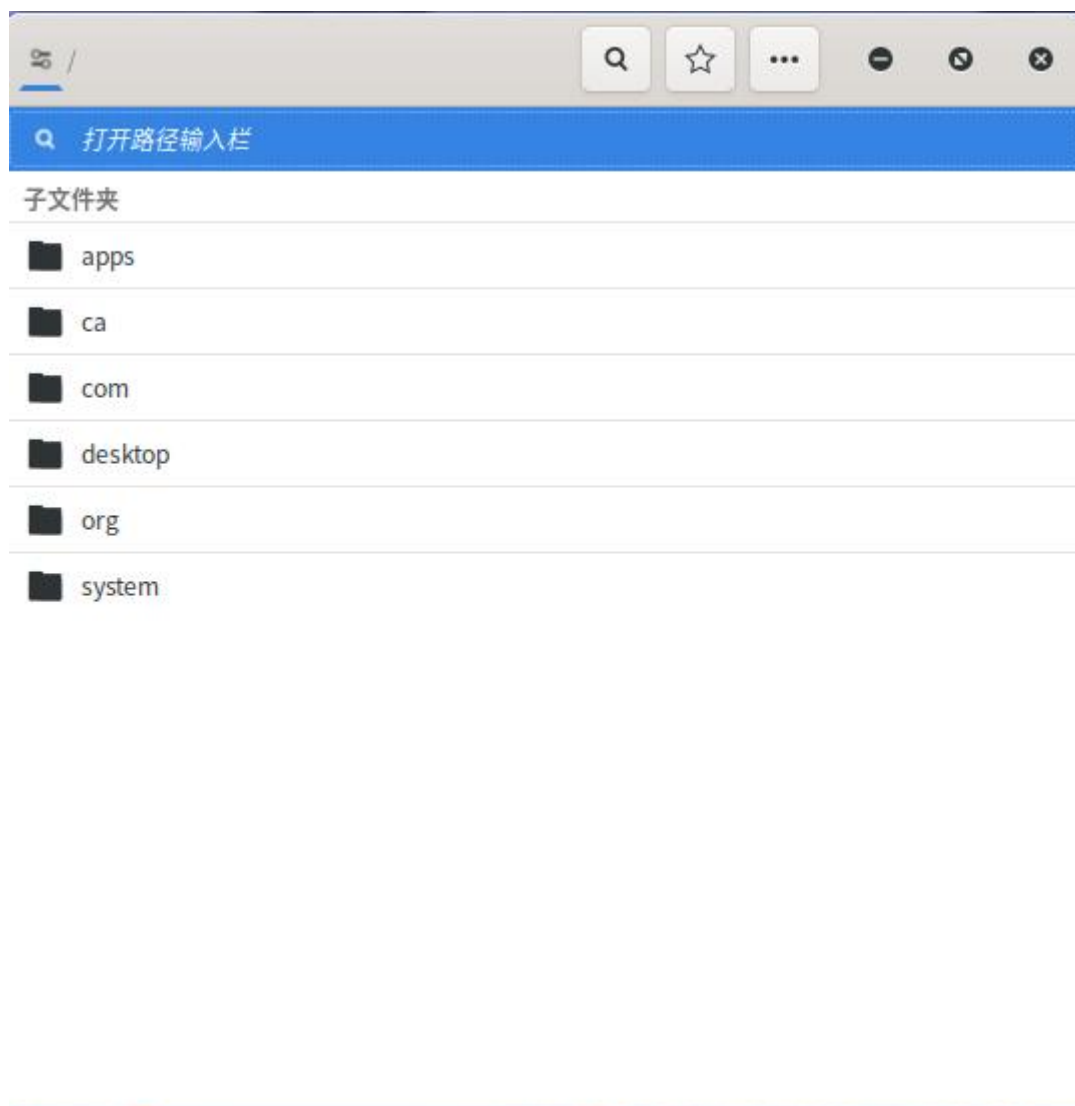


图 8-68 dconf 编辑器

8.3.17 打印设置

系统除了支持终端软件应用外，也与常见的打印机品牌和类型进行了适配，方便办公应用，支持添加多个打印机外设，并且支持网络打印机设备的使用；使用打印功能前需下载对应的驱动。

添加和管理打印机设备：系统使用了强大和易于配置的 cups 打印子系统。除了支持的打印机类型更多，配置选项更丰富外，cups 还能设置并允许任何联网的计算机通过局域网访问单个 cups 服务器。

点击“开始菜单”>“打印设置”可以打开设置页面，如下图 8-69 所示：



图 8-69 打印设置

8.3.18 软件用户手册、软件管理员手册

系统默认集成软件用户手册与软件管理员手册，如需查询相关系统、服务搭建信息，点击开始菜单>所有应用>软件用户手册/软件管理员手册即可查看，如图 8-70 所示：

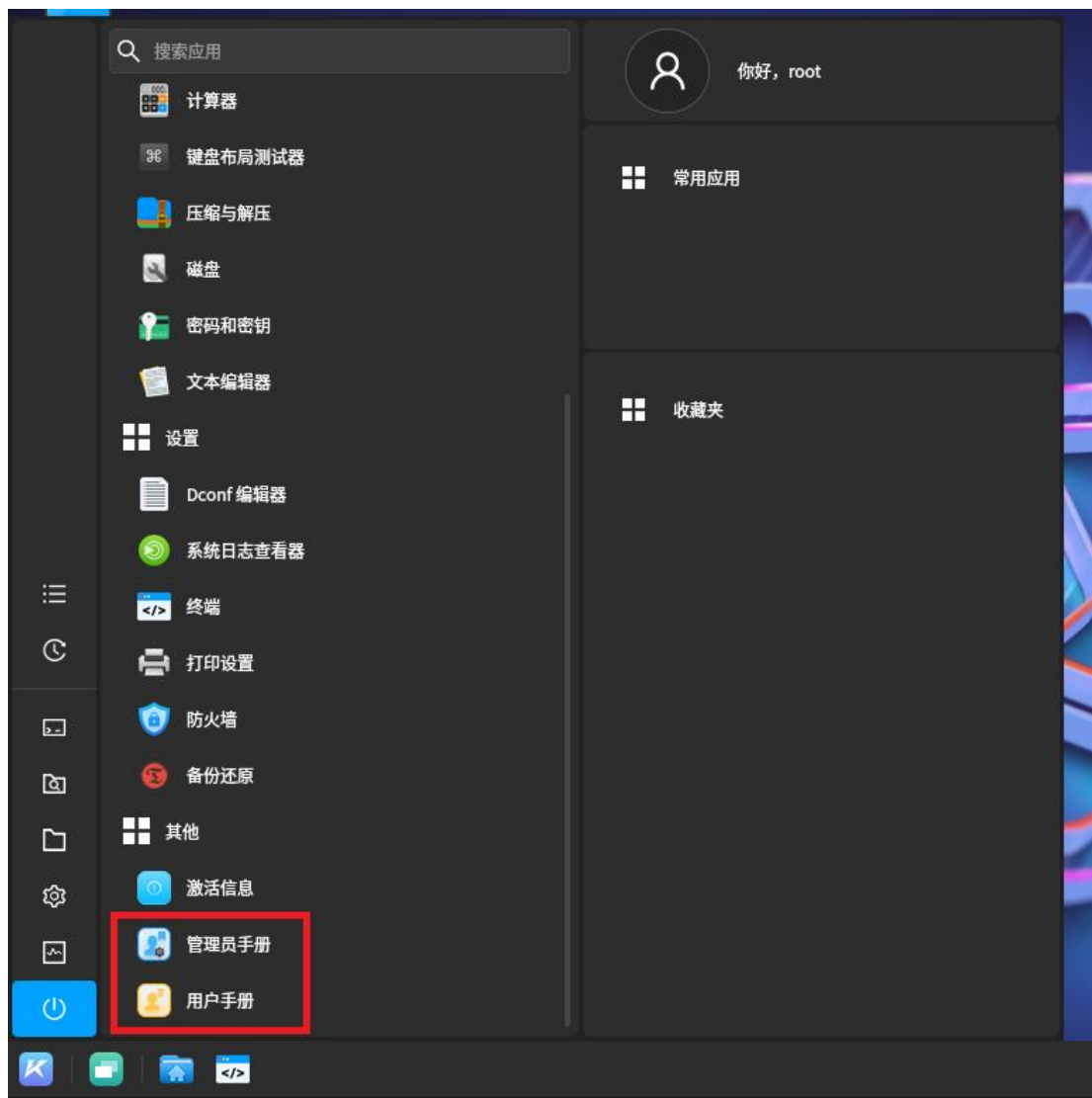


图 8-70 软件管理员手册和软件用户手册

8.3.19 激活信息

点击“开始菜单”>“激活信息”可以查看系统激活状态。如图 8-71 所示：



图 8-71 激活信息

8.4 安全中心

8.4.1 软件概述

将麒麟信安操作系统的安全功能进行产品化, 开发一款图形界面安全中心组件, 用于安全模块相关功能的控制、配置和展示, 增强安全模块功能在麒麟信安操作系统上使用的灵活性和便捷性。

8.4.2 软件综述

8.4.2.1 操作系统版本及硬件配置

请确定操作系统版本、硬件配置满足主机安全中心 V1 的运行要求, 如表 8-4 所示:

表 8-4 系统要求

内容	参数
ISO	KylinSec-Server-6-SP1
安装系统的设备	服务器/台式机/虚拟机
磁盘空间	50GB 及以上剩余空间
CPU	2 核 2.5GHz 及以上 (2 核及以上)
内存	2GB 及以上

8.4.2.2 软件安装

表 8-5 软件清单

序号	软件
1	kss-xxx.rpm
2	ks-sc-1.1.x-xxx.rpm

安装步骤说明：

配置 V6 SP1 系统的软件源。

使用 `yum install ks-sc` 安装安全中心软件，重启系统。

8.4.3 软件功能

8.4.3.1 设置

8.4.3.1.1 设置

点击右上角，点击【设置】

打开【设置】界面，打开【可信保护】开关，可信数据开始初始化，等待一段时间后重启机器，后续可使用可信保护功能模块，如图 8-72 和图 8-73：

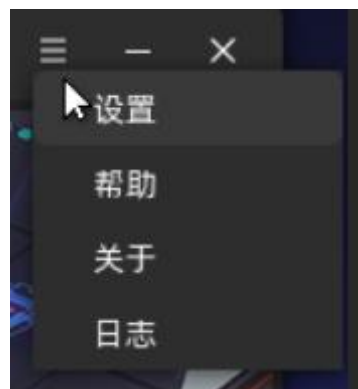


图 8-72 设置列表



图 8-73 设置窗口

8.4.3.1.2 帮助

点击右上角, 点击【帮助】可打开软件用户手册。

8.4.3.1.3 关于

点击右上角, 点击【关于】弹出软件版本信息, 如图 8-74:



图 8-74 关于窗口

8.4.3.1.4 日志

点击右上角, 点击【日志】弹出安全中心软件相关所有操作日志。如图

8-75:

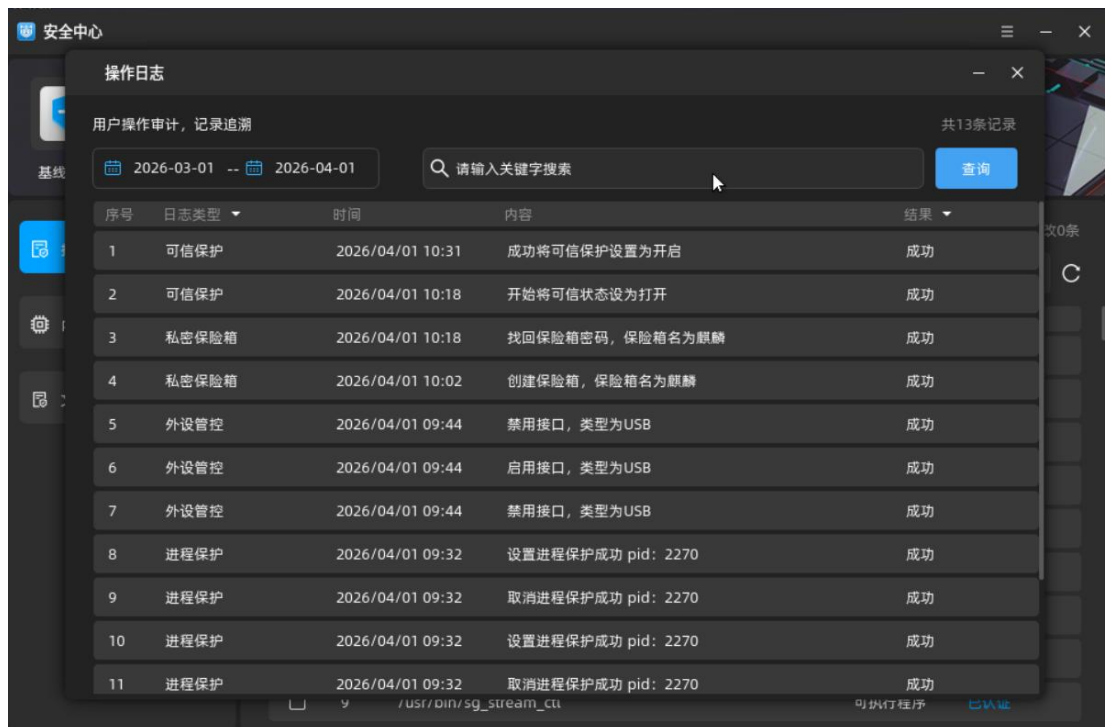


图 8-75 操作日志窗口

8.4.3.2 基线加固

基线加固功能提供系统安全配置的检测与修复能力。

扫描主要是为了对比加固项配置是否与系统标准一致。

加固前必须先进行一次扫描, 扫描前可以选择策略。若加固项配置与系统标准一致时, 扫描结果显示为“符合”, 加固项配置与系统标准不一致时, 扫描结果显示为“不符合”。

8.4.3.2.1 快速扫描

点击【快速扫描】勾选需要扫描的加固项, 点击【开始扫描】: 扫描完成, 进度为100%, 扫描开始时间和用时正确, 各个加固项的状态为“符合”或“不符合”, 如图 8-76和图 8-77:



图 8-76 基线加固

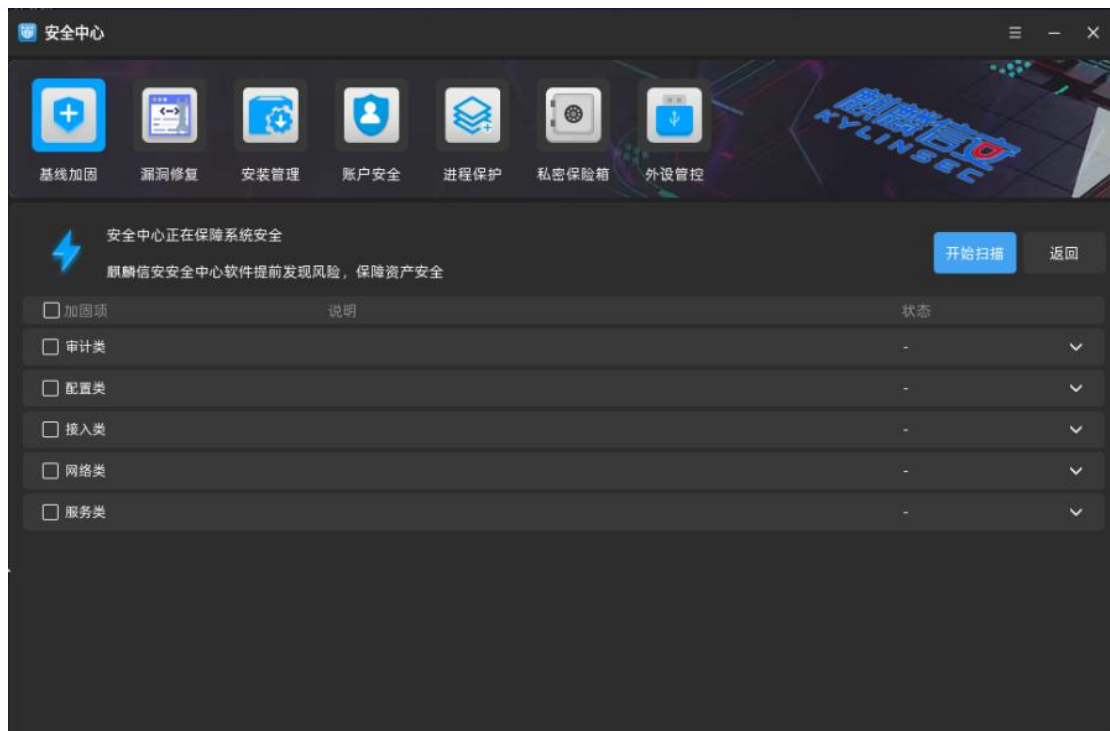


图 8-77 快速扫描

8.4.3.2.2 开始加固

快速扫描后点击【加固】会对所有加固项按照系统标准进行加固，如图图

8-78:



