|  |  |  |
| --- | --- | --- |
| 1. | True/False | |
|  | Q: | T/F: Dynamic load balancing must come from one central source, often known as a master |
|  |  |  |
|  | A: | False, “mini-masters” can certainly be used if the number of slaves will overwhelm a single master. |
|  |  |  |
|  | Ref: | Page 205 |

|  |  |  |
| --- | --- | --- |
| 2. | Multiple Choice | |
|  | Q: | Which of the following is not static load balancing scheme:   1. Genetic algorithms 2. Randomized algorithms 3. Line structure job loading 4. Round-robin |
|  |  |  |
|  | A: | C. Line structure job loading is intended for pipelined schemes/architectures, and is not really a static load balancing scheme |
|  |  |  |
|  | Ref: | Pages 202, 207 |

|  |  |  |
| --- | --- | --- |
| 3. | Fill in the Blank | |
|  | Q: | A great \_\_\_\_\_\_\_\_\_\_ of a centralized dynamic load balancing structure is that it is \_\_\_\_\_\_\_\_\_\_ for the master to know when to terminate. |
|  |  |  |
|  | A: | advantage; simple. Since the master always knows what tasks remain, the master can easily send out the signal to other processes to stop. |
|  |  |  |
|  | Ref: | Page 204 |

|  |  |  |
| --- | --- | --- |
| 4. | Short Answer/Code | |
|  | Q: | When is static load balancing superior to dynamic load balancing and vice versa? |
|  |  |  |
|  | A: | Static load balancing is superior when all problem and hardware parameters are known and the load can be pre-divided, resulting in greater performance due to a lack of message passing. Dynamic allocation is superior when it is not known if the hardware is balanced, when the job “size” is not known, etc., so a dynamic approach can more evenly distribute the work to minimize process idleness. |
|  |  |  |
|  | Ref: | Pages 201-207 |