**Chapter 10**

**Multiprocessor and Real-Time Scheduling**

**TRUE/FALSE QUESTIONS:**

T F 1)  A loosely coupled or distributed multiprocessor consists of a collection of

relatively autonomous systems, with each processor having its own main

memory and I/O channels.

T F 2)  With independent parallelism there is an explicit synchronization among processes.

T F 3)  A typical use of independent parallelism is in a time-sharing system.

T F 4)  A disadvantage of static assignment is that one processor can be idle, with an

empty queue, while another processor has a backlog.

T F 5)  In most traditional multiprocessor systems processes are dedicated to processors.

T F 6)  An application can be implemented as a set of threads, which cooperate and

execute concurrently in the same address space.

T F 7)  Because of the potential disadvantages, load sharing is not one of the commonly

used schemes in current multiprocessors.

T F 8)  The operating system, and in particular the scheduler, is perhaps the most important

component of a real-time system.

T F 9)  One useful measure of the ability of an operating system to function deterministically

is the maximum delay from the arrival of a high-priority device interrupt to when

servicing begins.

T F 10)  User control is generally much broader in an ordinary operating system than in a

real-time operating system.

T F 11)  Static table-driven scheduling is applicable to tasks that are periodic.

T F 12)  One of the more promising methods of resolving multitask scheduling conflicts

for periodic tasks is rate monotonic scheduling.

T F 13)  Real-time tasks are not handled any differently than non-real-time tasks in the

priority queues.

T F 14)  Linux provided a real-time scheduling capability coupled with a scheduler for

non-real-time processes that made use of the traditional UNIX scheduling algorithm.

T F 15)  In the real-time-priority class, all threads have a fixed priority that never changes and

all the active threads at a given priority level are in a round-robin queue.

**MULTIPLE CHOICE QUESTIONS:**

1)  The multiprocessor system which consists of a set of processors that share a common main

memory and are under the integrated control of an operating system is a \_\_\_\_\_\_\_\_\_ .

A)  tightly coupled multiprocessor

B)  cluster

C)  loosely coupled or distributed multiprocessor

D)  functionally specialized processor

2)  With coarse and \_\_\_\_\_\_\_\_\_\_ grained parallelism, there is synchronization among processes, but

at a very gross level.

A)  medium   B)  fine

C)  independent   D)  very course

3)  An example of \_\_\_\_\_\_\_\_\_\_ is an I/O processor.

A)  loosely coupled or distributed multiprocessors

B)  tightly coupled multiprocessors

C)  cluster multiprocessing

D)  functionally specialized processors

4)  A \_\_\_\_\_\_\_\_\_\_ is one that must meet its deadline, otherwise it will cause unacceptable damage

or a fatal error to the system.

A)  periodic task   B)  soft real-time task

C)  hard real-time task   D)  aperiodic task

5)  Typically, there will need to be a rather high degree of coordination and interaction among

the threads of an application, leading to a \_\_\_\_\_\_\_\_\_\_ level of synchronization.

A)  independent grain   B)  medium grain

C)  course grain   D)  fine grain

6)  The \_\_\_\_\_\_\_\_\_\_ approach is a set of related threads scheduled to run on a set of processors at

the same time, on a one-to-one basis.

A)  dynamic scheduling   B)  load sharing

C)  gang scheduling   D)  dedicated processor assignment

7)  A \_\_\_\_\_\_\_\_\_ has an associated deadline that is desirable but not mandatory.

A)  hard real-time task   B)  periodic task

C)  soft real-time task   D)  aperiodic task

8)  With the \_\_\_\_\_\_\_\_\_\_ approach the number of threads in a process can be altered during the

course of execution.

A)  dynamic scheduling   B)  load sharing

C)  gang scheduling   D)  dedicated processor assignment

9)  \_\_\_\_\_\_\_\_\_\_ perform static analysis of feasible schedules of dispatching with the result of the

analysis being a schedule that determines, at run time, when a task must begin execution.

