基础概念

什么是 Linux

定义

Linux 是一个类 Unix 操作系统，是基于 POSIX 的多用户、多任务、支持多线程和多 CPU 的操作系统。能运行主要的 UNIX 工具软件、应用程序和网络协议。支持 32 位和 64 位硬件。Linux 拥有广泛的硬件支持和用户群体。

特点

- **开源性**：Linux 的源代码可以被任何人自由地使用、修改和发布。
- **安全性**：Linux 提供了多种安全机制，包括访问控制、安全审计和加密。
- **多用户支持**：Linux 是一个多用户系统，一个系统可以有多个用户同时登录和使用。
- **多任务**：Linux 支持多任务操作，即同时运行多个任务。

Unix 和 Linux 有什么区别?

起源

- **Unix**：Unix 是在 20 世纪 60 年代末至 70 年代初由 AT&T 的贝尔实验室开发的。
- **Linux**：Linux 是在 1991 年由林纳斯·托瓦兹（Linus Torvalds）创建的，最初是作为对 Unix 的一个免费和开源的替代品。

开源性

- **Unix**：大多数 Unix 系统都是闭源、专有的。
- **Linux**：Linux 是开源的，任何人都可以下载源代码并对其进行修改和分发。

发行版和可移植性

- **Unix**：Unix 有多个版本，通常特定于硬件。
- **Linux**：Linux 可在多种硬件平台上运行，有众多发行版，如 Ubuntu、Fedora 和 Debian 等。

什么是 Linux 内核?

定义

Linux 内核是 Linux 操作系统的核心，**负责管理硬件资源**，提供各种程序运行的基本环境。

主要功能

- **进程管理**：调度进程、管理进程生命周期。
- **内存管理**：内存分配和回收，内存页管理。
- **文件系统**：提供文件的存储和访问。
- **设备控制**：管理各种硬件设备。

- **网络功能：** 实现网络协议栈，管理网络通信。

Linux 的基本组件

核心组件

- **内核（Kernel）：** 控制计算机硬件。
- **系统库（System Libraries）：** 这些特殊的函数或程序使用内核的功能，不需要内核代码的访问权。
- **系统工具（System Utilities）：** 这些是完成特定任务的应用程序。

用户界面

- **命令行界面（CLI）：** 通过命令行输入与系统交互。
- **图形用户界面（GUI）：** 提供图形化界面，用户通过点击、拖拽等方式与系统交互。

Linux 的体系结构

简述

Linux 的体系结构主要由以下几个部分组成：

1. **硬件层（Hardware）：** Linux 支持的物理硬件。
2. **内核空间（Kernel Space）：** 内核是与系统硬件直接交互的部分。
3. **用户空间（User Space）：** 运行用户进程和应用程序的地方，与内核空间隔离。

层次结构

- **硬件层：** 提供基本的计算资源，包括 CPU、内存、I/O 设备等。
- **内核层：** 包括系统调用接口、进程管理、内存管理、文件系统、网络管理等。
- **用户空间：** 用户空间包括**系统库（为应用程序提供核心功能）、系统工具和应用程序**。用户空间的程序通过系统调用与内核空间通信。

BASH 和 DOS 之间的基本区别

BASH (Bourne Again SHell)

- **平台：** 主要用于 Unix 和 Linux 系统。
- **功能：** 支持脚本编程，拥有强大的命令行编辑、任务控制、函数和数组等特性。
- **脚本扩展名：** 通常为 `.sh` 或 `.bash`。
- **大小写敏感：** 命令和文件名对大小写敏感。
- **文件系统导航：** 使用 `/` 作为路径分隔符。

DOS (Disk Operating System)

- **平台：** 主要用于 Windows 系统。
- **功能：** 功能相对简单，主要用于基本的文件操作、任务执行等。
- **脚本扩展名：** 通常为 `.bat` 或 `.cmd`。
- **大小写敏感：** 命令和文件名不区分大小写。
- **文件系统导航：** 使用 `\` 作为路径分隔符。

Linux 开机启动过程

步骤概览

1. **BIOS/UEFI**：自检并初始化硬件。
2. **Boot Loader（如 GRUB）**：加载内核。
3. **内核初始化**：初始化系统和硬件资源。
4. **Init 进程**：系统的第一个进程（PID 为 1）。
5. **运行级别脚本**：根据默认运行级别执行相应脚本。
6. **登录**：用户输入用户名和密码进行登录。

