Overview of Implementation:

The goal of this project was to implement the ls command for the stacsos operating system, enabling us to list the contents of directories. This involved the creation of a user space application and modifications to the TAR filesystem driver and the system call infrastructure to support directory listings.

Assumptions:

1. The TAR filesystem contains a nested hierarchy of directories and files.
2. The maximum filename length allowed is 50 characters long (hard coded in my implementation)

Implementation Details:

User Space Program: Created a new directory under user/ls and added the program to the Makefile. The program reads the command-line arguments to determine the directory path and whether to present a long listing (-l flag).

System Call Modification: Introduced a new system call /usr/ls to retrieve directory listings. This was necessary to handle the transmission of structured directory data from the kernel to user space.

TAR Filesystem: Modified the tarfs_node class to support reading directory contents. Implemented a method to gather directory entries and their attributes (size).

Data Transmission: Used buffer allocation strategies to safely copy data between kernel space and user space, ensuring boundary checks and data integrity.

Challenges Encountered:

While working on the practical, one of the challenges that became apparent was deciding how to transfer the directory data between the user's console and program. Originally, I attempted to transfer the data in the syscall_result output. But this proved problematic as I could not encode the information in u64 without errors. I realized that I could pass a char buffer to the syscall program for storing the data. This enabled easy access and retrieval and solved my issues.

Testing:

I tested the implementation using the directory structure, verifying that both standard and long listings correctly display the various contents. I also cross-verified file sizes with the actual sizes in the sysroot directory to ensure accuracy. I tested the output of /tree/c (which contains more than 1024 chars) which exceeds consoles output limit. By calling the console multiple times in ls/main.cpp, I enabled all the buffer to be printed. In my testing I have found my results reliable and accurate.