图 8-78 加固

8.4.3.2.3 生成报表

点击【生成报表】，生成报表的文件名称格式为【安全中心加固报告-主机名-主机 IP-时间戳】，如图 8-79 和图 8-80：



图 8-79 生成报表



图 8-80 生成报表位置

8.4.3.3 可信保护

8.4.3.3.1 执行文件保护

点击【可信保护】侧边栏中的【执行文件保护】

点击【添加】选择文件

点击【打开】，文件添加至白名单列表，对可执行文件进行防篡改保护，如

图 8-81 和图 8-82：

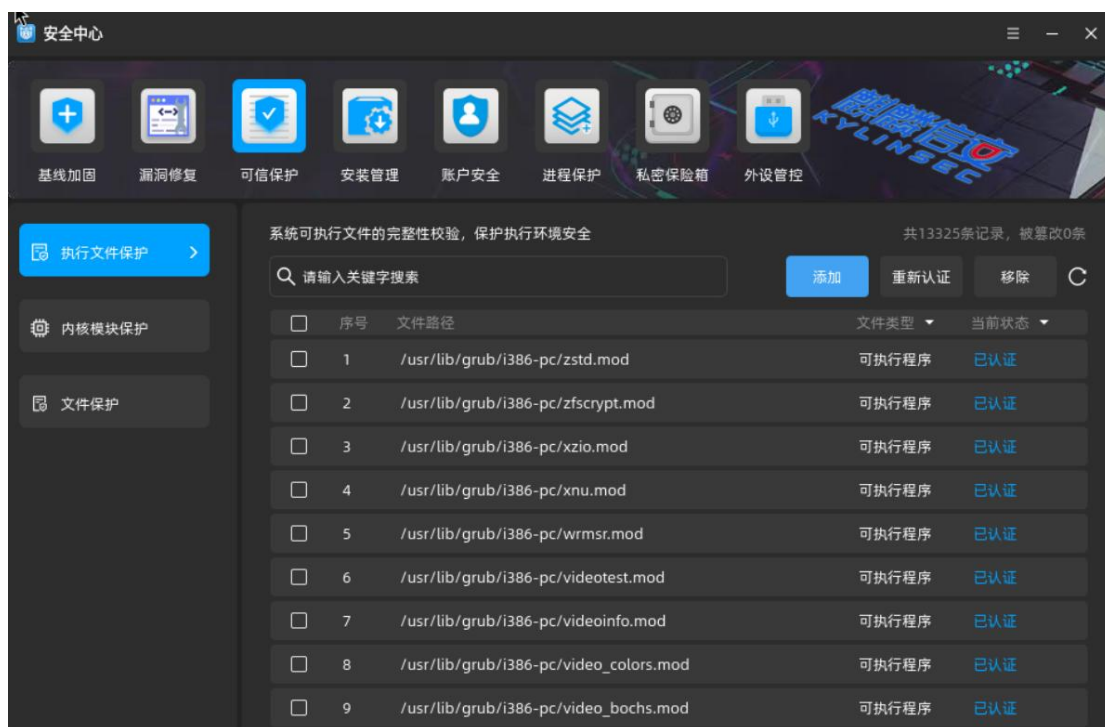


图 8-81 执行文件保护

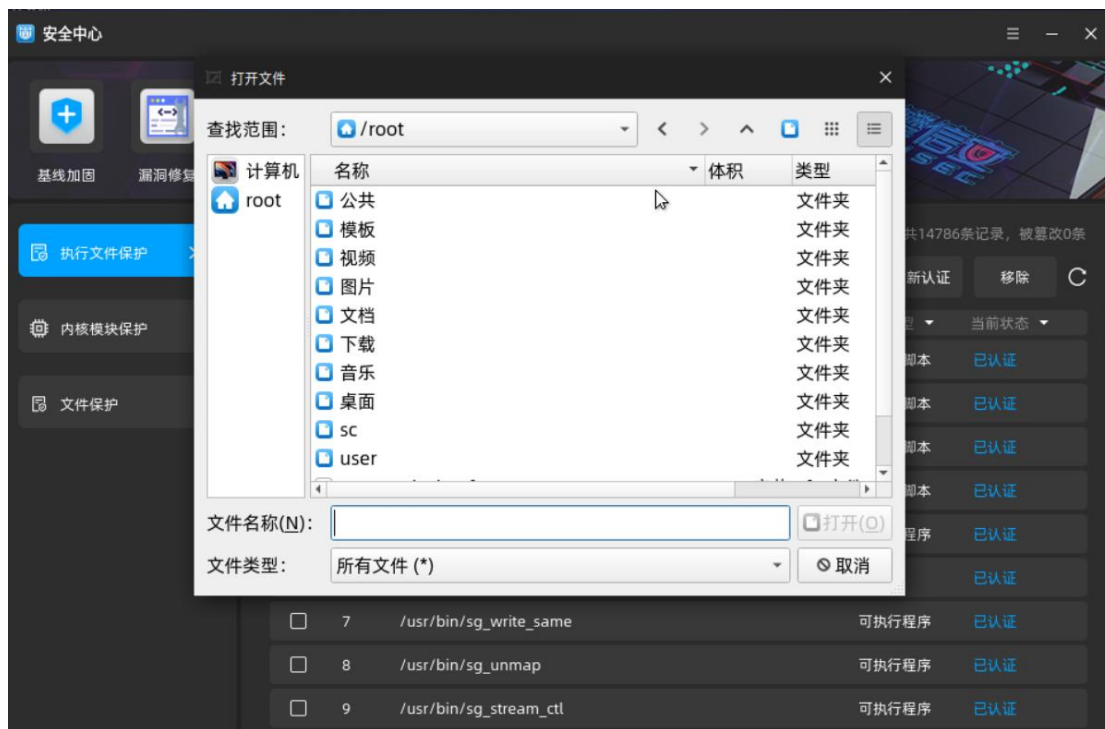


图 8-82 执行文件保护-添加文件

执行文件保护列表中文件编辑后无法执行且列表中会标注文件已被篡改, 如

图 8-83:

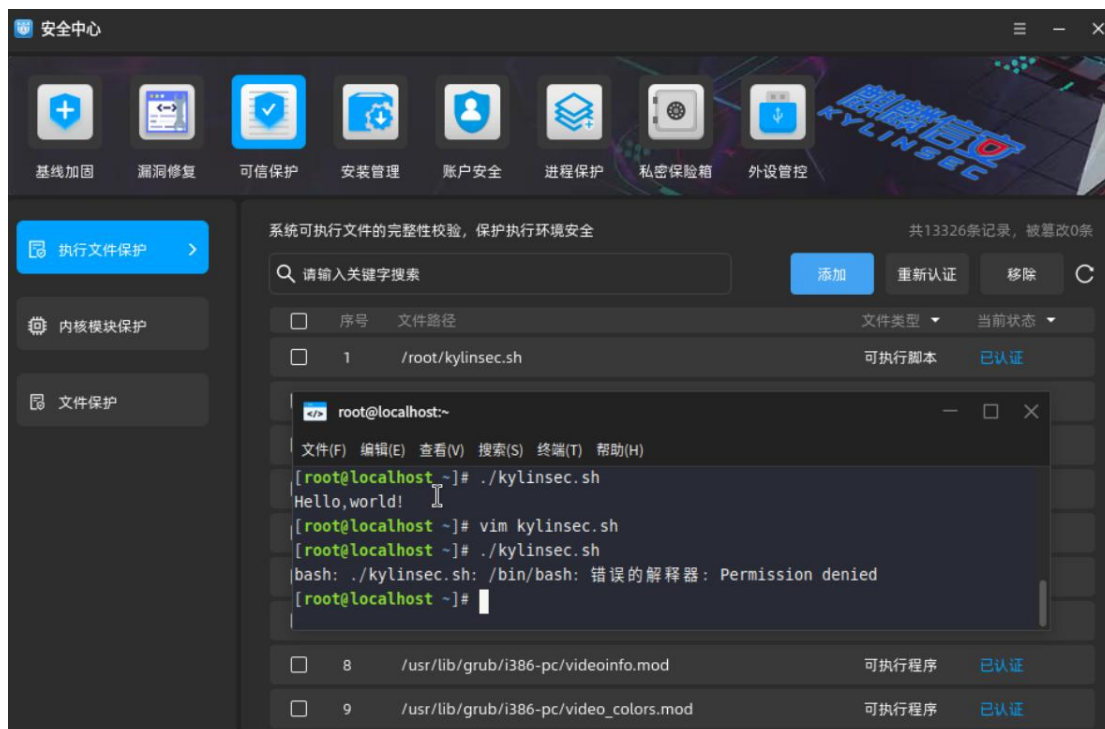


图 8-83 编辑执行文件

8.4.3.3.2 内核模块保护

点击【可信保护】侧边栏中的“内核模块保护”

点击【添加】选择文件

点击【打开】，文件添加至白名单列表,选择需要保护的文件，打开右侧的【防卸载】开关，所选择的文件受到保护，该文件的内核驱动模块无法卸载，如图 8-84、图 8-85 和图 8-86：

