Kernel Memory Allocator in BSD 4.3

Pankaj Kumar (00005026)
Manish Kumar (00005027)

under the guidance of
Prof. R. K. Joshi
Computer Science & Engineering Department,
Indian Institute of Technology,
Bombay.
November 13, 2003

Criterion for a kernel memory allocator

- Good utilization of physical memory
- Utilization in kernel more important than in user processes
- Kernel memory allocator needs to be fast
- Problems with slow memory allocator

Consideration Unique to a kernel allocator

- Kernel Memory is typically small part of the physical memory
- Dynamic memory allocated to kernel need not be very large as compared to user process which can get entire virtual memory
- Kernel can control its own address space
- All different uses of dynamic memory are known in advance

Implementation of the kernel memory allocator

- Hybrid of a fast memory allocator in 4.2 BSD C library and slower but more memory efficient first fit allocator
- Small Allocations
  - Power of two list (strategy)
  - A set of list that is ordered by increasing power of two
  - Each list contains a set of memory blocks of its corresponding size
  - To fulfill a memory request the size of request is rounded up to the next power of two(for 5 KB, the 8 KB space is given)
  - Freeing the memory is easy
  - Inefficient for large memory blocks

Implementation of kernel memory allocator

- Large allocation of memory
  - First round up to a multiple of page size
  - First fit algorithm is used to find the space in the kernel address space set aside for dynamic allocation.
  - So 5KB memory will use only 5 pages of memory rather than 8 KB as with the power of two strategy.
  - Memory is returned to free memory pool and address space is returned to kernel address space when memory is freed

References

- The Design and Implementation of the 4.4 BSD Operating System by Marshall Kirk McKusick, Keith Bostic, Michael J. Karels , John S. Quarterman
- Design of a General Purpose Memory Allocator for the 4.3BSD UNIX Kernel, *Proceedings of the San Francisco USENIX Confrence, pp, 295-303, June 1988*