**FreeBSD own algorithm memory management**

**Memory Classes**

**Active**

* Contains pages "actively" (recently) referenced by userland
* Contains a mix of clean and dirty pages
* Pages are regularly scanned by the page daemon (each page is visited once every vm.pageout\_update\_period seconds)
* Scans check to see if the page has been referenced since the last scan
* If enough scans complete without seeing a reference, the page is moved to the inactive queue
* Implements pseudo-LRU

**Inactive**

* Contains pages aged out of the active queue
* Contains pages evicted from the buffer cache
* Contains a mix of clean and dirty pages
* Pages are scanned by the page daemon (starting from the head of the queue) when there is a memory shortage:
  + Pages which have been referenced are moved back to the active queue or the tail of the inactive queue
  + Pages which are dirty are moved to the tail of the laundry queue
  + Unreferenced, clean pages may be freed and reused immediately
* Implements second-chance LRU

**Laundry**

* Queue for managing dirty inactive pages, which must be cleaned ("laundered") before they can be reused
* Managed by a separate thread, the laundry thread, instead of the page daemon
* Laundry thread launders a small number of pages to balance the inactive and laundry queues
* Frequency of laundering depends on:
  + How many clean pages the page daemon is freeing; more frees contributes to a higher frequency of laundering
  + The size of the laundry queue relative to the inactive queue; if the laundry queue is growing, we will launder more frequently
* Pages are scanned by the laundry thread (starting from the head of the queue):
  + Pages which have been referenced are moved back to the active queue or the tail of the laundry queue
  + Dirty pages are laundered and then moved close to the head of the inactive queue

**Free**

* Memory available for use by the rest of the system.

**Wired**

* Non-pageable memory: cannot be freed until explicitly released by the owner
* Userland memory can be wired by mlock(2) (subject to system and per-user limits)
* Kernel memory allocators return wired memory
* Contents of the ARC and the buffer cache are wired
* Some memory is permanently wired and is never freed (e.g., the kernel file itself)

**To make it short**,

Active memory is memory that is currently being used by processes or the kernel.

Wired memory is memory that has been explicitly marked as non-swappable, meaning that it cannot be moved to disk to free up memory.

Inactive memory is memory that has not been used recently but can still be swapped out to disk if needed.

Laundry memory is a list of dirty memory pages that need to be written to disk before they can be reused. Buffer memory is used to cache data from filesystems.

ARC (Adaptive Replacement Cache) is the cache used by ZFS to store frequently accessed data.

Free memory is memory that is not currently being used by any processes or the kernel. It can be used to allocate new memory for processes or to swap out inactive memory.