

## System Programming (CSE344) - HW1

### Advanced File System Search in Posix Environment

#### How I solved this problem ?

- I exploited **struct** and **enumerate** type to keep my algorithm neat then traverse the file system with recursive function and validate the current file or directory with validation function that contains some system calls and macros.

#### How I made design decision ?

- I needed to keep the data while traversing on file system due to printing layout. Tree was required. Therefore I decide record the file name in a tree structure. Algorithm splits the path then keep in array, send that array to add function which is recursive also.

#### Expected result

- All required tasks should work as expected.
- All errors and system calls were checked and handled in a good way.
- There should not be any kind of leak and arbitrary errors based on my tests.

#### Optimization that I made

- I wrote my own split function which returns array of string instead of **strtok**. Because **strtok** is not a **thread safe** and may cause an error in multithreaded programs.
- In case of permission denied to file, user is notified then at the end of the search, tree will be printed to screen.