# 账户管理后台设计说明书

**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** |
| 1.0 | 2020-12-09 | 唐杰 | A | ALL | 建立 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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

[账户管理后台设计说明书 1](#_Toc2017)

[一、 概述 3](#_Toc5031)

[二、 总体设计 3](#_Toc7860)

[三、 模块设计 3](#_Toc27294)

[3.1. 数据管理模块 3](#_Toc5923)

[3.2. 账户管理模块 4](#_Toc6780)

[3.2.1. 自动登录 4](#_Toc2215)

[3.2.2. 账户管理 5](#_Toc23726)

[3.3. 用户设置模块 5](#_Toc12145)

[3.3.1. PolicyKit验证规则 6](#_Toc2895)

[3.3.2. 用户类型 6](#_Toc31026)

[3.3.3. 密码模式 6](#_Toc28998)

[四、 接口设计 6](#_Toc12611)

[4.1. DBUS接口 6](#_Toc28751)

[4.1.1. 账户管理 6](#_Toc3862)

[4.1.2. 用户设置 8](#_Toc21549)

# 概述

本文档对账户管理后台程序的概要设计进行了阐述。账户管理后台是作为一个插件由控制中心后台(kiran-system-daemon)加载，目前主要应用于桌面环境，为上层的桌面应用提供必要的账户管理接口。

# 总体设计

账户管理后台分为三个部分：数据管理、账户管理和用户设置。

数据管理模块主要是对/etc/passwd、/etc/shadow和/etc/group三个系统配置文件的封装；账户管理的主要功能为创建/删除/查找用户。用户设置的主要功能为查找和设置用户信息，包括用户名、用户密码、用户主目录等。账户管理后台的整体流程如下：

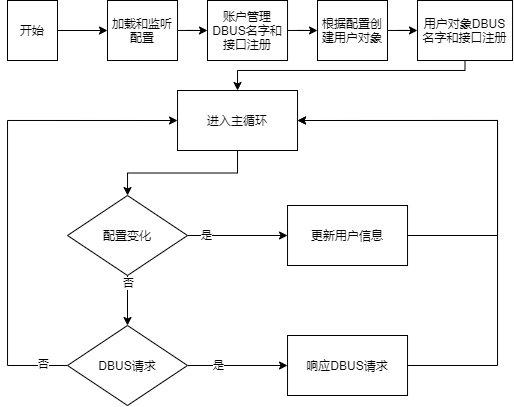


图 1整体流程

# 模块设计

## 数据管理模块

数据管理模块会在模块内部对每个配置维护一个数据结构，方便其它模块操作这些系统配置。同时，当配置发生变更时，数据管理模块会及时更新内部数据结构，并发送信号告知监听者文件有变化。

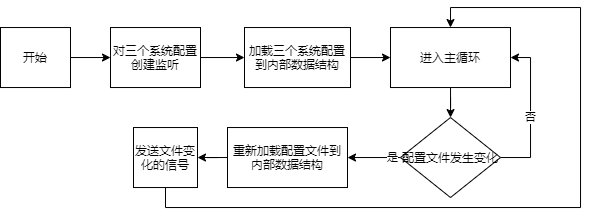


图 2 数据管理流程

## 账户管理模块

账户管理模块为整个账户管理后台的核心，用户信息的变更主要来自配置文件的变更和外部DBUS请求，具体流程如下：

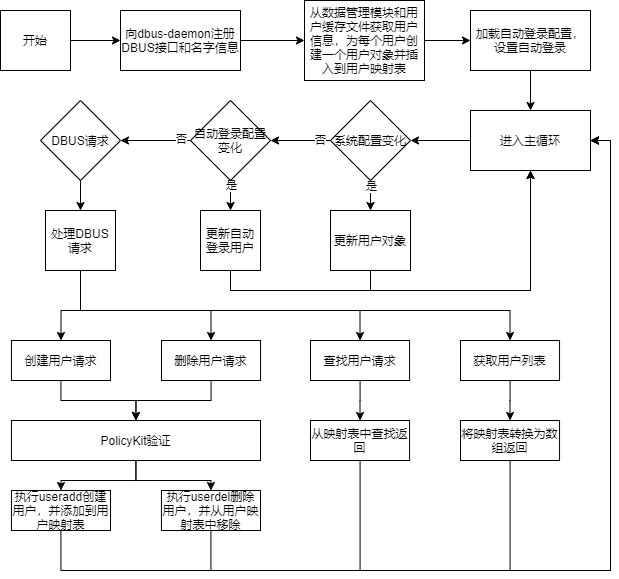


图 3 账户管理流程图

### 自动登录

自动登录配置为/etc/gdm/custom.conf，配置中的AutomaticLoginEnable和AutomaticLogin分别设定自动登录是否开启和自动登录账户名。

### 账户管理

账户管理默认只加载非系统用户，若满足以下条件之一则为系统用户：

1. 用户名为下列名字之一："bin", "root", "daemon", "adm", "lp", "sync", "shutdown", "halt", "mail", "news", "uucp", "nobody", "postgres", "pvm", "rpm", "nfsnobody", "pcap", "mysql", "ftp", "games", "man", "at", "gdm", "gnome-initial-setup"。
2. 登录shell名为空
3. 登录shell名为nologin或者false
4. uid小于1000

账户管理会为每个非系统用户建立一个缓存文件，缓存文件的目录为/var/lib/AccountsService/users中，缓存文件主要记录系统配置(/etc/passwd /etc/shadow /etc/group)中没有的信息，例如用户邮箱，账户头像等。

