

# 原理和实践

## 方向

- 版本号知识 如果要做到不同版本的升级 需要区分版本号
- 配置文件热更新:1: 把新配置加载到内存, 使用的地方从内存中读取,直接修改内存的值 实现不重启系统更新 2 :全局定时器 随时检查每个配置文件的版本 如果版本更替 通知使用者重新加载配置文件
- 用户配置文件和系统配置文件隔离 更新只更新系统文件 新的系统如果要用用户配置 给个默认
- 参考配置文件管理热更新:nacos
- 参考linux系统的软件包更新指令

## 文件系统initrd,kernel,rootfs

### initrd文件系统

是一个临时的根文件系统, 用于 Linux 启动过程中提供必要的文件系统支持, 使系统能够引导到真正的根文件系统-

它通常包含了一些必要的驱动程序和工具, 用于加载和配置硬件设备, 以及进行文件系统挂载等操作。

initrd 的主要作用是在启动过程中为系统提供一个临时的环境, 以便引导到真正的根文件系统。

### 原理

1. **引导过程**: 在 Linux 系统引导过程中, 内核会首先被加载到内存中并执行。然后, 内核会检测硬件设备、初始化系统并尝试挂载根文件系统。在这个阶段, 内核需要访问一些文件和工具来完成引导过程。 注:系统启动后 initrd会被内存卸载释放资源
2. **硬件依赖**: 在内核能够识别和初始化系统的硬件之前, 它需要一些基本的工具和驱动程序。这些工具和驱动程序通常被放置在根文件系统上。但在内核能够挂载根文件系统之前, 它无法访问这些文件。
3. **解决方案**: 为了解决这个问题, Linux 使用了 initrd。initrd 是一个临时的根文件系统, 它被包含在内核镜像中或者作为单独的文件加载到内存中。initrd 中包含了一些基本的文件和工具, 如命令行工具、驱动程序等, 以便内核能够顺利启动。
4. **内核启动过程**: 在启动过程中, 当内核加载 initrd 后, 它会将其挂载为根文件系统, 并执行其中的初始化脚本。这些脚本通常用于加载更多的驱动程序、检测硬件、设置网络等操作。
5. **切换根文件系统**: 一旦 initrd 完成了必要的准备工作, 内核就会卸载 initrd, 并将控制权转移到真正的根文件系统上。这个根文件系统可以是硬盘、网络存储或者其他存储介质上的文

件系统。

## 正常运行时修改initrd的配置信息

1. **挂载 initrd 文件系统：**首先，你需要将 initrd 文件系统挂载到一个临时目录中，以便能够访问其中的文件。你可以使用 `mount` 命令来挂载 initrd 文件。例如：

```
mkdir /tmp/initrd_mount mount -o loop /path/to/initrd_file /tmp/initrd_mount
```

2. **修改配置文件：**一旦成功挂载了 initrd 文件系统，你可以进入挂载点目录 `/tmp/initrd_mount`，并修改其中的配置文件或其他文件，根据你的需求进行修改。
3. **卸载 initrd 文件系统：**完成修改后，确保保存了所有的更改。然后，使用 `umount` 命令将 initrd 文件系统卸载。

```
umount /tmp/initrd_mount
```

4. **重新生成 initrd：**如果你修改了 initrd 中的关键文件，如初始化脚本或驱动程序，你可能需要重新生成 initrd。具体的生成方法取决于你的 Linux 发行版，通常可以使用 `mkinitrd` 或 `dracut` 等工具来重新生成 initrd。
5. **重启系统：**一旦完成了以上步骤，你可以重启系统以应用你的更改。

## kernel

- kernel 是 Linux 操作系统的核心，负责管理系统的各种资源和提供各种服务。
- 它包含了操作系统的核心功能，如进程管理、内存管理、文件系统管理、设备驱动程序等。
- kernel 被加载到内存中后，负责初始化系统并启动用户空间的进程，从而完成系统的启动过程。

## 功能

1. **资源管理：**内核负责管理计算机系统的各种硬件资源，包括处理器（CPU）、内存、外部设备（如硬盘、键盘、鼠标、网络接口卡等）以及其他的系统资源。内核通过调度算法管理 CPU 的使用，分配内存空间给不同的进程，并处理设备的输入输出请求。
2. **系统调用：**内核提供了系统调用（*system calls*）接口，允许应用程序通过调用内核提供的函数来执行特权操作，如文件操作、网络通信、进程管理等。通过系统调用，应用程序可以访问内核提供的服务和功能，从而与硬件交互并完成各种任务。
3. **设备驱动程序：**内核包含了各种设备驱动程序，用于控制和管理硬件设备。这些驱动程序允许内核与硬件交互，通过提供统一的接口使得应用程序可以与硬件设备进行通信。内核会自动加载适当的设备驱动程序以支持系统中存在的硬件设备。
4. **中断处理：**内核负责处理硬件中断，当硬件设备发生事件（如输入输出操作完成、定时器到期等）时，内核会中断当前执行的程序并处理相应的中断请求。这确保了系统能够及时响应硬件事件，并进行必要的处理。

