Simple Shell

Implement a Unix shell program that allows to run other programs and supports the command exit, commands with no arguments, commands with arguments, commands with or without arguments executed in the background using &.

Code organization

Global variables

- LogFile to write terminated child processes.
- A set of built-in commands (cd and exit).

Command Class

Public methods

- Constructor: takes command line as arguments and calls the function parse;
- execute(): decides whether the command is a built-in command or not and calls executeBuiltin() or executeOther() accordingly.

Private methods

- parse(): splits line command based on spaces and checks if the command is a background or a foreground command.
- executeBuiltin(): executes built-in commands
- executeOther(): executes non built-in commands by creating a new child process using the method fork() and then checks for pid if it is less than zero then an error occurred and a the child is not created, if pid is equal to zero then it is the child process and we execute the command using execvp method, if pid is bigger than zero ten it is the parent process and here we check if it is a foreground process we wait for the child process to terminate using wait method, write in the log file and check if the child process terminated normally.

Global methods

- SIGCHLD_handler(int sig): the signal handler responsible for the SIGCHLD signal, it waits for the terminated child background process to be freed, writes in the log file and checks if the child process terminated normally.
- printShellLine(): gets current directory path and prints it.
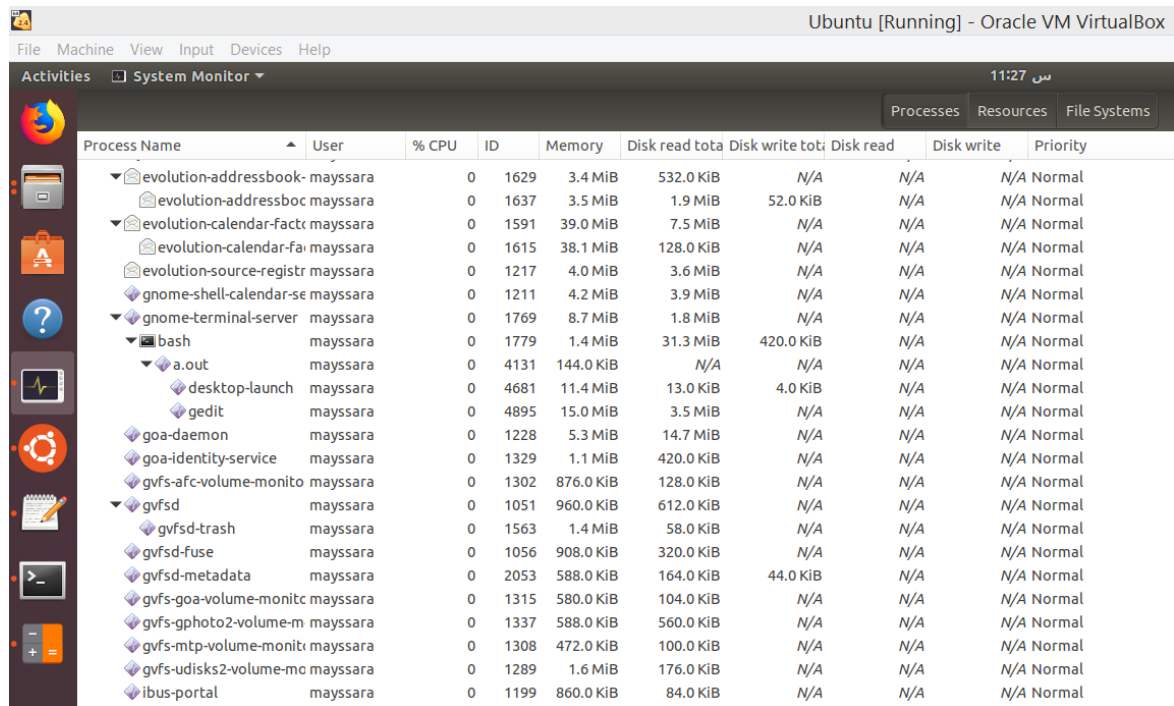- setBuiltins(): inserts built-ins commands in the built_in_commands set.

