

*Design of Data Structures:
Simulated Unix File System Project*

ENGR 3950U / CSCI 3020U: Operating Systems

Instructor

Dr. Kamran Sartipi

FEAS, UOIT

Simulated File System Project

- Disk Device interfaces:
 - The whole file system is a file: `simdisk.data`
 - 512 disk blocks (0 to 511), Each block is 128 bytes.
 - `Put_block(blk#, memLoc)`: copies 128 bytes of data from memory to a disk block.
 - `get_block(blk#, memLoc)`: does the reverse operation as `put_block(..)`
 - `Simdisk.data` is created automatically (all bits zero) if not exist before. Otherwise, the existing one is used.
- File System:
 - Unix-like hierarchical directory structure
 - Two types of file: directory and regular
 - Full path name of files/directories: `/foo/bar/zam` not `bar/zam`
 - Path-name to file-name translation to get I-node number of the file/directory
 - Use of I-node table and file-descriptor (one per-process table) to open a file several times (at least 4)
 - **At least** 64 files to exist in system, 4 open files, each file up to 512 byte long.
 - File name component max 6 characters.
 - Appending to a file is by setting start location of the file to -1. This is the only allowable way to increase the length of a file.

Interface of Simulated File System

(see `sfstest.c` for instruction on how to compile)

sartipi% sfstest

o: open a file

r: read from a file

w: write to a file

R: read from a directory

c: close a file

m: create (make) a new file

d: delete a file

s: get the size of a file

t: get the type of a file

i: initialize the file system

q: quit - exit this program

Command?

**Simulated
File System**

