Homework 6: Memory Management

Matt Bass

Due date: noon of Thursday Apr 14, 2022

**[Question 1]** Rank the memory levels following from fastest to slowest:

Fastest to slowest (1-5)

1. Main memory **3**
2. Cache **2**
3. Register **1**
4. Solid state drive **4**
5. Magnetic disk **5**

**[Question 2]** In a single statement, write the main advantage and disadvantage of each of the following memory management schemes:

1. Single-User Contiguous Memory: Is when the entire program is loaded into memory. Its advantages are that it has contiguous memory space and allocates space as needed; it processes jobs sequentially and its memory manger performs minimal work. It evaluates incoming process size and loads if small enough to fit, otherwise, it . It also Monitors occupied memory space rejects and evaluates next incoming process. Its disadvantages are multiprogramming or networking is not supported, no sharing of RAM between processes, process memory size is restricted by memory size and it has low utilization of RAM, can be thought of as internal fragmentation
2. Fixed Partitions: In this type of allocation, main memory is divided into a number of fixed-sized partitions where each partition should contain only one process. One of its large advantages is that it permits multiprogramming. One of its big disadvatages is that external fragmentation can occur which is, a phenomenon in which storage space is used inefficiently, reducing capacity or performance and often both. Total memory space exists to satisfy a request, but it is not contiguous.
3. Dynamic Partitions: A few big advantages of this method is that is helps prevent internal fragmentation because memory is dynamically assigned to jobs based on their need, this leads to another advantage is that there is no size limitation for a processes other than the total memory avalible, and because of this it is possible to have more multiprogramming. Its disadvatages are that there is full memory utilization, only when first jobs loaded; subsequent allocation leads to memory waste, and external fragmentation can occur
4. Relocatable Dynamic Partitions: This is the same as dynamic partitions expect it also gets all the empty memory blocks and merges them into one large memory block to hep with memory defragmentation.
5. Paging: It uses a page table to translate physical addresses into logical ones and then uses that to keep track of all free frames (using n frames to run n programs) each fram represent a free space in memory . Its advantages are that it avoids external fragmentation and it avoids problem of varying sized memory chunks.

**[Question 3]** Apply the following page replacement algorithms to the following reference string with 3 frames, show your full table and mark page faults: 1, 3, 2, 1, 5, 6, 4, 3, 1, 5, 6, 2, 1

* First-in First-out
* Least Recently Used
* Optimal Page Replacement

**FIFO**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **3** | **2** | **1** | **5** | **6** | **4** | **3** | **1** | **5** | **6** | **2** | **1** |
| 1 | 1 | 1 | 1 | 5 | 5 | 5 | 3 | 3 | 3 | 6 | 6 | 6 |
|  | 3 | 3 | 3 | 3 | 6 | 6 | 6 | 1 | 1 | 1 | 2 | 2 |
|  |  | 2 | 2 | 2 | 2 | 4 | 4 | 4 | 5 | 5 | 5 | 1 |
| **fault** | **fault** | **fault** |  | **fault** | **fault** | **fault** | **fault** | **fault** | **fault** | **fault** | **fault** | **fault** |

**LRU**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **3** | **2** | **1** | **5** | **6** | **4** | **3** | **1** | **5** | **6** | **2** | **1** |
| 1 | 1 | 1 | 1 | 1 | 1 | 4 | 4 | 4 | 5 | 5 | 5 | 1 |
|  | 3 | 3 | 3 | 5 | 5 | 5 | 3 | 3 | 3 | 6 | 6 | 6 |
|  |  | 2 | 2 | 2 | 6 | 6 | 6 | 1 | 1 | 1 | 2 | 2 |
| **fault** | **fault** | **fault** |  | **fault** | **fault** | **fault** | **fault** | **fault** | **fault** | **fault** | **fault** | **fault** |

**Optimal**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **3** | **2** | **1** | **5** | **6** | **4** | **3** | **1** | **5** | **6** | **2** | **1** |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|  | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
|  |  | 2 | 2 | 5 | 6 | 6 | 4 | 4 | 5 | 6 | 2 | 4 |
| **fault** | **fault** | **fault** |  | **fault** | **fault** | **fault** |  |  | **fault** | **fault** | **fault** |  |

**[Question 4]** Briefly answer**,** what is Belady’s Anomaly?

It is the anomaly in which increasing the number of page frames results in an increase in the number of page faults for certain memory access patterns. This anomaly is commonly experienced when using the FIFO page replacement.

**[Question 5]** Can Belady’s Anomaly occur with Least Recently Used? Why?

No Belady’s Anomaly can not occur with Least Recently Used because the set of pages in memory would be the n most recently referenced pages. If the number of frames increases then these n pages will still be the most recently referenced and so, will still be in the memory.

**[Submission]**

Add your document to the *homeworks* directory in your shared folder on Drive.