如果需要获取系统用户信息，需要显示调用查找用户的DBUS请求，该请求先从用户映射表中查找，如果查找不到且用户名合法则会创建一个用户对象并返回用户对象的DBUS路径。

## 用户设置模块

为了方便，阅读本文档可以不用太区分用户对象和用户设置的术语，从某种意义上说它们表达的意思相似。用户设置模块主要用于设置和获取用户信息。设置用户信息前需要通过PolicyKit模块的验证，获取用户信息不需要验证。具体流程如下：

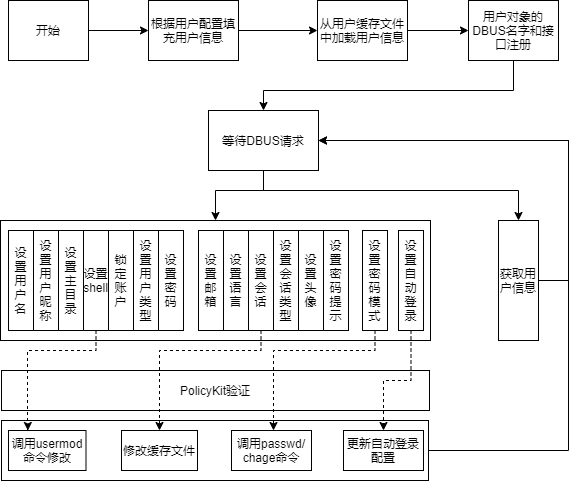


图 4账户设置流程图

### PolicyKit验证规则

如果修改的不是自己用户的信息，则需要管理员权限；如果修改的是自己的用户，下面几种情况需要管理员权限：

1. 修改用户名
2. 修改用户主目录
3. 修改用户登录Shell
4. 修改用户锁定
5. 修改用户自动登录

其它情况可直接设置。

### 用户类型

用户类型分为普通用户和管理员用户（不要与前面的系统用户和非系统用户混淆，两个概念不相关），满足下列条件为管理员用户，否则为普通用户：

1. 用户uid为0(即root用户)
2. 用户在用户组’wheel’中

### 密码模式

密码模式分为三种：正常、登录时需要设置密码和无密码登录。

当设置为登录时需要设置密码或者无密码登录时，会调用passwd -d命令清除当前密码，并删除密码提示；

当设置为登录时需要设置密码时，会继续调用chage -d 0命令；

当设置为正常时会对账户进行解锁。

# 接口设计

## DBUS接口

### 账户管理

#### 函数

##### 创建用户

ObjectPath CreateUser(String name, String realname, Int32 acccount\_type, Int64 uid);

函数说明：通过用户名，用户昵称，用户类型和用户唯一ID创建用户。

返回值：如果调用useradd命令失败，则返回错误，否则返回用户对象的DBus ObjectPath路径。

##### **删除**用户

DeleteUser(Uint64 uid, Boolean remove\_files);

函数说明：通过uid来查找用户并删除，如果remove\_files设置为真，则删除用户主目录。

返回值：如果调用userdel命令失败，则返回错误，否则返回成功。

##### **查找**用户

ObjectPath FindUserById(Uint64 uid);

函数说明：通过uid来查找用户，如果用户在映射表中则直接返回，否则如果用户存在(/etc/passwd中)，则创建一个新用户并返回。

返回值：查找成功则返回用户对象的DBus ObjectPath路径，否则返回失败。

ObjectPath FindUserByName(String name);

函数说明：通过用户名来查找账户，其它逻辑跟FindUserById函数相同。

返回值：查找成功则返回用户对象的DBus ObjectPath路径，否则返回失败。

##### **获取**用户**列表**

Array[ObjectPath] GetNonSystemUsers();

函数说明：获取所有非系统用户。

返回值：返回所有非系统用户的DBus ObjectPath路径。

#### **信号**

UserAdd(ObjectPath user);

信号说明：当有新用户创建时触发信号。

参数：用户的DBus ObjectPath路径

UserDeleted(ObjectPath user);

信号说明：当有用户被删除时触发信号。

参数：用户的DBus ObjectPath路径

### 用户设置

#### 设置用户信息

// 设置用户类型

SetAccountType(Int32 account\_type);

// 设置自动登录

SetAutomaticLogin(Boolean enabled);

// 设置邮箱

SetEmail(String Email);

// 设置主目录

SetHomeDirectory(String homedir);

// 设置图标

SetIconFile(String filename);

// 设置语言

SetLanguage(String language);

// 设置用户锁定

SetLocked(Boolean locked);

// 设置密码

SetPassword(String password, String hint);

// 设置密码提示

SetPasswordHint(String hint);

// 设置密码模式

SetPasswordMode(Int32 mode);

// 设置昵称

SetRealName(String name);

// 设置会话

SetSession(String session);

// 设置会话类型

SetSessionType(String session\_type);

// 设置登录shell

SetShell(String shell);

// 设置用户名

SetUserName(String name);

// 设置X会话

SetXSession(String x\_session);

这些设置命令会更新系统配置文件(/etc/passwd等)、用户缓存文件或者自动登录配置。如果更新失败则返回错误，否则返回成功，返回成功不带返回值。

#### **获取用户信息**

上面可设置的用户信息都可以从Properties中获取，除此以外，还可以获取用户UID，用户是否为系统用户和用户的过期策略。