|  |  |  |
| --- | --- | --- |
| **CS3800: Intro to Operating Systems** |  |  |
| **Memory Management Worksheet** | **Name:** | Evan Richard |

* Answer the following: [15pts]
* What is the difference between Fixed Partitioning and Dynamic Partitioning? What are the positives and negatives of using either?

|  |
| --- |
| *Fixed partitioning divides memory into set-size blocks before execution, while dynamic partitioning allocates memory blocks to processes based on their specific size during execution.* |

* What is the cause of thrashing? How does the system detect thrashing? Once it detects thrashing, what can the system do to eliminate this problem?

|  |
| --- |
| *Thrashing occurs when a system spends more time swapping pages in and out of memory than executing processes, usually due to insufficient memory or too many active processes. The system detects it by monitoring a high page fault rate. To fix thrashing, it can reduce the number of active processes, increase RAM, or apply working set or page-fault frequency algorithms.* |

* On a system with paging, a process cannot access memory that it does not own; why? Could the OS allow access to other memory? Why or why not

|  |
| --- |
| *Each process has its own page table managed by the OS, ensuring memory protection. A process can’t access others' memory to prevent corruption and security breaches. The OS could allow it (e.g., shared memory), but only under strict control and for specific purposes like inter-process communication.* |

* Consider a fixed partitioning scheme with equal-size partitions of 216 bytes and a total main memory size of 224 bytes. A process table is maintained that includes a pointer to a partition for each resident process. How many bits are required for the pointer? [10pts]

|  |
| --- |
| *256 partitions or 8 bits are required.* |

* Consider a memory-management system based on paging. In this system we are going to use 32 bit memory addresses. (Note, there is an example similar to this in the memory management slides, in the notes of the paging section ) [20]
* If we wanted to use 12 bits of our address to indicate the offset into a page, what should our page size be?

|  |
| --- |
| *4 KB* |

* Assuming the 12 bit offset of part a, what is the maximum number of pages a process can have in this system? What is the maximum process size?

|  |
| --- |
| *1,048,576 pages and a max process size of 4 GB* |

* What is the maximum amount of memory that this system can use (i.e., that it can address) using the 12 bit offset from above?

|  |
| --- |
| *4 GB* |

* If we change the size of the offset region in the address to 8 bits, does that change the maximum amount of memory that can be used in the system?

|  |
| --- |
| *No, it would still be 4 GB.* |

* Answer the following: [18]
* In a fixed partitioning scheme, what are the advantages of using unequal-size partitions?

|  |
| --- |
| *It helps accommodate processes of varying sizes more efficiently, reducing internal fragmentation and making better use of memory.* |

* What is the difference between internal and external fragmentation?

|  |
| --- |
| *Internal fragmentation is unused space inside allocated memory blocks while external fragmentation is unused space between allocated blocks, making it hard to find large enough contiguous free blocks.* |

* What is the difference between a page and a frame?

|  |
| --- |
| *A page is a fixed-size block of a process's logical memory and a grame is a fixed-size block of physical memory.* |

* What is the difference between a page and a segment?

|  |
| --- |
| *Page is a fixed-size block for memory management and a segment is a variable-size block representing logical division of a program.* |

* In paging, external fragmentation is not a concern, why?

|  |
| --- |
| *Because any free frame can be used, pages do not need to be contiguous in physical memory.* |

* In paging, internal fragmentation is only a concern on one specific page? Which page? Why only that one?

|  |
| --- |
| *It only affects the last page of a process, since the process likely won't perfectly fill the entire last page, leading to some unused space.* |

* Given N frames, M references, K distinct page numbers [10pts]
* What is the lower bound on the number of page faults? Explain.

|  |
| --- |
| *K, this assumes an optimal replacement strategy and that all distinct pages fit in memory.* |

* What is the upper bound on the number of page faults? Explain.

|  |
| --- |
| *M, this happens when there are fewer frames than needed, and every access requires reloading a page.* |

* Answer the following questions using the figure below. The grey regions indicate allocated space. [20pts]



* Given the following memory partitioning, indicate where a 6 Mb allocation would be made using
* Best Fit

|  |
| --- |
| *C* |

* First Fit

|  |
| --- |
| *A* |

* Next Fit

|  |
| --- |
| *D* |

* Is there Internal or External Fragmentation? If External Fragmentation, how can it be fixed and what would the resulting empty space be? If Internal Fragmentation, how much memory is being wasted? Note: assume the allocation from part a has not happened yet.

|  |
| --- |
| *External fragmentation, which can be fixed by using compaction. Resulting in a total new free memory space of 22MB.* |