

Program #2

By: Alex Herman

Professor: Dr. McGough

Due Date: March 14, 2016

Description

This program is a shell. It is a command line that takes commands and arguments from the user through a text interface. There are six intrinsic commands in the shell: systat, cmdnm, pwd, cd, signal, and exit.

These commands do the following:

- systat - Displays information about the hardware and operating system
- cmdnm - Displays the command used to start a chosen process
- pwd - Prints the current working directory of the shell
- cd - Changes the working directory to the specified directory
- signal - Sends a signal to a process
- exit - Exits the the shell
- hb - Prints the system timestamp every chosen interval

Any command issued by the user that is not intrinsic is executed in a new process. If the chosen command is not found, an error message is printed. Input and Output redirection is also supported.

These can be used with the following syntax:

- [command] > [file] - Redirect standard out to the chosen file
- [command] < [file] - Redirect standard in to the chosen file
- [command] | [command] - Redirect standard out from the first command to standard in for the second command
- [command])) [port] – Redirect standard out to the specified port
- [jcommand] (([address] [port] – Redirect standard in to the specified ip and port combination

****Note** – All redirection works on unix commands AS WELL as intrinsic functions.

Contents

dsh.c – The source file that contains all of the functions in the program.

Makefile – This is the file that specifies compilation instructions that are carried out by the ‘make’ command.

Usage

Compilation: Run 'make' command

Running the program: Run './dsh' from the command line

Usage: Run commands similar to any other shell. Type the name of the command followed by any arguments separated by spaces.

Functions

Function: usage()

Description: This function displays usage statements based on the command entered by the user. This is only executed when a command is entered with the incorrect number of arguments.

Parameters:

command - The command that the user passed into the shell

Function: systat()

Description: This function displays some system information obtained from the proc file system. It displays the linux version, system uptime, memory info, and some information about the cpu in the computer.

Parameters:

arg – This argument contains no information but is required to use pthreads

Function: signal()

Description: This function sends a signal to a selected process. It utilizes the kill() function to send the desired signal.

Parameters:

signalNum - This is the actual signal that is sent to the selected process

pid - This is the id of the process that the signal is sent to

Function: pwd()

Description: This function returns the current working directory of the shell. It uses the getcwd() function to obtain this information.

Function: printCurrentTime()

Description: This function prints the system's current time. It displays it in hh:mm:ss format.

Function: pipeCommands()

Description: This function is used to create a pipe between two processes. It takes the output of the command represented by the lhs vector and uses it as the input of the command represented by the rhs vector.

Parameters:

lhs – The vector of command strings on the left hand side (lhs) of the pipe

rhs – The vector of command strings on the right hand side (rhs) of the pipe

Function: isNotBlank()

Description: This function takes a character array and checks to see if it contains any characters aside from white space. It stops searching after it encounters a null terminator. It returns true if the string contains non-whitespace characters. If the string is all whitespace, it returns false.

Parameters:

str - A character pointer to the first letter of the string to be checked.

Function: isInteger()

Description: This function determines whether a string is actually an integer. It simply checks if every character in the string is a digit, and if a single one is not then it returns false

Parameters:

c – the string to check

Function: hb()

Description: This function prints out the current system time at an increment specified in the structure passed in the argument. This increment can be either seconds or milliseconds, also specified in the argument.

Parameters:

pthreadArg – This is a void pointer to a structure that contains:

tinc – The number of intervals between each timestamp prints

tend – The number of intervals to print for

tval – The interval unit, either s or ms (seconds or milliseconds)

Function: handleRedirection()

Description: This function is used to handle all of the possible redirection operators that exist within a command. It can handle input and output redirection, command piping, and remote piping. If any output is redirected, the corresponding referenced Boolean value is set.

