《操作系统》课下作业 (OS-HW9)

中国人民大学 信息学院 崔冠宇 2018202147

P597, 12.2 One scheme to avoid the problem of preallocation versus waste or lack of contiguity is to allocate portions of increasing size as the file grows. For example, begin with a portion size of one block, and double the portion size for each allocation. Consider a file of n records with a blocking factor of F, and suppose a simple one-level index is used as a file allocation table.

a. Give an upper limit on the number of entries in the file allocation table as a function of F and n.

b. What is the maximum amount of the allocated file space that is unused at any time?

解.

a. 根据题意, 记录占用的块数 $= \frac{n}{F}$. 当 $0 < \frac{n}{F} \leq 1$ 时, 只有一项; $1 < \frac{n}{F} \leq 2$ 时, 有两项; $2 < \frac{n}{F} \leq 4$ 时, 有三项. 找到规律, 当 $2^{i-1} < \frac{n}{F} \leq 2^i$ 时, 有 $i+1$ 项, 所以至多需要 $\lceil \log_2 \frac{n}{F} \rceil + 1$ 项.

b. 在任何时刻, 未使用的空间不超过已分配空间的一半.

P597, 12.7 A sequential file is stored in a disk occupying 100 contiguous disk blocks.The disk has an average rotational delay of 2.5 ms. The time taken to seek the head of the drive to the required cylinder is 25 ms and the time taken to read a block is 0.25 ms. Find the minimum, maximum, and average time to search for a record using a linear search process.

解. 我有两种理解, 其一是定位磁头以及旋转延迟为平均情况, 仅考虑搜索的最优最差情况; 二是定位磁头及旋转延迟和搜索等因素都考虑.

理解一.(仅考虑搜索记录) 如果在任何情况下寻道时间几乎是不能缩短的, 旋转延迟按平均情况计算, 这就至少消耗了 $25 + 2.5 = 27.5$ ms. 由于是线性查找:

最少时间. 被查找的记录就在第一条, 只需要读 1 个块, 则总耗时为 $27.5 + 0.25 = 27.75$ ms.

最长时间. 被查找的记录在最后一条, 需要读 100 个块, 则总耗时为 $27.5 + 0.25 \times 100 = 52.5$ ms.

平均时间. 根据线性查找平均查找长度为总长度的一半, 需要读 50 个块, 则总耗时为 $27.5 + 0.25 \times 50 = 40$ ms.

理解二.(考虑定位和搜索记录)

最少时间. 磁头位置恰好在第一条记录的位置, 寻道时间为 0, 最小旋转延迟为 0, 被查找的记录就在第一条, 只需要读 1 个块, 则总耗时为 0.25 ms.

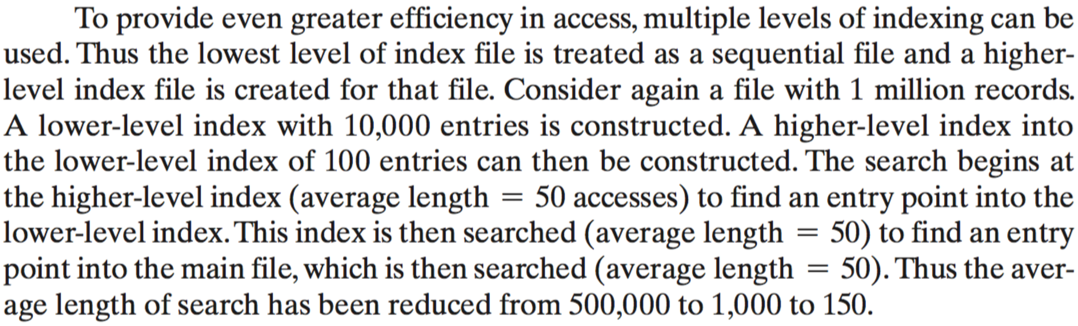
最长时间. 考虑寻道时间为 25 ms, 最大旋转延迟为 $2.5 \times 2 = 5$ ms, 被查找的记录在最后一条, 需要读 100 个块, 则总耗时为 $25 + 5 + 0.25 \times 100 = 55$ ms.

平均时间. 考虑寻道时间为 25 ms, 平均旋转延迟为 2.5 ms, 根据线性查找平均查找长度为总长度的一半, 需要读 50 个块, 则总耗时为 $25 + 2.5 + 0.25 \times 50 = 40$ ms.

P598, 12.12 A sequential file has 10 million records. How does efficiency in access improve by using a two-level index? Assume 100 entries in a higher-level index and 10,000 entries in a lower-level index.

解. 参考原书 P560 相似内容. 不设置索引, 顺序查找平均查找长度为 5,000,000 次. 设置双层索引后, 高级索引至低级

索引为1-100的关系, 低级索引至文件块为1-1000的关系. 查找时, 高级索引平均查找50次, 找到对应索引项; 进入后查找索引项对应的低级索引, 平均查找50次, 找到对应文件块表, 平均查找500次, 共计600次. 效率提升近1万倍.



To provide even greater efficiency in access, multiple levels of indexing can be used. Thus the lowest level of index file is treated as a sequential file and a higher-level index file is created for that file. Consider again a file with 1 million records. A lower-level index with 10,000 entries is constructed. A higher-level index into the lower-level index of 100 entries can then be constructed. The search begins at the higher-level index (average length = 50 accesses) to find an entry point into the lower-level index. This index is then searched (average length = 50) to find an entry point into the main file, which is then searched (average length = 50). Thus the average length of search has been reduced from 500,000 to 1,000 to 150.