

OSN Assignment 2

2021111003

General

Compile via Makefile, with command `make`. Executable named `shell`. Maximum size for strings (buffer) is set to 5000 in the `Headers.h` file. Changing here will change all occurrences of `size`. `homeDir` and some helper functions are called by `extern` and not included into each other. `base_comm.c` is used to tokenise and call respective functions for all inputs, along with containing some helper functions. `clear` and `exit` are also implemented.

Specification 1

This segment of the code is present in `prompt_comm.c` and can be called by `promptText()`. It uses `setDirectory()` function as a helper function on setting the path on the prompt.

Specification 2

This segment is in `builtin_comm.c` and called by `pwdCommand()`, `echoCommand()` and `cdCommand()`. The `cd` function uses four helper functions `removeSpaces()`, `spaceCheck()`, `slashGet()` and `stepBack()`. The three functions work with quotes as well in all combinations, but do not work with `\n` character in the quoted text. The directory from which the shell is invoked is considered as the 'home' directory.

Specification 3

The segment is in `ls_comm.c` file and called by `lsComm()`. The `ls` function has `spaceGet()`, `sort()`, `getColor()` and `sprintfData()` as helper functions for separating (tokenising), sorting, setting print color and formatting data, resp. The `ls` command only works with a maximum of 200 files per directory displayed. If the directory is bigger, it cannot display and files.

Specification 4

The system command function `sysComm()` is in `system_comm.c` file. The function uses `spaceGet()` function for tokenising. The function does not run builtin commands in background. Time for running is not calculated here. It is found in the `assign2.c` file and passed to `promptText()`.

Specification 5

The `pinfoComm()` is in the `pinfo_comm.c` file. The function uses `/proc` to get data on processes. So specs 4, 5 and 6 do not a limit on how many processes can be run from the code side. `pinfo` command without argument prints the data for `./shell`.

Specification 6

This segment is in `bg_done_comm.c` and called by `handleBgProcess()`. It checks for completed background processes and prints id of completed ones. This function when ends will have to be recovered

to prompt by pressing 'Enter' from the user.

Specification 7

This segment is in `discover_comm.c` and command declaration is `discoverComm()`. It uses a recursive call function `printChildren()` to recursively check and print required files and directories. It also uses `spaceGet()` to tokenise the input command. Default directory path is `cwd` and if no destination file is given, all files matching flags are printed. If no destination is given and neither of the flags are given, then they are both by default set to be active.

Specification 8

Contained in `history_comm.c` file and called by `historyComm()`. The commands are stored in `.history_temp.txt`. Type 0 of the command is used to update the hidden file and type 1 is used to print the history elements. Negative values are not allowed. If number is greater than 20, only the 20 stored are printed.