Linux 系统缺省的运行级别

- **运行级别 0**：关机。
- **运行级别 1**：单用户模式。
- **运行级别 2**：多用户模式，没有网络服务。
- **运行级别 3**：完全的多用户模式，有网络服务，通常是缺省运行级别。
- **运行级别 4**：未使用，用户可自定义。
- **运行级别 5**：X11，图形界面。
- **运行级别 6**：重启。

Linux 使用的进程间通信方式

- **信号（Signals）**
- **管道（Pipes）**
- **消息队列（Message Queues）**
- **共享内存（Shared Memory）**
- **信号量（Semaphores）**
- **套接字（Sockets）**

Linux 有哪些系统日志文件?

- **/var/log/message**：系统启动后的信息和错误消息。
- **/var/log/auth.log**：用户授权信息。
- **/var/log/kern.log**：内核日志。
- **/var/log/cron.log**：定时任务cron的日志。
- **/var/log/maillog**：邮件服务器日志。
- **/var/log/boot.log**：系统启动日志。
- **/var/log/dmesg**：系统启动时内核检测到的硬件和启动信息。

Linux 系统安装多个桌面环境有帮助吗?

- **优点**：
  - 用户可以选择最适合自己的桌面环境。
  - 开发者可以在不同的环境中测试软件。
- **缺点**：
  - 占用更多的磁盘空间。

    - 可能导致系统配置和管理更加复杂。
    - 不同桌面环境间可能存在兼容性问题。

什么是交换空间?

- **定义：** 交换空间是硬盘驱动器上的一部分空间，当物理内存不足时，系统会将内存中的数据临时存储到交换空间中。
- **作用**：
    - 扩展物理内存，允许系统运行更多的应用程序。
    - 有助于内存管理，提高系统的灵活性和稳定性。

什么是 root 帐户

定义

- **root 帐户**：在 Linux 和 Unix 系统中，root 是最高权限的用户帐户。通常被称为超级用户或管理员帐户。

特点

- **无限权限**：root 帐户可以访问系统上的所有文件和命令。
- **系统管理**：用于执行系统级的任务，如安装软件、更改关键配置文件、管理用户帐户等。
- **风险性**：由于其强大的权限，不当使用 root 帐户可能导致系统不稳定或安全问题。

什么是 LILO?

定义

- **LILO（Linux Loader）**：是一种早期的 Linux 启动加载器，负责加载 Linux 操作系统到内存中，启动操作系统。

特点

- **功能性**：LILO 可以处理多种操作系统的启动过程。
- **配置文件**：配置文件为 `/etc/lilo.conf`，在更改后需要运行 `lilo` 命令来激活更改。
- **替代品**：现在 LILO 很少使用，大多被 GRUB（GRand Unified Bootloader）所替代，后者提供更多功能和灵活性。

什么是 BASH?

定义

- **BASH（Bourne Again SHell）**：是 Linux 默认的命令行界面和脚本语言解释器。

特点

- **兼容性**：BASH 兼容于原始的 Bourne Shell（sh），同时加入了许多新特性。
- **功能性**：支持命令行历史、自动补全、文件名通配、管道、背景执行等特性。

什么是 CLI?

定义

- **CLI（Command Line Interface）**：是一种允许用户通过输入文本命令来与程序交互的界面。

特点

- **资源消耗少**：与图形界面相比，CLI 占用更少的系统资源。
- **自动化和脚本**：非常适合自动化任务和脚本编写。
- **学习曲线**：可能需要时间学习，但提供了强大的灵活性和控制能力。

什么是 GUI?

定义

- **GUI（Graphical User Interface）**：是一种允许用户通过图形图标和视觉指示器与程序交互的界面。

特点

- **易用性**：直观，易于理解和使用，特别是对初学者。
- **资源消耗**：通常比 CLI 消耗更多资源。
- **功能性**：适合多任务操作，易于导航。

开源的优势是什么?