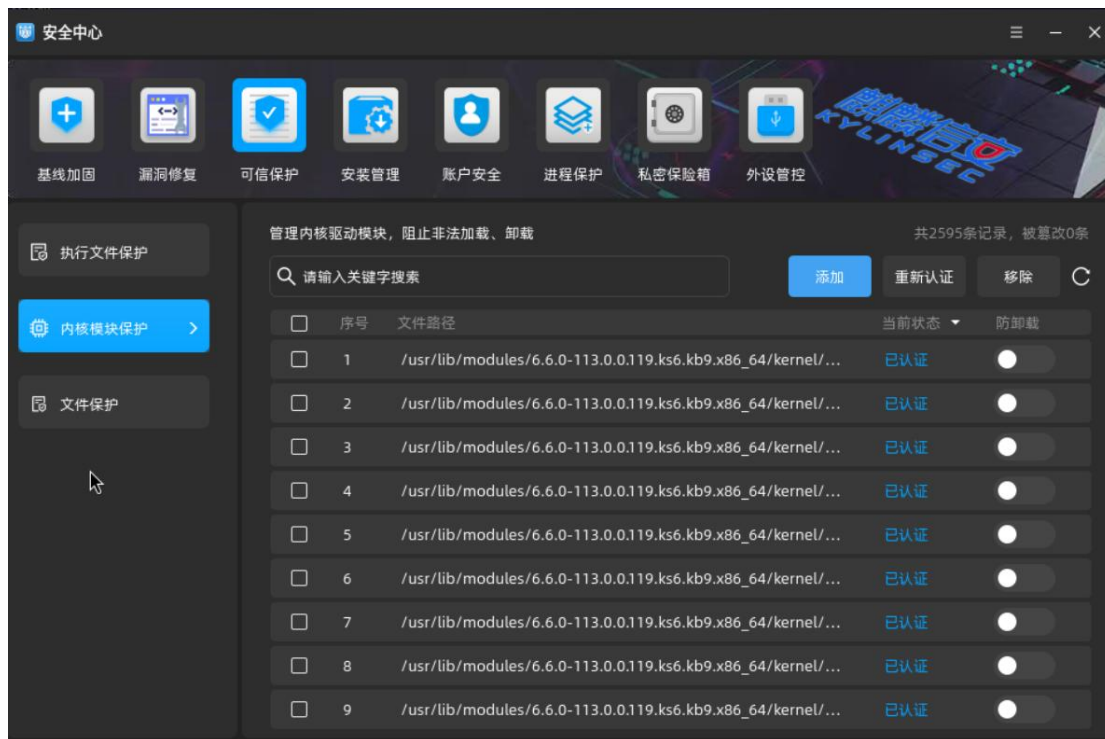


图 8-84 内核模块保护

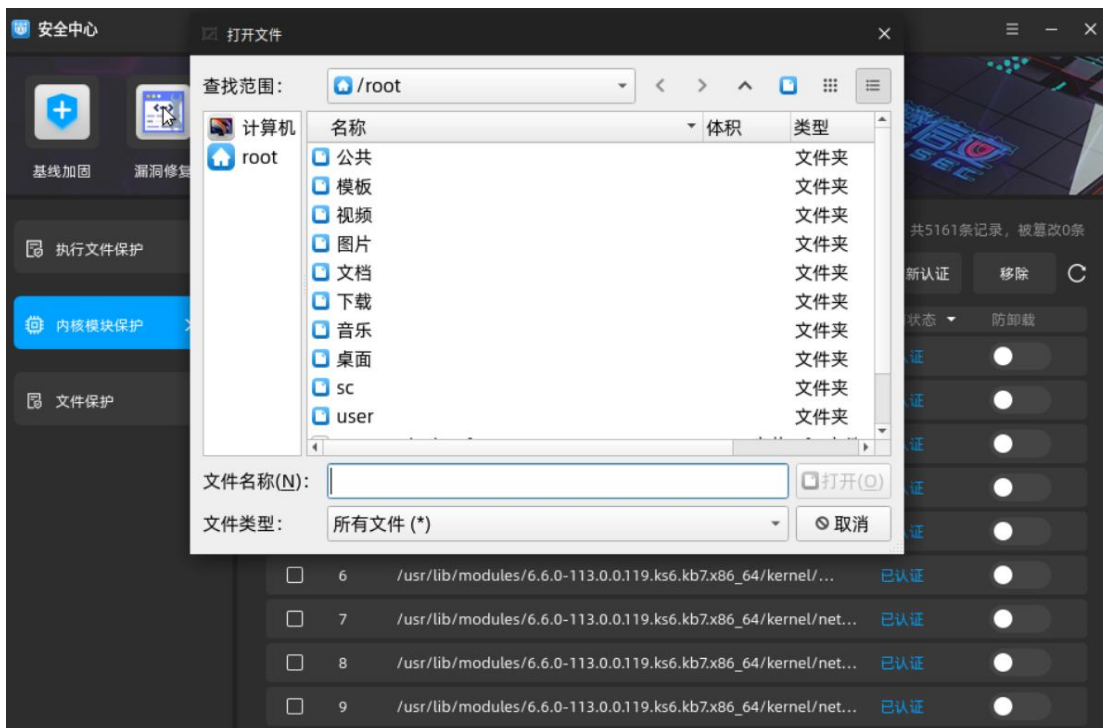


图 8-85 内核模块保护-添加文件

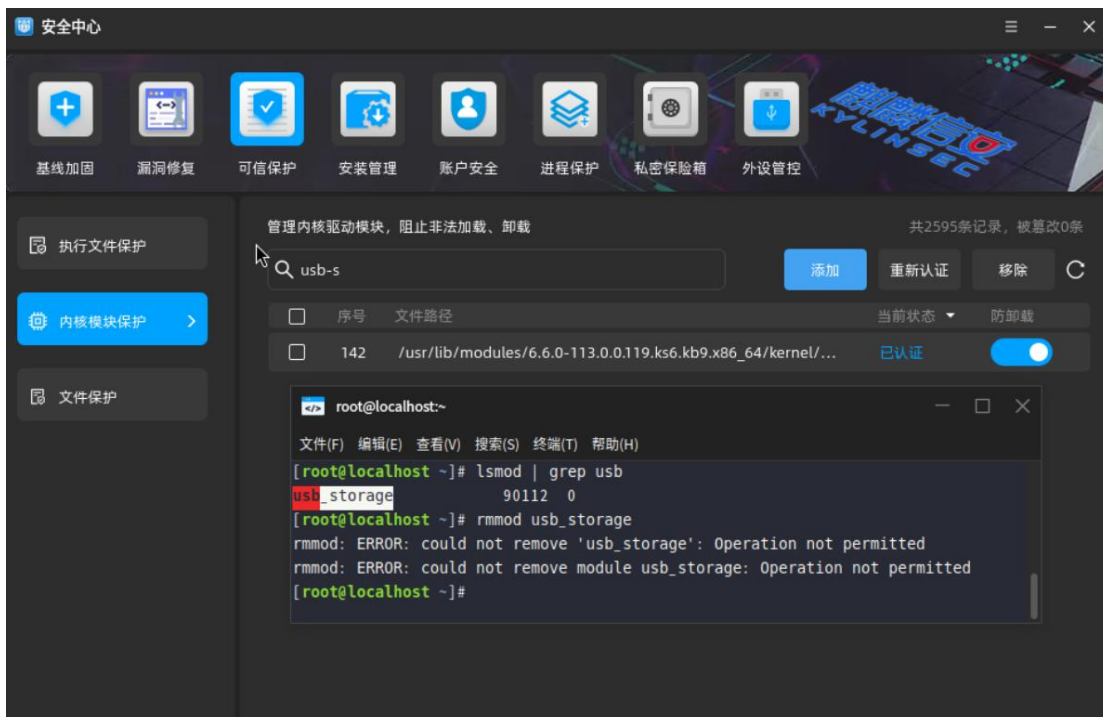


图 8-86 无法卸载已开启防卸载的模块

8.4.3.3 文件保护

点击【可信保护】侧边栏中的【文件保护】

点击【添加】选择文件

点击【打开】，文件添加至白名单列表，添加到列表的文件，不允许篡改和删除，如图 8-87、图 8-88、图 8-89 和图 8-90：

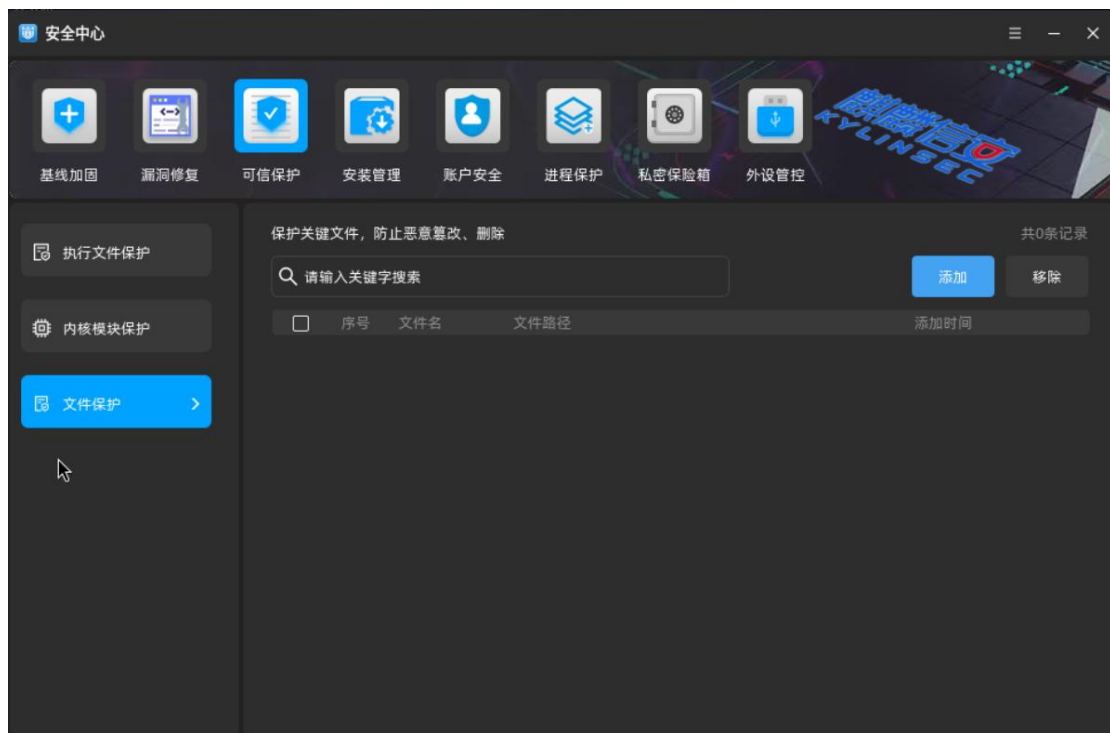


图 8-87 文件保护

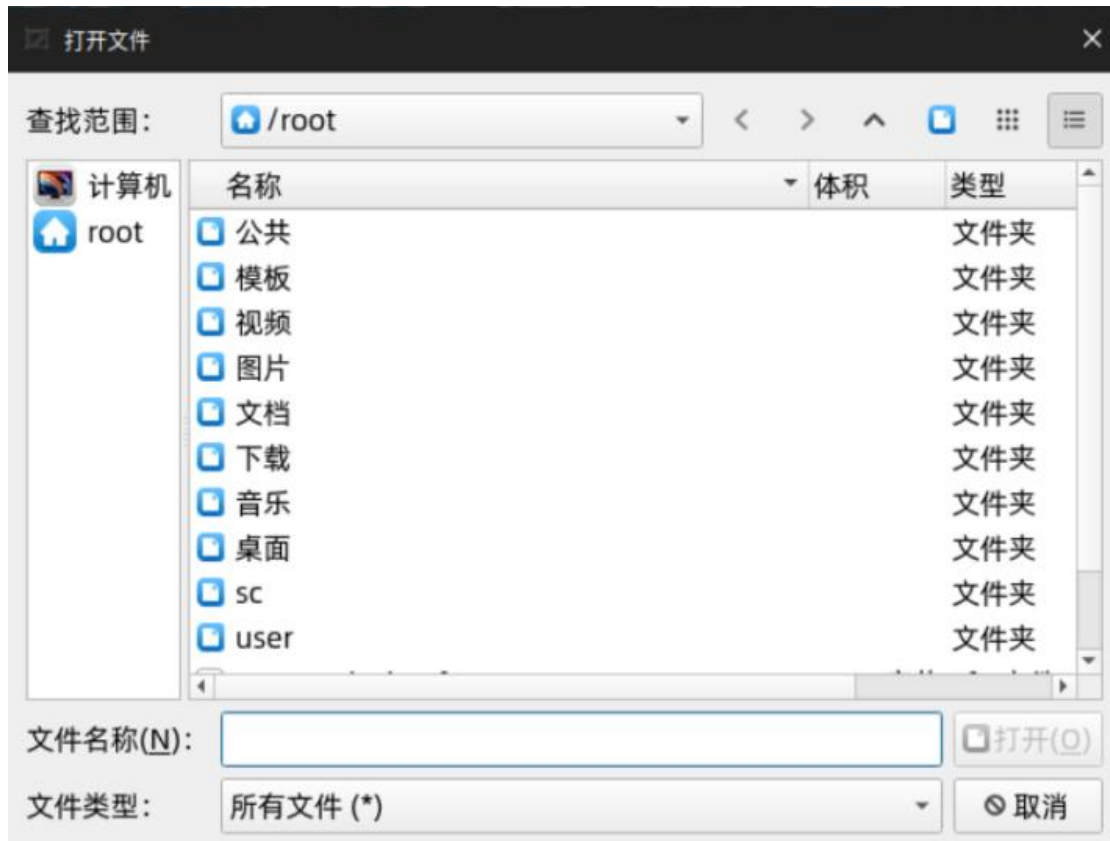


图 8-88 文件保护-添加文件

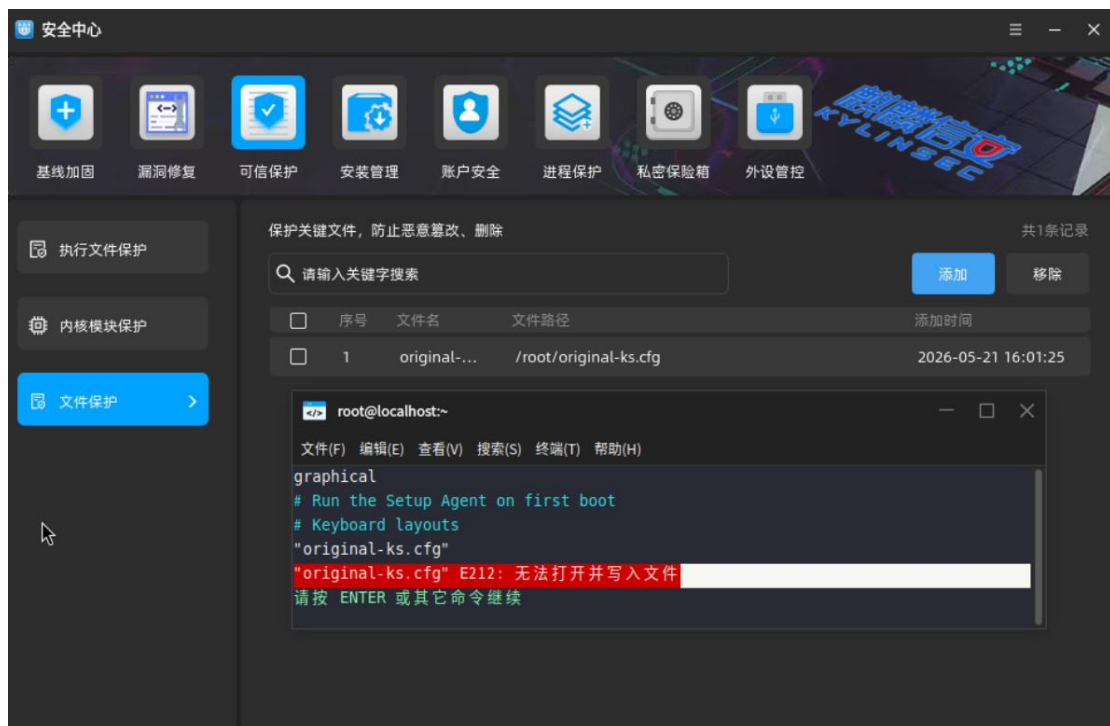


图 8-89 无法编辑文件保护列表中的文件

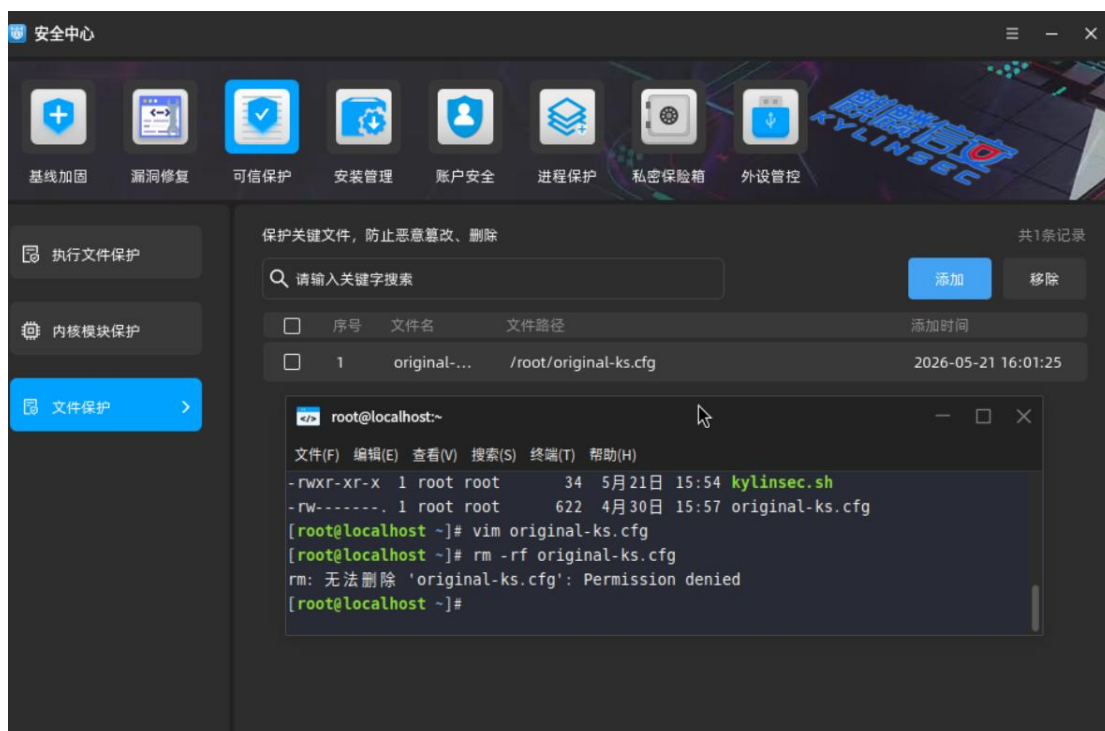


图 8-90 无法删除文件保护列表中的文件

8.4.3.4 账户安全

8.4.3.4.1 三权分立

点击【账户安全】侧边栏中的【三权分立】

开启【三权管理员开关】控制，如安装系统已选三权，则默认开启；如无三权管理员，开启开关则创建三权用户，需要手动设置三权管理员密码，如图 8-91 和图 8-92：



图 8-91 三权分立

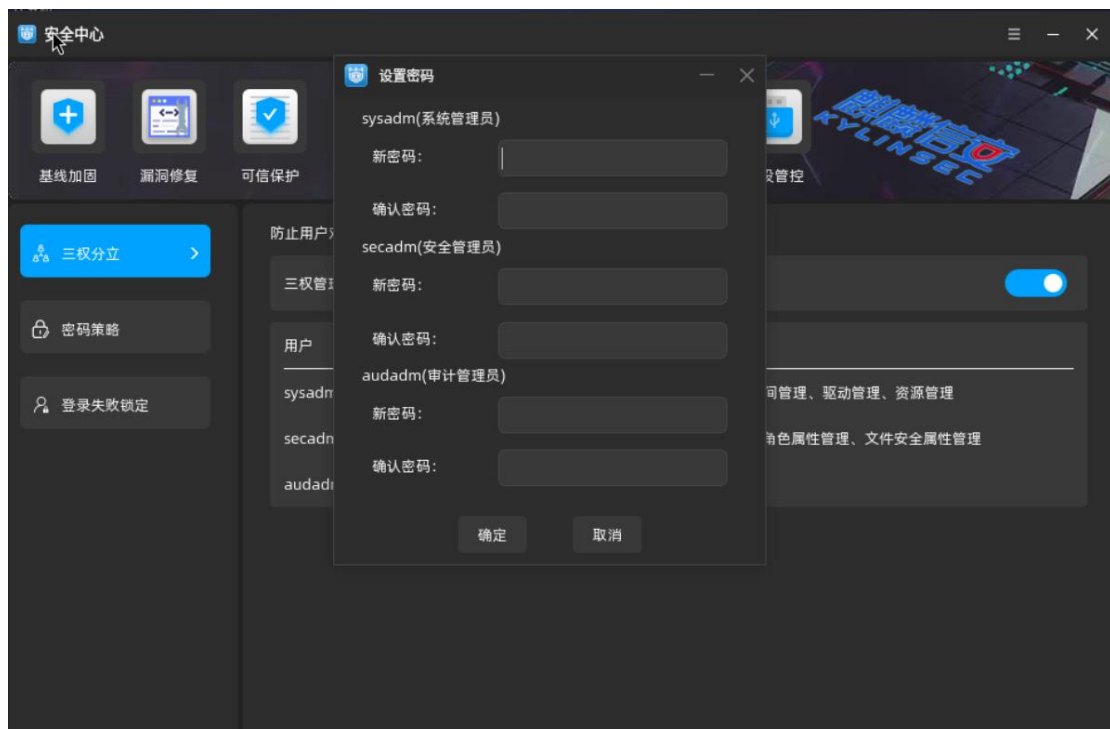


图 8-92 创建三权用户

8.4.3.4.2 密码策略

点击【账户安全】侧边栏中的【密码策略】。

打开【密码强度要求】开关，相关配置全部展示。配置如下：密码强度要求、密码最小长度、密码至少包含字符类型数量、密码有效期、修改密码的最小间隔天数、密码到期提醒、密码加密，如图 8-93：



图 8-93 密码策略

8.4.3.4.3 登录失败锁定

点击【账户安全】侧边栏中的【登录失败锁定】。

打开【登录失败锁定】开关，相关配置全部展示。配置如下：密码错误次数、锁定后自动解锁时间、是否包含 root 用户，如图 8-94：



图 8-94 登录失败锁定

8.4.3.5 漏洞修复

8.4.3.5.1 漏洞扫描

选择【漏洞修复】，点击【开始扫描】，扫描中会有扫描进度条、漏洞信息展示，扫描过程中也可以点击【取消】停止扫描，如图 8-95：



图 8-95 漏洞扫描

8.4.3.5.2 漏洞修复

扫描后，漏洞列表显示扫描结果，勾选全部漏洞，点击【一键修复】，修复中会有扫描进度条、修复漏洞信息展示，扫描过程中也可以点击【取消】停止修复，如图 8-96：



图 8-96 漏洞修复

8.4.3.5.3 进程保护

8.4.3.5.4 进程保护

点击【进程保护】，列表展示当前系统所有进程，列表字段包括序号、进程号、进程名称、进程路径、进程参数，可悬浮查看详细信息。在进程列表中或者通过搜索找到需要保护的进程，打开进程保护开关，如图 8-97：

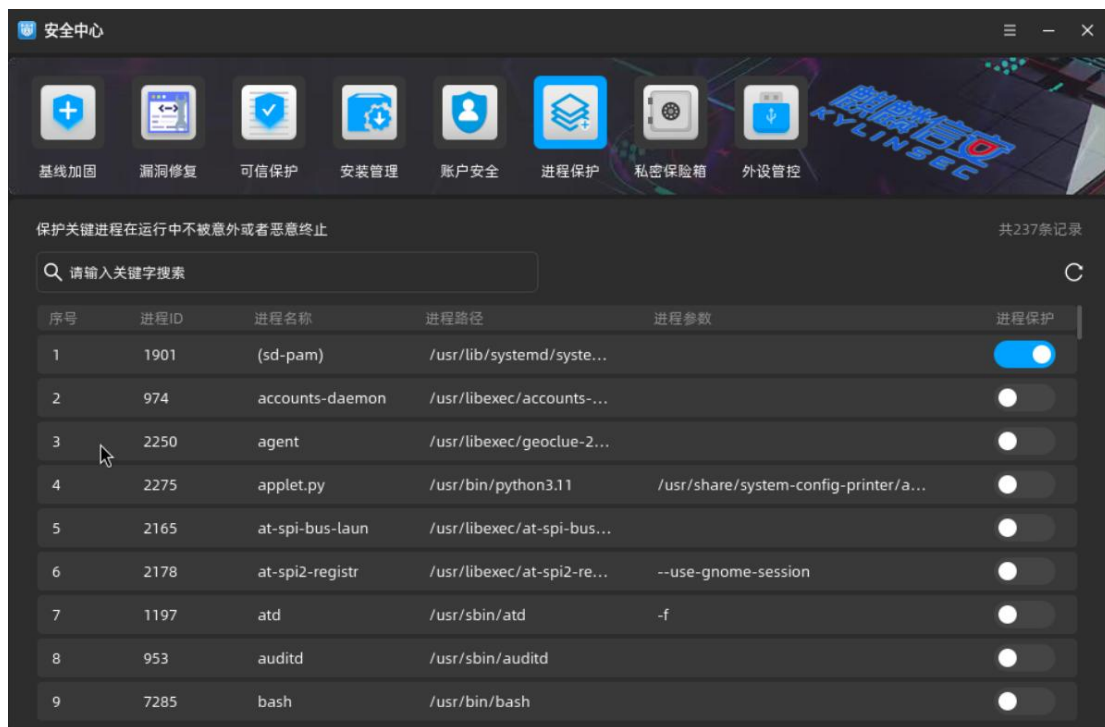


图 8-97 进程保护

8.4.3.6 外设管控

8.4.3.6.1 设备管理

点击【外设管控】侧边栏中的【设备管理】。

设备默认不管控,设备接入之后为启用状态,如需管控可将设备设置为禁用,设备状态仅包括启用和禁用状态。

点击需要修改状态的设备【操作】一栏的【编辑】,状态为禁用的设备在操作系统无法正常使用,如图 8-98 和图 8-99:



