CSE 344 HW2 REPORT

Bahadir Etka Kilinc

1901042701

The program is a simple shell that allows the user to execute multiple commands in a pipeline, separated by the "|" character. The shell supports a maximum of 20 commands in a pipeline, and it logs the commands and their respective process IDs (PIDs) to a file with a timestamped filename.

The log_commands function takes an array of command strings, an array of PIDs, and the number of commands as input, and logs them to a file. The function first opens the file with the specified filename in write mode. Then, it iterates over the commands and PIDs arrays and writes each entry to the file using the fprintf function. Finally, the function closes the file.

The timestamp function returns the current time in the format "YYYY-MM-DD_HH-MM-SS" as a dynamically allocated string. The function uses the localtime and strftime functions from the standard C library to format the current time and returns a duplicate of the formatted string using the strdup function.

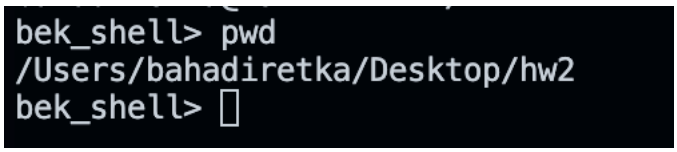
The execute_command function takes a command string, an input file descriptor, and an output file descriptor as input, and executes the command using the execl function. If the input or output file descriptors are not the standard input or output file descriptors, respectively, the function redirects the standard input or output to the specified file descriptors using the dup2 function before closing them. If the execl function fails, the function prints an error message using the perror function and exits with an error code.

The run_commands function takes a pipeline string as input and executes the pipeline by forking a child process for each command in the pipeline. The function first parses the pipeline string into an array of command strings using the strtok function. Then, it iterates over the command strings and forks a child process for each command. For each child process, the function calls the execute_command function with the command string, the input file descriptor of the previous command's output (if any), and the output file descriptor of the next command's input (if any). The function also maintains an array of PIDs for each child process and closes any file descriptors that are not the standard input or output file descriptors. Finally, the function waits for all child processes to terminate using the wait function and logs the commands and PIDs to a file using the log_commands function with a timestamped filename.

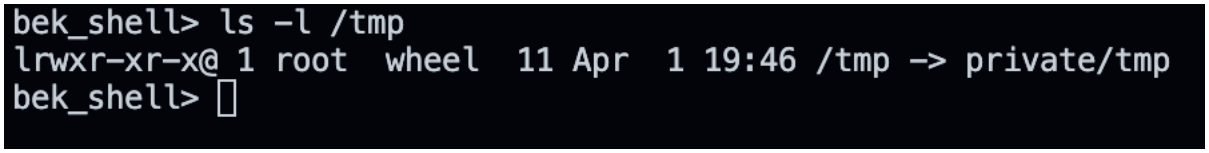
The main function takes no command line arguments and reads input from the user using the fgets function. It repeatedly prompts the user for input with the ">" prompt until the user enters the ":q" command to exit the shell. Otherwise, it calls the run_commands function with the user's input string.

Test Cases

Single command with no arguments



Single command with arguments



Two commands with pipe, with arguments

```
bek_shell> ps aux | grep root
root               881   0.8  0.4 410767712  31088   ??  Ss   Wed09AM  10:12.85 /Syst
/A/Support/mds_stores
root               518   0.7  0.2 408527840  18576   ??  Ss   Wed09AM   7:50.85 /Syst
mds
root               569   0.6  0.1 408277440  10224   ??  Ss   Wed09AM   3:15.43 /usr/
root                 1   0.2  0.2 409220000  15824   ??  Ss   Wed09AM  16:34.71 /sbin
root               514   0.1  0.0 408069232   2128   ??  Ss   Wed09AM   0:04.45 /usr/
root               531   0.1  0.1 408270448  10320   ??  Ss   Wed09AM   2:27.69 /Syst
root               491   0.1  0.2 408343968  16384   ??  Ss   Wed09AM   6:19.64 /usr/
root             35597   0.0  0.1 408227072   4304   ??  Ss    6:16AM   0:00.10 /usr/
root             35560   0.0  0.0 408197744   2960   ??  Ss    6:16AM   0:00.06 /usr/
root             35550   0.0  0.0 408226912   3328   ??  S     6:16AM   0:00.06 /usr/
```