优势

- **成本效益**：通常是免费的，减少了软件购买和升级的费用。
- **透明性**：源代码开放，用户可以理解软件的工作原理。
- **可定制性**：用户可以根据自己的需求修改软件。
- **安全性**：更多的眼睛检查代码可能意味着更少的安全漏洞。
- **社区支持**：庞大的开发者社区支持，提供帮助和资源。

GNU 项目的重要性是什么?

重要性

- **开源运动**：GNU 项目是开源和自由软件运动的先驱。
- **软件自由**：致力于用户的自由使用、学习、修改和分享软件。
- **工具和系统**：提供了许多重要的工具和系统，如 GCC（GNU Compiler Collection）、GNU Bash shell 和其他许多工具，这些都是现代 Linux 发行版的重要组成部分。
- **GNU 通用公共许可证（GPL）**：推动了一个广泛的软件许可模型，保护了用户的自由，同时鼓励软件的共享和改进。

Shell

Shell 脚本

定义

- **Shell 脚本**：是一种用于自动化执行多个命令的脚本语言。是由一系列命令组成的文本文件，通过 Shell 程序解释执行。

特点

- **自动化任务**：可以用于自动化常见的任务，如文件管理、程序执行等。
- **简洁性**：相比其他编程语言，Shell 脚本通常更简洁。
- **跨平台性**：大多数 Shell 脚本可以在不同的 Unix/Linux 系统上无缝运行。

变量

变量类型

- **局部变量**：在单个脚本或命令中定义和使用。
- **环境变量**：在所有程序和 Shell 会话中全局有效。
- **位置参数**：脚本命令行中的参数，如 `$1`，`$2` 等。
- **特殊变量**：由 Shell 预定义的特殊变量，如 `$0` (脚本名称)，`$#` (参数数量)，`$$` (脚本的进程 ID) 等。

if

语法示例

```
if [ condition1 ]; then
    if [ condition2 ]; then
        # 执行当 condition1 和 condition2 都为真时的命令
    else
        # 执行当 condition1 为真，但 condition2 为假时的命令
    fi
else
    # 执行当 condition1 为假时的命令
fi
```

case

语法示例

```
case $variable in
    pattern1)
        # 如果 $variable 匹配 pattern1，执行这里的命令
        ;;
    pattern2)
        # 如果 $variable 匹配 pattern2，执行这里的命令
        ;;
    *)
        # 如果 $variable 与任何模式都不匹配，执行这里的命令
        ;;
esac
```

for

语法示例

- **基于列表的循环**

```
for var in list
do
    # 对列表中的每个元素执行的命令
done
```

- **基于范围的循环（C 风格）**

```
for (( i=0; i<10; i++ ))
do
    # 对从 0 到 9 的每个数字执行的命令
done
```

while

语法示例

```
while [ condition ]
do
    # 当 condition 为真时，执行的命令
done
```

break

定义

- **break 命令**：用于立即退出循环（for、while、until 循环），不再执行剩余的循环体。

示例

```
for i in {1..5}
do
    if [ "$i" -eq "3" ]; then
        break  # 当 i 等于 3 时退出循环
    fi
    echo "Number is $i"
done
```

continue

定义

- **continue 命令**：用于跳过当前循环的剩余部分，直接进入下一次循环。

示例

```
for i in {1..5}
do
    if [ "$i" -eq "3" ]; then
        continue  # 当 i 等于 3 时，跳过当前循环，继续下一个循环
    fi
    echo "Number is $i"
done
```

使脚本可执行

步骤

1. **添加 shebang**：在脚本文件的第一行添加 `#!/bin/bash`。
2. **修改权限**：使用 `chmod +x script_name.sh` 命令使脚本具有执行权限。

调试 Shell 脚本

方法

- **使用 `-x` 选项**：运行脚本时使用 `bash -x script_name.sh`，会在执行每一条命令前打印该命令。
- **在脚本中设置**：在脚本中添加 `set -x` 来开始调试，添加 `set +x` 来停止调试。

