



Trabalho Prático 1

Linux Recycle Bin Simulation – Students Quick Reference Guide¹

Quick Start (5 Minutes)

Step 1: Create Your Script

```
touch recycle_bin.sh
chmod +x recycle_bin.sh
```

Step 2: Add Shebang and Basic Structure

```
#!/bin/bash

RECYCLE_BIN_DIR="$HOME/.recycle_bin"
METADATA_FILE="$RECYCLE_BIN_DIR/metadata.db"

main() {
    echo "Hello, Recycle Bin!"
}

main "$@"
```

Step 3: Test It

```
./recycle_bin.sh
```

✓ If you see "Hello, Recycle Bin!" you're ready to go!

¹ The text of this project proposal had AI contributions to its completion.



Implementation Checklist

Week 1: Foundation

- ☐ Create script file and make executable
- ☐ Add shebang line (`#!/bin/bash`)
- ☐ Define global variables
- ☐ Implement `initialize_recyclebin()`
- ☐ Test: Directory structure created

Week 2: Core Features

- ☐ Implement `delete_file()`
- ☐ Generate unique IDs
- ☐ Store metadata
- ☐ Implement `list_recycled()`
- ☐ Test: Delete and list working

Week 3: Advanced Features

- ☐ Implement `restore_file()`
- ☐ Handle restoration conflicts
- ☐ Implement `search_recycled()`
- ☐ Implement `empty_recyclebin()`
- ☐ Test: All core features work

Week 4: Polish

- ☐ Add error handling everywhere
- ☐ Write README.md
- ☐ Write TECHNICAL_DOC.md
- ☐ Write TESTING.md
- ☐ Create test suite
- ☐ Take screenshots
- ☐ Final testing

Essential Shell Commands

File Information

```
# Get file size
stat -c %s file.txt
stat -f%z file.txt
```

```
# Linux
# macOS
```

```
# Get permissions
stat -c %a file.txt
```

```
# Returns: 644
```

```
# Get owner
```



```
stat -c %U:%G file.txt          # Returns: user:group

# Get absolute path
realpath file.txt               # Returns: /full/path/to/file.txt

# Get filename only
basename /path/to/file.txt     # Returns: file.txt

# Get directory only
dirname /path/to/file.txt      # Returns: /path/to
```

Date and Time

```
# Current timestamp
date +%s                       # Returns: 1696234567

# Formatted date
date "+%Y-%m-%d %H:%M:%S"     # Returns: 2024-10-02 14:30:22

# Date arithmetic
date -d "30 days ago" +%s     # 30 days ago timestamp
```

String Operations

```
# Substring
${var:0:10}                    # First 10 characters

# Length
${#var}                        # Length of string

# Replace
${var/old/new}                 # Replace first occurrence
${var//old/new}                # Replace all occurrences
```

File Tests

```
[ -e file ]      # File exists
[ -f file ]      # Is regular file
[ -d dir ]       # Is directory
[ -r file ]      # Is readable
[ -w file ]      # Is writable
[ -x file ]      # Is executable
```

CSV Operations

```
# Read CSV line by line
while IFS=',' read -r col1 col2 col3; do
    echo "Column 1: $col1"
done < file.csv

# Skip header line
tail -n +2 file.csv

# Append to CSV
echo "$id,$name,$path" >> file.csv
```



Code Snippets You'll Need

1. Generate Unique ID

```
generate_unique_id() {
    local timestamp=$(date +%s%N)
    local random=$(cat /dev/urandom | tr -dc 'a-z0-9' | fold -w 6 | head -
n 1)
    echo "${timestamp}_${random}"
}

# Usage
id=$(generate_unique_id)
echo "$id" # Output: 1696234567123456_abc123
```

2. Initialize Directory Structure

```
initialize_recyclebin() {
    if [ ! -d "$RECYCLE_BIN_DIR" ]; then
        mkdir -p "$FILES_DIR"
        echo "ID,ORIGINAL_NAME,ORIGINAL_PATH,DELE-
TION_DATE,FILE_SIZE,FILE_TYPE,PERMISSIONS,OWNER" > "$METADATA_FILE"
        echo "Recycle bin initialized"
    fi
}
```

3. Check If File Exists

```
if [ ! -e "$file_path" ]; then
    echo "Error: File does not exist"
    return 1
fi
```

4. Get File Metadata

```
# Get all metadata at once
filename=$(basename "$file_path")
abs_path=$(realpath "$file_path")
file_size=$(stat -c %s "$file_path")
permissions=$(stat -c %a "$file_path")
owner=$(stat -c %U:%G "$file_path")
deletion_date=$(date "+%Y-%m-%d %H:%M:%S")

# Determine file type
file_type="file"
[ -d "$file_path" ] && file_type="directory"
```

5. Move File to Recycle Bin

```
unique_id=$(generate_
```
