

# Operating System 2021

## Quiz #10

- **links in file systems**
  - A directory with a link count of 2 is empty.
  - Symbolic links can cross file system boundaries while hard links are constrained to a single file system.
  - A hard link is resolved at creation time while a symbolic link is resolved when a the name is used to access a file system object.
- **about file names**
  - All names of files in a directory must be unique.
  - File systems have different sets of reserved characters (e.g., path separators).
  - On Unix file systems, the path `/usr/./bin/..` refers to the root of the file system.
- **special file systems**
  - A device file system is used to expose **devices** that have been detected by the kernel as device files to user space applications.
  - A process file system is used to expose the kernel's process list and related information to user space processes.
  - An user space file system is used to expose a file system implemented in user space to user space applications.
- **about file systems using index nodes (inodes)**
  - Index node file systems provide fast sequential access and good random access.
  - The attributes of a file system object are stored in the index node., It is possible to have files where not all data blocks are actually allocated (sparse files).
  - Some file system operations may require multiple changes to the inodes of a file and hence there can be inconsistencies until all block updates have been committed to stable storage.
- **permissions in file systems**
  - The file permissions `-r-xrw-r-` indicate that a regular file can be read and executed by the owner, it can be read and written by group members, and it can be read by anybody else.
  - Changing the name of a file only requires write permissions to the directory but not to the file itself., The file permissions `—r-r-` prevent the owner of the file from reading the file content but the owner retains the rights to change the file permissions.
  - File permissions can control under which rights a program is executes.
- **about mounting file systems**
  - An entire file system can be mounted read-only, which prevents any write attempts by regular processes, irrespective of the permissions associated with file system objects.
  - Mounting a file system on a directory usually implies that any content of the directory becomes inaccessible.
  - User space processes may expose a file system that can be mounted by the kernel., Unmounting a file system while processes have files of the file system open can cause file system corruption and data loss.
- **POSIX `fcntl` file locks**

- File locks are advisory, i.e., they can be ignored by applications., Locks can cover entire files or only a region of a file (record locks).
- Applications sometimes use the existence of files as way to indicate that some other file is locked.
- Locks are associated to a process and they will be released when the process terminates.
- about virtual file systems
  - Virtual file systems provide an abstraction to simplify the integration of new file systems.
  - Concrete file systems do not necessarily access the local block storage to store data.
  - The virtual file system resolves names to virtual inodes and maintains suitable directory entry caches.
- about special file events
  - File events describe file system accesses and changes.
  - File events cannot be obtained for all mounted file systems.
  - File events may be incomplete if, for example, buffers are filling up., On Linux systems, file system events are signaled via special file descriptors.
- \* File creation time → {index node},
- \* Directory entries → {data node},
- \* File permissions → {index node},
- \* Directory name → {data node},
- \* Directory permissions → {index node},
- \* File name → {data node},
- \* File size → {index node}