Mobile Recharge System: Function Explanations

Generated on August 18, 2025

Document Overview

This document explains the purpose and behavior of each function in the C program `main.c`, which implements a mobile recharge card management system. The program supports admin and user roles, managing recharge cards (40, 60, or 100 minutes) stored in text files. Functions are grouped by category for clarity. The system uses ANSI color codes for console output and file-based storage for persistence.

1 Global Variables

- `unsuccessful_recharge`: Integer tracking failed recharge attempts (default 3). Used in `recharge()` to block users after three invalid card entries.

2 Helper Functions

2.1 is_exist(char number[], FILE *reference)

- **Purpose**: Checks if a string (e.g., card or phone number) exists in a file.
- **Input**: String to search for, open file pointer.
- **Output**: Integer (number of matches, typically 0 or 1).
- **Behavior**: Reads lines, trims newlines, compares using `strcmp`. Closes the file before returning.
- **Used In**: Card validation in `recharge()`, block checks in `User()`.

3 Main Entry

3.1 main()

- **Purpose**: Program entry point and main menu loop.
- **Input**: None (user inputs via `scanf`).
- **Output**: None (exits on choice 3).
- **Behavior**: Displays a menu with options: Admin, User, Exit. Admin access requires a password (`2403172`). Calls `Admin()` or `User()`. Uses ANSI colors for console output (e.g., MAGENTA for prompts).

4 Admin-Related Functions

4.1 Admin()

- **Purpose**: Admin menu loop for managing system operations.

- **Input**: None.
- **Output**: Returns 0 on exit.
- **Behavior**: Provides options: New card, Delete card, Unlock number, History, Statistics, Search, Exit. Validates phone numbers where required. Calls respective functions based on user choice.

4.2 generate_cards(int count, int minute)

- **Purpose**: Generates random recharge cards and stores them in unused card files.
- **Input**: Number of cards, minute value (40, 60, or 100).
- **Output**: None (appends to file).
- **Behavior**: Opens file (e.g., `unused_cards/40.txt`) based on minutes. Generates 20-digit random strings (digits 1–9 using `rand()`). Writes to file with newlines. Prints success message with count and minutes.

4.3 delete_card()

- **Purpose**: Deletes a specific card from an unused card file.
- **Input**: User-prompted minute type and card number.
- **Output**: None.
- **Behavior**: Prompts for minute (40/60/100) and 20-digit card number. Uses a temporary file (`unused_cards/temp.txt`) to copy non-matching lines. Renames temp file if card is found; removes temp file if not. Prints success or not-found message.

4.4 unblock_number(char number[])

- **Purpose**: Removes a phone number from the blocked list.
- **Input**: Phone number string (11 digits).
- **Output**: None.
- **Behavior**: Uses a temporary file (`blocked_numbers/temp.txt`) to copy non-matching lines from `blocked_numbers/numbers.txt`. Renames files regardless of result. Prints success or not-found message.

4.5 statistics()

- **Purpose**: Displays counts of used/unused cards and revenue in Taka.
- **Input**: None.
- **Output**: None (prints statistics).
- **Behavior**: Counts lines in used (`used_cards/*.txt`) and unused (`unused_cards/*.txt`) files for each minute type. Calculates revenue (40min = 50 TK, 60min = 70 TK, 100min = 120 TK). Outputs in text format.

4.6 history()

- **Purpose**: Displays transaction history.
- **Input**: None.
- **Output**: None (prints lines).
- **Behavior**: Reads and prints all lines from `admin/history.txt` (format: Card NO, Date, Time, Min, Mobile NO). Creates file if missing. Prints "No history found" if empty.

4.7 search()

- **Purpose**: Searches transaction history by phone number.
- **Input**: User-prompted phone number.
- **Output**: None (prints matching records).
- **Behavior**: Validates number (11 digits, starts with "01"). Parses `admin/history.txt` lines using `sscanf`. Prints matching transactions (omitting mobile number in output for brevity).

5 User-Related Functions

5.1 User()

- **Purpose**: User menu loop for account operations.
- **Input**: None.
- **Output**: Returns 0 on exit or invalid number.
- **Behavior**: Prompts for phone number (validates as 11 digits starting with "01"). Checks if number is blocked before each operation. Options: Check balance, Recharge, Exit. Calls respective functions.

5.2 check_balance()

- **Purpose**: Displays the user's balance.
- **Input**: None.
- **Output**: None (prints balance).
- **Behavior**: Reads an integer from `user/balance.txt` and prints it as the current balance in minutes.

5.3 recharge(char u_number[])

- **Purpose**: Processes account recharge using a card.
- **Input**: User's phone number.
- **Output**: None.

- **Behavior**: Prompts for package (40/60/100 minutes). Allows 3 attempts to enter a valid unused card. Validates card using `is_exist`. On success: moves card to used file, updates `user/balance.txt`, logs transaction in `admin/history.txt` with date/time. On failure (3 invalid attempts): appends number to `blocked_numbers/numbers` Uses temporary files for safe updates.

6 Appendix: Program Assumptions

- File paths (e.g., `unused_cards/`, `used_cards/`) must exist or be created implicitly by `fopen`.
- Balance is stored in a single file (`user/balance.txt`), not per-user, which may cause issues in a multi-user system.
- Transaction dates/times use local system time via `strftime`.
- ANSI color codes (`RED`, `GREEN`, etc.) are used for console output but may not work on all terminals.