标准输出和错误输出同时重定向到同一位置

语法示例

```
command > file 2>&1
# 或者
command &> file
```

这两个命令都会将标准输出（stdout）和标准错误（stderr）重定向到同一个文件。

定义函数

语法示例

```
function_name() {
    # 函数体
}
```

算术运算

方法

- **使用 `expr`**：

  ```
  result=`expr $a + $b`
  ```

- **使用双括号**：

  ```
  result=$((a + b))
  ```

命令

文件管理命令

cat 命令

定义

- **cat (concatenate)**：用于查看、创建或合并文件的内容。

常用选项

- **查看内容**：`cat file.txt` - 显示文件内容。
- **合并文件**：`cat file1.txt file2.txt > merged_file.txt` - 合并文件内容。
- **创建文件**：`cat > file.txt` - 创建一个新文件并等待输入内容。
- **追加内容**：`cat >> file.txt` - 在文件末尾追加内容。

chmod 命令

定义

- **chmod (change mode)**：用于改变文件或目录的访问权限。

常用选项

- **使用符号**：`chmod u+x file.txt` - 给文件所有者增加执行权限。
- **使用数字**：`chmod 755 file.txt` - 将文件的权限设置为 755 (rwxr-xr-x)。

chown 命令

定义

- **chown (change owner)**：用于改变文件或目录的所有者和所属组。

常用选项

- **改变所有者**：`chown user file.txt` - 改变文件所有者为 user。
- **改变所有者和组**：`chown user:group file.txt` - 同时改变文件的所有者和组。
- **递归改变**：`chown -R user:group directory` - 递归改变目录及其内部所有文件的所有者和组。

cp 命令

定义

- **cp (copy)**：用于复制文件或目录。

常用选项

- **复制文件**：`cp source_file target_file` - 将 source_file 复制到 target_file。
- **递归复制目录**：`cp -r source_directory target_directory` - 将整个目录及其内容复制到新位置。
- **保留属性**：`cp -p file1 file2` - 复制时保留原文件的属性。

find 命令

定义

- **find**：用于在目录树中查找文件，并对找到的文件执行指定的操作。

常用选项

- **按名称查找**：`find /path -name filename` - 在指定路径下查找名为 filename 的文件。
- **按类型查找**：`find /path -type f` - 查找所有文件。
- **按修改时间**：`find /path -mtime +10` - 查找在过去 10 天内被修改过的文件。
- **执行操作**：`find /path -type f -exec cat {} \;` - 对查找到的每个文件执行 cat 命令。

head 命令

定义

- **head**：用于查看文件的开头部分内容，默认显示前10行。

常用选项

- **指定行数**：`head -n 5 file.txt` - 显示文件的前5行。

less 命令

定义

- **less**：用于分页查看文件内容，可以前后翻页浏览。

常用操作

- **打开文件**：`less file.txt` - 打开文件进行查看。
- **导航**：使用方向键或 `PgUp/PgDn` 进行导航。

ln 命令

定义

- **ln**：用于创建链接，包括硬链接和软链接（符号链接）。

常用选项

- **创建硬链接**：`ln source_file link_name` - 创建一个指向 source_file 的硬链接。硬链接是对文件的另一种引用。指向文件系统中的数据块（或 inode），与原始文件共享相同的 inode。
- **创建软链接**：`ln -s source_file link_name` - 创建一个指向 source_file 的软链接。软链接是指向文件或目录的路径的引用。类似于 Windows 中的快捷方式。

locate 命令

定义

- **locate**：用于快速查找文件系统中的文件，基于事先构建的数据库。

常用选项

- **查找文件**：`locate filename` - 快速显示所有包含该文件名的文件路径。

more 命令

定义

- **more**：用于分页显示文本文件内容，只能向前翻页。

常用操作

- **打开文件**：`more file.txt` - 打开文件进行查看。
- **导航**：使用空格键翻页，使用 `q` 退出。

mv 命令

定义

- **mv (move)**：用于移动文件或目录，或重命名文件或目录。

常用选项

- **移动文件**：`mv source_file target_directory` - 将文件移动到指定目录。
- **重命名文件**：`mv old_name new_name` - 将文件或目录重命名。

rm 命令

定义

- **rm (remove)**：用于删除文件或目录。

常用选项

- **删除文件**：`rm file.txt` - 删除文件。
- **递归删除**：`rm -r directory` - 递归删除目录及其内容。
- **强制删除**：`rm -f file.txt` - 强制删除文件，不提示确认。