A)  Static priority-driven preemptive approaches

B)  Static table-driven approaches

C)  Dynamic planning-based approaches

D)  Dynamic best effort approaches

10)  \_\_\_\_\_\_\_\_\_\_ is the time required to execute the task to completion.

A)  Processing time   B)  Priority time

C)  Resource time   D)  Ready time

11)  The basic idea of \_\_\_\_\_\_\_\_\_ is that a lower-priority task inherits the priority of any higher-priority

task pending on a resource they share.

A)  priority ceiling   B)  priority inversion

C)  unbounded priority inversion   D)  priority inheritance

12)  In the \_\_\_\_\_\_\_\_\_\_ approach, a priority is associated with each resource.

A)  priority ceiling   B)  unbounded priority inversion

C)  priority inversion   D)  priority inheritance

13)  Processes at \_\_\_\_\_\_\_\_\_ priority levels are guaranteed to be selected to run before any time-sharing

process but must defer to real-time processes.

A)  kernel   B)  time-shared

C)  real time   D)  load sharing

14)  \_\_\_\_\_\_\_\_\_\_ is the lowest-priority process, intended for user applications other than real-time

applications.

A)  Kernel   B)  Variable priority

C)  Real time   D)  Time-shared

15)  The \_\_\_\_\_\_\_\_\_\_ class is intended for applications that will only consume processor time when

no other threads are ready to execute.

A)  idle user   B)  time-sharing

C)  bottom-half kernel   D)  real-time

**SHORT ANSWER QUESTIONS:**

1)  A good way of characterizing multiprocessors and placing them in context with other

architectures is to consider the synchronization \_\_\_\_\_\_\_\_\_\_ , or frequency of synchronization,

between processes in a system.

2)  Linux uses \_\_\_\_\_\_\_\_\_ , in which threads are moved from a queue for one processor to a queue

for another processor.

3)  In a \_\_\_\_\_\_\_\_\_\_ architecture, the kernel can execute on any processor, and each processor

does self-scheduling from the pool of available processes.

4)  The \_\_\_\_\_\_\_\_\_ approach is the opposite of the load-sharing approach and provides implicit

scheduling defined by the assignment of threads to processors.

5)  \_\_\_\_\_\_\_\_\_\_ is useful for medium-grained to fine-grained parallel applications whose

performance severely degrades when any part of the application is not running while

other parts are ready to run.

6)  \_\_\_\_\_\_\_\_\_\_ computing is defined as that type of computing in which the correctness of the

system depends not only on the logical result of the computation but also on the time at

which the results are produced.

7)  An \_\_\_\_\_\_\_\_\_ has a deadline by which it must finish or start, or it may have a constraint on

both start and finish time.

8)  An operating system is \_\_\_\_\_\_\_\_\_\_ to the extent that it performs operations at fixed,

predetermined times or within predetermined time intervals.

9)  \_\_\_\_\_\_\_\_\_ is a characteristic that refers to the ability of a system to fail in such a way as

to preserve as much capability and data as possible.

10)  No feasibility analysis is performed with \_\_\_\_\_\_\_\_\_\_ approach, and the system tries to

meet all deadlines and aborts any started process whose deadline is missed.

11)  \_\_\_\_\_\_\_\_\_\_ is the time at which a task becomes ready for execution.

12)  For \_\_\_\_\_\_\_\_\_\_ scheduling, the highest-priority task is the one with the shortest period,

the second highest-priority task is the one with the second shortest period, and so on.

13)  \_\_\_\_\_\_\_\_\_\_ occurs when circumstances within the system force a higher-priority task to

wait for a lower-priority task.

14)  Processes at \_\_\_\_\_\_\_\_\_ priority levels are guaranteed to be selected to run before any kernel

or time-sharing process.

15)  A thread is considered to be \_\_\_\_\_\_\_\_\_ if the ratio of its voluntary sleep time versus its runtime

is below a certain threshold.