

Notes on Cache Memory

Basic Ideas

The cache is a small mirror-image of a portion (several "lines") of main memory.

- cache is faster than main memory ==> so we must maximize its utilization
- cache is more expensive than main memory ==> so it is much smaller

How do we keep that portion of the current program in cache which maximizes cache utilization?

1. Direct Mapping

Each block is supposed to fit-in in the specific line of a cache.

Rigid Mapping

Inefficient utilization of cache memory.

2. Associative Mapping

3. Set Associative Mapping

Calculating Internal Fragmentation

Page Size = 2048 Bytes

Process Size = 72766 Bytes

35 Pages + 1086 Bytes

Internal Fragmentation = $2048 - 1086 = 962$ Bytes

1. Worst Case Internal Fragmentation = 1 frame – 1 Byte = $2048 - 1 = 2047$ Bytes
2. So small frame size desirable?
 - a. But each page table entry takes memory to track?
 - b. Page sizes growing over time.
 - c. Solaris support two values for page sizes – 8 KB and 4 MB

State True/False with justification

(A) Non-contiguous memory allocation schemes are advantageous, because, unlike the contiguous memory allocation schemes, they do not suffer from Internal Fragmentation or External Fragmentation Problems.

Solution: FALSE.

There are basically two types of non-contiguous memory allocation schemes, viz., Paging and Segmentations. Any fixed-size partition-based memory allocation scheme suffers from internal fragmentation. This problem arises when the size of a process is not a whole multiple of the partition size. In such case, for the last portion of the process, we need to allocate an entire partition whose part is only used by the process. Paging is also a fixed-size partition scheme, hence, suffers from internal fragmentation problem. External fragmentation exists when there is enough total memory space available to accommodate a process or part of a process, but the available spaces are not contiguous; storage is fragmented into a large number of small blocks of free memory. Segmentation may cause external fragmentation, when all blocks of free memory are too small to accommodate a segment, but the sum of the free spaces is larger than the segment size. Hence, paging suffers from internal fragmentation and segmentation suffers from external fragmentation problem. However, paging does not suffer from external fragmentation and segmentation does not suffer from internal fragmentation problem.

(B) For sequential files, linked allocation is preferred over contiguous allocation.

Solution: FALSE

For sequential access, the address of the k th block of the file which starts at block b can easily be obtained as $(b+k)$. Using contiguous allocation make the access extremely fast since the number of seeks are minimal because of contiguous allocation of file blocks.