5. **虚拟内存管理**：内核负责管理系统的虚拟内存，包括内存分页、页面置换、内存映射等功能。通过虚拟内存管理，内核可以为每个进程提供独立的虚拟地址空间，实现内存隔离和保护，并通过页面置换策略优化内存使用效率。
6. **安全性和权限控制**：内核负责实施安全性和权限控制机制，确保系统资源和敏感数据不被未授权的应用程序访问。通过用户身份验证、访问控制列表（ACLs）、权限位等机制，内核可以限制用户和应用程序对系统资源的访问权限。

## 运行时候修改系统文件

1. **了解内核配置文件**：了解你所使用的操作系统的内核配置文件的位置和格式。通常情况下，内核的配置文件位于 `/boot` 目录下的 `config` 文件，但具体位置和名称可能会有所不同。
2. **编译和安装内核**：完成内核配置后，需要将配置保存并编译成新的内核镜像。编译完成后，将新的内核安装到系统中，并更新引导加载程序（如 GRUB）的配置，以便系统能够启动新的内核。
3. **选择配置工具**：使用适合你的操作系统和喜好的配置工具。常用的配置工具包括 `make menuconfig`、`make xconfig`、`make gconfig` 等。这些工具提供了图形化或者命令行的界面，可以方便地进行内核配置

## rootfs

rootfs 是真正的根文件系统，包含了完整的文件系统结构和用户空间程序。它是操作系统的根目录，包含了所有的用户空间程序、配置文件、库文件以及其他系统文件。rootfs 负责提供用户空间的环境，使用户能够进行各种操作和运行各种应用程序。

## 文件更新

感觉普通的文件io

## 指令分类

### Genshin check

获取网上地址 检查当前系统的三个文件系统是否是最新版本 如果是输出当前已经是最新版本 如果不是 给出当前文件系统的版本号和最新版本的版本号 并给出更新指令 和更新内容等

### Genshin update

检查当前是否是最新版本 如果不是拉取网上资源并开始对本地文件系统进行覆写操作 并且实时给出进度

### Genshin getmsg

从网上拉取各个版本号和信息 分类给出稳定版 测试版

# Genshin back（版本号）

回滚版本号操作 从网上拉取固定版本的资源进行覆盖

## OTA升级

通过无线方式实现一个Linux系统的一些重要系统文件的版本升级,第一题是initrd, kernel, rootfs三类系统文件 考虑到设置每次的版本号 和 系统更新的权限 可以先从普通文件的在线升级和版本号管理开始

## 阶段一:普通文件的升级

可以根据linux系统的 软件包更新指令 从网上下载文件到指定目录

## 自定义一个linux指令

在Linux上创建自定义指令通常涉及以下步骤:

1. **选择指令名称:** 首先,你需要选择一个唯一的指令名称,确保它不会与系统中已有的指令重名。
2. **创建可执行文件:** 使用你喜欢的编程语言编写一个可执行文件,该文件将包含你的自定义指令的实际功能代码。你可以使用任何你熟悉的编程语言,如Shell脚本、Python、C、C++等。
3. **添加执行权限:** 确保你的可执行文件具有执行权限,以便在命令行中运行它。你可以使用 `chmod` 命令添加执行权限,例如:

复制代码

```
chmod +x your_command
```

4. **将可执行文件添加到PATH中:** 为了让系统能够识别并执行你的自定义指令,你需要将包含该可执行文件的目录添加到系统的PATH环境变量中。可以将该目录添加到 `~/.bashrc` 或 `~/.bash_profile` 文件中,例如:

复制代码

```
export PATH=$PATH:/path/to/your/directory
```

保存文件后,执行以下命令使更改生效:

复制代码

```
source ~/.bashrc
```

5. **测试指令:** 现在,你可以在命令行中使用你的自定义指令了。输入指令名称并按下回车键,系统应该能够执行你的指令并显示相应的输出。

请注意，如果你希望你的自定义指令在任何目录中都可用，需要将可执行文件放置在系统PATH中的某个目录中，例如 `/usr/local/bin`。

## 阶段二:拉取网上配置文件

通过linux下的c++编写程序 通过libcurl库函数进行拉取网上的资源更新文件系统

## 阶段三:对rootfs系统文件进行升级

首先准备好心的文件系统 与该发行版的文件系统的目录结构一致，  
然后挂载到新的rootfs文件系统 可以用mount指挂载到一个临时目录，因为要保证用户配置文件一致性 所以要把现在的rootfs的etc下的用户文件进行拷贝到新的文件系统，  
切换到新的rootfs 使用chroot命令切换到新的rootfs文件系统

## 阶段四：对内核kernel文件系统进行升级

内核官网 [kernel.org](http://kernel.org)

## 用户应用配置一致性

根据调研 用户应用配置 一般是一些固定的文件 目前感觉能通过对一个文件系统中 只需要对部分系统文件进行替换更新 涉及到用户配置的文件就不需要替换更新

## 用户配置目录

在rootfs文件系统里面 用户配置目录是/etc下面的一些文件 基本上都是用户配置文件 最好不要动

## 如果监测系统配置异常,如何回滚

通过在升级的过程中检测是否有报错情况 如果有结束升级 并且吧已经升级过的文件替换成原来的文件