

# CSCI 4061 Discussion 6

2/26/18



UNIVERSITY OF MINNESOTA  
**Driven to Discover<sup>SM</sup>**

# Overview

- Hard Links
- Soft Links
- Dynamic Memory Management
- Exercise



# Links

- A reference to a file.
- Cheaper than duplicating
- Changes made via any link affect the content seen via all links.



# Hard Links

- Reference a file's inode.
- Are not effected by name changes.
- Count of hard links maintained by OS.
  - When count == 0, file/inode is deleted.



# Soft Links

- Reference a file by name.
- Soft links can be 'bad' if the file/link they referenced is deleted/renamed.
- Count is not maintained.
- A file with soft links may be deleted.



# Istat and readlink

- `lstat(char*, struct stat*)` performs `stat()`, but if the file is a soft link, the data is about the link, not the file.
  - `S_ISLNK(stat->st_mode);`
- `readlink(char* path, char* buff, int size)` places the name of the file referenced by path in buff. Does not null-terminate the string.



# Dynamic Memory Management

- Memory allocated **statically** is placed on the stack.
  - Without the use of malloc
  - Does **not** persist after returning from function calls.
- Memory allocated **dynamically** is placed on the heap.
  - Allocated using malloc.
  - Persists after function calls.
  - If not freed, can lead to memory leaks.



# Exercise

- For this exercise, we will observe the performance of performing a ‘shallow copy’ with `link()` and a deep copy using `read/write()`. Takes in 3 arguments, the old directory and the names of two new directories for the shallow and deep copies.
- Finish writing `shallowCopy()`, which creates hard links in a new directory to every file in the old directory.
- Finish writing `deepCopy()`, which actually copies all the files in the old directory to the new one.
- Record the time it takes to copy the whole directory in each case and print out the results.

