COMP0020 Functional Programming

Lecture 19

Memory Allocation Techniques

Contents

- Pointer increment
- Free list – sequential fits
- Segregated free lists
- Buddy systems

Pointer increment

- Ideal world :
  - all allocated memory would be at one end
  - All free memory at other end
- Single pointer to boundary
- To allocate a block of size N, increment the boundary pointer by N and return the original value of the boundary pointer
- Problems : to achieve an “ideal world” may be costly

Free list : sequential fits

- Assume all free blocks chained together into a linked list.
  - A pointer holds the start of this chain
  - Freed blocks either added to start (LIFO) or end (FIFO) of chain, or inserted into chain in order of address (AO)
- First fit allocation
  - Larger blocks near start of list tend to be split first, resulting in many small fragments at start of list
- Next fit allocation
  - Tends to increase fragmentation in practice
- Best fit allocation
  - Tiny fragments ? Not in practice.
  - Slow ? Not always.
- Worst fit ? Good fit ?

Segregated free lists

- Set of free lists
  - each for blocks of a specific size
  - or for a range of sizes
    - Use first-fit or next-fit etc within free list
- "Good fit"
  - Or even "worst fit"
- Use an array of pointers to the free lists
- Or a tree of pointers to the free lists

Buddy systems

- Variant of segregated free lists
  - Aimed to optimize splitting and coalescing
- A free area may only be merged with its buddy
- When block freed - (unique) buddy found by simple address arithmetic
  - wholly in use, or wholly free (so can merge)
- Binary buddies
  - problem with internal frags (25%)
- Double buddies
  - uses powers-of-two (2, 4, 8...) AND double-from-three (3, 6, 12...)
  - Better frags (12.5%) but restricted coalescing

Summary

- Pointer increment
- Free list – sequential fits
- Segregated free lists
- Buddy systems

END OF LECTURE