CSN08116 :: Coursework Documentation

Produced by: Stewart Anderson (40345422), Robert Galloway (40397559) & Connor Grattan (40416106)

Method Name	Method Description
display_title	Expected to take in no arguments and displayed the opening title to the user – only used within the initial boot of the program.
display_ui	Another method which expected to take in no arguments. This is displayed after most operations where the user interacts with the program.
prompt_user	Also taking in no arguments, this function will ask the user for a numeric value.
check_numeric	Taking in 1 argument - the user's input - which is checked if the value entered is numeric. If the value entered is not numeric, the user will be met with an error message and will be re-prompted until a number is entered.
error	This function is expected to take in a string which is parsed as a single argument and displayed to the user as an error message if there is an unexpected input.
list_running_jobs	Taking in no arguments, the function displays all running cron jobs to the user which run on their profile.
insert_cron_job	This function takes in no arguments but is called when the user enters '2'. Takes in multiple values to write to a cron file - which is created if it doesn't exist.
remove_cron_job	Similar to the insert function above, this function lists all current running tasks to the user and removes whichever item the user requests.
edit_cron_jobs	This is an advanced function that I felt like adding in. This function just allows the user to add a cron job if they know the syntax or are curious and wish to know/ learn the syntax of crontab.
remove_all_jobs	As the name of the function suggests, this function removes all jobs from the machine for the current user.
quit	Again, this function does exactly as this name suggests and automatically quits the program when called.
is_invalid	Taking in 2 parameters, this function will check if the value you're trying to enter is valid for crontab ('*', 0-59 etc.).
input_error	This function makes use of the 2 parameters passed into is_invalid is called to return an error to the user reminding them of the valid inputs for their current entry stage
special_string	This function allows the user to select a special function from a list of predefined variables from a specified list for defining timescales or to just use our insert function.

Task Distribution

Stewart
I began the initial layout for the coursework and started off by designing the title and main UI prompt through ASCII. This is where I proceeded to implement colours and create my table-like design which was inspired through my use of MySQL. Post-design, I began to work on the 2 basic functions of listing and removing all cron jobs as well as the quit & error functions.

After a week or so had passed, another member of the team started to work on the insert, removing and edit functions of the coursework, we got a bit stuck until I had implemented file I/O (which originally was Robert's idea, but scrapped it). This re-enabled us to continually test and edit the script as we saw fit.

Prior to hand-in, I have been working on the listing function to make it completely easy and simple for the beginner user to read and I have also been adding in small details such as displaying a message to the user to inform them all of their cron jobs have been removed.

Robert
I began planning each method. I firstly created the insert function. The idea was to ask the user the periodicity of the task they want to set up: asking the user for the minutes, hours, etc. Each time the user enters an input it would concatenate onto a string, creating the whole task the user wants. This would get appended to the crontab -l list. As I encountered problems doing this me and Stewart worked around it by making a temporary text file which would hold the list of tasks, appending the new task to the text file then pushing the text file to crontab -l.

The second method I created was the delete function. The idea was to ask the user which line number containing the task they wanted to delete. The list of tasks would get pushed to a text file, and using sed, deleting the line the user enters. The text file would then get pushed to the crontab -l list: updating it.

The final method I created was the edit function. It used both the delete and insert functions functionality. It first asks for the line, containing the task, the user would like to edit. The application would delete this task and then call on the insert function to get the users input and update the crontab -l list.

I then helped validating some inputs, making sure functions worked and tested for errors.

Connor
I started with adding and making use of an input validation function to the insert task function. Firstly, I added basic validation to ensure that the user was using numbers instead of strings to specify the timings of new functions. Once the correct type of input was being validated, I added checks to ensure that the input was within a specified range. An exception was made if the user entered an asterisk on its own as that is an acceptable string for crontab.

I then modified the delete and list functions so that they were more readable. With the delete function I added labels so that the user could associate each of the functions with a number, and a clear description of what the user is supposed to enter to delete a specific task. With the list function I modified how each task was stored before output so that each section of the task definition could have a label attached to tell the user what it represents.

Finally, I went back to the insert function to add a method for inputting crontab special strings. To save the user time, they are asked before entering the numbers for a new task if they want to use a special string. If the user does enter a special string, the manual number input section is skipped, and the user is simply asked what task to attach the special string to. If the user enters nothing the special string prompt is skipped and numbers can be added as normal, if the user enters something that isn't a special string a prompt appears asking for them to reenter a valid special string.