tail 命令

定义

- **tail**：用于查看文件的尾部内容，默认显示最后10行。

常用选项

- **指定行数**：`tail -n 5 file.txt` - 显示文件的最后5行。

touch 命令

定义

- **touch**：用于创建空文件或修改文件时间戳。

常用选项

- **创建空文件**：`touch new_file.txt` - 创建一个新的空文件。
- **修改时间戳**：`touch -t 202101010000 file.txt` - 修改文件的时间戳。

vim 命令

定义

- **vim**：一种模式编辑器，具有多种功能，用于编辑文本文件。

常用操作

- **打开文件**：`vim file.txt` - 打开文件进行编辑。
- **模式切换**：从普通模式切换到插入模式（`i`），从插入模式返回普通模式（`Esc`）。
- **保存和退出**：在普通模式下，`:w` 保存文件，`:q` 退出，`:wq` 保存并退出。

whereis 命令

定义

- **whereis**：用于定位二进制文件、源代码文件和手册页文件的位置。

常用选项

- **查找文件**：`whereis program` - 显示程序的二进制文件、源代码和手册页的位置。

which 命令

定义

- **which**：搜索用户的 `PATH` 环境变量中列出的目录，返回可执行文件的路径。

常用选项

- **查找可执行文件**：`which program` - 显示程序的完整路径。

文档编辑命令

grep 命令

定义

- **grep (global regular expression print)**：用于搜索文件内容，并打印出匹配的行。

常用选项

- **基本搜索**：`grep 'pattern' filename` - 在文件中搜索模式。
- **递归搜索**：`grep -r 'pattern' directory` - 在目录及其子目录中递归搜索。
- **忽略大小写**：`grep -i 'pattern' filename` - 搜索时忽略大小写。
- **显示行号**：`grep -n 'pattern' filename` - 显示匹配行及其行号。

wc 命令

定义

- **wc (word count)**：统计文件的行数、单词数和字符数。

常用选项

- **行数**：`wc -l filename` - 显示文件的行数。
- **单词数**：`wc -w filename` - 显示文件的单词数。
- **字符数**：`wc -c filename` - 显示文件的字符数。

磁盘管理命令

---

cd 命令

定义

- **cd (change directory)**：用于切换当前工作目录。

常用用法

- **切换目录**：`cd /path/to/directory` - 切换到指定目录。
- **回到主目录**：`cd ~` 或 `cd` - 切换到用户的主目录。
- **上级目录**：`cd ..` - 切换到上级目录。

df 命令

定义

- **df (disk free)**：显示磁盘空间的使用情况。

常用选项

- **显示所有文件系统**：`df -a` - 显示所有文件系统的磁盘使用情况。
- **人类可读格式**：`df -h` - 以易于阅读的格式（如 KB、MB、GB）显示信息。

du 命令

定义

- **du (disk usage)**：统计目录或文件的磁盘使用空间。

常用选项

- **统计目录**：`du /path/to/directory` - 显示指定目录的磁盘使用量。
- **人类可读格式**：`du -h /path/to/directory` - 以易于阅读的格式显示信息。

ls 命令

定义

- **ls (list)**：列出目录内容。

常用选项

- **显示详细信息**：`ls -l` - 详细列出文件和目录信息。
- **显示隐藏文件**：`ls -a` - 包括隐藏文件在内的所有文件和目录。
- **按时间排序**：`ls -lt` - 按修改时间排序。

mkdir 命令

定义

- **mkdir (make directory)**：用于创建新目录。

常用选项

- **创建目录**：`mkdir directory_name` - 创建一个新目录。
- **递归创建目录**：`mkdir -p /path/to/directory` - 创建路径中的所有必需的父目录。

pwd 命令

定义

- **pwd (print working directory)**：显示当前工作目录的完整路径。

rmdir 命令

定义

- **rmdir (remove directory)**：用于删除空目录。

常用选项

- **删除目录**：`rmdir directory_name` - 删除空目录。
- **递归删除**：`rmdir -p /path/to/directory` - 若目录为空，递归删除路径中的所有父目录。

