

ASSIGNMENT 1: SHELL SCRIPTING

CS3424 - Systems Programming

Rocky Slavin - UTSA

For this assignment, you will use **bash** create a simple inventory system. The system will store basic information about items and allow the user to create, read, update, and delete them.

This assignment requires only the utilities used so far in the lecture notes. **Do not** use sed, awk, find, grep, or any programming languages not yet covered in the lecture notes.

Storing Item Information

Item information will be stored in text files.

1. Files will be stored inside a directory called **data** within the same directory as your script.
2. Each file will be named based on the item number, an integer with exactly **four** digits, followed by the extension **.item**.
3. An item file consists of *exactly* three lines:
 - simple_name (string **with no whitespace**) item_name (string)
 - current_quantity (integer) max_quantity (integer)
 - description (string)
4. Example file named 3923.item

```
b_water Bottled Water
35 99
The finest spring water you can purchase!
```

Script Execution

When the script is run, the following should occur.

1. Upon running your script, the user should be presented with the following menu:
Enter one of the following actions or press CTRL-D to exit.
C - create a new item
R - read an existing item
U - update an existing item
D - delete an existing item
2. The user then enters a one-character action (upper or lowercase), leading to one of the following.
 - C: an item file is created
 - (a) From the terminal, read the following one at a time
 - i. Item number (four digit integer)
 - ii. Simple name (string with **no whitespace**)

- iii. Item name (string)
 - iv. Current quantity (integer)
 - v. Maximum quantity (integer)
 - vi. Description (string)
- (b) Using the values entered by the user, create a new file in the **data** directory based on the instructions above.
- (c) Update **data/queries.log** by adding the following line:


```
CREATED: simple_name - date
```

 where *simple_name* is the item's short name and *date* is the output from the **date** command.
- (d) If the item number already exists, print the following error and continue with the script.


```
ERROR: item already exists
```
- R: read an existing item's information
 - (a) Prompt the user for an item number:


```
Enter an item number:
```
 - (b) Search for the specified item using the item number.
 - (c) Print the item information in the following format:


```
Item Name:  item_name
Simple Name: simple_name
Item Number: item_number
Qty:  current_quantity/max_quantity
Description: description
```
 - (d) If the item is not found, print the following error and continue with the script.


```
ERROR: item not found
```
- U: update an existing item
 - (a) Prompt the user for the following one at a time
 - i. Item number (four digit integer)
 - ii. Simple name (string **with no spaces**)
 - iii. Item name (string)
 - iv. Current quantity (integer)
 - v. Maximum quantity (integer)
 - vi. Description
 - (b) Search for the specified item using the item number.

- (c) Update each of the corresponding fields based on the user input. **If the user input is blank for a particular field (except item number), keep the original value from the file.**
- (d) Update `data/queries.log` by adding the following line:
`UPDATED: simple_name - date`
 where `simple_name` is the item's short name and `date` is the output from the `date` command.
- (e) If the item is not found, print the following error and continue with the script.
`ERROR: item not found`
- D: delete an existing item
 - (a) Prompt the user for an item number:
`Enter an item number:`
 - (b) Delete the specified item's file
 - (c) Update `data/queries.log` by adding the following line:
`DELETED: simple_name - date`
 where `simple_name` is the item's short name and `date` is the output from the `date` command.
 - (d) Print the following message with the item's simple name:
`simple_name was successfully deleted.`
 - (e) If the item is not found, print the following error and continue with the script.
`ERROR: item not found`
- If an invalid character is entered, print the following error and continue with the script.
`ERROR: invalid option`
- 3. After an action is completed, display the menu again. This should go on indefinitely until CTRL-D or the end of a file is reached.

Assignment Data

An initial data set can be found in `/usr/local/courses/rslavin/cs3424/assignment_input/assign1`. Copy this to your own assignment's directory.

Script Files

Your program should consist of five bash files:

- `assign1.bash` - the main file which is initially invoked
- `create.bash` - logic for the create option
- `read.bash` - logic for the read option
- `update.bash` - logic for the update option

- `delete.bash` - logic for the delete option

Verifying Your Program

Your program must work with the input provided in `a1Input.txt`. To test it:

1. Verify that your assignment directory has a `data` directory with the initial data set.
2. Execute your script and **redirect** `a1Input.txt` into it. You should not be copying or typing the contents of `a1Input.txt` into your terminal. Redirection must work.
3. Verify that the output and files are as expected.

Submission

Turn your assignment in via Blackboard. Your zip file, named `abc123.zip` with your personal abc123 id, should contain only your five bash files.