Jeremy Holloway
CPSC-3220-001
6/12/2019

Pre-class Assignment #18

1. Define the following terms:

   -non-volatile storage: Unlike D-Ram, memory that is durable and retains its state across crashes and power outages.

   -file system: An operating system abstraction that provides persistent, named data.

   -file: A named collection of data in a file system.

   -file metadata: Information about a file that is managed by the operating system, but not including the file contents.

   -directory: a list of human-readable names plus a mapping from each name to a specific file or sub-directory.

   -hard link: The mapping between a file name and the underlying file, typically when there are multiple path names for the same underlying files.

   -symbolic link: directory mappings from a a filename to another filename and then uses the target name to open the file.

   -volume: A collection of physical storage blocks that form a logical storage device.

   -mount: A mapping of a path in the existing file system to the root directory of another file system volume.

   -file descriptor: A handle to an open file, device, or channel.

   -device driver: Operating system code to initialize and manage a particular I/O device.

   -block device: An I/O device that allows data to be read or written in fixed size blocks.

2. Give at least three distinct examples of information maintained in a file's metadata.

   A file’s size, modification time, owner, and it’s security information.

3. Identify at least three distinct types of files.

   Regular, Directory, Special File.

4. Identify three distinct conventions for indicating the file type.

Magic number, Meta-data, extension.

5. What is the purpose of an explicit file open()?

This is done for two reasons, path parsing and permission checking can be done when the file is opened which saves repeated checks, and the OS can create a data structure that stores information about the file.

6. What is the purpose of the fsync() system call?

To force to disk, all buffered dirty pages for the file associated with fileDescriptor.

7. Give an example where prefetching might make sense for file access.

Sequential access to a file, prefetching can reduce storage device overheads by repackaging a large number of smaller requests into one larger one.