

AP-Assignment 4

October 2022

kxs806

qmc887

Design of the code

Answer to questions:

how you deal with unreliable analytics functions.

We use a try-catch to handle those function which may have an error while running and if it catch sth it will escape it and keep running the following Functions. In that way it won't crash the whole system, just ignore a step of the procedure.

You should also describe what state your server (and other relevant processes) maintain.

As there're 4 things we need to store and each of them is contains different information, we designed a state like {Shortname,Emoji,[Alias],[Fun]}. The Alias list contains all the names related to this Emoji, including it's own Shortname. The Function list contains some information about the functions stored in the Shortname, each of them is a tuple like {Label,Func,State}.

The usage is simple, if create a new shortname it will be {Name,Emoji,[Name],[]}. When you create an alias it will first add the alias name into the [Name] list, and then add a new item like{AliasName,none,Name,[]}. Here we use none to show it's an alias item, and in the list position there's it's real name, this will help to delete it in the future and also helpful when searching for it's real name.

The Function list contains tuples of function informations as shown above, this will also be delivered if an alias is create on a shortname so you will be able to use the function under the alias name. The State is firstly the Init value being inputed, and will be changed when running the function.

How do the processes interact, and how many (kinds) are there in the system?

When you run start function, it will spawn a loop process (which is the main server process) and feed it with the initial parameters list. Then the process is active and ready to receive a lot of words such as newshortcode,delete,alias.... And we can get the Pid of this process as E.

Then,with those main operator functions we can send those words and variables to the loop process.Whenever the server receive a tuple it needs it will run some function the operate needs and re-run the loop server process with new(if the operate changed it) parameter list.

Followings are some implementations we think may need to say about:

(Following all the "The List" means the list of the loop function unless otherwise noted)

getRealName function: To search for a given name's real shortcode name.It will search for the List of the loop for the given Shortname. If there's on tuple's Shortname matchs then we check the Emoji. If it's none then it means it's an alias, then we just return the name stored before at the Alias position.If it's sth else we just return it's real name.

getshortCodes function:It will get all the Shortcodes and Alias currently registered by searching The list and add all the Alias part together.

New_shortcode part: After receiving the Short we first search the shortcode list if there's already a shortcode. If not, we add a tuple into The List and re-run the loop function.

Alias part: After receiving the names we first check shortcode duplicate. Then we check alias duplicate. After these we first get the shortcode's real name in case of the user register an alias on an alias, then we add an alias on the real name tuple and also add an alias tuple into The List.

Delete part: After receiving the name we first get it's real name, and then use a helper function getAliasList to get a list of the real name's all alias name + it's real name. Then we just use deletehelper to delete every tuple in The List which has the name inside the alias list.

Lookup part: First we get it's real name, and then we find the emoji with the real name and return it via signal. Then we need to run the function. To run the function we need it's real name to find the function list, then we also need the shortname to feed in the function. So we have a runFun function to search for The List for the function list, then use another function runhelper to run the function and change it's state. runhelper runs each function in the list and change the state with the run result using try-catch. If there's a function failed to run it will just skip it and continue run the next.

set,get,delete analytics part: Set analytics is just like set alias, after two check of duplicate it will add the function tuple into the real name's function list. get analytics needs to get all the status of the functions, so we need to use a helper to recursively check each tuple in the function list, get their {label,State} and put them together. Remove analytics first check the given realname's function list, and then check for the label's tuple, and remove it from the list.

And btw. During coding we found a bug (we don't really know if it's a bug) that is, in this function and many others

```
212 % remove all those with Shortcode in alias list
213 deletehelper(Aliases,List)->
214     case List of
215         []->[];
216         [{Short,Emoji,Alias,Fun}|Tail]->
217             case lists:member(Short,Aliases) of
218                 true->deletehelper(Aliases,Tail);
219                 false->[{Short,Emoji,Alias,Fun}|deletehelper(Aliases,Tail)];
220             end
221     end.
222
```

If we use {S,E,A,F} to match it will crash some times like the photo below:

```
139 % Remove all those with shortcode in alias list
140 deletehelper(Aliases,List)->
141     io:format("Alias: ~p~nList: ~p~n",[Aliases,List]),
142     case List of
143     []->[];
144     [{S,E,A,F}|Rest]->
145         case lists:member(S,Aliases) of
146         true->deletehelper(Aliases,Rest);
147         false->[{S,E,A,F}|deletehelper(Aliases,Rest)]
148         end
149     end.
150
151 % find the emoji of the shortcode
```

PROBLEMS 58 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER: VARIABLES

```
Alias: ["a","aa"]
List: [{"a",none,"aa",[]}]
41> =ERROR REPORT==== 7-Oct-2022::12:37:19.141290 ===
Error in process <0.138.0> with exit value:
{{case_clause, [{"a",none,"aa",[]}]},
 {{emoji,deletehelper,2, [{file,"emoji.erl"},{line,142}]},
 {{emoji,loop,1, [{file,"emoji.erl"},{line,118}]}}}

41> emoji:deletehelper(["a","aa"], [{"aa",123,["a","aa"],[]}, {"a",none,"aa",[]}] ).
Alias: ["a","aa"]
List: [{"aa",123,["a","aa"],[]}, {"a",none,"aa",[]}]
Alias: ["a","aa"]
List: [{"a",none,"aa",[]}]
** exception error: no case clause matching [{"a",none,"aa",[]}]
    in function emoji:deletehelper/2 (emoji.erl, line 142)
42> □
```

It shows also in other functions, so we have to change all the variables into longer word and it will be solved.

Assessment

Completeness

We completed all of the tasks in the assignment.

Correctness

We did some tests while developing in terminal but when we try to use the test suit it always return fail, I think maybe it's because we used it wrongly. But in case of onlineTA we passed all the tests on onlineTA. And when developing we tested the code and it performed well, so we assume the correctness is not so bad.....

Efficiency

We used pattern match a lot and we think it may not only make the code much more easy to read but also fasten the speed of the code. However, we used a lot of recursive to find things in all kinds of lists and we believe there's a better way to do it.

Robustness

In loop function we have a separate function to handle those undefined signal, and also in each signal process we have error messages to send back instead of just crash the system.

Maintainability

We made the code line as short as possible to make it readable. Also we add some comments on the functions to make it easier to understand. We also split some very commonly used module into functions so that the code become easier to read and change.