图 8-98 设备管理

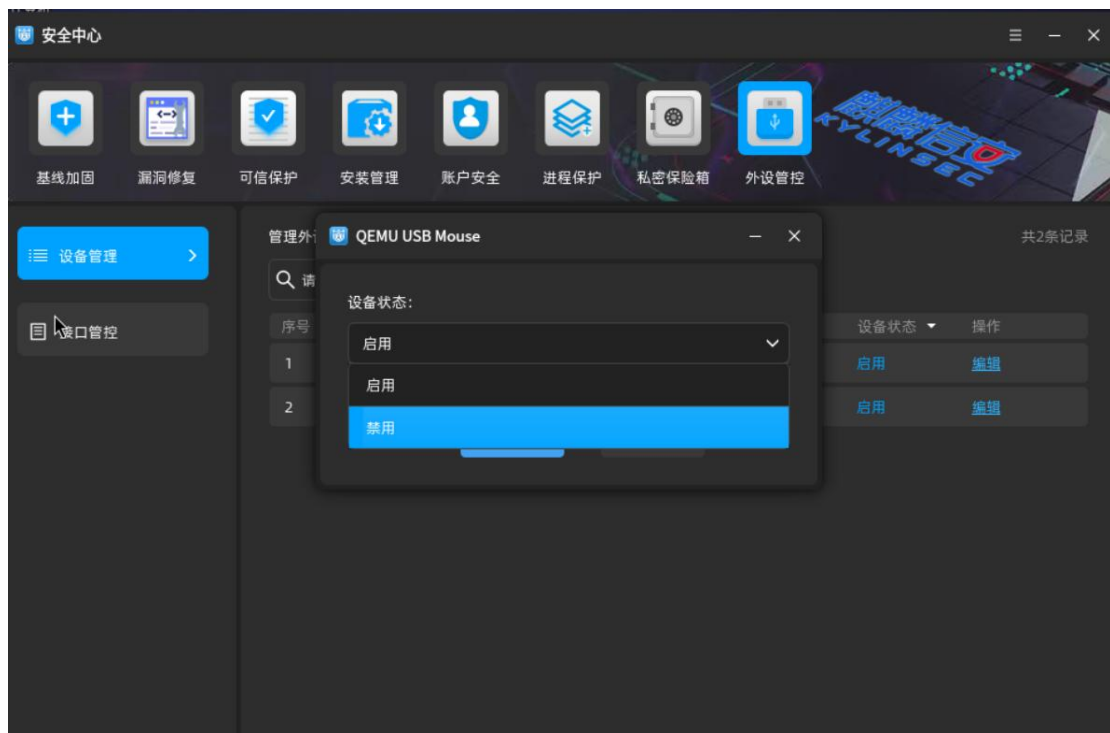


图 8-99 编辑设备状态

8.4.3.6.2 接口管理

点击【外设管控】侧边栏中的【接口管控】。

接口默认不管控，所有接口管控开关为关闭状态，如需管控可将接口管控开关置为打开状态。

打开各类别接口管控旁开关即可对该接口进行控制。

开启【USB 接口管控】后，会额外显示【键盘接口管控】和【鼠标接口管控】两个开关来控制键鼠的接入，如图 8-100 和图 8-101：

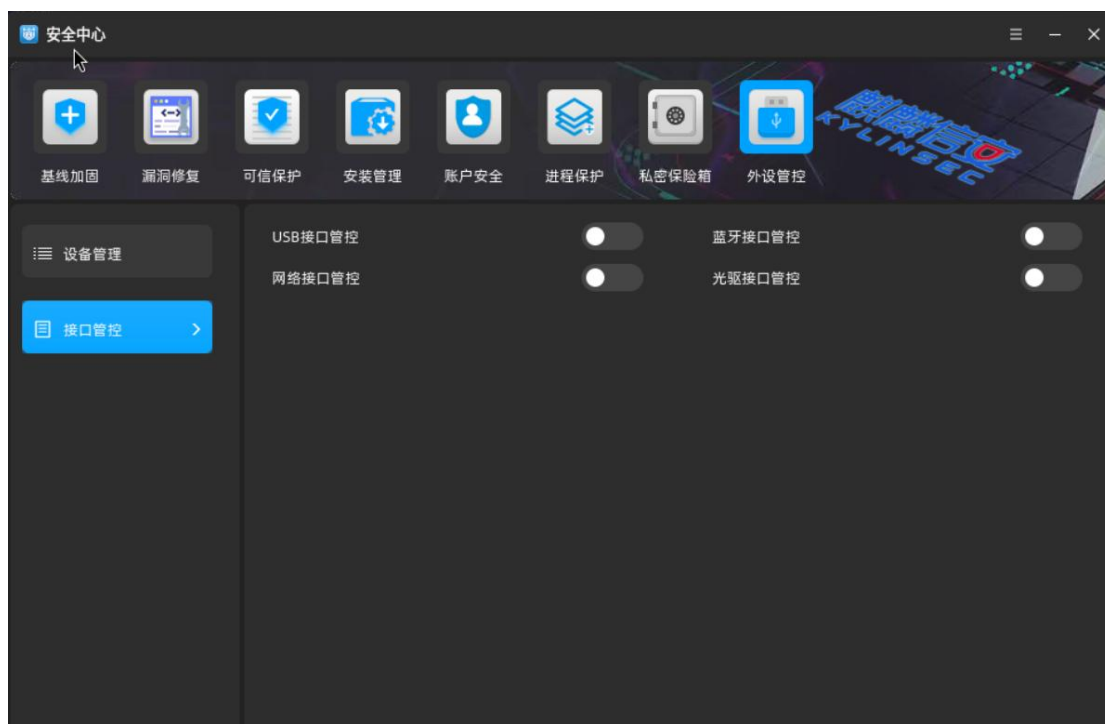


图 8-100 接口管控



图 8-101 开启 usb 接口管控

8.4.3.7 安装管理

8.4.3.7.1 安装管理

点击【安装管理】，打开【软件包签名检测】开关，系统将对所有途径下的软件包进行签名检测，签名不合法的安装包无法安装，只有通过签名校验的软件包才能安装，默认不开启。

支持软件源启停管理，打开【仓库源】中各仓库信息后的【仓库启停】开关，系统可正常使用对应的仓库源下载，如图 8-102：

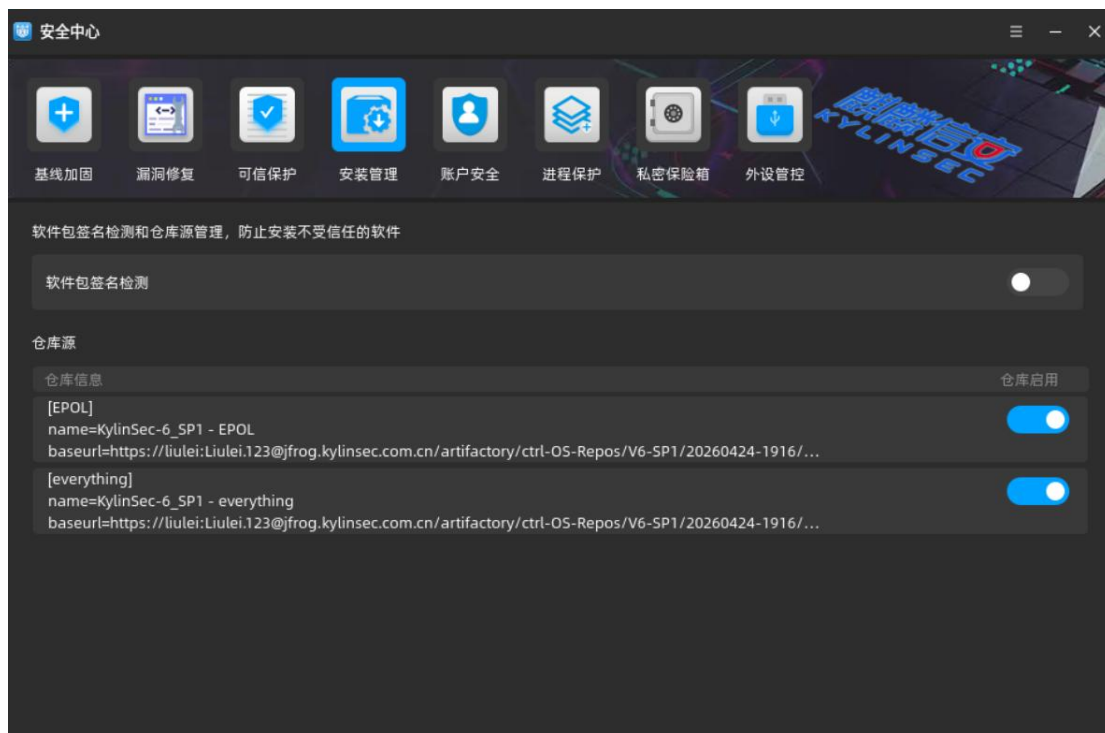


图 8-102 安装管理

8.4.3.8 私密保险箱

8.4.3.8.1 创建保险箱

点击【私密保险箱】

界面点击【新建】打开【创建保险箱】窗口

输入名称、密码和确认密码即可创建保险箱，如图 8-103、图 8-104 和图

8-105：

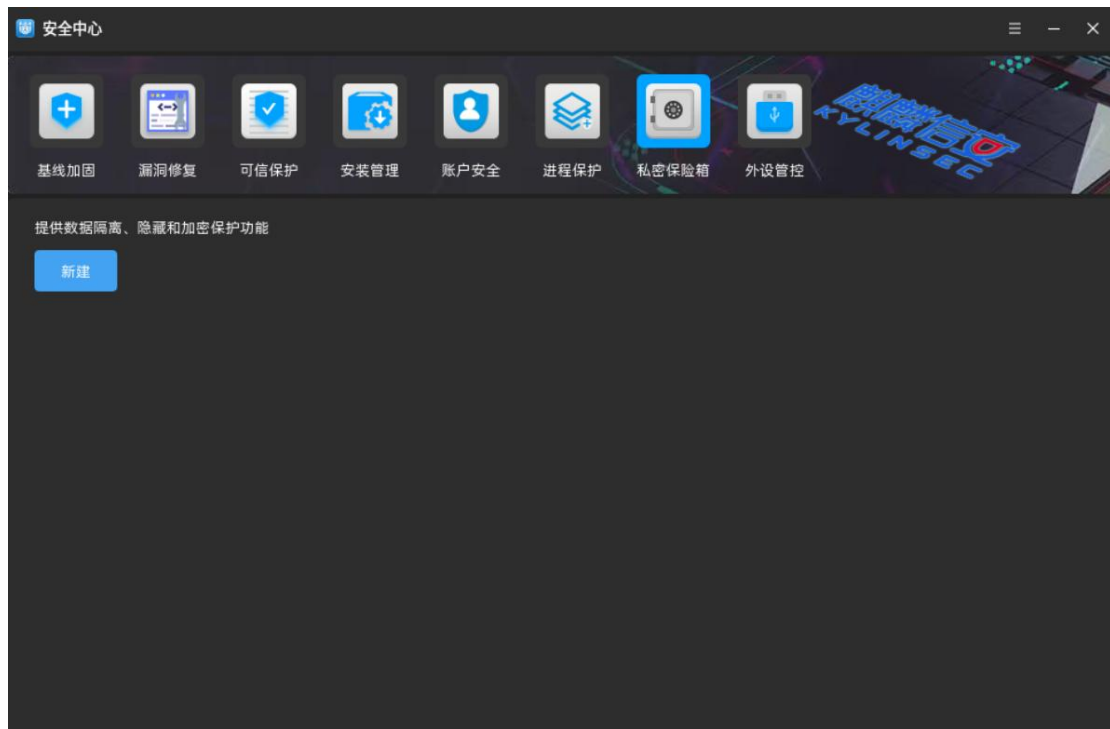


图 8-103 私密保险箱

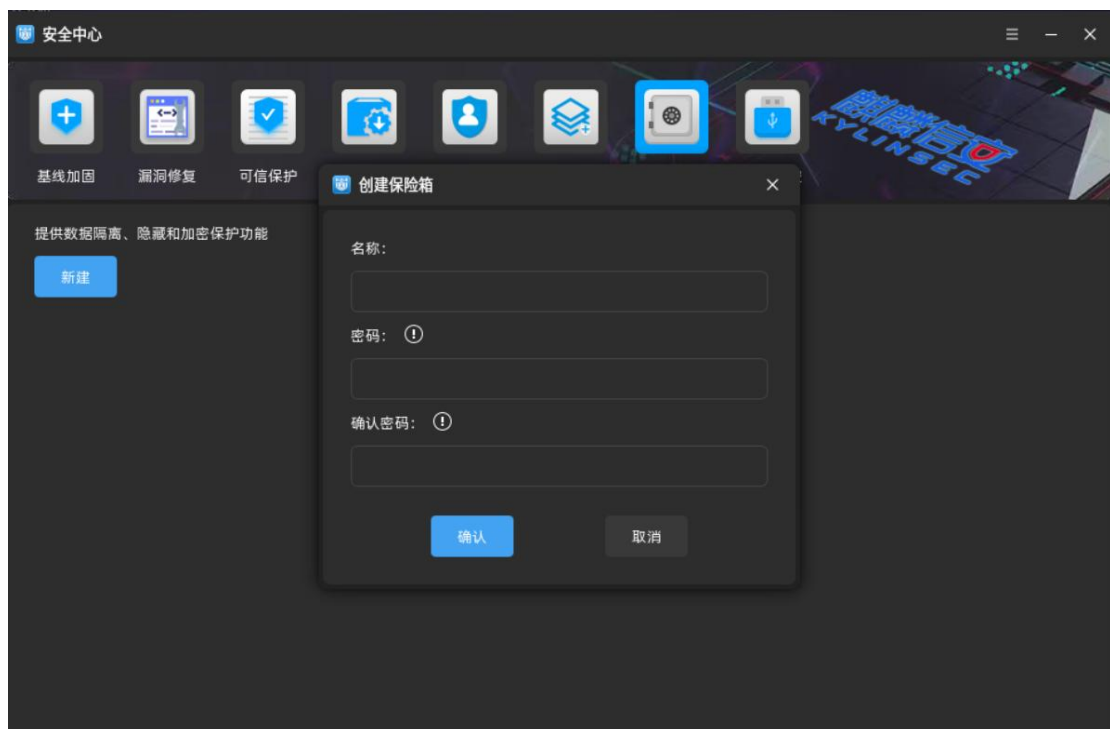


图 8-104 创建保险箱

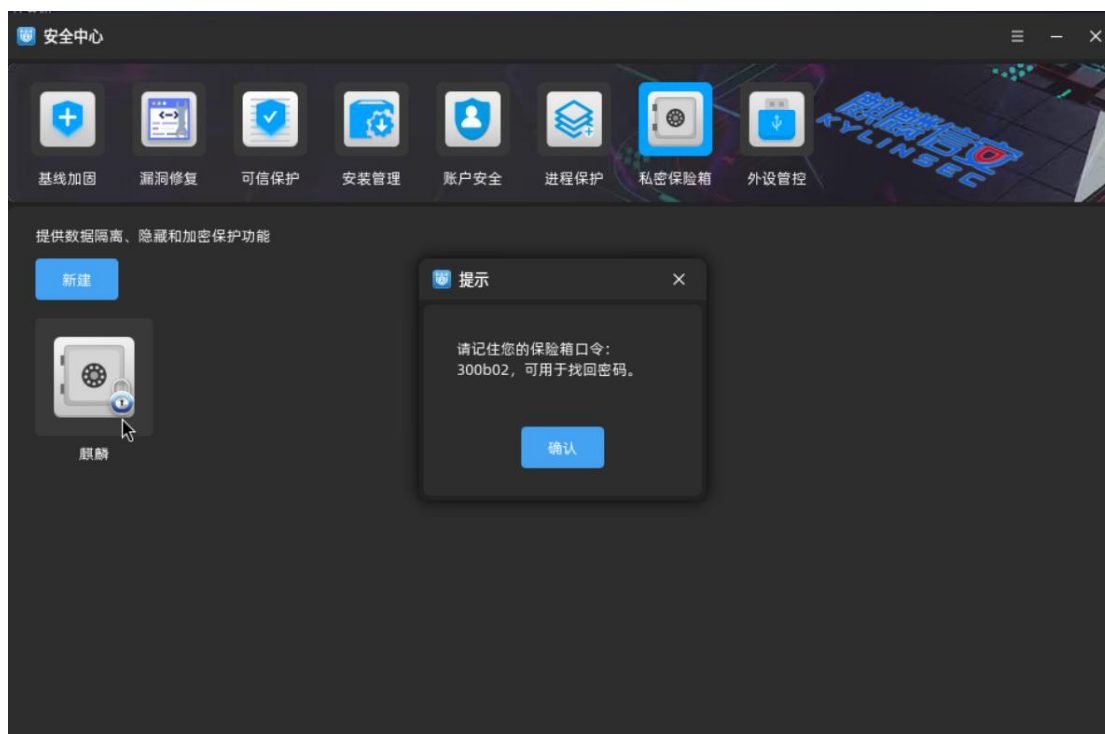


图 8-105 保险箱口令

8.4.3.8.2 解锁

选择需要解锁的保险箱点击鼠标右键，选择【解锁】，如图 8-106。

弹出【解锁】窗口，输入解锁密码可解锁并使用保险箱，如图 8-107。

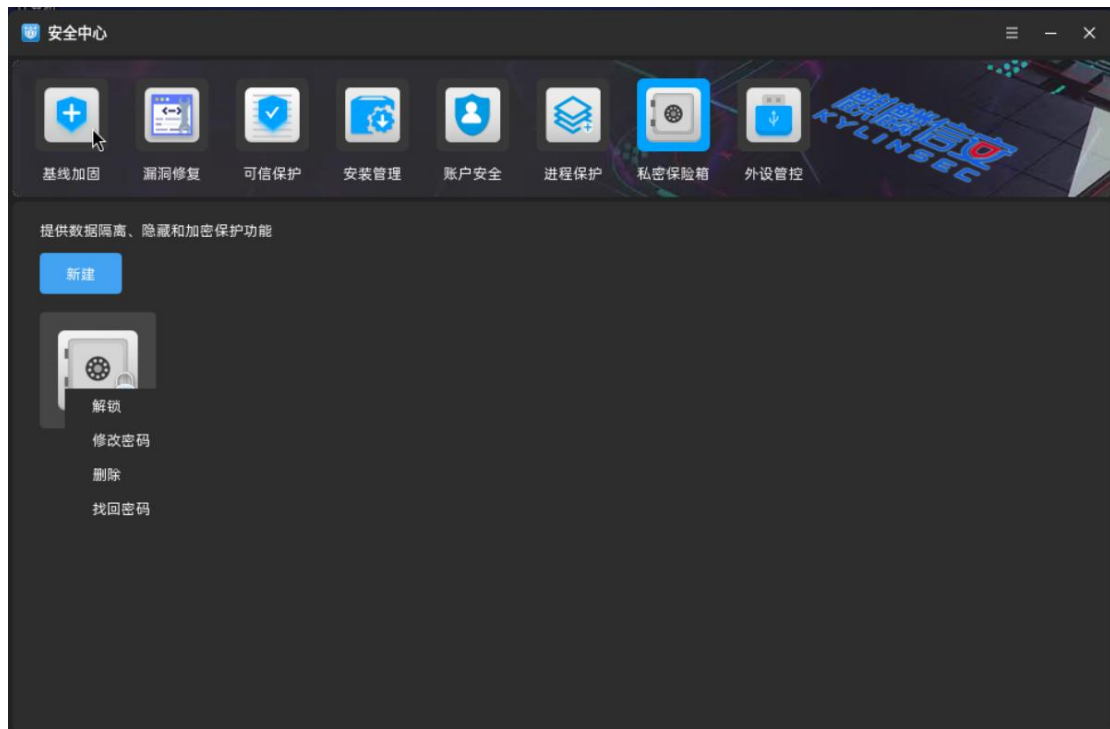


图 8-106 保险箱右键功能

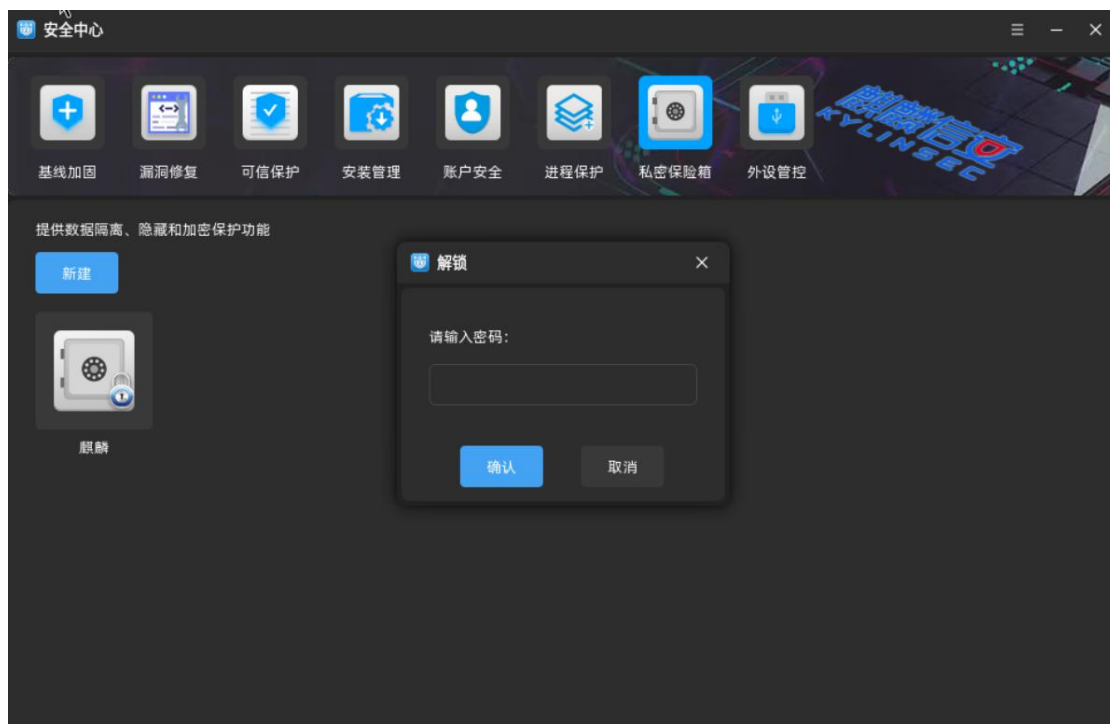


图 8-107 解锁保险箱

8.4.3.8.3 修改密码

选择需要修改密码的保险箱点击鼠标右键，选择【修改密码】。

弹出【修改密码】窗口，输入正确的当前密码、新密码和确认密码，保险箱密码修改为新密码，如图 8-108：



图 8-108 修改保险箱密码

8.4.3.8.4 删除

选择需要删除的保险箱点击鼠标右键，选择【删除】。

弹出【删除保险箱】窗口，输入正确的密码，可成功删除保险箱，如图 8-109。



图 8-109 删除保险箱

8.4.3.8.5 找回密码

选择需要找回密码的保险箱点击鼠标右键，选择【找回密码】。

弹出【找回密码】窗口，输入创建保险箱时给出的保险箱口令，可成功删除保险箱，如图 8-110。



图 8-110 找回保险箱密码

8.4.4 FAQ

问：三权用户开启的操作系统，安全中心关闭三权用户后重启再打开三权，重启后系统在标签阶段循环。

答：操作系统安全机制原因，无法从 disabled 直接转换到 enforcing 状态，需要中间转换成 permissive 过度。

8.5 远程控制

麒麟信安服务器操作系统提供远程控制管理工具，支持 VNC 协议，方便用户进行文本或图形化形式的远程连接及维护。

8.5.1 VNC 配置

8.5.1.1 服务端安装 vnc

```
yum install tigervnc-server -y
```

8.5.1.2 复制服务配置文件

```
cp /lib/systemd/system/vncserver@.service /etc/systemd/system/
```

8.5.1.3 修改 vncserver@.service 的配置

将用户名进行替换,<USER>替换成用户名(下划线的为需要修改的配置项):

[Unit]

Description=Remote desktop service (VNC)

