```
3. what parts go into an inode?

inode: some d-s to describe a "file" on some media

ownership: user, group

times: last modified (mtime), atime, ctime

pointers to data-blocks (inode is just the m-d of the file)

permissions: who can read/write/execute or otherwise access the file

type: what kind of inode this is? file, directory, symbolic link, char/bloc
device, fifo.

device that inode belongs to

file size: how big file is (in bytes, and in 512B blocks)

inode number: unique identifier in the file system.

syscall: stat(2) gives you m-d about a file
- statfs(2): give you info about whole f/s (e.g., capacity, used, type)

file name?!
- same named file but in different directory
- hard links use multiple names to the same inode.
- not part of inode. represents a "namespace"
- will discuss more with linux VFS.

4. MMU?

memory management unit
translates b/t physical and virtual addresses (fast)
monitors for access permissions to phys/virt pages:
- unmapped or non-existent mappings (result in page fault, SEGV, core dump)
- if process violated page protections (read, write, execute, none)
  e.g., TEXT segment and "STATIC variables" are in readonly segments

5. buffer cache?

buffer cache: caches I/O data into RAM, often from slow storage devices.

page cache:
- pages of information brought from disk
- pages of executables/binaries loaded from disk
- pages of running processes' segments: HEAP, STACK, etc.

modern OSs, marge the buffer and page caches.

cache management:
- overhead, speed
- validity and staleness of data
- whether data in cache is "newer" than on its source (dirty data)
```