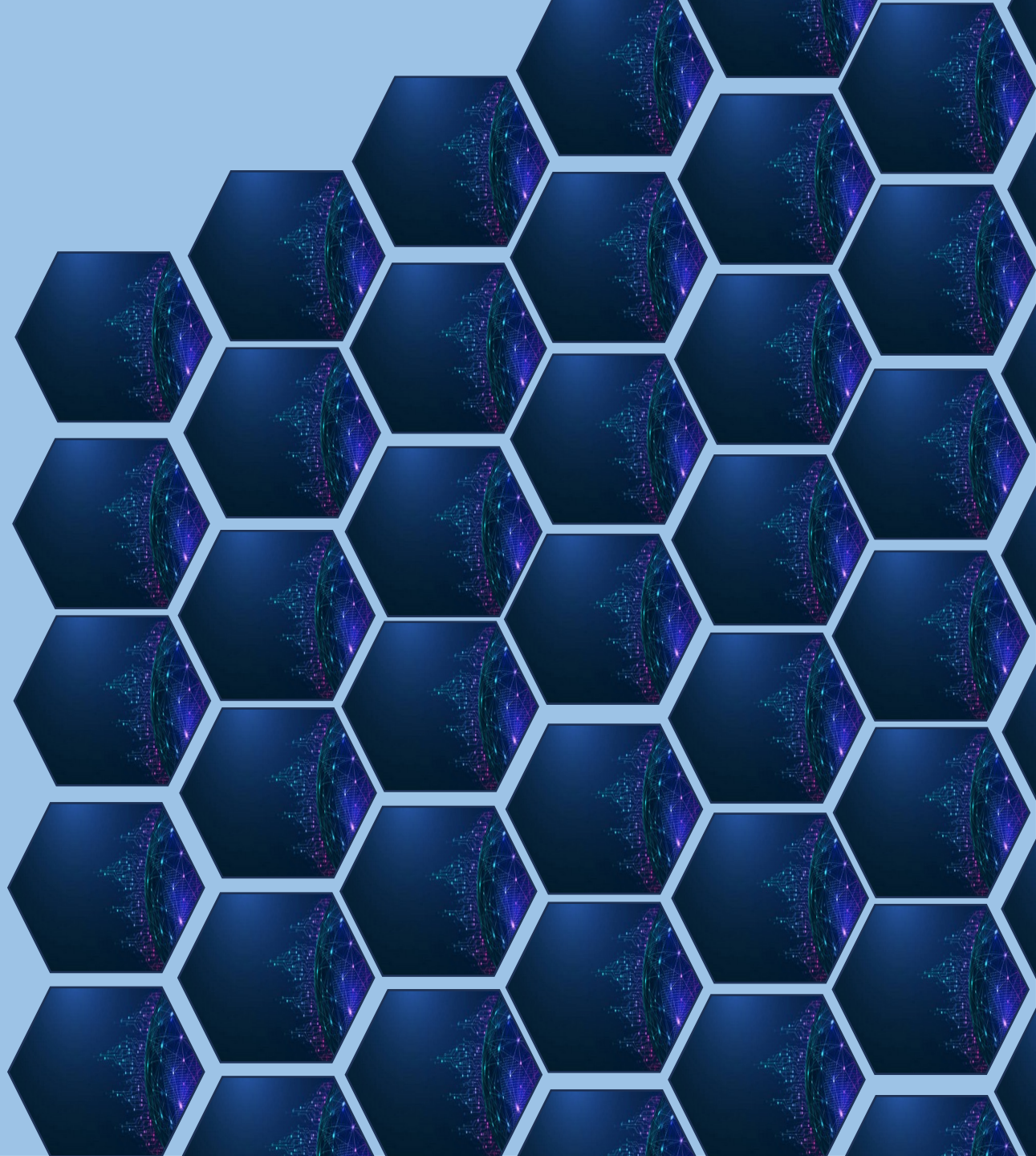
ROBOTICS CORNER

Robotics Corner



Robotics Corner

Bash Mini-Project

01 Project Overview

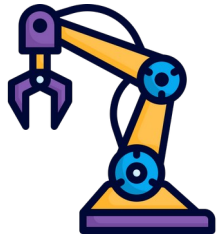
02 Add a Task

03 Deleting a task

04 View Tasks

05 Mark a Task

ROBOTICS CORNER



Task Manager application with bash

Make a shell script that manages tasks with a text file by doing the following:

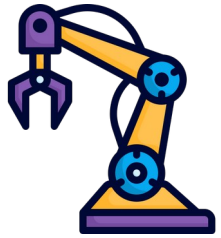
1) Add a task description and status columns to the text file.