After=syslog.target network.target

[Service]

Type=forking

WorkingDirectory=/root

User=root

Group=root

PIDFile=/root/.vnc/%H%i.pid

ExecStartPre=/bin/sh -c '/usr/bin/vncserver -kill %i > /dev/null

2>&1 || :'

ExecStart=/usr/bin/vncserver -autokill %i

ExecStop=/usr/bin/vncserver -kill %i

Restart=on-success

RestartSec=15

```
[Install]
```

```
WantedBy=multi-user.target
```

8.5.1.4 加载 systemd 服务单元

```
systemctl daemon-reload
```

8.5.1.5 设置 VNC 服务为自启动

```
#systemctl enable vncserver@:1.service
```

 ps:注意冒号

8.5.1.6 设置当前用户的 vnc 登录密码

```
vncpasswd
```

```
Password:
```

```
Verify:
```

```
Would you like to enter a view-only password (y/n)? n
```

8.5.1.7 启动一次 vncserver, 以便生成相关配置文件

```
vncserver :1
```

8.5.1.8 杀死 vncserver

```
vncserver -kill :1
```

8.5.1.9 修改 vnc 启动配置文件

```
vim /root/.vnc/xstartup
```

```
#!/bin/sh
```

```
unset SESSION_MANAGER
```

```
unset DBUS_SESSION_BUS_ADDRESS
```

```
[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
```

```
[ -r $HOME/.Xresources ] && xrdp $HOME/.Xresources  
  
xsetroot -solid grey  
  
vncconfig -iconic &  
  
#x-terminal-emulator -geometry 80x24+10+10 -ls -title "$VNCDES  
KTOP Desktop" &  
  
kiran-session-manager &
```

8.5.1.10 启动 vnc 服务

```
vncserver :1
```

8.5.1.11 在客户端上连接 vnc

```
vncviewer 192.168.120.93:1
```

9 备份还原

Timeshift 是一款功能强大的开源工具，可帮助您保护数据，它允许您创建文件系统的增量快照，可以使用文件管理器进行浏览。在 RSYNC 模式下，快照是通过 rsync 和硬链接进行的。普通文件在快照之间共享，这样可以节省磁盘空间。每个快照都是一个完整的系统备份，可以用文件管理器进行浏览。

该工具可以通过图形、命令两种方式进行操作。

9.1 图形方式

打开“开始菜单”>“Timeshift”，即可打开备份还原工具。

9.1.1 创建

当系统做了一些操作后、创建了某些文件，想对当前环境与文档进行备份时，点击“创建”按钮，如图 9-1 所示：

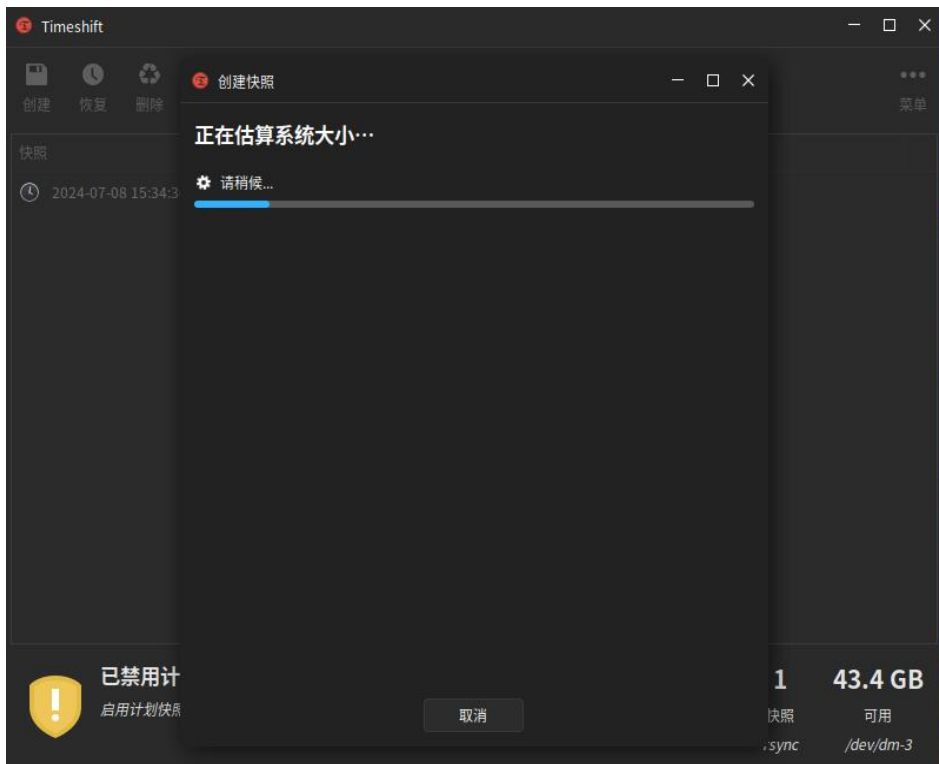


图 9-1 创建快照

此时软件会根据时间生成一个带有时间戳的快照，需要等待一段时间，建议期间不要进行其他操作。

9.1.2 恢复

选择想要恢复的快照点，点击“恢复”>“下一步”，直至完成重启，进入系统后，即恢复快照点数据，如图 9-2 所示；

注意：开始菜单列表缓存在.cache 目录下，该目录不会进行备份还原，备份还原后可通过手动删除.cache 目录下列表生成的备份文件（例如：'ksy**'）来清理缓存的应用程序列表；

