Page Replacement Policy in Linux

Dina Thomas(00D05015)

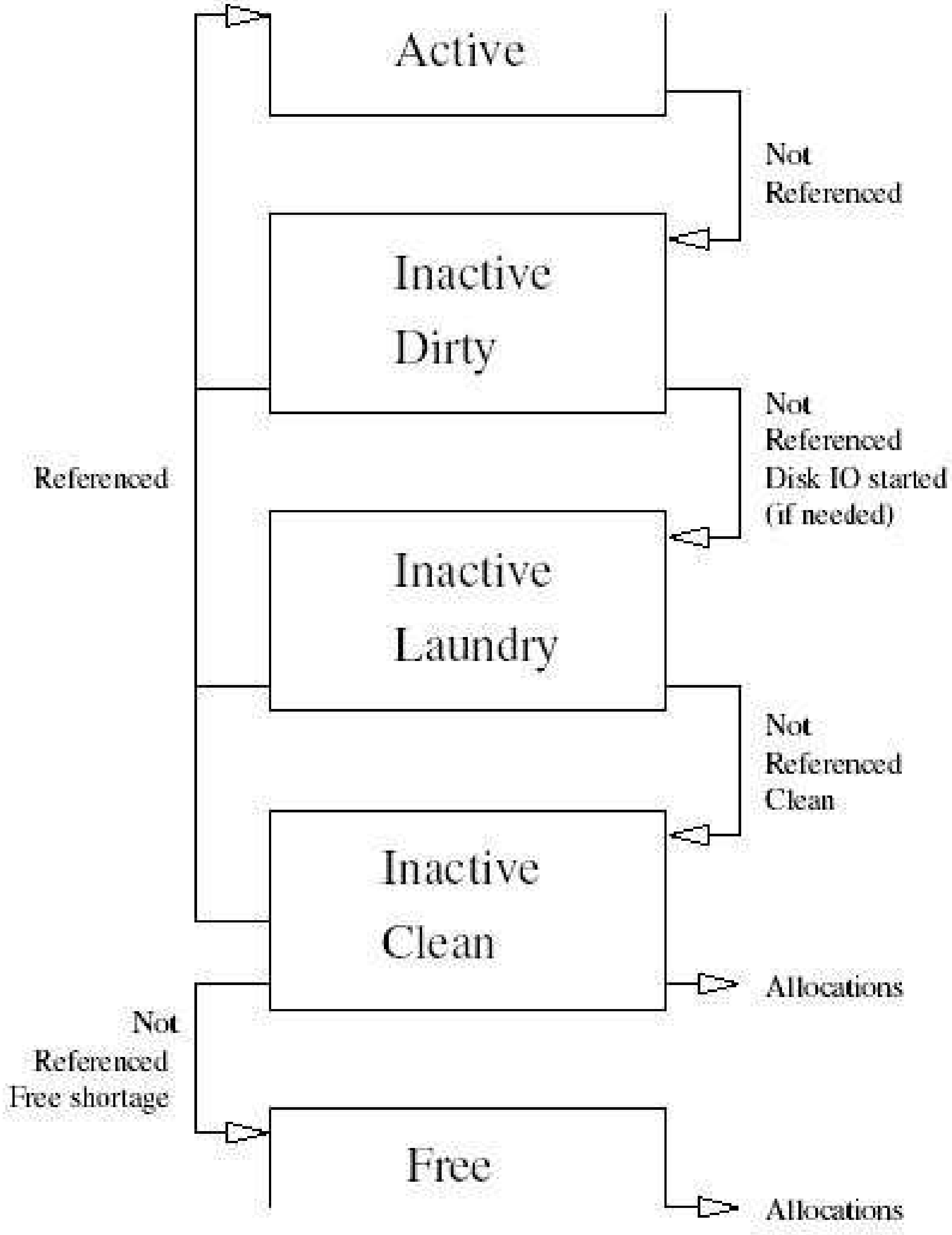
Priyanka Garg(00D05010)

Page Aging

- combines good points of NRU and NFU
- when a page is used, increase page_age
  - page_age += constant
- when a page is not used, decrease page_age
  - page_age= $\frac{page\_age}{2}$
- when page_age=0 consider for eviction

Implementation in Linux

- search for eviction page is process by process and then page by page.
- **Process by process:**
  - based on NRU
  - unreferenced pages $\Rightarrow$ Active-Page-List
- **Page by page:**
  - Page aging on Active-Page-List.
  - page_age=0 $\Rightarrow$ Inactive-Page-List.


Active
Not Referenced
Inactive Dirty
Not Referenced Disk IO started (if needed)
Referenced
Inactive Laundry
Not Referenced Clean
Inactive Clean
Allocations
Not Referenced Free shortage
Free
Allocations

Variants

- **Kernel 2.0:**
  - NFU
  - penalize a process, not more than a constant.
  - contribution is distributed
  - no one can build up its working set.
- **Kernel 2.2:**
  - NRU
  - selected process continously contribute,
  - penalize the memory usage of one process at a time.
- **Kernel 2.4:** only 6% address space of a process is scaned to select NRU pages.

LRU approximations

- NRU
  - Not Recently Used
  - reference bit based
  - idea: don't evict pages which were just used
- NFU
  - Not Frequently Used
  - based on frequency of access
  - accessed page : page_age +=constant_1
  - not accessed page : page_age -=constant_2
  - idea: don't evict pages which are used frequently

References

- Adaptive Page Replacement to Protect Thrashing in Linux.
- Page Replacement in Linux 2.4 memory management.
- The Linux Memory Management Home Page.

THANK YOU!!