2) Add actual tasks in that text file.

3) Delete a task.

4) Mark a task as completed.

5)View tasks from the text file.

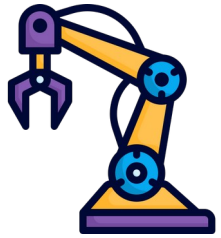


A sample of the text file and terminal

tasks.txt
~/Desktop

```
1 Description |  Status
2 Do Session_1 assignment | Not_Completed
3 Do Science lap | Not_Completed
```

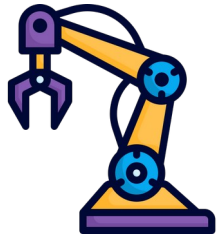
```
1      Description                    Status
2      Do Session_1 assignment        Not_Completed
3      Do Science lap                 Not_Completed
```



Adding a task

```
filename="tasks.txt"

add_task(){
        num_lines=$(wc -l <"$filename")
        if [ "$num_lines" = 0 ]; then
                echo "Description |  Status" >> "$filename"
        fi
        echo "Please Enter the Description of the task, 30 characters max:"
        read desc
        length=${#desc}
        if [ "$length" -le 30 ]; then
                echo "$desc | Not_Completed" >> "$filename"
        else
                echo "Long Description!!"
        fi
}
```

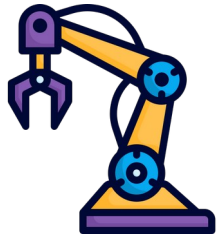
ROBOTICS CORNER



sed command

1) The sed command, short for "stream editor," is a powerful tool in Unix-like operating systems for performing basic text transformations and editing on files or input streams.

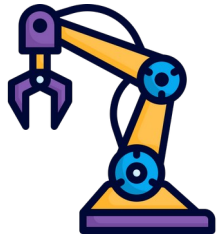
2) It operates by reading input line-by-line, applying specified editing commands, and outputting the result.



Deleting a task

```
sed -i "${num}d" "$filename"
```

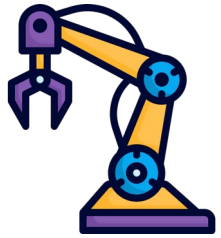
- (-i): Save changes back to the original file.
- ${num}: is a variable that contains the line number to delete.
- The d command tells sed to delete that specific line.



Deleting a task

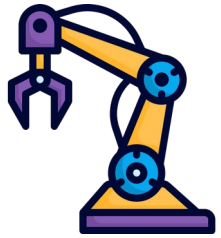
```
delete_task(){
        num_lines=$(wc -l <"$filename")
        if [ "$num_lines" = 0 ]; then
                echo "File is Empty!!, Please add a task first"
                return
        fi
        view_task
        echo "Please Enter the number of the task you want to delete:(all -> A)"
        read num
        if [ "$num" = "A" ]; then
                truncate -s 0 "$filename"
        else
                if [ "$num" -le "$num_lines" ] && [ "$num" -gt 1 ]; then
                        sed -i "${num}d" "$filename"
                else
                        echo "Wrong line number!!"
                fi
        fi
}
```

ROBOTICS CORNER



Awk command

- awk is a powerful text processing and pattern scanning tool in Unix-like systems
- It can extract specific fields or columns from structured data, like CSV files.
- It is useful for generating formatted reports, such as summarizing data or creating formatted output.



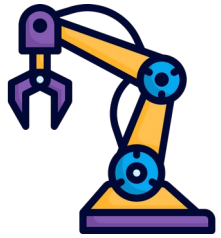
awk -F'|' '{printf " %-5s %-30s %-10s\n", NR, $1, $2}' "$filename"

1) -F'|': This means awk will split each line of the input file into fields based on the | delimiter.