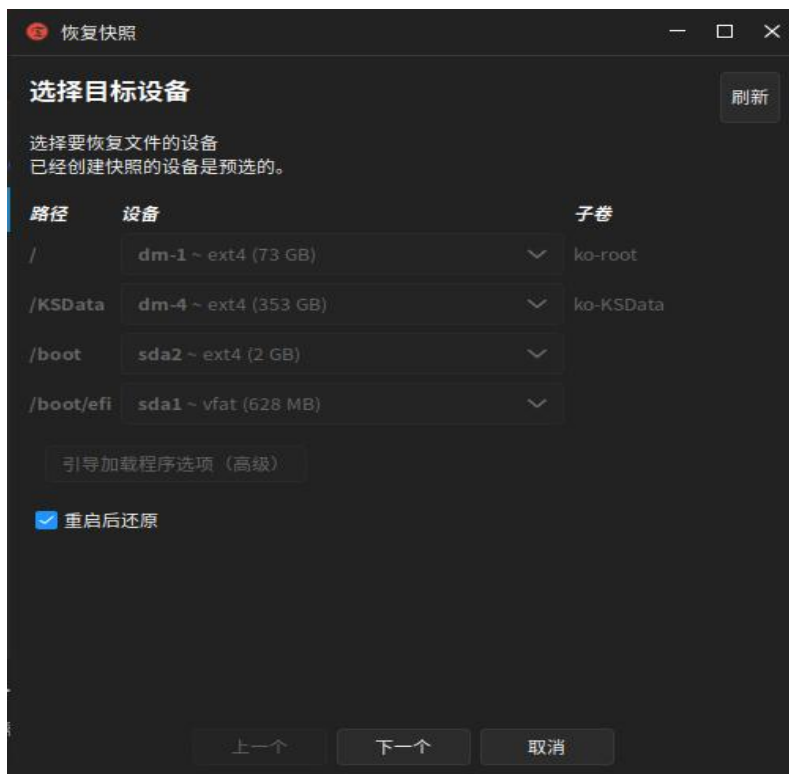


图 9-2 恢复快照

9.1.3 删除

选中快照点后，点击“删除”即可删除快照。如图 9-3 所示：



图 9-3 删除快照

9.1.4 浏览

选中快照点，点击“浏览”，即可浏览该快照点所有已经备份目录文件，如

图 9-4 所示：

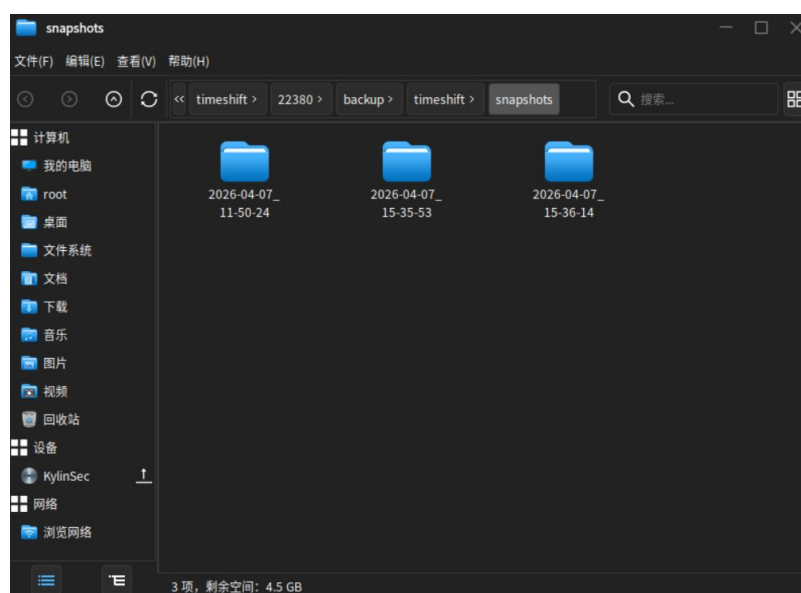


图 9-4 浏览

9.1.5 设定

根据需要对快照的类型、位置、计划、用户等进行设置。

9.1.5.1 类型

设置快照类型，默认且当前只支持“RSYNC”，如图 9-5 所示：

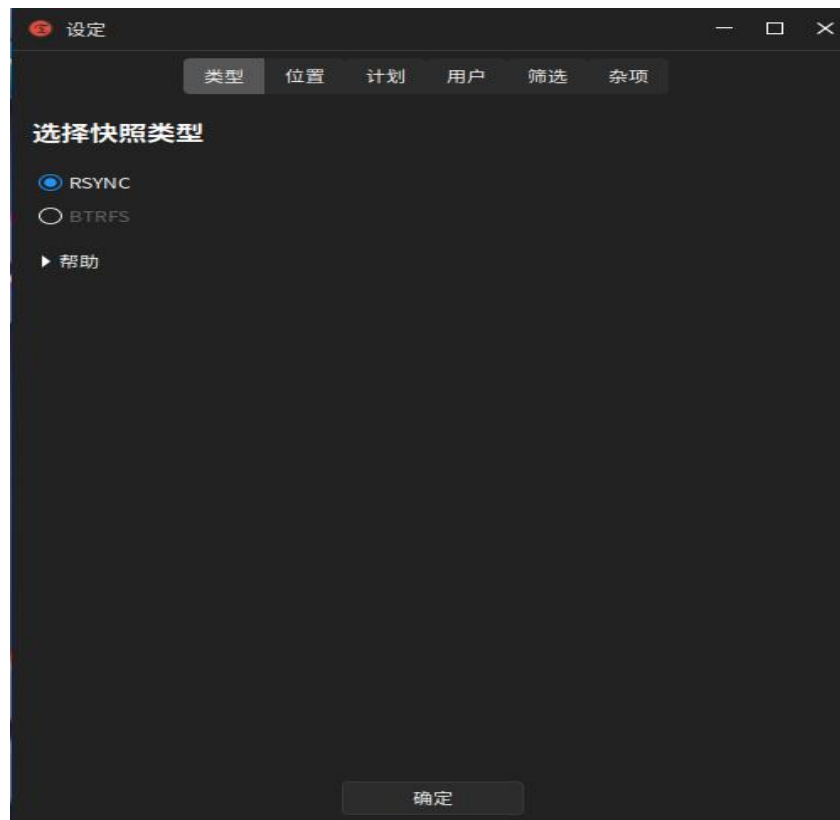


图 9-5 类型

9.1.5.2 位置

可根据需要选择快照存储位置，会显示各个磁盘、分区及分区类型、大小、空闲空间等，如图 9-6 所示：

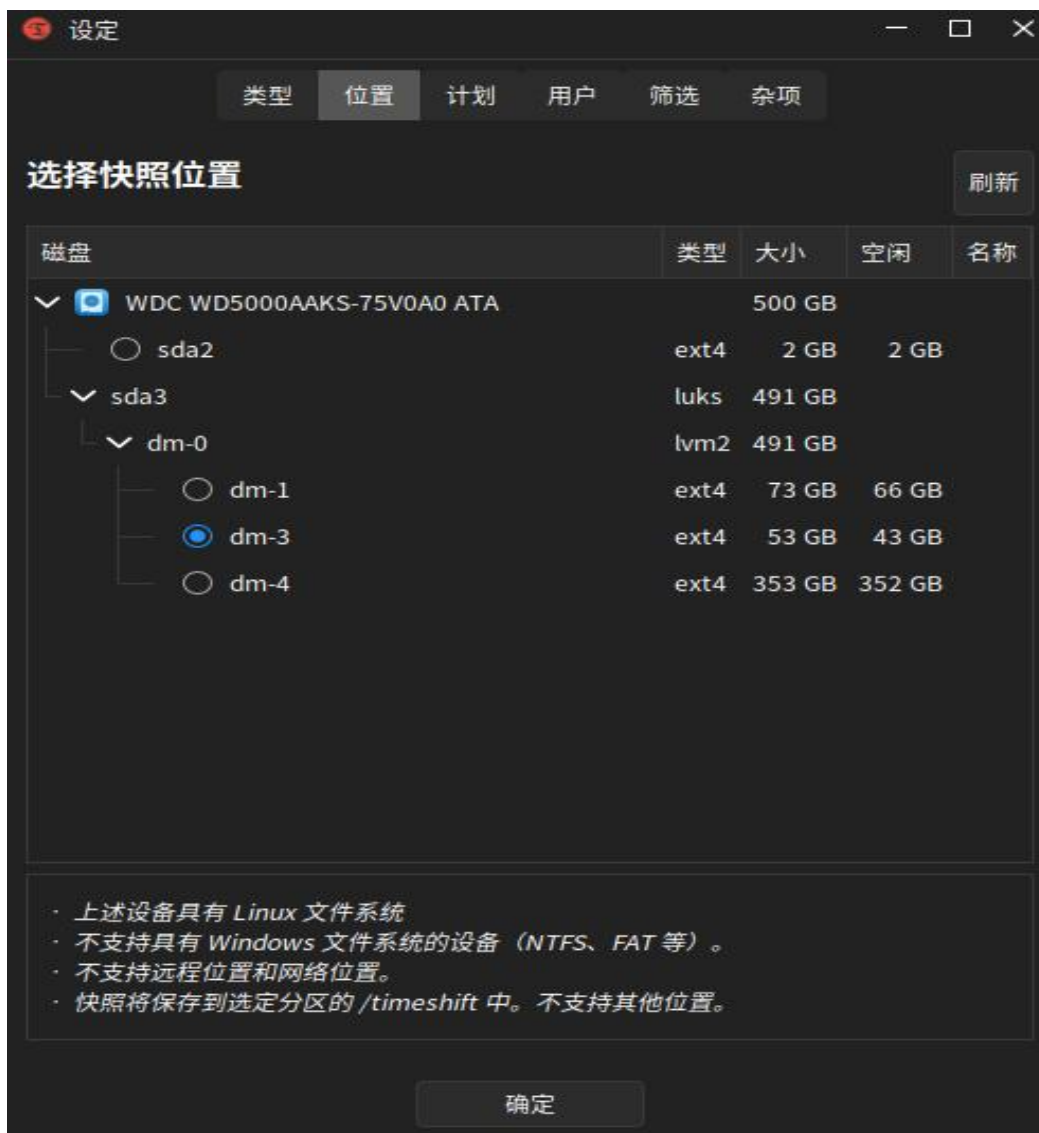


图 9-6 类型

9.1.5.3 计划

会在选项内不定期做备份；磁盘充足的情况下可设置该选项，因为磁盘占用空间会不断增长，需要定期清理，磁盘不够充足的情况下，建议不设置该项，如图 9-7 所示：

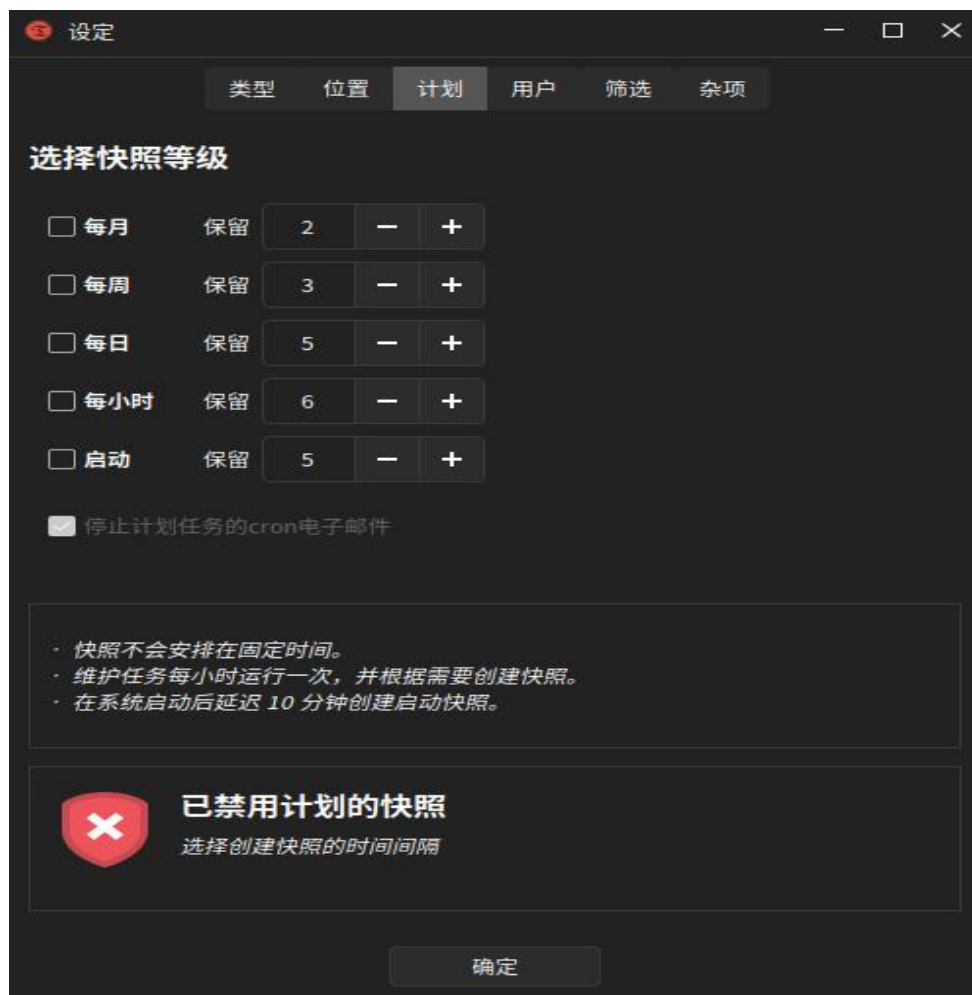


图 9-7 计划

9.1.5.4 用户

选择需要备份的用户分区，默认情况下，会排除用户主目录，如图 9-8 所示：



图 9-8 用户

9.1.5.5 筛选

可自定义备份文件或文件夹，如图 9-9 所示：



图 9-9 筛选

9.1.5.6 杂项

可设置快照时间戳格式，如图 9-10 所示：

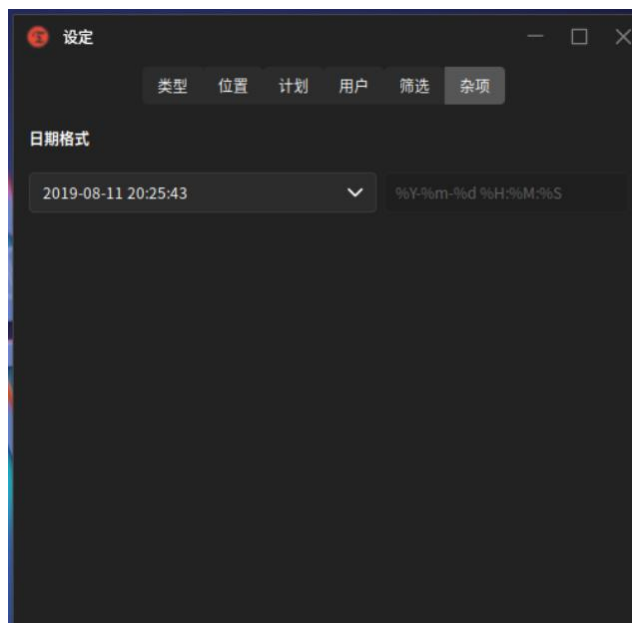


图 9-10 杂项

9.2 命令方式

查看当前备份列表及其相关配置：timeshift --list，如图 9-11 所示：

```
[root@localhost 桌面]# timeshift --list
Mounted '/dev/dm-4' at '/run/timeshift/4128/backup'
Device : /dev/dm-4
UUID   : ab31ebdc-cefd-4939-ab1c-77ba74330a73
Path    : /run/timeshift/4128/backup
Mode    : RSYNC
Status  : OK
1 snapshots, 14.2 GB free

Num      Name                      Tags Description
-----
0    > 2026-04-07_11-50-24    0    backup_of_base_os
```

图 9-11 备份列表

9.2.1 创建快照

timeshift --create [OPTIONS]

可以根据需要设置 tag、comments 与 snapshot-device 等其他参数，具体可使用 timeshift --help 查看。

```
timeshift --create --comments "X" --tags X
```

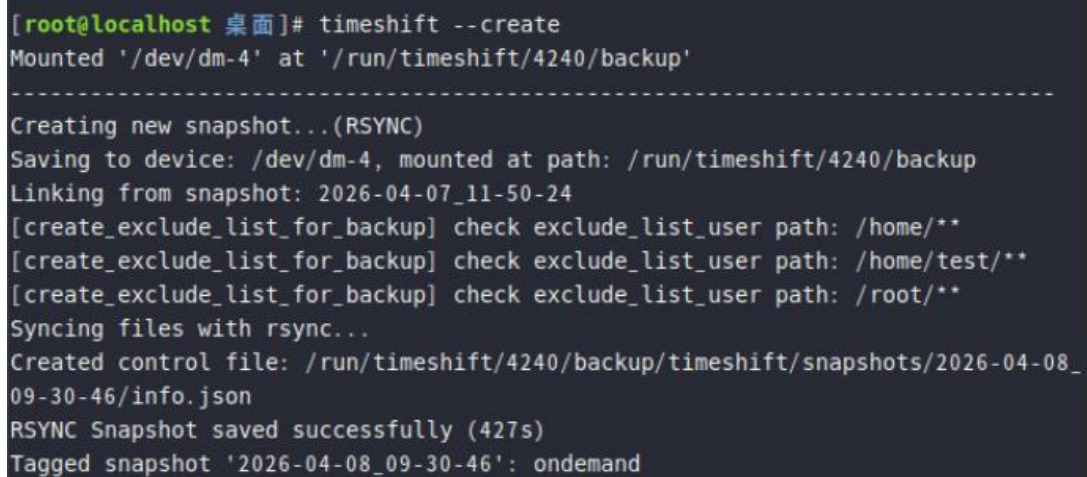
--comments "X": 快照点描述, 根据需要选择填写;

--tags X: 指定快照类型, 不设置默认为 O, 其中包括: D (每日备份)、W (每周备份)、M (每月备份)、O (按需备份);

--snapshot-device: 选择快照分区;

例: `timeshift --create --comments "Fisrt Backup" --snapshot-device /dev/dm-5`

正常情况下无特殊要求, 参数不需要全部填写, 如图 9-12。



```
[root@localhost 桌面]# timeshift --create
Mounted '/dev/dm-4' at '/run/timeshift/4240/backup'
-----
Creating new snapshot...(RSYNC)
Saving to device: /dev/dm-4, mounted at path: /run/timeshift/4240/backup
Linking from snapshot: 2026-04-07_11-50-24
[create_exclude_list_for_backup] check exclude_list_user path: /home/**
[create_exclude_list_for_backup] check exclude_list_user path: /home/test/**
[create_exclude_list_for_backup] check exclude_list_user path: /root/**
Syncing files with rsync...
Created control file: /run/timeshift/4240/backup/timeshift/snapshots/2026-04-08_09-30-46/info.json
RSYNC Snapshot saved successfully (427s)
Tagged snapshot '2026-04-08_09-30-46': ondemand
```

图 9-12 创建快照

9.2.2 还原快照

```
timeshift --restore
```

选择需要还原的快照数字, 例 1, 之后按 ENTER 键继续, 输入 y/n 选择重新安装或跳过 grub 引导加载程序, 最后根据提示输入 y 后, 重启后进入系统及还原快照成功, 如图 9-13 所示:

```
[root@localhost 桌面]# timeshift --restore
Mounted '/dev/dm-4' at '/run/timeshift/4876/backup'
Select snapshot:

Num      Name                Tags Description
-----
0  >  2026-04-07_11-50-24  0      backup_of_base_os
1  >  2026-04-08_09-30-46  0

Enter snapshot number (a=Abort, p=Previous, n=Next): 1

*****
To restore with default options, press the ENTER key for all prompts!
*****

Press ENTER to continue...

Re-install GRUB2 bootloader? (y/n):
```

图 9-13 还原

9.2.3 删除快照

删除指定快照：timeshift --delete --snapshot '2021-10-12_16-29-08'

删除所有快照：timeshift --delete-all

```
[root@localhost 桌面]# timeshift --delete-all
[Warning] Deleted invalid lock
Mounted '/dev/dm-4' at '/run/timeshift/5315/backup'
-----
Removing '2026-04-07_11-50-24'...
Removed '2026-04-07_11-50-24'
-----
Removing '2026-04-08_09-30-46'...
Removed '2026-04-08_09-30-46'
-----
[root@localhost 桌面]#
```

图 9-14 删除快照

10 系统急救模式

麒麟信安服务器操作系统提供了系统急救模式，以供在系统发生因安全配置错误等原因而导致系统故障时进行系统维护。在系统急救模式下，所有强制访问控制策略均失效，所有访问均返回允许。插入光盘后使用光盘引导进入，选择“Troubleshooting”后如下图 10-1 所示：

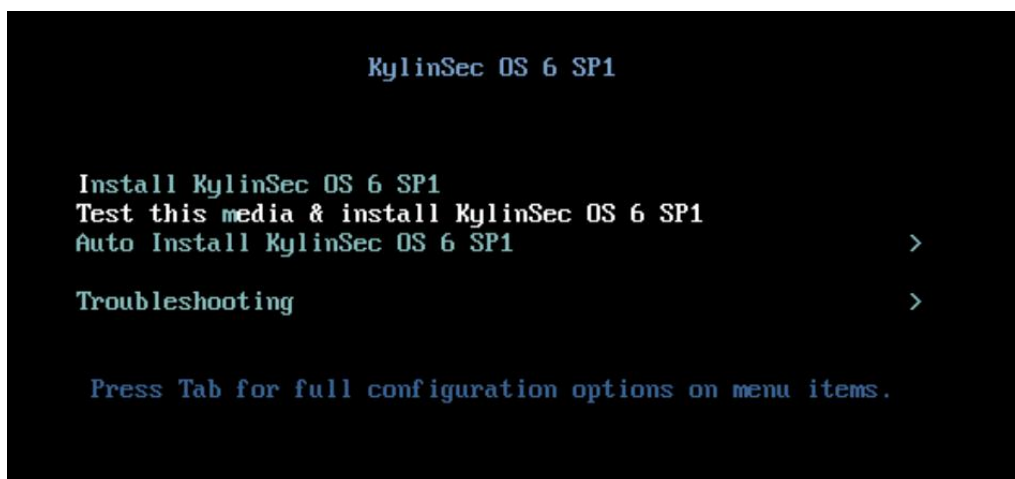


图 10-1 系统救援模式界面

选择“Rescue a KylinSec OS system”进入救援模式，显示如图 10-2：

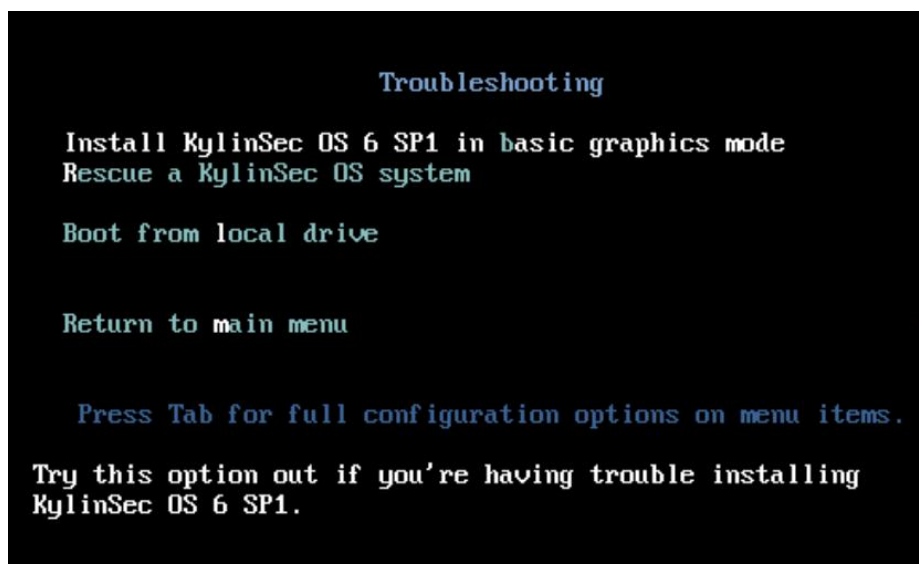


图 10-2 成功进入救援模式

Continue：输入 1 系统进入挂载模式，挂载在 /mnt/sysroot 下，可使用

chroot /mnt/sysroot 可以切到系统原根环境。在该模式下进行输入 kylin-repair 进行系统修复。

Read only mount: 输入 2 进入只读挂载模式，不可向文件系统写入。

Skip to shell: 输入 3 进入救援模式的 shell，但不会挂载系统。

Quit: 退出救援模式，系统将重启。

使用实例：模拟 grub 引导损坏；

问题描述：无法找到 grub 配置文件，无法正常进入系统；

前提条件：准备刻录好的 U 盘或光盘；将系统/boot/efi/EFI/kylinsecos/grub.cfg 改名或移动到其他目录；

操作步骤：（1）插入光盘后使用光盘引导进入，选择“Troubleshooting”；（2）选择“Rescue a KylinSec OS system”进入救援模式；（3）输入 1 系统进入 挂载模式，挂载在/mnt/sysroot 下，使用 chroot /mnt/sysroot 切到系统原根环境；（4）进入系统引导分区 cd /boot/efi/EFI/kylinsecos/，执行命令重新生成系统引导文件 grub2-mkconfig -o /boot/efi/EFI/kylinsecos/grub.cfg；（5）修复完成后，reboot 重启可正常进入系统。

11 bond3+功能

麒麟信安服务器操作系统提供 bond3+功能。在 bond3 模式下，所有数据包都会从所有绑定的网络接口发出。接收端会根据发送节点时间的链路通断状态，设置一条线路为活动线，另一条为备份线。接收端将从活动线收到的报文上送到协议栈，而备份线的报文则被丢弃。

11.1 配置 bond

1.加载驱动 modprobe bonding

2.两台双网卡机器配置 bond, A 和 B 机器创建 bond3 设备

A: bond3tool new dev bond0 slave1 网卡名 slave2 网卡名 ip 12.

12.254.100 netmask 255.255.255.0

B: bond3tool new dev bond0 slave1 网卡名 slave2 网卡名 ip 12.

