GROUP 72

Members:

Prakriti Jain (2022362)

Varsha Manoj (2021297)

Contribution:

Both of us worked on the assignment together

Implementation:

- In the Simple Shell file, the main function initializes the shell and sets up a signal handler for interrupt signals (Ctrl+C).
- The shell_loop function implements the main loop of the shell, where it continuously prompts the user for input and processes commands until the user exits.
- A shared memory object is created.
- The launch function reads user input, tokenizes it, and processes the command or pipeline accordingly.
- The change_directory changes the current working directory to the specified path.
- The create_process_and_run function forks a child process and executes the specified command using execvp
- The execute_pipeline function handles the execution of a sequence of piped commands. It is responsible for setting up pipes for inter-process communication and coordinating the execution of each command in the pipeline.
- The submit function creates a child process, write the process information in the shared memory and calls the start_scheduler() function.
- The start_scheduler() function sets the child process as a new session leader to run as a daemon and execute the Scheduler file.
- A structure named Process is created to represent a process. It contains process PID, priority, start time and end time. A priority queue structure is also created.
- The ready processes are moved to the running state, the process is started on the first availabe CPU, process is signaled to start execution, sleeps for the time quantum and check and reschedule processes.
- The addHistory, showHistory, and showPID functions manage and display command history.
- The my_handler function handles the SIGINT signal (Ctrl+C) by displaying the command history before exiting.
- All the basic commands are working except the high level modern shell commands which includes backslash, & , etc

Link to Github Repository:

https://github.com/varsha21297/OS_Assignment