FINAL EXAM – CSC 456 – Spring 2021

Record Answers on FINAL QUIZ in D2L Question 1 (5 points)

If we take a system and change its page/frame size to a larger size then Question 1 options:

|  |  |
| --- | --- |
|  | The TLB reach with be smaller |
|  | The number of page table entries for a process will be smaller |
|  | The Logical address register will need to get smaller |
|  | The Physical address register will need to get larger |

**Question 2** (5 points)

If the size of the “delta” ( /\_\ = ?) as part of the Working Set model is set larger Question 2 options:

|  |
| --- |
| Its working set will also get larger |
| Its working set will get smaller |
| The locality of Reference will not change |
| Thrashing will occur |

**Question 3** (5 points)

Assuming a 1-KB (1024 bytes) page size, what is the page number (p) and offset(d) for the following address reference (values are in decimal) - Address : **698**

Question 3 options:

|  |
| --- |
| p = 1 d = 698 |
| p = 0 d = 698 |
| p = 3 d = 193 |
| p = 0 d = 24 |

**Question 4** (5 points)

Consider the logical address space of **1024** pages of **1024** words each, mapped into a Physical memory of **128** frames. How many **bits** are there in the **Logical** address?

Question 4 options:

|  |
| --- |
| 20 |
| 16 |
| 1024 |
| 2048 |

**Question 5** (5 points)

Consider the following segment table

|  |  |  |
| --- | --- | --- |
| **Segment** | **Base** | **Length** |
| 0 | 1200 | 350 |
| 1 | 2200 | 150 |
| 2 | 150 | 110 |

What is the physical address of the following logical address? Address : (**2 , 140)**

Question 5 options:

|  |
| --- |
| 160 |
| 290 |
| There is an error in the segment table! |
| Segmentation Fault ! |

**Question 6** (5 points)

Consider the following page reference string: **1,2,3,1,3,1,3,1,2,1,2**

Assuming **pure** demand paging and 2 **frames** allocation, how many times will the Page Replacement Algorithm need to be run?

Use the First-in-First-Out (**FIFO**) algorithm. Question 6 options:

|  |
| --- |
| 3 |
| 2 |
| 5 |
| 7 |

**Question 7** (5 points)

Consider the following page reference string: **1,2,1,3,4,3,1,3,4,3,1,2**

Assuming **pure** demand paging and **3 frames** allocation, how many page faults will the Least Recently Used (**LRU**) algorithm produce.

Question 7 options:

|  |
| --- |
| 3 |
| 4 |
| 5 |
| 6 |

**Question 8** (5 points)

As the quantum (q) time for the Round Robin (**RR**) short term Schedule is set smaller and smaller Question 8 options:

|  |
| --- |
| It will become just like First-Come-First-Served Algorithm |
| No process will ever run |
| It could increase the number of Context switches |
| The degree of Multi-programming will increase |

**Question 9** (5 points)

The Scheduling Criteria “**Waiting Time**” for a process is

Question 9 options:

|  |
| --- |
| the total time it Waits in the “Ready” state |
| the total time the user waits for it to get done |
| the total time of the context switching it waits for |
| None of the others |

**Question 10** (5 points)

Using an **Internal** Priority Preemptive short-term scheduling algorithm Question 10 options:

|  |
| --- |
| can lead to deadlocks |
| can lead to starvation |
| will lead to the convoy effect |
| could never lead to starvation |

**Question 11** (5 points)

How many **new** processes are created by the program below?

int main(void)

{

fork(); fork()

fork()

return 0;

}

Question 11 options:

|  |
| --- |
| 0 |
| 7 |
| 3 |
| 14 |
| 9 |
| None of the other answers |

**Question 12** (5 points)

Is it possible to have concurrency but not parallelism? Question 12 options:

|  |
| --- |
| No |
| Yes |
| Yes, but only if you have multiple processors |
| Only sometimes |

**Question 13** (5 points)

What does the **Mutual Exclusion** requirement for the Critical Section problem address ? Question 13 options:

|  |
| --- |
| Correctness |
| Efficiency |
| Fairness |

**Question 14** (5 points)

What does the **Progress** requirement for the Critical Section problem address? Question 14 options:

|  |
| --- |
| Correctness |
| Efficiency |
| Fairness |
| All of the above |

**Question 15** (5 points)

Assume you have a system with many processors. You are asked to design a solution to the following problem.

*Given a “VERY” large Data set of N numbers,find the* ***largest*** *of all the numbers in the set and the* ***sum*** *of the number in the set . In creating a Parallel solution you develop a separate function to do each.*

What form of Parallelism are you using?

Question 15 options:

|  |
| --- |
| Data |
| Task |
| long term |
| short term |

**Question 16** (5 points)

What is a **Spin Lock**? Question 16 options:

|  |
| --- |
| A lock that suspends the process, removing it from the mix |
| A lock that has the process spins in the "WAIT" state |
| A lock that has the process spin in the "Ready" state |
| A lock that has the process spin in the "RUN" state |

**Question 17** (5 points)

Implementing Multi-Threading makes sense Question 17 options:

|  |
| --- |
| if we want to decrease the Degree of Multi-programming |
| only if we have multiple processors to schedule them on |
| if we are in a Distributed OS system |
| if we want to try and increase CPU utilization |

Question 18 (5 points)

Which of these is NOT a necessary condition for Deadlock ? Question 18 options:

|  |
| --- |
| Mutual Exclusion |
| Bounded Wait |
| Hold & Wait |
| Circular Wait |

Question 19 (5 points)

If the user whole program must be in main memory and loaded into contiguous main memory then Question 19 options:

|  |
| --- |
| Paging will help reduce Internal Fragmentation |
| External Fragmentation will be eliminated |
| only one page per process would be needed |
| only a Base Address register would be needed to map the Logical Address to the Physical address |

Question 20 (5 points)

A Page Fault occurs because Question 20 options:

|  |
| --- |
| the page being referenced does not have a valid Frame # in the Page Table |
| the page being referenced is not found in the TLP |
| all the frames allocated to a process are full when a new page needs to be brought in from the backing store |
| a page displacement value is out of range |