网络通讯命令

ifconfig 命令

定义

- **ifconfig (interface configuration)**：用于配置和显示系统网络接口的参数。

常用选项

- **查看所有接口**：`ifconfig -a` - 显示所有接口，包括未激活的。
- **配置接口**：`ifconfig interface ip_address` - 为指定接口设置 IP 地址。

iptables 命令

定义

- **iptables**：是 Linux 上的一个命令行防火墙工具，用于设置、维护和检查 IP 数据包过滤规则。

常用选项

- **查看规则**：`iptables -L` - 列出所有规则。
- **添加规则**：`iptables -A chain_rule` - 向指定链添加规则。
- **删除规则**：`iptables -D chain_rule` - 从指定链删除规则。

netstat 命令

定义

- **netstat (network statistics)**：显示网络连接、路由表、接口统计、伪装连接等网络相关信息。

常用选项

- **显示所有端口**：`netstat -a` - 显示所有端口的监听状态。
- **显示路由表**：`netstat -r` - 显示路由表。

ping 命令

定义

- **ping**：用于测试主机之间网络的连通性。

常用选项

- **发送请求**：`ping hostname_or_ip` - 向指定的主机名或 IP 地址发送 ICMP ECHO_REQUEST。

telnet 命令

定义

- **telnet**：用于远程登录到服务器。是一个不安全的协议，建议使用 SSH 替代。

常用选项

- **远程登录**：`telnet hostname_or_ip` - 连接到指定的主机名或 IP 地址。

系统管理命令

date 命令

定义

- **date**：显示或设置系统日期和时间。

常用选项

- **显示当前日期时间**：`date` - 显示当前系统的日期和时间。
- **设置日期时间**：`date MMDDhhmm[[CC]YY][.ss]` - 设置系统的日期和时间。

free 命令

定义

- **free**：显示系统内存的使用情况。

常用选项

- **显示内存情况**：`free -h` - 以人类可读的格式显示内存使用情况。

kill 命令

定义

- **kill**：用于终止进程。

常用选项

- **终止进程**：`kill pid` - 终止指定 PID 的进程。
- **强制终止**：`kill -9 pid` - 强制终止指定 PID 的进程。

ps 命令

定义

- **ps (process status)**：显示当前进程的状态。

常用选项

- **显示所有进程**：`ps -e` - 显示系统中所有的进程。
- **详细信息**：`ps aux` - 显示进程的详细信息。

rpm 命令

定义

- **rpm (Red Hat Package Manager)**：用于管理 Red Hat 系统上的软件包。

常用选项

- **安装软件包**：`rpm -i package.rpm` - 安装 RPM 包。
- **卸载软件包**：`rpm -e package` - 卸载 RPM 包。

top 命令

定义

- **top**：实时显示系统中各个进程的资源占用状况。

常用操作

- **查看进程**：直接运行 `top`。
- **退出**：按 `q` 键退出。

yum 命令

定义

- **yum (Yellowdog Updater, Modified)**：用于 RPM 兼容的 Linux 系统上的包管理。

常用选项

- **安装软件包**：`yum install package` - 安装软件包。
- **更新软件包**：`yum update package` - 更新软件包。
- **卸载软件包**：`yum remove package` - 卸载软件包。

备份压缩命令

---

bzip2 命令

定义

- **bzip2**：用于压缩和解压文件，使用 Burrows-Wheeler 块排序文本压缩算法和 Huffman 编码。

常用选项

- **压缩文件**：`bzip2 filename` - 压缩文件。
- **解压文件**：`bzip2 -d filename.bz2` - 解压文件。

gzip 命令

定义

- **gzip (GNU zip)**：用于压缩和解压文件。

常用选项

- **压缩文件**：`gzip filename` - 压缩文件。
- **解压文件**：`gzip -d filename.gz` - 解压文件。

tar 命令

定义

- **tar (tape archive)**：用于打包和解包文件。

常用选项

- **创建归档文件**：`tar -cvf archive_name.tar directory_or_file` - 创建归档文件。
- **解包归档文件**：`tar -xvf archive_name.tar` - 解包归档文件。
- **查看归档内容**：`tar -tvf archive_name.tar` - 查看归档文件中的内容。

unzip 命令

定义

- **unzip**：用于解压缩 ZIP 压缩文件。

常用选项

- **解压文件**：`unzip filename.zip` - 解压 ZIP 文件。
- **列出压缩文件内容**：`unzip -l filename.zip` - 列出 ZIP 文件的内容。