12.254.200 netmask 255.255.255.0

11.2 bond3+功能

成功配置 bond3+后, 拔掉其中一根网线, 对网络 (TCP、UDP) 速率不会存在影响。

12 运维工具集

12.1 工具概述

麒麟信安服务器操作系统基于 eBPF（Berkeley Packet Filter）实现运维与故障分析工具集可以提供实时的、高效的系统监控和分析功能，以辅助客户现场问题的排查与定位。整个运维与故障分析工具集当前包含如下 4 个子工具：

CPU 负载观测与分析工具

文件缓存观测与分析工具

内存观测与分析工具

文件 I/O 观测与分析工具

12.2 软件安装

监测操作系统支持麒麟信安服务器操作系统 V6 SP1，使用“yum install 软件包名称”或者“yum update 软件包名称”的方式进行升级，opsinsight 软件包依赖 bcc 等软件包，安装前需 yum install bcc，如图 12-1：

```
sysadm@localhost:~  
文件(F) 编辑(E) 视图(V) 搜索(S) 终端(T) 帮助(H)  
[sysadm@localhost ~]$ sudo yum install bcc  
Last metadata expiration check: 0:00:27 ago on 2025年01月14日 星期二 13时03分50秒.  
Dependencies resolved.  
=====
```

Package	Architecture	Version	Repository	Size
Installing:				
bcc	x86_64	0.29.1-2.ks6	test	901 k
Installing dependencies:				
bcc-tools	x86_64	0.29.1-2.ks6	test	501 k
kernel-devel	x86_64	6.6.0-28.0.0.34.kb8.ks6	test	16 M
python3-bpfcc	noarch	0.29.1-2.ks6	test	127 k
python3-netaddr	noarch	0.10.1-1.ks6	test	848 k

```
Transaction Summary  
=====
```

Install 5 Packages			
Total download size: 18 M			
Installed size: 75 M			
Is this ok [y/N]: y			
Downloading Packages:			
(1/5): bcc-0.29.1-2.ks6.x86_64.rpm	2.8 MB/s	901 kB	00:00
(2/5): bcc-tools-0.29.1-2.ks6.x86_64.rpm	1.4 MB/s	501 kB	00:00
(3/5): python3-bpfcc-0.29.1-2.ks6.noarch.rpm	1.6 MB/s	127 kB	00:00
(4/5): python3-netaddr-0.10.1-1.ks6.noarch.rpm	3.5 MB/s	848 kB	00:00
(5/5): kernel-devel-6.6.0-28.0.0.34.kb8.ks6.x86_64.rpm	7.8 MB/s	16 MB	00:02
Total	8.9 MB/s	18 MB	00:02

```
Running transaction check  
Transaction check succeeded.  
Running transaction test  
Transaction test succeeded.
```

图 12-1 安装依赖包 bcc

使用“yum install rpm 软件包名称”进行安装 opsinsight，如图 12-2：

```
[sysadm@localhost ~]$ sudo yum provides opsinsight
epol                                     8.0 MB/s | 4.3 MB   00:00
extra                                   1.2 MB/s | 292 kB   00:00
opsinsight-0.0.1-kb4.ks6.x86_64 : opsinsight
Repo                                   : extra

[sysadm@localhost ~]$ sudo yum install opsinsight
test                                   162 kB/s | 4.6 kB   00:00
Dependencies resolved.
=====
Package                               Architecture      Version           Repository        Size
=====
Installing:
opsinsight                           x86_64            0.0.1-kb4.ks6    extra             27 k
=====
Transaction Summary
=====
Install 1 Package

Total download size: 27 k
Installed size: 88 k
Is this ok [y/N]: y
Downloading Packages:
opsinsight-0.0.1-kb4.ks6.x86_64.rpm   343 kB/s | 27 kB    00:00
-----
Total                                   331 kB/s | 27 kB    00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                                     1/1
  Installing     : opsinsight-0.0.1-kb4.ks6.x86_64    1/1
  Running scriptlet: opsinsight-0.0.1-kb4.ks6.x86_64 1/1
  Verifying      : opsinsight-0.0.1-kb4.ks6.x86_64    1/1

Installed:
opsinsight-0.0.1-kb4.ks6.x86_64

Complete!
```

图 12-2 安装 opsinsight 安装包

12.3 CPU 负载观测与分析功能

12.3.1 作用

实时观测指定周期内系统最高 CPU 负载情况。

12.3.2 输入

opsinsight cpu [-i] [-d] [-c] [-f] [-s] [-l] [-h]

-i (--interval): 指定观测的周期, 单位为秒, 默认为 10 秒, 可配置范围[10,+
∞)。

-d (--detail): 指定显示负载情况的详细程度。默认为 0。

0: 显示系统最高负载时刻 R 状态和 D 状态线程的个数。

1: 显示系统最高负载时刻 R 状态和 D 状态线程的个数+线程名。

2: 显示系统最高负载时刻 R 状态和 D 状态线程的个数+线程名+调用栈。

-c(--console): 指定显示是否输出到屏幕。默认为 1。

0: 不输出到屏幕

1: 输出到屏幕

2: 输出到屏幕并且清屏

-f(--logfile): 指定输出的日志文件。默认为空，表示不输出到日志文件。

-r(--rotate): 指定轮转日志文件的个数。默认为 2，可配置范围[0,+∞)。

-s(--logsize): 指定日志文件的最大大小，单位为 MB，默认为 1024。当日志文件超过最大大小时将会清除内容重新开始记录，可配置范围[0,+∞)。

-l(--limit): 显示输出最高负载信息的阈值。默认为 0，只有当负载大于指定的阈值时才显示到屏幕和输出到日志文件，可配置范围[0,+∞)。

-h(--help): 显示工具帮助信息。

12.3.3 输出

CPU 负载观测与分析工具的输出按照命令行或配置文件中参数的指定有三种详细程度。三种情况下的输出日志格式与内容如下：

12.3.3.1 详细程度为 0 的情况

Timestamp	Running Tasks	Uninterruptable Tasks
2026-07-08 04:34:28.464	15	0

图 12-3 详细程度为 0 的示意图

如图 12-3：第一行为表头。

第二行为系统最高 CPU 负载情况。

第一列显示观测周期内系统最高负载时间戳。

第二列显示观测周期内系统最高负载时就绪态线程的个数。

第三列显示观测周期内系统最高负载时不可中断阻塞态线程的个数。

12.3.3.2 详细程度为 1 的情况

Timestamp	Running Tasks	Uninterruptable Tasks
2026-07-08 04:35:42.184	19	0
R-2-QXcbEventQueue-2970		
R-2-kiran-polkit-ag-2985		
R-2-QXcbEventQueue-2949		
R-2-QXcbEventQueue-3091		
R-2-QXcbEventQueue-2934		
R-0-kauditd-45		
R-2-fcitx5-6775		
R-3-CPU 1/TCG-153810		
R-2-QXcbEventQueue-2936		

图 12-4 详细程度为 1 示意图

如图 12-4：第一行为表头。

第二行为系统最高 CPU 负载情况。

以 R 开头的显示就绪态线程所在的 CPU、线程名称和 PID

以 D 开头的显示不可中断阻塞态线程所在的 CPU、线程名称和 PID

12.3.3.3 详细程度为 2 的情况

Timestamp	Running Tasks	Uninterruptable Tasks
2026-07-08 04:37:19.371	15	0
R-1-CPU 1/TCG-153810		
stack(7854):		
enqueue_task_fair		
activate_task		
ttwu_do_activate		
sched_ttwu_pending		
__flush_smp_call_function_queue		
__sysvec_call_function_single		
sysvec_call_function_single		
asm_sysvec_call_function_single		
pv_native_safe_halt		
default_idle		
default_idle_call		
cpuidle_idle_call		
do_idle		
cpu_startup_entry		
start_secondary		

图 12-5 详细程度为 2 示意图

如图 12-5，第一行为表头。

第二行为系统最高 CPU 负载情况，增加调用栈显示。

以 R 开头的显示就绪态线程所在的 CPU、线程名称和 PID

以 D 开头的显示不可中断阻塞态线程所在的 CPU、线程名称和 PID

12.3.3.4 /etc/opsinsight.conf 配置文件

```
[cpu]

interval=10

detail=0

console=1

logfile= ""

#/var/log/opsinsight-cpu.log

rotate=2

logsize=1024

limit=0
```

12.4 文件缓存观测与分析功能

12.4.1 作用

实时观测指定周期内系统文件缓存占用情况。

12.4.2 输入

```
opsinsight cache [-i] [-c] [-f] [-s] [-n] [-t] [-w]
```

-i(--interval): 指定观测的周期，单位为秒，默认为 10 秒。可配置范围[10,+
∞)。

-c(--console): 指定显示是否输出到屏幕。默认为 1。

0: 不输出到屏幕

1: 输出到屏幕

2: 输出到屏幕并且清屏

-f(--logfile): 指定输出的日志文件。默认为空, 表示不输出到日志文件。

-r(--rotate): 指定轮转日志文件的个数。默认为 2, 可配置范围 $[0, +\infty)$ 。

-s(--logsize): 指定日志文件的最大大小, 单位为 MB, 默认为 1024。当日志文件超过最大大小时将会清除内容重新开始记录, 可配置范围 $[0, +\infty)$ 。

-n(--numbers): 指定输出多少项。默认为 10, 可配置范围 $[0, +\infty)$ 。

-t(--type): 指定统计的方式。默认为 0。按进程 ID 进行统计。

0: 按进程 ID 进行统计

1: 按线程 ID 进行统计

2: 按文件名进行统计

3: 按调用栈进行统计

-w(--watch): 指定观测的内容。默认为 0, 表示观测文件缓存的占用情况。

0: 观测文件缓存占用

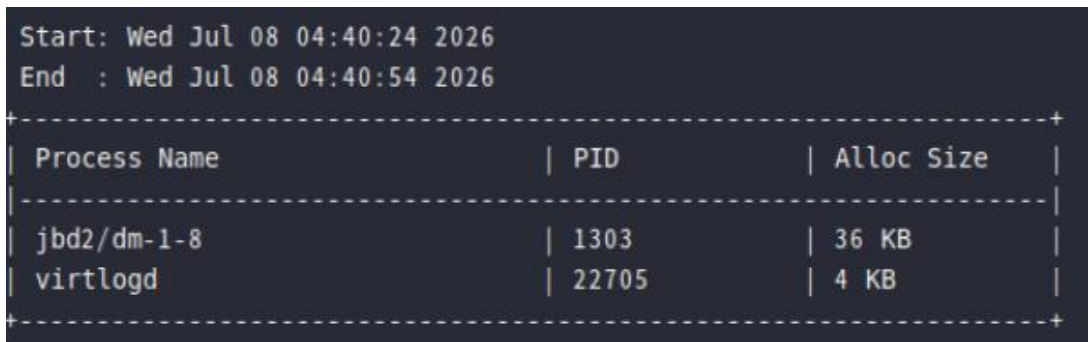
1: 观测文件缓存释放

12.4.3 输出

文件缓存负载观测与分析工具的输出按照命令行或配置文件中参数的指定有四种统计方式。四种情况下的输出日志格式与内容如下:

12.4.3.1 按进程 ID 进行统计的情况

按进程 ID 进行统计主要用于分析哪些进程在观测周期内使用/释放了文件缓存，如图 12-6：



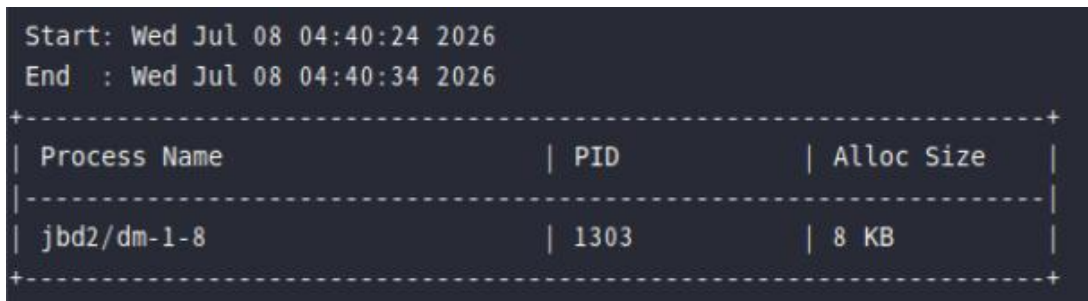
```
Start: Wed Jul 08 04:40:24 2026
End : Wed Jul 08 04:40:54 2026
```

Process Name	PID	Alloc Size
jbd2/dm-1-8	1303	36 KB
virtlogd	22705	4 KB

图 12-6 进程 ID 统计示意图

12.4.3.2 按线程 ID 进行统计的情况

按线程 ID 进行统计主要用于分析哪些线程在观测周期内使用/释放了文件缓存，如图 12-7：



```
Start: Wed Jul 08 04:40:24 2026
End : Wed Jul 08 04:40:34 2026
```

Process Name	PID	Alloc Size
jbd2/dm-1-8	1303	8 KB

图 12-7 线程 ID 统计示意图

12.4.3.3 按文件名进行统计的情况

按文件名进行统计主要用于分析观测周期内使用/释放了文件缓存所属的文件，如图 12-8：

```
Start: Wed Jul 08 04:40:24 2026
End : Wed Jul 08 04:40:44 2026
```

Process Name	PID	Alloc Size
jbd2/dm-1-8	1303	24 KB

图 12-8 文件名统计示意图

12.4.3.4 按调用栈进行统计的情况：

按调用栈进行统计主要用于分析观测周期内使用/释放了文件缓存的函数调用位置，以便排查什么原因导致了文件缓存的使用与释放，如图 12-9：

```
Start: Wed Jul 08 04:43:21 2026
End : Wed Jul 08 04:45:41 2026

Alloc: 92 KB
jbd2/dm-1-8      1303
  __filemap_add_folio
  filemap_add_folio
  __filemap_get_folio
  grow_dev_page
  __getblk_gfp
  jbd2_journal_get_descriptor_buffer
  journal_submit_commit_record
  jbd2_journal_commit_transaction
  kjournald2
  kthread
  ret_from_fork
  ret_from_fork_asm
```

图 12-9 调用栈统计示意图

12.4.4 /etc/opsinsight.conf 配置文件

[cache]

interval=10

console=1

```
logfile= ""
```

```
#!/var/log/opsinsight-cache.log
```

```
rotate=2
```

```
logsize=1024
```

```
numbers=10
```

```
type=0
```

```
watch=0
```

12.5 内存观测与分析功能

12.5.1 作用

在观测周期内，实时检测进程/线程申请或释放内存的情况。

12.5.2 输入

```
opsinsight mem [-i] [-w] [-t] [-l] [-A] [-f] [-r] [-s] [-c] [-h]
```

-i (--interval): 指定观测的周期，单位为秒，默认为 1 秒。可配置范围[1, 10000]。

-w(--watch): 指定显示类型。默认为 0，表示按进程显示。

0: 按进程显示

1: 按线程显示

-t(--type): 指定排序方式。默认为 0，按单个进程或线程分配的总大小来排序。

0: total_size 按单个进程或线程分配的总大小排序

1: delta_size 按申请内存的变化量的绝对值大小排序

2: alloc_size 按分配大小排序

3: free_size 按释放大小排序

-n(--numbers): 指定输出多少项。默认为 10, 可配置范围[1, 1000]。

-A(--accu): 指定统计模式为累计模式, 默认为非累计模式

-f(--logfile): 指定输出的日志文件。默认为空, 表示不输出到日志文件。

-r(--rotate): 指定轮转日志文件的个数。默认为 2, 可配置范围[1, 10]。

-s(--logsize): 指定日志文件的最大大小, 单位为 MB, 默认为 1024。当日志文件超过最大大小时将会清除内容重新开始记录, 可配置范围[1, 1024]。

-c(--console): 指定显示是否输出到屏幕。默认为 1。

0: 不输出到屏幕

1: 输出到屏幕且每次不清屏

2: 输出到屏幕且每次清屏

-h(--help): 显示帮助信息。

12.5.3 输出

Start-Time:2026-07-08 04:47:57		alloc_size: 192.00K			
End-Time:2026-07-08 04:47:58		free_size: 180.00K			
PID	ALLOC_SIZE	FREE_SIZE	<TOTAL_SIZE>	ABS_SIZE	COMMANDS
5258	76.00K	68.00K	144.00K	8.00K	mate-terminal
1833	52.00K	68.00K	120.00K	16.00K	NetworkManager
153797	48.00K	32.00K	80.00K	16.00K	CPU 1/TCG
2930	12.00K	0B	12.00K	12.00K	kwin_x11
2931	4.00K	4.00K	8.00K	0B	kiran-shell
2085	0B	8.00K	8.00K	8.00K	Xorg

图 12-10 内存观测示意图

如图 12-10: PID/TID: 进程 pid 或线程 tid

TOTAL_SIZE: 分配总大小, 单个周期内分配内存+释放内存大小, 单位自适

应

ALLOC_SIZE: 分配内存大小, 单位自适应

FREE_SIZE: 释放内存大小, 单位自适应

ABS_SIZE: 申请内存-释放内存的大小, 单位自适应

COMMANDS: 执行的程序名字

12.5.4 /etc/opsinsight.conf 配置文件

```
[mem]
```

```
interval=1
```

```
watch=0
```

```
type=0
```

```
numbers=10
```

```
accu=1
```

```
logfile=""
```

```
#/var/log/opsinsight-mem.log
```

```
rotate=2
```

```
logsize=1024
```

```
console=1
```

12.6 io 观测与分析功能

12.6.1 作用

实时观测指定周期内系统 io 的情况。

12.6.2 输入

opsinsight io [-i][-n] [-w][-t] [-A] [-f][-r] [-s][-c][-h]

-i(--interval): 指定观测的周期，单位为秒，默认为 1 秒，可配置范围[1, 10000]。

-n(--numbers): 指定输出多少项。默认为 10，可配置范围[1, 1000]。

-w(--watch): 指定显示类型。默认为 0，表示按进程显示。

0: 按进程显示

1: 按线程显示

-t(--type): 指定排序的方式。默认为 0。按磁盘写模式进行排序。

0: 按磁盘写模式排序

1: 按磁盘读模式排序

2: 按 io 模式排序

-A(--accu): 指定统计模式为累计模式，默认为非累计模式。

-f(--logfile): 指定输出的日志文件。默认为空，表示不输出到日志文件。

-r(--rotate): 指定轮转日志文件的个数。默认为 2，可配置范围[1, 10]。

-s(--logsize): 指定日志文件的最大大小，单位为 MB，默认为 1024。当日志文件超过最大大小时将会清除内容重新开始记录，可配置范围[1, 1024]。