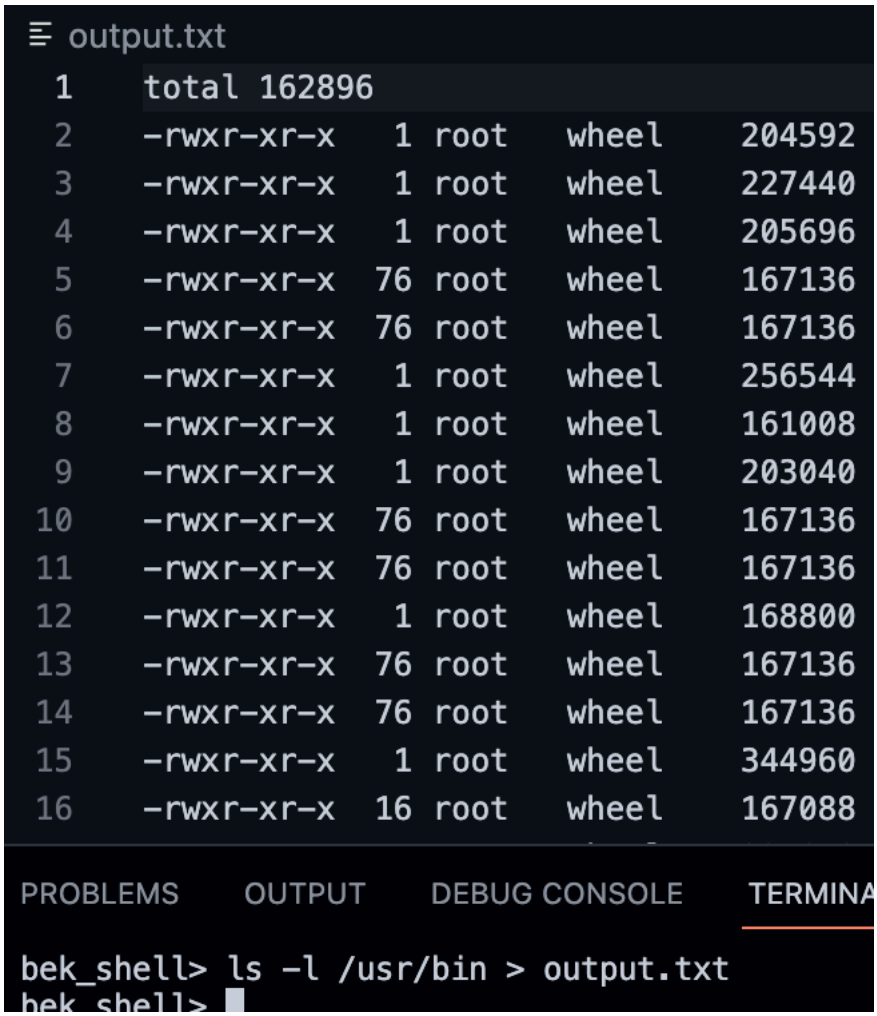
Command with input redirection, with arguments

```
1   PID: 37652, Command: grep 'hello' < input.txt
2

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

bek_shell> grep 'hello' < input.txt
bek_shell>
```

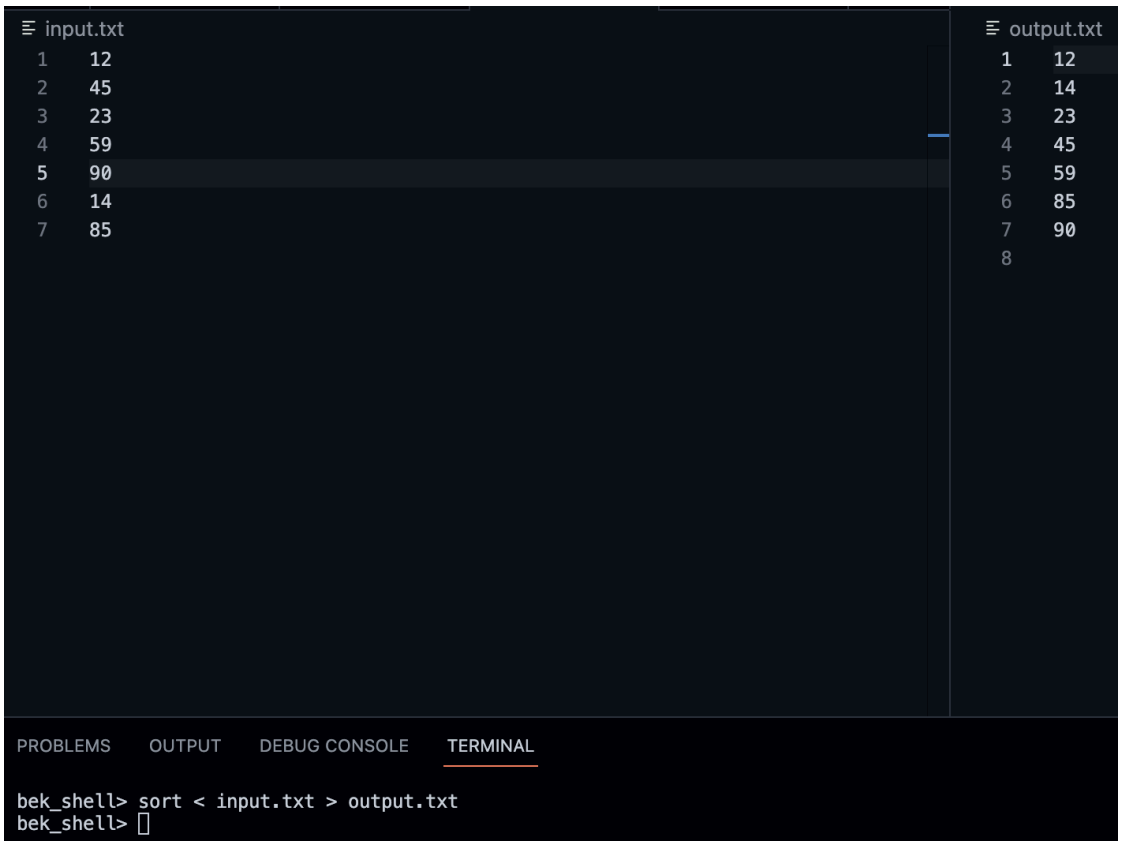
Command with input redirection

```
1   PID: 37494, Command: cat < input.txt
2

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

bek_shell> cat < input.txt
```

Command with output redirection, with arguments



Command with both input and output redirection, with arguments



Multiple commands with pipes and input/output redirection, with arguments

Extras