2) printf: Used for formatted output. It allows precise control over how the data is printed.

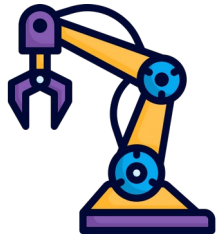
3) " %-5s %-30s %-10s\n": This is the format string that specifies how each field should be formatted.

4)"filename": This is the input file to be processed by awk.



View Task

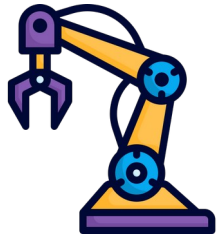
```
view_task(){
        num_lines=$(wc -l <"$filename")
        if [ "$num_lines" = 0 ]; then
                echo "File is Empty!!, Please add a task first"
                return
        fi
        awk -F'|' '{printf " %-5s %-30s %-10s\n", NR, $1, $2}' "$filename"
}
```



awk -F'|' -v r=$num 'BEGIN {OFS = FS} NR==r {$2=" Completed"}1' "$filename" > temp && mv temp "$filename"

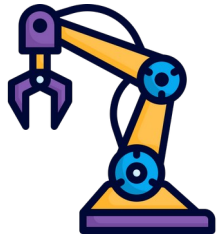
1) -F'|': This means awk will split each line of the input file into fields based on the | delimiter.
2) -v r=$num: Passes the value of the shell variable num to awk as a variable named r. This is used within the awk script.
3) BEGIN {OFS = FS}: In the BEGIN block, sets the output field separator (OFS) to be the same as the input field separator (FS). This ensures that fields in the output will be separated by |, matching the input format.
4) NR==r {$2=" Completed"}: For the record (line) number that matches r, sets the second field ($2) to " Completed". NR is the current record number, and r is the line number passed from the shell.
5) 1: A shorthand in awk for {print $0}, meaning print the entire line. In this context, it ensures that all lines are printed, with the modification applied to the specified line.
6) "$filename": This is the input file.
7) > temp: Redirects the output of the awk command to a temporary file named temp.
8) && mv temp "$filename": mv temp "$filename": Moves the temporary file temp to the original file name ($filename).

ROBOTICS CORNER



Mark a Task

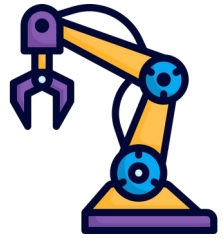
```
mark_task(){
        num_lines=$(wc -l <"$filename")
        if [ "$num_lines" = 0 ]; then
                echo "File is Empty!!, Please add a task first"
                return
        fi
        view_task
        echo "Please Enter the number of the task you want to Mark Completed:"
        read num
        if [ "$num" -le "$num_lines" ] && [ "$num" -gt 1 ]; then
                awk -F'|' -v r=$num 'BEGIN {OFS = FS} NR==r {$2=" Completed"}1' "$filename" > temp && mv temp "$filename"
        else
                echo "Wrong line number!!"
        fi

}
```



Main loop

```
tries=3
touch tasks.txt

while [ $tries -gt 0 ]
do
        echo "Please Enter your command (Add Task -> A, View tasks -> V, Mark a task -> M, Delete a task(s) -> D, Exit -> E)"
        read cmd

        if [ "$cmd" != "A" ] && [ "$cmd" != "V" ] && [ "$cmd" != "M" ] && [ "$cmd" != "D" ] && [ "$cmd" != "E" ]; then
                tries=$((tries - 1))
                echo "Wrong command, number of tries left" $tries
                echo " "
        else
                tries=3
        fi

        if [ "$cmd" = "E" ]; then
                exit 1
        elif [ "$cmd" = "A" ]; then
                add_task
        elif [ "$cmd" = "V" ]; then
                view_task
        elif [ "$cmd" = "M" ]; then
                mark_task
        elif [ "$cmd" = "D" ]; then
                delete_task
        fi


done
```

ROBOTICS CORNER



Thank You

Do you have any questions?

ROBOTICS CORNER