University of
St Andrews

CS3104 Practical 3

200016227

Due date: November 29, 2024

1 Implementing ls in STACSOS

1.1 Overview

For this practical, the *ls* command was to be implemented for the STACS operating system. This included creating a new struct, creating new system calls and modifying *fs-node* and its implementations to support directory listings.

1.2 Completeness

The *ls* command is supported in my implementation, as well as two unique flags:

- `r`
  Enables recursive directory listing
- `l`
  Enables long listing with the specified format

1.3 Strategy and Implementation Choices

The main function in *user/ls/src/main.cpp* parses the arguments looking for any flags and the directory to be used in the listing. Once the arguments are parsed, the recursive function *listPath()* is called with the path and the options as the arguments.

The *listPath()* function verifies the path, then makes a system call (*open_directory()*) to obtain an object id which is then passed to the *read_directory()*, this systemcall can then find the object using the id and advance it by setting it to its child node. This behavior is intended and closely resembles iterators from Java.

Once the fs_node has been advanced and all file/directory data of the new node populated, the *list_path()* function reads whether the node is a directory or a file and depending on whether the long listing flag is enabled, it will print out the relevant information accordingly.

The iterator continues until *read_directory()* returns -1 due to the systemcall code not returning *syscall_result_code::ok*, at that point the function has finished listing and returns a 0 for successful exit code.

1.4 Conclusion

My final solution seems to be working as specified. In addition, the recursive flag has been implemented which works as intended and recursively lists out the files inside of directories. I enjoyed working on this practical and learning about how the kernel space interacts with the user space as that was the biggest hurdle to overcome through this practical.