

《操作系统》课下作业 (OS-HW5)

中国人民大学 信息学院 崔冠宇 2018202147

P361, 7.5 Another placement algorithm for dynamic partitioning is referred to as worst-fit. In this case, the largest free block of memory is used for bringing in a process.

- Discuss the pros and cons of this method compared to first-, next-, and best-fit.
- What is the average length of the search for worst-fit?

解.

a. 优点: 如果系统维护内存块的信息时保持大小降序, 则最坏适配(worst-fit)方法相对于其它方法来说具有查找快的特点, 而且不会像最佳适配等方法产生大量小碎片. 缺点: 首先分配较大的块, 可能导致浪费; 除此之外, 若系统不对块进行排序, 则查找长度退化为 n , 最大.

b. 若各内存块已经按照从大到小的顺序排序, 则由于分配时先与最大块进行比较, 故只需比较一次, 若成功即分配, 否则不会分配, 所以平均查找次数为 1; 在没有排序的前提下, 则需要从头到尾扫描一遍, 平均查找次数为 n (内存块个数).

P361, 7.7 A 512 KB block of memory is allocated using the buddy system. Show the results of the following sequence of requests and returns in a figure that is similar to Figure 7.6: Request A: 100; Request B: 40; Request C: 190; Return A; Request D: 60; Return B; Return D; Return C. Also, find the internal fragmentation at each stage of allocation/de-allocation.

解. (1)

初始	512KB			
请求 A	A	128KB		256KB
请求 B	A	B	64KB	256KB
请求 C	A	B	64KB	C
释放 A	128KB	B	64KB	C
请求 D	128KB	B	D	C
释放 B	128KB	64KB	D	C
释放 D	256KB			C
释放 C	512KB			

(2) 请求 A 时, 产生大小为 $128\text{KB}-100\text{KB}=28\text{KB}$ 的内部碎片; 请求 B 时, 产生大小为 $64\text{KB}-40\text{KB}=24\text{KB}$ 的内部碎片; 请求 C 时, 产生大小为 $256\text{KB}-190\text{KB}=66\text{KB}$ 的内部碎片; 释放 A 时, 对应的内部碎片(28KB)被合并回收; 请求 D 时, 产生大小为 4KB 的内部碎片; 释放 B 时, 对应的内部碎片(24KB)被合并回收; 释放 D 时, 对应的内部碎片(4KB)被合并回收; 释放 C 时, 对应的内部碎片(66KB)被合并回收.

P362, 7.12 Consider a memory-management system based on paging. The total size of the physical memory is 2 GB, laid out over pages of size 8 KB. The logical address space of each process has been limited to 256 MB.

- Determine the total number of bits in the physical address.
- Determine the number of bits specifying page replacement and the number of bits for page frame number.
- Determine the number of page frames.
- Determine the logical address layout.

解.

- $2 \text{ GB} = 2^{31} \text{ Byte}$, 所以物理地址需要 **31** 位表示.
- 一个进程的地址空间最多 $256 \text{ MB} = 2^{18} \text{ KB}$, 即一个进程最多有 $2^{18} \text{ (KB)} / 2^3 \text{ (KB/页)} = 2^{15}$ 逻辑页, 所以进程页表需要 **15** 位来表示页号. $2 \text{ GB} = 2^{21} \text{ KB}$, 页数 $= 2^{21} \text{ (KB)} / 2^3 \text{ (KB/页)} = 2^{18}$ 页, 故需要 **18** 位表示页框号.

c. 由上面的计算, 有 **2^{18}** 个页框.

d. 逻辑地址 = $\underbrace{\text{逻辑页号}}_{15\text{位}} + \underbrace{\text{页内偏移}}_{13\text{位}}$. (此处加号表示拼接, 不是数学加法)

P362, 7.14 Consider a simple segmentation system that has the following segment table:

Starting Address	Length(bytes)
830	346
648	110
1,508	408
770	812

For each of the following logical addresses, determine the physical address or indicate if a segment fault occurs:

- a. 0,228; b. 2,648; c. 3,776; d. 1,98; e. 1,240

解.

- 对段 0, $228 < 346$, 故物理地址为 $830 + 228 = 1058$.
- 对段 2, $648 > 408$, 发生段错误.
- 对段 3, $776 < 812$, 故物理地址为 $770 + 776 = 1546$.
- 对段 1, $98 < 110$, 故物理地址为 $648 + 98 = 746$.
- 对段 1, $240 > 110$, 发生段错误.