1. What is "internal fragmentation" in paged memory architecture? What happens when page frames sizes are too big or small?

**Internal fragmentation is when the page sizes are too large or too small. When the page sizes are too large, you end up wasting memory because there is a larger potential that there is more wasted space in each page. When you use pages that are too small it could be inefficient, and data could carry over to other pages.**

1. When the operating system reuses memory, it must first zero out the contents of the memory or disk. Why?

**Because it needs to ensure that the memory is not waiting on another process to write to it and it needs to make sure that there’s no bad data pre-existing.**

1. Almost all multi-level address translation systems use paging as the lowest level of the architecture. What are the benefits?

**Some of the benefits are that the individual page tables are smaller which means its faster and easier to access data and assign space. The page tables can be paged easily, its more specific and more organized.**

1. Please work on Chapter 9: page 458, Problem 4.

**Suppose an application is assigned 4 pages of physical memory and the memory is initially empty. It then references pages in the following sequence:**

ACBDBAEFBFAGEFA

* 1. Show how the system would fault pages into the four frames of physical memory, using the LRU replacement policy.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **A** | **C** | **B** | **D** | **B** | **A** | **E** | **F** | **B** | **F** | **A** | **G** | **E** | **F** | **A** |
| **1** | **A** |  |  |  |  | **+** |  |  |  |  | **+** |  |  |  | **+** |
| **2** |  | **C** |  |  |  |  | **E** |  |  |  |  | **G** |  |  |  |
| **3** |  |  | **B** |  | **+** |  |  |  | **+** |  |  |  | **E** |  |  |
| **4** |  |  |  | **D** |  |  |  | **F** |  | **+** |  |  |  | **+** |  |

* 1. Show how the system would fault pages into the four frames of physical memory, using the MIN replacement policy.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **A** | **C** | **B** | **D** | **B** | **A** | **E** | **F** | **B** | **F** | **A** | **G** | **E** | **F** | **A** |
| **1** | **A** |  |  |  |  | **+** |  |  |  |  | **+** |  |  |  | **+** |
| **2** |  | **C** |  |  |  |  | **E** |  |  |  |  |  | **+** |  |  |
| **3** |  |  | **B** |  | **+** |  |  |  | **+** |  |  | **G** |  |  |  |
| **4** |  |  |  | **D** |  |  |  | **F** |  | **+** |  |  |  | **+** |  |

* 1. Show how the system would fault pages into the four frames of physical memory, using the clock replacement policy.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **A** | **C** | **B** | **D** | **B** | **A** | **E** | **F** | **B** | **F** | **A** | **G** | **E** | **F** | **A** |
| **1** | **A1** | **1** | **1** | **1** | **1** | **+1** | **E1** | **1** | **1** | **1** | **1** | **0** | **+1** | **1** | **1** |
| **2** |  | **C1** | **1** | **1** | **1** | **1** | **0** | **F1** | **1** | **+1** | **1** | **0** | **0** | **+1** | **1** |
| **3** |  |  | **B1** | **1** | **+1** | **1** | **0** | **0** | **+1** | **1** | **0** | **G1** | **1** | **1** | **1** |
| **4** |  |  |  | **D1** | **1** | **1** | **0** | **0** | **0** | **0** | **A1** | **1** | **1** | **1** | **+1** |