Parameters:

tokens – This is a vector of strings, each of which is an argument passed into the shell by the user. This includes the name of the command.

fd – This is an array that contains the file descriptors for input and output.

pipd – A Boolean value passed by reference to tell the calling function that two commands were piped

redirectIn – A Boolean reference to tell the calling function that input was redirected

redirectOut – A Boolean reference to tell the calling function that output was redirected

Function: execute()

Description: This function is called when a non-intrinsic command is entered by the user. It starts by forking itself. The child process then runs an exec command to start the process specified by the user. The original process

prints the pid of the child process, and once the child has finished executing it displays some statistics about the child.

Parameters:

tokens - This is a vector of strings, each of which is an argument passed into the shell by the user. This includes the name of the command.

Function: createSocketServer()

Description: This function creates a socket that is used to transmit the output of the command to another process. It transmits this information over the port specified. This function returns a file descriptor that the output of the command is redirected to.

Parameters:

port – The number of the port that the information is sent across

Function: createSocketClient()c

Description: This function creates a socket that is used to receive data from the specified ip address and port combination. It returns a file descriptor that the input of the command is redirected to.

Parameters:

address – The ip address of the device sending the information

port - The number of the port that the information is sent across

Function: cmdnm()

Description: This function stands for "command name" and returns the contents of the cmdline file in the proc file system. This file contains the command that was issued in order to start the process.

Parameters:

pid - The id of the process whose name is being retrieved

Function: chooseCommand()

Description: This function is the decision making engine. Given a vector of strings that represent the arguments passed into the shell by the user, it either calls the correct intrinsic function, or it calls a function to execute the command in a new process. If it returns true, then the exit command has been entered and it will exit the program.

Parameters:

tokens - The vector of string arguments passed by the user into the shell

Function: main()

Description: This is the main entry point of the program. It is in charge of displaying the prompt and parsing the user's arguments, which it then passes to another function that determines the course of action.

Parameters:

argc - The number of arguments passed to the program from the command line

argv - An array of strings passed to the program from the command line

Libraries

iostream

License: GNU General Public License

Use: Provides functionality for displaying information to standard IO

vector

License: GNU General Public License

Use: This gives access to the vector class, used in place of arrays in several places

iterator

License: GNU General Public License

Use: This class is used to gain access to a string iterator that is used to parse user arguments

sstream

License: GNU General Public License

Use: This library is used to create a string stream, which is used to parse user arguments

unistd.h

License: Mozilla Public License

Use: This library gives access to the following unix commands: fork, execvp, chdir, and getcwd

fstream

License: GNU General Public License

Use: This library provides functionality for doing IO with files on the disk.

algorithm

License: GNU General Public License

Use: This class contains the atoi function, which converts strings to integers, the exit function, which closes the program, and the transform method, which is used to convert a string to lower case.

sys/wait.h

License: GNU General Public License

Use: This contains functionality that is used to wait for the child process to finish executing.

sys/resource.h

License: GNU General Public License

Use: This library is used for its access to the getrusage function, which is used to access runtime statistics of the child processes.

arpa/inet.h

License: Apple Public Source License Version 1.1

Use: Used in definitions of internet addresses

ctime

License: GNU General Public License

Use: This class is used for getting the current time of the system

io manip

License: GNU General Public License

Use: This is used for basic output formatting

netinet/in.h

License: Copyright (c) 1982, 1986, 1990 Regents of the University of California.

Use: This library is used for socket communication

pthread.h

License: GNU General Public License

Use: This library is used to run commands in separate threads

string.h

License: GNU General Public License

Use: This class is used for assorted string manipulation

sys/socket.h

License: GNU General Public License

Use: This is used to create a socket through a system call

Stdio.h

License: GNU General Public License

Use: This library is used for input redirection capabilities

Testing Process

The testing process for this program was quite simple. I input blank entries, entries with the wrong arguments, and non-existent commands, into each intrinsic function. Every external command has its own error handling so that part of the process is not in my control. Then, I tried all of this again but including each kind of input redirection. This process covered all possible areas of error that I could think of.