# Article information:

内存管理(三)——内存分页 - 知乎  
<https://zhuanlan.zhihu.com/p/549919985>

# Article summary:

1. Memory paging divides the virtual and physical memory space into fixed-sized pages and frames respectively, with a typical page size of 4KB in Linux.

2. Page tables are used to map virtual addresses to physical addresses, allowing for efficient memory management and swapping of pages between memory and disk.

3. Multi-level page tables can be used to reduce the memory space required for storing page tables, with two-level paging being commonly used in 32-bit systems.

# Article rating:

Appears moderately imbalanced: The article provides some useful information, but is missing several important points or pieces of evidence that would be required to present the discussed topics in a balanced and reliable way. You are encouraged to seek a more balanced perspective on the presented issues by exploring the provided research topics and looking at different information sources.

# Article analysis:

The article titled "内存管理(三)——内存分页" provides an overview of memory paging in computer systems. While the article offers some useful information, it lacks depth and fails to address certain important aspects.

One potential bias in the article is its focus on Linux as the operating system for discussing memory paging. This narrow focus limits the scope of the discussion and may not be applicable to other operating systems or environments. The article could have provided a more balanced view by mentioning other popular operating systems like Windows or macOS.

Furthermore, the article makes unsupported claims without providing evidence or references. For example, it states that memory swapping to disk is efficient because only a few pages are written at a time. However, no data or studies are cited to support this claim. Without supporting evidence, readers may question the validity of such statements.

The article also fails to explore counterarguments or potential risks associated with memory paging. It does not mention any drawbacks of paging, such as increased overhead due to page table lookups or potential performance issues when there is heavy swapping between physical memory and disk.

Additionally, the article lacks technical details and explanations for concepts like page tables and page table entries. It assumes prior knowledge on these topics, which may make it difficult for readers who are unfamiliar with memory management concepts to understand the content fully.

Another issue is that the article does not provide a comprehensive comparison between segmentation and paging. It briefly mentions segmentation in the introduction but does not discuss its advantages or disadvantages compared to paging. A more balanced analysis would have included a comparison between these two memory management techniques.

Overall, while the article provides a basic introduction to memory paging, it lacks depth, supporting evidence, and a balanced perspective. It would benefit from addressing potential biases, providing more detailed explanations, exploring counterarguments, and presenting a broader view of memory management techniques beyond Linux-specific examples.

# Topics for further research:

* Memory paging in Windows operating system
* Drawbacks of memory paging in computer systems
* Page table lookup overhead in memory paging
* Performance issues with heavy swapping between physical memory and disk
* Comparison between memory segmentation and paging
* Technical details of page tables and page table entries in memory management

# Report location:

<https://www.fullpicture.app/item/f46e6701f8eab11e6f1615467da26725>