Main

- Sets signal handler for SIGCHLD signal.
- Sets built-ins commands.
- Opens log file.
- Prints shell line and takes new commands inputs and executes them.
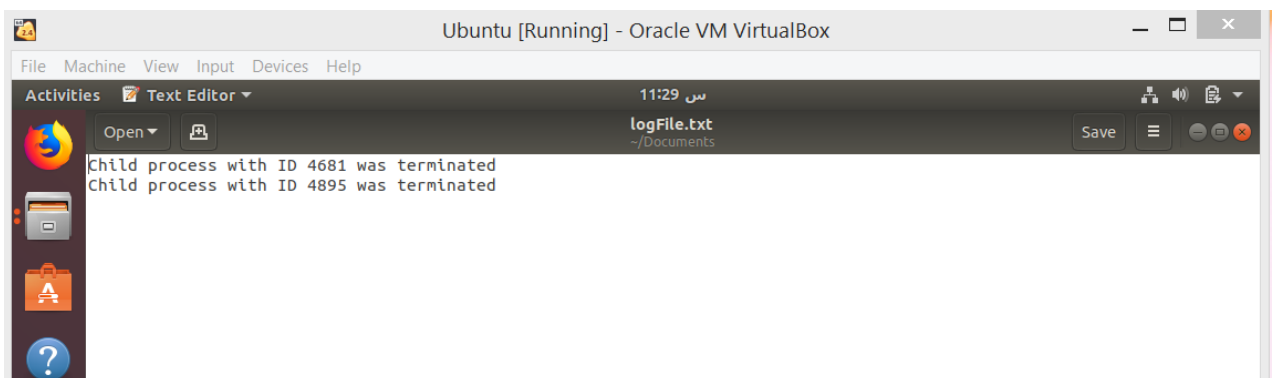- Closes log file.

Sample runs
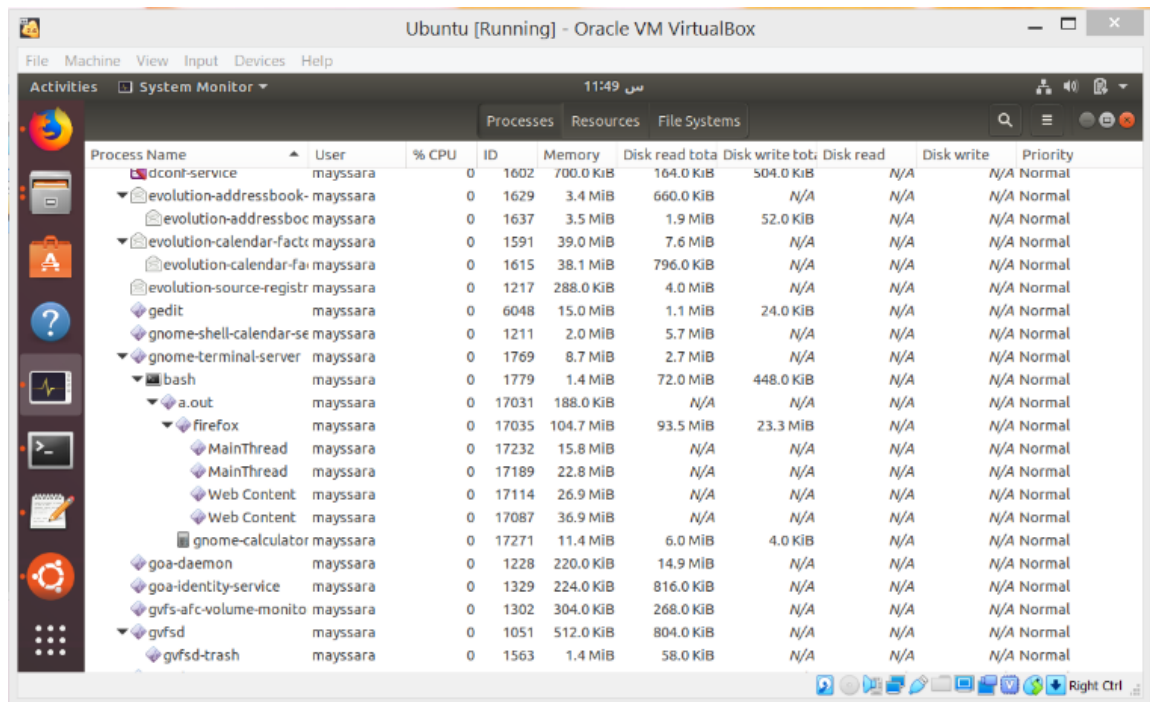
commands

- gnome-calculator &
- gedit &

Commands

- firefox &
- gnome-calculator &