OS lab Exercise 2 Design Document

‘By: Kareem Kassab
to Dr Karim Sobh

Kareem Kassab Exercise 2 OS Lab 900182771

Design Overview:

This program simulates the Linux shell through a series of functions and conditionals that are made by the assistance of some flags. The execve handles around half the functions, and the rest is supported trough the series of the following data structures and functions:

Major Data Structures:

Arrays&Buffers: var, val, dirpath

Var& val are used to store values in custom variable defined by the user in the assignvar function

Dirpath is used to print the directory we are in in the

Flags: inflag, outflag, redirectionflag, assignflag, pipeflag.
These are mainly used to define the mode of operation to indicate the conditional tree that we are going to operate on

Major Functions:

Getoperator
Purpose: to get the operator and its order in the submitted command
Input: argc, argv, the command string
Output: the order of the operator
Assumptions: none

Redirectin
Purpose: redirects the reading command
Input: argc, argv, operator order
Output: just returns 1 and continues the program just in case there is a pipe after
Assumptions there is a “>” operator in the command

RedirectOut
Purpose: redirects the writing command
Input: argc, argv, operator order
Output: just returns 1 and continues the program just in case there is a pipe after
Assumptions there is a “<” operator in the command

assignVar
Purpose: to assign values to custom variables if the command demands so
Input”arg, argv, 2 buffers for variable and value
Output: the value of the variable
Assumptions: there is a “=” operator, and that the first element of the command buffer argv[0] is the variable

pipecommands
Purpose: to handle the commands that have a pipe and need to be done together
Input: argc, argv,
Output:
Assumptions: there is at least one | in the command