-c(--console): 指定显示是否输出到屏幕。默认为 1。

0: 不输出到屏幕

1: 输出到屏幕且每次不清屏

2: 输出到屏幕且每次清屏

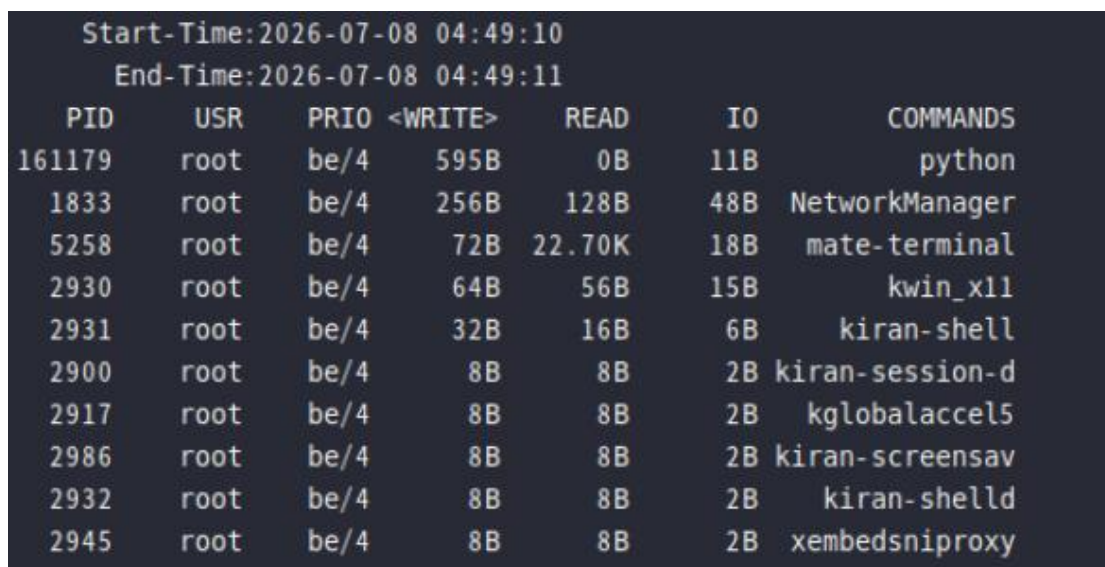
-m(--mode): 指定采集模式，默认为 0

0: 文件 io, 侧重分析通用设备的读写问题

1: 磁盘 io, 侧重分析块设备读写落盘问题

-h (--help) : 显示工具帮助信息。

12.6.3 输出



```
Start-Time:2026-07-08 04:49:10
End-Time:2026-07-08 04:49:11
```

PID	USR	PRIO	<WRITE>	READ	IO	COMMANDS
161179	root	be/4	595B	0B	11B	python
1833	root	be/4	256B	128B	48B	NetworkManager
5258	root	be/4	72B	22.70K	18B	mate-terminal
2930	root	be/4	64B	56B	15B	kwin_x11
2931	root	be/4	32B	16B	6B	kiran-shell
2900	root	be/4	8B	8B	2B	kiran-session-d
2917	root	be/4	8B	8B	2B	kglobalaccel5
2986	root	be/4	8B	8B	2B	kiran-screensav
2932	root	be/4	8B	8B	2B	kiran-shelld
2945	root	be/4	8B	8B	2B	xembedsniproxy

图 12-11 io 观测示意图

如图 12-11: 第一列为显示项的名字:

PID: 进程 id

USR: 用户名

PRIO: io 优先级

WRITE: 写大小, 单位自适应

READ: 大小, 单位自适应

IO: io 的次数

COMMANDS: 运行程序名字

12.6.4 /etc/opsinsight.conf 配置文件

[io]

interval=1

watch=0

type=0

numbers=10

accu=1

logfile=""

#logfile=/var/log/opsinsight-io.log

rotate=2

logsize=1024

console=1

mode=0

13 软件源

麒麟信安服务器操作系统提供可用软件源，包括网络源和镜像源，配置源后可以安装各种软件包。

13.1 网络源配置

在终端查看源文件/etc/yum.repo/kylinsec.repo，如下图 13-1 所示：

```
[everything]
name=KylinSec-$releasever - everything
gpgcheck=1
enabled=1
mirrorlist=https://mirrorlist.kylinsec.com.cn/mirrors/?osversion=$osversion&repo=BaseOS&arch=$basearch
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-kylinsec-release

[EPOL]
name=KylinSec-$releasever - EPOL
gpgcheck=1
enabled=0
mirrorlist=https://mirrorlist.kylinsec.com.cn/mirrors/?osversion=$osversion&repo=EPOL&arch=$basearch
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-kylinsec-release

[CDROM]
name=KylinSec-$releasever - CDROM
gpgcheck=1
enabled=0
baseurl=file:///run/media/$user/KylinSec/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-kylinsec-release

[update]
name=KylinSec-$releasever - update
gpgcheck=1
enabled=1
mirrorlist=https://mirrorlist.kylinsec.com.cn/mirrors/?osversion=$osversion&repo=update&arch=$basearch
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-kylinsec-release

~
~
~

"/etc/yum.repos.d/kylinsec.repo" 27L, 854B
```

图 13-1 系统源文件

无需修改默认源地址，连接好网络确保能访问外网后在终端输入 `dnf clean all && dnf makecache` 更新源。

13.2 镜像源配置

镜像源分为 U 盘和光盘源以及 ISO 镜像源，第一种先介绍如何使用光盘/U 盘源：首先插入刻录好的 U 盘或光盘，在终端查看挂载目录：`df -Th`，如下图 13-2

所示：

```
[root@localhost ~]# df -h
文件系统      大小  已用  可用  已用% 挂载点
/dev/mapper/ko-root 69G  7.3G  58G   12% /
devtmpfs        4.0M    0  4.0M    0% /dev
tmpfs           3.7G    0  3.7G    0% /dev/shm
tmpfs           4.0M    0  4.0M    0% /sys/fs/cgroup
tmpfs           1.5G  9.3M  1.5G    1% /run
tmpfs           3.7G   16K  3.7G    1% /tmp
/dev/sda2       974M  331M  576M   37% /boot
/dev/mapper/ko-home 316G   32K  299G    1% /home
tmpfs           743M   36K  743M    1% /run/user/0
/dev/sr0        3.9G  3.9G    0  100% /run/media/root/KylinSec
```

图 13-2 查看系统文件系统信息

查看到插入的光盘设备是/dev/sr0，挂载点是/run/media/root/KylinSec。

修改源文件/etc/yum.repo/kylinsec.repo，修改 CDROM 源中的 baseurl 为 U 盘或光盘挂载目录，以这里为例：baseurl=file:///run/media/root/KylinSec,设置 CDROM 的 enabled=1，其他源里的 enabled=0，修改后的文件参考下图 13-3：

```
[everything]
name=KylinSec-$releasever - everything
gpgcheck=1
enabled=0
baseurl=https://liulei:Liulei.123@jfrog.kylinsec.com.cn/artifactory/ctrl-OS-Repo/V6-SP1/20260330-2117/os/$basearch
mirrorlist=https://mirrorlist.kylinsec.com.cn/mirrors/?osversion=$osversion&repo=BaseOS&arch=$basearch
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-kylinsec-release

[EPOL]
name=KylinSec-$releasever - EPOL
gpgcheck=1
enabled=0
mirrorlist=https://mirrorlist.kylinsec.com.cn/mirrors/?osversion=$osversion&repo=EPOL&arch=$basearch
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-kylinsec-release

[CDROM]
name=KylinSec-$releasever - CDROM
gpgcheck=1
enabled=1
baseurl=file:///run/media/$user/KylinSec/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-kylinsec-release

[update]
name=KylinSec-$releasever - update
gpgcheck=1
enabled=0
mirrorlist=https://mirrorlist.kylinsec.com.cn/mirrors/?osversion=$osversion&repo=update&arch=$basearch
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-kylinsec-release
```

图 13-3 修改源路径为光盘源

在终端输入 `dnf clean all && dnf makecache` 更新源，如下图 13-4:

```
[root@localhost 桌面]# dnf clean all && dnf makecache
1 file removed
KylinSec-6_SP1 - CDROM 39 MB/s | 15 MB 00:00
Last metadata expiration check: 0:00:01 ago on 2026年04月08日 星期三 10时36分30秒.
Metadata cache created.
```

图 13-4 更新系统源

第二种是镜像源设置，创建一个挂载目录，如：`mkdir /mnt/ISO`，将下载好的 ISO 镜像挂载到该目录：`mount ***.ISO /mnt/ISO`，输入 `df -Th` 命令查看是否挂载成功，如下图 13-5 所示：

```
[root@localhost ~]# ls
公共 视频 文档 音乐 anaconda-ks.cfg KylinSec-Server-6-SP1-beta-260330-2117-x86_64.iso
模板 图片 下载 桌面 initial-setup-ks.cfg
[root@localhost ~]# df -Th
文件系统      类型      大小  已用  可用  已用% 挂载点
/dev/mapper/ko-root ext4      21G   12G   8.5G   58% /
devtmpfs      devtmpfs  4.0M    0   4.0M    0% /dev
tmpfs         tmpfs     2.7G    0   2.7G    0% /dev/shm
tmpfs         tmpfs     4.0M    0   4.0M    0% /sys/fs/cgroup
tmpfs         tmpfs     1.1G   9.3M   1.1G    1% /run
tmpfs         tmpfs     2.7G   4.0K   2.7G    1% /tmp
/dev/vda2     ext4     974M  332M  575M   37% /boot
/dev/mapper/ko-home ext4      11G   44K   9.7G    1% /home
tmpfs         tmpfs     540M   16K   540M    1% /run/user/0
/dev/sr0      iso9660   3.9G   3.9G    0  100% /run/media/root/KylinSec
/dev/loop0    iso9660   3.9G   3.9G    0  100% /mnt/ISO
[root@localhost ~]#
```

图 13-5 挂载 ISO 镜像到本地

然后修改源文件/etc/yum.repo/kylinsec.repo, 修改 CDROM 源中的 baseurl 为挂载目录, 这里以/mnt/ISO 为例: baseurl=file:///mnt/ISO, 设置 CDROM 的 enabled=1, 其他源里的 enabled=0, 修改后的文件参考下图 13-6 本地挂载目录:

```
[everything]
name=KylinSec-$releasever - everything
gpgcheck=1
enabled=0
baseurl=https://liulei:Liulei.123@jfrog.kylinsec.com.cn/artifactory/ctrl-OS-Repas/V6-SP1/20260330-2117/os/$basearch
mirrorlist=https://mirrorlist.kylinsec.com.cn/mirrors/?osversion=$osversion&repo=BaseOS&arch=$basearch
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-kylinsec-release

[EPOL]
name=KylinSec-$releasever - EPOL
gpgcheck=1
enabled=0
mirrorlist=https://mirrorlist.kylinsec.com.cn/mirrors/?osversion=$osversion&repo=EPOL&arch=$basearch
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-kylinsec-release

[CDROM]
name=KylinSec-$releasever - CDROM
gpgcheck=1
enabled=1
baseurl=file:///mnt/ISO
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-kylinsec-release

[update]
name=KylinSec-$releasever - update
gpgcheck=1
enabled=0
mirrorlist=https://mirrorlist.kylinsec.com.cn/mirrors/?osversion=$osversion&repo=update&arch=$basearch
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-kylinsec-release
```

图 13-6 本地挂载目录

然后在终端输入 dnf clean all && dnf makecache 更新源, 如下图 13-7

所示：

```
[root@localhost ~]# dnf clean all && dnf makecache
6 files removed
KylinSec-6_SPl - CDR0M                                40 MB/s | 2.6 MB    00:00
Last metadata expiration check: 0:00:01 ago on 2026年04月08日 星期三 11时03分09秒.
Metadata cache created.
```

图 13-7 更新源

13.3 安装软件包

源配置完成后，可以下载安装需要的软件包，如下载 gcc-c++：首先可以在终端输入“dnf list 包名”查找源里是否存在该包，若存在则输入“dnf install 包名”进行安装，如下图 13-8 所示：

```
[root@localhost ~]# dnf list gcc-c++
Last metadata expiration check: 0:01:12 ago on 2026年04月08日 星期三 11时03分09秒.
Available Packages
gcc-c++.x86_64                                12.3.1-98.ks6.kb4                                CDR0M
[root@localhost ~]# dnf install gcc-c++
Last metadata expiration check: 0:01:33 ago on 2026年04月08日 星期三 11时03分09秒.
Dependencies resolved.
=====
Package                                Architecture    Version                                Repository    Size
=====
Installing:
gcc-c++                                x86_64          12.3.1-98.ks6.kb4                    CDR0M         13 M
Installing dependencies:
libstdc++-devel                        x86_64          12.3.1-98.ks6.kb4                    CDR0M         2.6 M
Transaction Summary
=====
Install 2 Packages
Total size: 16 M
Installed size: 51 M
Is this ok [y/N]: y
Downloading Packages:
Running transaction check
Transaction check succeeded.
```

图 13-8 安装 gcc-c++包

以安装 gcc-c++包为例，输入“dnf list gcc-c++”查看源里有该包，然后输入“dnf install gcc-c++.x86_64”进行安装，输入“y”确定安装，显示“Complete!”表示安装完成。

14 免责声明

我们尊重版权，也致力于保护版权，如果您在使用我们的试用期麒麟信安服务器操作系统，请尽快激活。试用期时间为半年，试用期过期后系统会出现提示

“请转到‘控制面板’激活您的系统”，请尽快注册正式版麒麟信安服务器操作系统。详细激活方法请参考 8.2.1.1.1 章节。由于用户未及时注册系统，导致系统问题，造成的一切后果，本公司概不负责。

附录

常见问题解答

问：麒麟信安服务器操作系统的产品定位是什么？

答：麒麟信安服务器操作系统 V6 SP1 紧跟 openEuler 社区发展，本次发布的 KylinSec-Server-6-SP1（以下简称 KylinSec 6 SP1）是以 openEuler 24.03-LTS-SP2 社区发行版为上游供应链，通过整合自研组件和开源软件研制，是一款针对云桌面、轻量级服务器虚拟化、应用虚拟化、分布式存储、集群管理、容器、数据库、大数据、人工智能等应用场景的具有自主性、安全性、高可靠性、高性能的自研服务器操作系统。

问：麒麟信安操作系统的版本是如何划分的？

答：麒麟信安操作系统分为服务器操作系统、桌面操作系统以及工控操作系统。

问：麒麟信安服务器操作系统支持哪些数据库？

答：麒麟信安服务器操作系统支持 openGauss、PostgreSQL、MariaDB、MySQL 等主流数据库，具体信息请在我司官网中进行查询。

问：麒麟信安服务器操作系统容易安装吗？

答：麒麟信安服务器操作系统为用户提供了友好的图形化安装界面，用户根

据安装界面中的提示即可顺利完成系统的安装。

问：麒麟信安服务器操作系统软碟通刻录应该选什么写入方式？

答：麒麟信安服务器操作系统使用软碟通刻录镜像时，选择 raw 写入方式

问：麒麟信安服务器操作系统中有隐藏文件的功能吗？

答：在麒麟信安服务器操作系统中可以通过将文件重命名为“.xxx”来隐藏文件，但是系统现在暂时不支持图形化的文件隐藏工具。

问：在麒麟信安服务器操作系统下怎样显示 Windows 文本文件？

答：在麒麟信安服务器操作系统中，文本编辑器可以自动识别文本文件的编码格式，用户可以通过点击【开始菜单→工具→文本编辑器】打开文本编辑器来显示 Windows 文本文件。

问：麒麟信安服务器操作系统怎样提示安装各种驱动？

答：在麒麟信安服务器操作系统安装完毕之后可以使用第三方软件安装工具安装由第三方提供的驱动程序，如 NVIDIA、ATI 显卡驱动。若在系统安装过程中出现缺少驱动的现象，重新启动安装界面手动输入 Linux dd 命令，选择需要加载的驱动，这些驱动的搜索范围需要由光盘或者 U 盘提供。

问：在麒麟信安服务器操作系统中如何对弹出菜单后的屏幕进行截图？

答：在麒麟信安服务器操作系统中若需要对弹出菜单后的屏幕进行截屏，使用截图工具，设置截图延迟时间，在时间范围内弹出需要的菜单，等待预先设定的延迟秒数后系统会自动截图并显示“保存图像”对话框。

问：安装第三方 QT 库造成同名库冲突问题如何避免？

答：对第三方安装的 Qt 库配置 LD_LIBRARY_PATH 环境变量时，请勿配置

到全局上，需缩小范围应用到具体业务上，以避免引起冲突。可通过 `ldd` 命令设置验证仅目标应用加载第三方 Qt 库，避免影响其他应用。使用 `ldd /path/to/your_application | grep -i qt` 查看应用依赖的 Qt 库及其解析路径，使用 `LD_LIBRARY_PATH=/path/to/third_party/qt/lib:$LD_LIBRARY_PATH` 进行临时设置。

问：为什么在文档开启的情况下仍然可以删除文档但开启的该文档窗口依然存在？

答：麒麟信安服务器操作系统的编辑器 vi、vim 以及 gedit 提供了正在读写的文件可删除的灵活机制，这在某些方面给用户提供了方便，但是对于普通用户来说，删除文件之前要注意该文件是否被打开。若用户使用 gedit 编辑器时错误删除正在打开的文件，在关闭编辑器时系统将提示是否保存当前文件。若使用的编辑器为 vi 或 vim，请在错误删除文件后选择保存退出。

问：湖南麒麟信安的客户服务热线号码是什么？

答：400-012-6606。

问：客户在运行麒麟信安服务器操作系统出现问题时需要提供的信息是什么？

答：请提供产品版本、产品的运行环境和问题的详细描述。

📍 总部：湖南省长沙市岳麓区麒麟科技园
📈 股票代码：688152
🌐 www.kylinsec.com.cn
☎ 咨询热线：400-012-6606
湖南麒麟信安科技股份有限公司

