Here’s an updated project report for your directory and file search utility that includes the new features:

---

\*\*Advanced Directory and File Search Utility in C\*\*

Developed a comprehensive C-based utility to search and display information about files and directories within a specified location. The tool features enhanced functionality for filtering by file size, displaying detailed file information, and executing external commands on selected files.

- \*\*Project Features:\*\*

- \*\*Basic Directory and File Listing:\*\* Recursively traverses directories and prints names of all files and subdirectories.

- \*\*File Size Filtering:\*\* Implements a `-s` flag to list files larger than a specified size in bytes.

- \*\*Detailed File Information:\*\* Uses the `-S` flag to show detailed file information, including size, permissions, and last modified timestamp.

- \*\*External Command Execution:\*\* Integrates the `-e` flag to execute specified external commands (`ls -l` for detailed listing or `wc -l` for line count) on files meeting the size criteria.

- \*\*Command-Line Flexibility:\*\* Supports various flags and arguments for customized directory and file reporting.

- \*\*Recursive Functionality:\*\* Handles nested directories efficiently, ensuring that the entire file hierarchy is explored.

- \*\*Key Implementations:\*\*

- \*\*Dynamic Function Handling:\*\* Utilizes function pointers for dynamic selection of directory listing functions based on command-line arguments.

- \*\*External Command Execution:\*\* Uses `fork` and `exec` system calls to run external commands on files, showcasing understanding of process management in Unix-like systems.

- \*\*File Handling and System Calls:\*\* Applies `stat` for file attributes and `fseek`/`ftell` for file size determination, demonstrating proficiency in system-level programming.

- \*\*Technologies:\*\* C, File Handling (`dirent.h`, `stat`), Process Management (`fork`, `exec`), Makefile for build automation.

- \*\*Skills Demonstrated:\*\*

- Proficient in recursive algorithms and system-level file handling.

- Experience with command-line argument parsing and dynamic function invocation.

- Knowledge of process creation and execution in Unix-like environments.

- Ability to create and maintain a Makefile for project compilation and management.

This project highlights advanced skills in C programming, systems programming, and process management, making it a valuable addition to a resume for positions requiring low-level programming expertise.

---

This report emphasizes your advanced skills in C and system programming, showcasing the complexity and flexibility of your project. It should make a strong impression on potential employers or internship providers.